KINGSTON UNIVERSITY LONDON

DOCTORAL THESIS

Visual Memories

Author:
Jiri FAJTL

Supervisor:
Professor Paolo REMAGNINO
Professor Vasileios ARGYRIOU
Professor Dorothy MONEKOSSO

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

RoViT - Robot Vision Team
Faculty of Science, Engineering and Computing

June 21, 2021

Declaration of Authorship

I, Jiri FAJTL, declare that this thesis titled, "Visual Memories" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date: 21/6/2021

KINGSTON UNIVERSITY LONDON

Abstract

Faculty of Science, Engineering and Computing

Doctor of Philosophy

Visual Memories

by Jiri FAJTL

Despite the rapid progress in the field of artificial intelligence, there are still important new areas to be explored and existing methods enhanced to make machines think like humans. This thesis conducts research in four machine learning and computer vision areas in this direction.

First, we study what makes some images more memorable than others and propose a new machine learning method to learn and predict image memorability, closely matching human performance. A spatial attention function is learnt to localize image regions responsible for the image retention in memory.

To identify meaningful temporal segments in a video stream, we study episodic segmentation in our memory and design a novel algorithm for video summarization to mimic human capabilities. A soft, self-attention method without a recurrent network is used to learn frame importance scores for the video summarization. This simple algorithm demonstrates a performance superior to the current state-of-the-art methods.

Inspired by our brain's ability to project high dimensional visual information to computationally efficient, meaningful representations, we propose a method for latent binary representations learning and methods for operations in this discrete latent space such as interpolation, novel image generation, and attribute modification outperforming more complex published methods.

To advance methods targeting catastrophic interference, one of the most fundamental problems of artificial neural networks, we study elementary neural mechanisms mitigating this phenomenon in our brain's memory. Building on our insights on the function of pattern separation in the hippocampus, we propose a conceptually simple and resource-efficient method to learn high dimensional sparse binary representations for continual learning. By performing elementary binary operations *or* and *and* over a continual stream of sparse representations of novel classes, our method exhibits performance significantly exceeding the current state-of-the-art meta-learning methods on identical benchmarks.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my main supervisor Professor Paolo Remagnino for his constant, multidimensional support during my studies. Equally, I would like to thank my other supervisors Professor Vasileios Argyriou and Professor Dorothy Monekosso for many friendly, stimulating discussions.

Further on, I would like to thank all my colleagues from the Robot Vision Team (RoViT) research laboratory and, in particular, Dr. Rob Dupré for always being available for discussions and his collaboration on the *Improving dataset volumes and model accuracy with semi-supervised iterative self-learning* paper.

My high appreciation also goes to my Research Student Co-ordinator Rosalind Percival for her kindness and solid support in all administrative matters.

Contents

List of Figures

List of Tables

Acronyms

AAE Adversarial Autoencoders

AE autoencoder

AGI Artificial General Intelligence

ANN artificial neural network

AP Action Potential

CF Catastrophic Forgetting

CIL class-incremental Learning

CL Continual Learning

CLS Complementary Learning Systems

CNN Convolutional Neural Network

CVAE Conditional Variational Autoencoder

DG Dentate Gyrus

DIL domain-incremental Learning

EC Entorhinal Cortex

EEG electroencephalogram

EPSP Excitatory Postsynaptic Potential

FID Fréchet Inception Distance

GAN Generative Adversarial Networks

GMM Gaussian Mixture Models

GRU gated recurrent unit

GT ground truth

HOG Histogram of Oriented Gradients

HTM Hierarchical Temporal Memory

IID independent and identically distributed

KID Kernel Inception Distance

LBAE Latent Bernoulli Autoencoder

LSTM long short-term memory

LTM Long Term Memory

MEG Magnetoencephalography

ML machine learning

MSE Mean Squared Error

NLP natural language processing

NME Nearest Mean Exemplars

ood out-of-distribution

ReLU rectified linear unit

RGC retinal ganglion cells

RNN Recurrent Neural Network

SBR Sparse Binary Representation

SDM Sparse Distributed Memory

SDR Sparse Distributed Representation

SGD stochastic gradient descent

SIFT Scale Invariant Feature Transform

SOTA state-of-the-art

SSIM Structural Similarity Index Measure

STM Short Term Memory

SVM Support Vector Machine

SVR Support Vector Regression

TIL task-incremental Learning

VAE Variational Autoencoder

VQ-VAE Vector Quantised-Variational Autoencoder

WAE Wasserstein Autoencoders

To my amazing wife Mika and my beautiful daughters Aimi and Nicola for their everlasting, unconditional love and support.

Chapter 1

Introduction

The ability to remember our experiences, recall them later with cues, or revisit events of the past in our thoughts and learn from them is the hallmark of our memory system and the cornerstone of intelligence (Halford et al., 2007). All information to process and retain in the memory travel via sensory stimuli. Out of all our senses, vision provides the richest representation of our environment, which makes it a vital sense for navigation and observational and interactive learning. Consequently, vision is also a critical sense for autonomous systems, particularly those collaborating with humans due to their need to observe and recognize the same world features as our vision system does.

However, processing visual information is a complex and resource demanding task. The human retina has approximately 120 million cells (Schacter et al., 2011) connected by over 1 million neural fibers (Jonas et al., 1992) to the visual cortex (area 17) with about 300 million neurons (Leuba and Kraftsik, 1994). This is compared to 30,000 fibers within the cochlear nerve of our ear and 100 million neurons in the auditory cortex (A1) (MacGregor, 1993). Given the amount of information to process, the visual system is very fast, able to identify known image patterns within 13 ms (Potter et al., 2014).

To efficiently operate with such a vast stream of visual information our brain projects visual data to high dimensional but low activity representations (Quiroga, 2016) and stores them in a short-term memory. To consolidate new experiences to the long term memory without interference with already retained knowledge, our brain sets the stage to many sophisticated pattern replays, transformations, and associations among several brain regions (Káli and Dayan, 2004). At the end of this process, not all visual stimuli will experience the same memory retention (Isola et al., 2011a), some images will consolidate in our memory while others will rapidly fade away. Furthermore, we do not remember or recall arbitrarily long events. Our brain segments the continuous visual stream into coherent, shorter episodes characterized by few keypoint events (Kurby and Zacks, 2008).

At first sight, the way our brain encodes visual information, segments important sequences and stores them without interference may appear unremarkable. However, these functions are the bedrock of a vast majority of machine learning methods,

particularly in computer vision such as methods for object detection, visual question answering, video summarization, few shots and continual learning, autonomous navigation, and path planning and many others.

The primary goal of this thesis is to advance the following machine learning methods for computer vision by applying techniques inspired by the functions of our brain. The objectives are:

- study how our memory retains images as a function of their content and apply the findings in the design of a machine learning method to mimic human memory function. While this is not a new problem, the goal here is to design novel methods with performance superior to the prior work. This work is discussed in Chapter 2 where a learned spatial soft attention function is used in combination with a recurrent neural network to model image memorability, setting new state-of-the-art results.

- explore mechanisms behind episodic memory segmentation in our brain and attempt to apply them to a machine learning method for video summarization. A novel method for video summarization is proposed in Chapter 3 where a soft, self-attention network learns relations within the training video stream as a function of given frame importance scores annotated by humans. The learnt attention is then used to predict frame scores to summarize other video streams leading to results superior to the current state-of-the-art.

- investigate neural coding schemas and attempt to learn similar encoding with a neural network. A novel method for unsupervised binary representation learning is proposed in Chapter 4. This work also includes novel methods for random images generation, image interpolation, and attributes modification that outperform current methods.

- study catastrophic interference in the brain and its mitigation, and apply the insights to a machine learning method for continual learning of images for classification. A novel method to learn sparse binary distributed representations along with a dynamic pattern memory is introduced in Chapter 5. This conceptually straightforward and resource-light method significantly outperforms the current state-of-the-art methods, continually learning over 600 classes in the challenging, task-free incremental class learning scenario.

The main inspirations behind the propositions in this work are drawn through the lens of relevant functions of our brain, the only example of a working intelligent system in existence. However, it is imperative to note that the goal of this thesis is not to model any brain system or make conclusions on its functions but rather to study the brain processes applicable to the computation methods.

1.1 Model Complexity Overview

We focus on artificial neural networks (ANNs) as the core machine learning method in our work. As one of the lesser objectives, we also try to prioritize low complexity neural network methods.

In the context of this thesis, we define a low complexity method as a non-iterative, homogeneous (all layers are assemblies of a limited subset of identical building blocks such as the up/down convolution, basic activation, normalization, and scaling functions), single, end-to-end trained ANN without explicit objective functions in the neural pathways and non-hybrid architectures.

On the other side of this spectrum stand the high complexity methods, for example, methods with explicit objective functions in the pathway (Jaderberg et al., 2015, Liu et al., 2016), hybrid architectures (Zarezadeh et al., 2017, Wang et al., 2015) or meta-learning methods (Finn et al., 2017, Nichol et al., 2018) or procedures requiring to independently train multiple neural models (Kemker and Kanan, 2018, Hayes et al., 2020). The complexity of these methods increases as more diverse algorithms are utilized, and the more training or inference methods are required to be executed to achieve the method's objective.

Due to ANN homogeneity, low complexity methods can be implemented in software and hardware by following a single and straightforward design pattern, for example, a combination of a few fixed-kernel size 2D convolutions, multiply and accumulate, maximum and average operators. This homogeneous structure is akin to a biological neural network, where the anatomical structures on the level of neurons and their assemblies are also similarly homogeneous (cytoarchitectural homogeneity) despite their functional diversity (Haak and Beckmann, 2020).

According to the above definition, low complexity methods are also inherently easy to optimize (e.g., optimization within a single building block, such as performing FFT convolution, extends to the entire model), with parallelization being perhaps the most important (Upadhyaya, 2013, Seiffert, 2004).

Moreover, due to the absence of any explicit prior during the optimization and the end-to-end training, low complexity methods are typically not subjected to any algorithmic bias (in contrast to, for example, the pre-defined anchor boxes in the SSD method (Liu et al., 2016)), only to biases expressed in the training data.

1.2 Thesis Overview

There are four main chapters in this thesis. Each chapter commences with a brief exploration of neurological processes in our brain underpinning the studied problem. Literature review of related machine learning (ML) research follows. Then, building on the prior ML methods and insights gained from the functions of our brain, a novel ML method is proposed, followed by evaluation and results. Each chapter ends with a conclusion summarizing the achieved contributions. A Pictorial overview of this thesis is presented in Figure 1.1.

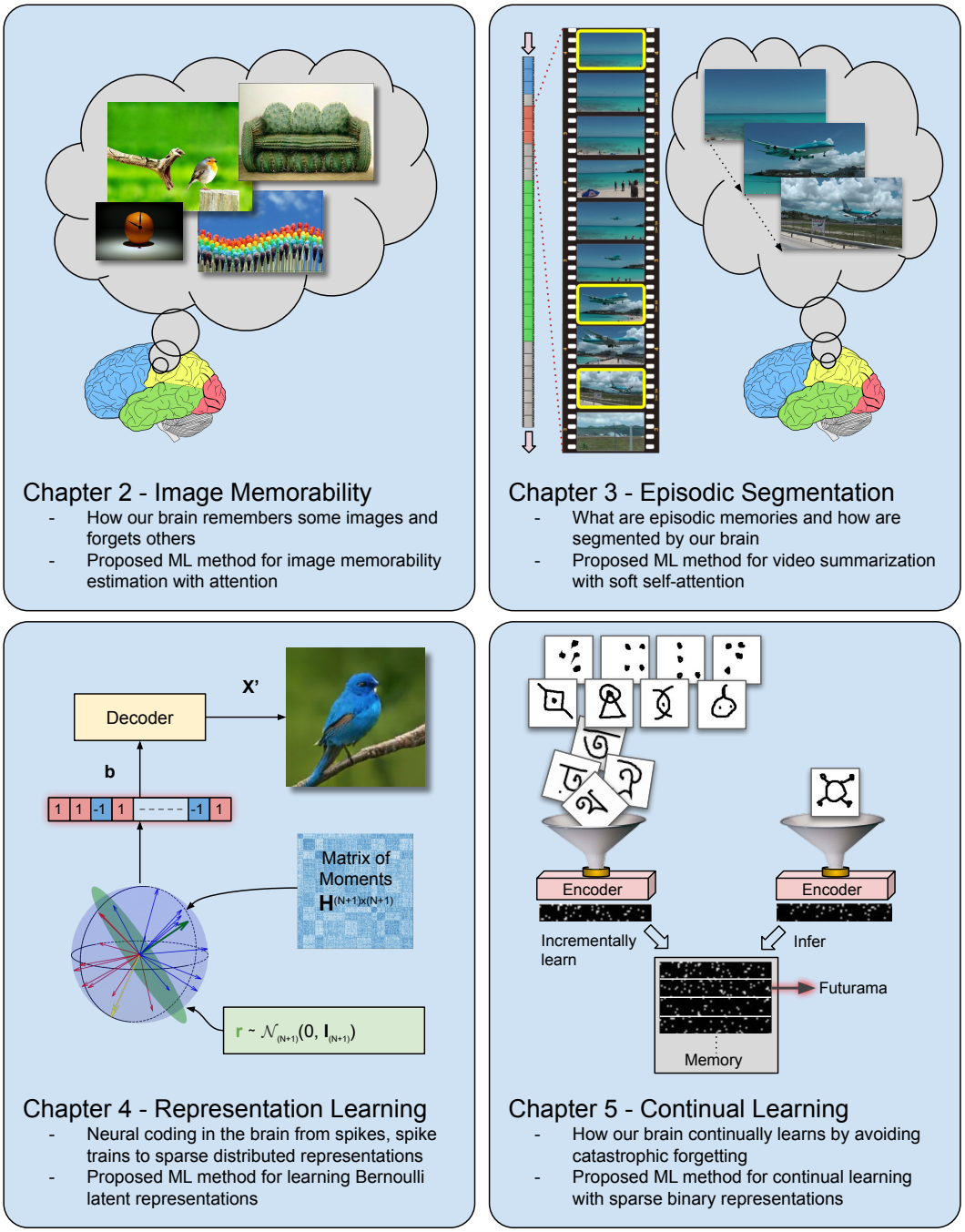


FIGURE 1.1: An overview of this thesis.

1.3 Publications

1.3.1 Published Work Included in this Thesis

- Amnet: Memorability Estimation with Attention
  Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), pages 6363–6372. This publication is included in Chapter 2.

- Summarizing Videos with Attention
  Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Asian Conference on Computer Vision (ACCV 2018), pages 39-54, Springer, 2018b. This publication is included in Chapter 3.

- Latent Bernoulli Autoencoder
  Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. International Conference on Machine Learning (ICML 2020), pages 2964-2974, 2020. This publication is included in Chapter 4.

1.3.2 Work Submitted for Publication

- Sparse Binary Representations for Continual Learning
  Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Submitted for publication to IEEE Transactions on Neural Networks and Learning Systems. This publication is included in Chapter 5.

1.3.3 Published Work not Included in this Thesis

- Deep Residual Network with Subclass Discriminant Analysis for Crowd Behaviour Recognition. Bappaditya Mandal, Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. IEEE International Conference on Image Processing (ICIP 2018), pages 938-942, 2018

- Improving Dataset Volumes and Model Accuracy with Semi-Supervised Iterative Self-Learning. Rob Dupre, Jiri Fajtl, Vasileios Argyriou, and Paolo Remagnino. IEEE Transactions on Image Processing, volume 29, pages 4337-4348, 2019

- A Comparison of Embedded Deep Learning Methods for Person Detection. Chloe Kim, Mahdi Oghaz, Jiri Fajtl, Vasileios Argyriou, and Paolo Remagnino. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), volume 5, pages 459-465, 2019

- Unsupervised Methods for a Personalised Route Recommendation System. Jiri Fajtl, Lorenzo Vitali, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP 2020), pages 1–4, IEEE, 2020b

Chapter 2

Image Memorability Estimation

Computational memory models for Artificial General Intelligence (AGI) agents typically draw inspiration from the brain, whether in the term of the storage capacity, autoassociative recall or fast continual learning without catastrophic interference. An essential part of memory is a circuitry responsible for regulating what information to retain and for how long. It may seem tempting to believe that, in an ideal case, our memory should retain all sensed information without distortion. Remembering everything, however, has been shown (Gaigg et al., 2008) not optimal since the information overload impairs the inter and intra-associations between new sensory information and existing memories during memory replay. This has a detrimental effect on the higher-level cognitive processes. From the evolutionary perspective, our brain learned to extract and store only the most relevant information to maximize our survival and minimize energy expenditure. If we understand what experiences exhibit higher memory retention in our brain, we may be able to replicate this process in an artificial memory. In particular, we would like to understand which elementary features and their structures in the core of these experiences define memory retention. This may lay the path to the emergence of memory models with more abstract, predictive and associative properties and reduce sudden performance drop on old tasks after learning new, known as catastrophic interference, significantly affecting current neural network architectures.

In this work, we study image memorability as a conduit to understanding visual memory retention. Specifically, we focus on a method allowing us to replicate the memorization behavior of our visual cortex with artificial neural networks. Further on, we attempt not only to perform the estimate for the entire image content, but also to learn a sequence of image regions whose memorability progression constitutes the memorability.

Aside from being a valuable building block of machine learning models, image memorability estimation can be utilized by many industrial applications. For example, to curate images for advertisement, a tool to illustrate educational material or presentations, as a memorability indicator of photographed scenes in a view-finder of future cameras, organize and tag photos in albums, or to help improve the memorability of critical elements of user interfaces. The image memorability estimation

could also be used to monitor a decline in memory capacity of patients affected by dementia, such as Alzheimer's and later stages of Parkinson's disease. Parkinson's disease affects the central nervous system responsible for planning and executing motion (Underwood and Parr-Brownlie, 2021), leading to symptoms such as shakiness, muscle stiffness, and lack of facial expression. The spread of the disease typically leads to a decline of mental functions and the development of dementia (Hely et al., 2008).

This chapter opens up with a brief introduction to the image memorability in Section 2.1, followed by a review of the current research on image properties corresponding to the memorability in Section 2.1.2. Section 2.1.3 shows how image memorability is measured and introduces the latest image datasets annotated with the memorability labels. The rest of the chapter centers on the image memorability estimation with machine learning techniques, leading with literature review in Section 2.2 and subsequently a proposal of our method for the memorability estimation with an attention component in Section 2.3. Evaluation with results follows in Sections 2.4 and 2.5 with an application to the image aesthetics estimation presented in Section 2.6. The specific role of attention on the memorability estimation is presented in Section 2.7. Implementation details in Section 2.8 and conclusion in Section 2.9 end this chapter.

2.1 Introduction to Image Memorability

Recalling images from our past is a ubiquitous, everyday experience, along with its quirks such as quickly forgetting an important scene we want to remember while on other occasions spontaneously picturing seemingly meaningless views from the past. We can remember thousands of images but not equally well. Images with close connections to our past experiences, particularly repeated ones, or experiences evoking strong emotions, stick more in our memory.

Surprisingly, there are image properties that make some images more memorable than others, regardless of subjective perception. Isola et al. (2014) revealed that the image memorability is an intrinsic image property, stable over observers and time; the general population tends to remember the same images equally well. Some images draw more attention than others, which creates an impression that these images may be primed to exhibit higher memory retention than the rest. This, counter to our intuition, has been shown not to be the case (Bainbridge, 2020). Moreover, memorable images typically do not even raise higher attention. Another interesting aspect of the image memorability is that it is not affected by a cognitive control of active remembering or forgetting. In this case, people still retain the same images equally well even though they were asked to remember or forget them. Perhaps the most intriguing observation is that individuals are unable to judge on the memorability level (Isola et al., 2014); there are no intuitive clues in the image content indicative

of memory retention. While image memorability has distinct, underlying neurobiological processes (Khaligh-Razavi et al., 2016), there is a known connection with the active memory retention. The intrinsic image memorability constitutes about 50% of the overall memory performance (Bainbridge et al., 2013).

Research on the image memorability has established an existence of a distinct, low-level neurobiological mechanism that our brain uses to encode and store visual memories. While there is no comprehensive account of the underlying processes, Isola et al. (2011b) devised a method to measure and quantify the image memorability and, in particular, to estimate it with machine learning methods.

In our work, we build on the current body of research and propose a new method that improves on the memorability estimation accuracy and, with equal importance, sets up a tool that assists with its interpretability.

2.1.1 Memory Retention in Visual Cortex

Understanding the information capacity of human memory is crucial in determining the computational constraints on visual tasks. If found, we may establish a maximum memory capacity required by the autonomous system to mimic the capabilities of the human visual system. Already in the early days of the visual cortex research, it has been found that our visual system has an enormous capacity to store and recall a large number of images (Standing, 1973). Recently, Brady et al. (2008) conducted a systematic examination of the long-term visual memory storage and concluded that the memory capacity is even larger than originally believed, and that in the term of memorization and recall of individual images as well as the image details. On a number of experiments with a dataset of 2500 images, Brady et al. (2008) attempted to establish bounds on the information capacity of human memory (Landauer, 1986) by measuring recall of details in images presented to participants. The authors concluded that despite the substantial increase in the memorized and recalled visual information in their experiment, the maximum capacity of human memory was still not reached.

To delineate between the general visual memorability and specific image memorability, Khaligh-Razavi et al. (2016) used Magnetoencephalography (MEG) to analyze how neurological signatures and timing of the perception and memory encodings correlate with the high and low memorabilities. Results have shown that the neurological signature of visual memorability appears in the late stages of the perception and before the memory encoding. Further on, they discovered that the neurological signatures of images with high and low memorabilities are distinct even for images subsequently not retained in the long-term memory. This signifies that memorability is a high-level, intrinsic image property with a neurological dynamic independent from other known memory stimuli.

In another work, Bainbridge (2020) studied the effects of the bottom-up and top-down processes on the memorability function. Over a set of experiments, he documented that, contrary to general belief, cognitive control does not override the image

memory imprint. Perhaps more surprisingly, memorable images are not associated with automatic attention capture. Finally, the study reveals that priming by repetition has little influence on the memorability function. The work concludes that image memorability is, indeed, a separable aspect of the memory encoding process with more a subconscious, bottom-up flow that is resilient to the top-down alterations.

While perhaps obvious, it is good to remind that image memorability targets Long Term Memory (LTM). On the other hand, Short Term Memory (STM) is known to have a capacity of up to 4 self-contained visual items for up to 15 seconds without active maintenance or rehearsal (Cowan, 2001). The STM is highly susceptible to the recency effect, where the recall rapidly declines over time and exposure to new information. The recency effect applies to all images equally. The STM encoding happens in the hippocampal circuits, followed by gradual consolidation in the LTM in the cortex. As discussed above, Bainbridge (2020) has shown that various cognitive controls, active during the STM encoding, have little to no influence on the memorability. This suggests that the memorability control occurs during the consolidation stage, but likely not during the long-term encoding (Khaligh-Razavi et al., 2016).

Further on, current research indicates that for some images, the visual memory encoding does carry a large amount of details, rapidly retained and available for long-term recall (Brady et al., 2008) along a path resilient to the top-down influence (Bainbridge, 2020). However, it still remains to be identified what particular image features give rise to the image memorability, their neurological encoding, and the memory traces they stimulate.

2.1.2 What Makes Images Memorable

As we learned, not all images exhibit the same memory imprint. The obvious questions to ask are; what possible image compositions, low-level visual features, and other image attributes may correlate with the image memorability. Answers to these questions could shed some light on image structures that are more biologically important, thus prone to memory retention. Consequently, we could use this knowledge to devise a computational model exhibiting similar performance, hopefully, with the memory retention closer to our visual cortex.

Early in the research on the memorability Isola et al. (2011a, 2014) and later Dubey et al. (2015) discovered that perhaps not surprisingly, pictures of people, central objects, and salient features are more memorable than landscapes and non-distinct images. A study on the memorability correlation with other image attributes was conducted by Isola et al. (2011b) and Khosla et al. (2015), investigating particularly causal relationships with aesthetics, emotions, popularity, and saliency image attributes.

The results show that popularity scores of highly memorable images on social media (Flickr[1]) are higher than the rest. Given that image memorability is an intrinsic property and that the memorability annotation procedure can detect images recalled but not yet presented (false positives, where the annotator has already seen the images somewhere else), the relation between memorability and popularity is causal. Therefore, we can conclude that image memorability is a contributing factor to the image popularity score. This, however, does not hold good for other attributes.

Contrary to our intuition, image aesthetics has low to no correlation with memorability (Isola et al., 2014). This observation may appear more understandable under the lens of other experiments relating emotions evoked by an image to its memorability. In a recent study, Khosla et al. (2015) found that strong emotions, particularly negative ones, have a close link with image memory retention. This is likely controlled by the amygdala as a response to a fear from a perceived thread (LaBar, 2007). Figure 2.1 shows the correlation between memorability and four image attributes analyzed by Khosla et al. (2015).

Saliency is a good predictor of image memorability, however, only for images with a simple context of a single or few objects (Khosla et al., 2015, Dubey et al., 2015, Mancas and Le Meur, 2013). In scenes with complex compositions, the link is weaker. This indicates that the fewer fixation points in the image, the higher memory retention is triggered.

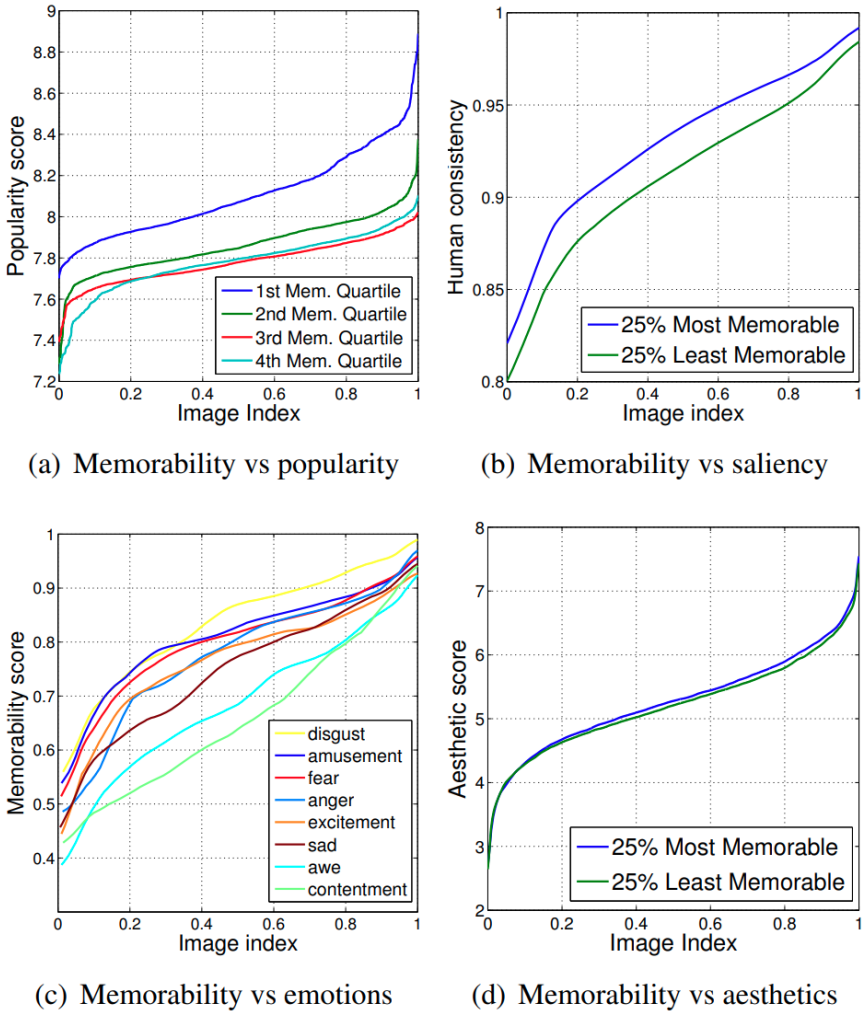
Some images instantly *pop-up*, while others require a closer inspection. Given this experience, we could be inclined to believe that images quickly capturing our attention would be also easy to recall later. However, it has been shown that, in general, such images do not exhibit higher memorability scores (Bainbridge, 2020). On the other hand, regions of images with simple compositions (with one or few objects), targeted by the first eye's fixations, do show a contribution to the memorability score. A relationship between visual attention and memorability was partially addressed in the works of Mancas and Le Meur (2013), Khosla et al. (2012) and Isola et al. (2011b), however not thoroughly investigated.

Ideally, we would like to identify low-level image features stimulating memory retention since computational methods could learn these. However, this may not be attainable because long-term memory does not capture low-level perceptual features well (Konkle et al., 2010, Brady et al., 2013). The connection between global color features was studied in Isola et al. (2014). The work reports a weak correlation with memorability for mean hue in the red regions, with a decline in the memorability as the hue shifts towards the blue.

2.1.3 Measuring and Quantifying Memorability

Image memorability is quantified as a probability of recalling an image that we have seen in the past. To calculate this probability, each image is viewed and recalled by a

[1] www.flickr.com



(a) Memorability vs popularity (b) Memorability vs saliency

(c) Memorability vs emotions (d) Memorability vs aesthetics

FIGURE 2.1: Correlation between image memorability and popularity, saliency, emotions and image aesthetics. As the number of images for each plot is different, the image index has been normalized to range between 0 and 1. Sourced with permission from Khosla et al. (2015).

large number of annotators and labelled as True="I have seen this image" or False= "I do not remember seeing the image." The memorability score of each image is then given as an average over the memorabilities reported by all annotators and scaled to the range between zero and one.

To measure image memorability, and subsequently assemble annotated datasets, Isola et al. (2011b) proposed a memory game where the player is presented with a sequence of images, some repeating over the game duration, and is asked to indicate whether the current image has already been presented. Each image was shown for 1 second with 1.4 seconds gaps. Each game included 120 images and took 4.8 minutes to perform. A total of 2222 target and 8220 filler images were sampled from the SUN dataset (Xiao et al., 2010) and annotated by 665 participants on the Amazon Mechanical Turk. Filler images where inserted between the repetition of the target images. Each target image was presented exactly twice, with a gap 91-109 images, while the fillers only once, or twice, with a gap 1-7 images, for the 'vigilance' images.

The 'vigilance' images are known to participants, who must acknowledge them every time they are presented. This is to keep the participant attentive to the task. The game also detects false positives - images identified as repeats but never presented.

In follow-up work, Khosla et al. (2015) noticed a decline in the memorability scores as the gap between the target repeats increased. This memorability decline affected all images by a similar amount, thus preserving the memorability rank ordering. Based on this observation, Khosla et al. (2015) updated the memory game protocol to compensate for the repeat delay, which allowed to dynamically change the repeat gap from 30 images up to 100 and still produce accurate, absolute memorability scores. The game flow is shown in Figure 2.2. Subsequently, the authors collected a new dataset LaMem with close to 60000 memorability annotated images. Figure 2.3 shows examples of images from this dataset with low, medium and high memorability scores.

To study the visual memory retention over a much longer interval (Goetschalckx et al., 2018), conducted a study with repeat delays reaching up to 20 minutes, one day, and one week instead of the typical ∼5 minutes intervals. The authors reported image memorability scores in line with the previous work, consistent across observers and stable over time.

2.2 Prior Work on Image Memorability Estimation

In a pioneering work on image memorability, Isola et al. (2011b) and Isola et al. (2011a) demonstrated that the ability of our cognition system to remember specific images and forget others is congruent among independent observers, despite considerable variability in the image content. The authors reached a consensus that memorability is a stable property intrinsic to images. Based on this premise, Isola et al. (2011b) investigated factors giving rise to the image memorability effects and subsequently applied them to memorability prediction with a machine learning method. The machine learning method was based on a mixture of global image features GIST (Oliva and Torralba, 2001), Scale Invariant Feature Transform (SIFT) (Lowe, 2004), Histogram of Oriented Gradients (HOG) (Dalal and Triggs, 2005), Structural Similarity Index Measure (SSIM) (Shechtman and Irani, 2007), and pixel histograms. To further improve the memorability prediction with computational

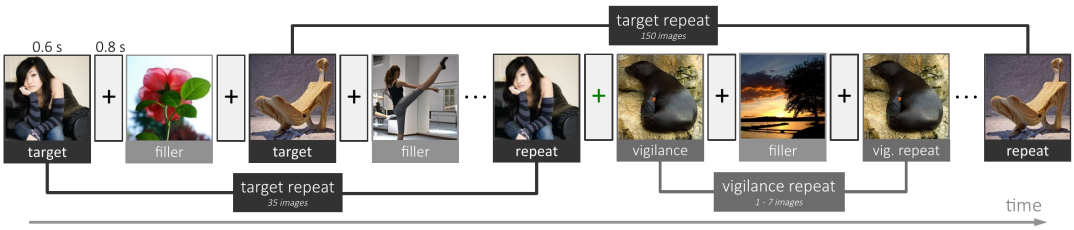


FIGURE 2.2: Memory game for the memorability annotation procedure. Sourced with permissions from Khosla et al. (2015).

FIGURE 2.3: Example images from the LaMem dataset with different high, medium and low memorability scores (shown bellow images).

methods, researchers analyzed the relationship between memorability and various visual features (Khosla et al., 2012), image classes (Isola et al., 2014) and saliency (Dubey et al., 2015). Bylinskii et al. (2015) conducted several experiments to better understand the intrinsic and extrinsic effects on image memorability, concluding that the primary memorability is grounded in the intrinsic properties of images and all extrinsic effects contribute only marginally.

Deep learning was first applied to the memorability problem by Baveye et al. (2016), who proposed a MemoNet model based on GoogLeNet (Szegedy et al., 2015) trained on the ImageNet (Russakovsky et al., 2015) dataset. Zarezadeh et al. (2017) used CNN features with Support Vector Regression (SVR) (Drucker et al., 1997) to predict memorability with accuracy comparable to MemoNet (Baveye et al., 2016).

To achieve higher accuracy with deep learning techniques, Khosla et al. (2015) annotated a large memorability dataset LaMem with 60k images and introduced deep learning model MemNet. This model is based on the Hybrid-CNN, which is the AlexNet CNN (Krizhevsky et al., 2012) pre-trained on the ImageNet (Russakovsky et al., 2015) and the Places (Zhou et al., 2014) datasets (∼3.6 million images in total). The MemNet, trained on the LaMem dataset, established a new state of the art for image memorability learning and inference and became a "de facto" standard model for this task.

Researchers also tried to improve memorability prediction by other techniques,

such as the adaptive transfer learning from external sources (Jing et al., 2017) or predicting image memorability by multi-view adaptive regression (Peng et al., 2015), none exceeding the performance of the MemNet (Khosla et al., 2015).

The relationship between the visual saliency, attention, and memorability was already suggested by Isola et al. (2011b) but was not further investigated. Mancas and Le Meur (2013) studied this link and found that the most memorable images have uniquely localized regions, while less memorable either do not have distinct regions of interest or have several of them. Based on these findings, Mancas and Le Meur (2013) devised new attention-related features that improved the memorability prediction by 2% compared to the non-attention-based models reported on by Isola et al. (2011b). In a similar work, Celikkale et al. (2013) applied an attention-driven spatial pooling based on SIFT (Lowe, 2004) and HOG (Dalal and Triggs, 2005) features and bottom-up and object-level saliency detectors. Their results, albeit only moderate, still indicate a benefit of the attention-based approach. Importance of the memorability regions was explored by Khosla et al. (2012) who introduced the concept of attention maps that relate image regions to memorability. These maps are learned directly as clusters of gradients, textures, and color features with the Support Vector Machine (SVM) solver (Joachims, 2006), with results highlighting the benefits of the attention function on memorability prediction.

2.2.1 Current State-of-the-Art, its Limitations and Promising Research Directions

The current state-of-the-art method for image memorability learning and estimation with machine learning is MemNet (Khosla et al., 2015). This method is based on a CNN pre-trained on the ImageNet and Places datasets and fine-tuned on the LaMem dataset. The high performance of this method can be attributed to a deep neural model trained end-to-end on a large annotated dataset.

The primary limitation of all current approaches lies in their inherent lack of provision for interpretability of the underlying causes of the image memorability. Our work partially addresses this limitation by visualizing learned attention weights, which highlights the contrast between more and less memorable regions in the images.

An inability of the current methods to predict a confidence value for the estimated memorability score could also be considered a limitation. We address this problem in our Future Work in Section 2.9.1.

As highlighted by the work of Khosla et al. (2015), a promising research direction appears to be a collection of even larger annotated image memorability datasets. In addition to the seen/not seen per-image labels created during the manual data collection, the annotation could be extended for other meta-data, for example, electroencephalogram (EEG) data. The EEG has been shown to measure brain activity highly correlated with the seen/not seen images (Stothart et al., 2021). Another interesting research direction, which we follow in our work, is to investigate a correlation

between the memorability score of an image and its spatial regions. Intuitively, not all places in an image are equally memorable, therefore utilizing machine learning to localize these regions and their sequence as a function of the image memorability should improve the model performance.

2.3 Method

Prior art methods learn the image memorability score as a function of the entire image area, that is, consider all places in the scene with equal importance. Our visual system, on the other hand, perceives the environment as a sequence of eye fixations. This work draws inspiration from the neuroscience underlying this mechanism and proposes a machine learning method that learns a sequence of image regions whose content maximizes the ground truth memorability score.

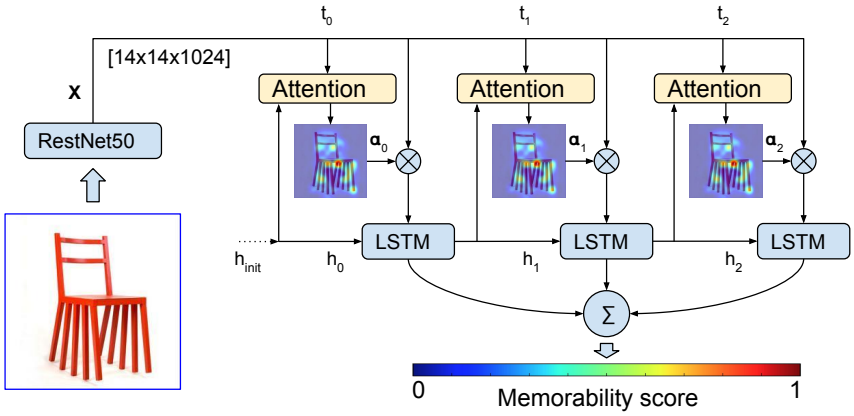


FIGURE 2.4: AMNet architecture. Memorability is learned and estimated over three recurrent steps t_0, t_1, t_2, each focusing on different image region localized by learned attention $\boldsymbol{\alpha_0}, \boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}$.

The idea behind the proposed AMNet (Attention for Memorability estimation Network) method (Figure 2.4) is based on four main components: a deep CNN, trained on a large-scale image classification task, a visual soft attention network and a LSTM (Hochreiter and Schmidhuber, 1997) recurrent neural network, followed by a fully connected neural network for memorability score regression.

2.3.1 Contributions to the Field of Neural Architectures

The key novel elements of the AMNet neural architecture are as follows:

- Attention layer over the spatial domain of CNN feature maps learned as a function of CNN features and previous LSTM state.
- Training objective encouraging high attention to a tight image region at each LSTM step and diverse among all LSTM steps.
- A regression layer over discrete memorabilities estimated at each LSTM step.

To the best of our knowledge, the presented approach has not been attempted before.

In the following sections we introduce the core principles behind the transfer learning, soft attention and the LSTM network for memorability regression. Finally, we outline the training procedure and finish with the data augmentation process and the AMNet implementation details.

2.3.2 Transfer Learning for Memorability Estimation

It is common practice to use a pre-trained CNN as a fixed feature extractor or to fine tune it for a similar application (Sharif Razavian et al., 2014), primarily to reduce training time and overfitting on tasks with small datasets.

This technique is readily applied to computer vision problems centered around semantic features, such as objects detection and segmentation. However, little is known about the transfer learning without fine tuning for the image memorability estimation since there is no clear understanding of what visual features trigger the effects of remembering and forgetting.

Khosla et al. (2015) has already shown the benefits of fine tuning deep CNN models for this domain. Rather than fine tuning the CNN, we propose to use a much deeper model as a fixed feature extractor. In this setup the CNN is trained on image classification datasets but not updated during the training on image memorability datasets. Our results show that the features optimized for image classification are also highly suitable for the memorability task. In our work we use ResNet50 (He et al., 2016) CNN trained on the ImageNet dataset with top 1 error 24.7%.

2.3.3 Soft Attention Mechanism

The ability of a neural network to learn which discrete information elements to focus on within a given training sample was first applied to machine translation by Bahdanau et al. (2014). This mechanism is called soft attention due to the fact that it produces a probability weight for every information element rather than a hard, binary decision boundary. The benefit of the soft attention method is that it is a fully differentiable function and as such it can be learned end-to-end with gradient based optimization methods.

The soft attention mechanism has two components, a network that learns probabilities for each information element within the input data space and a gating function that uses these probabilities to weight the input data for further processing.

An alternative technique to the soft attention, not applied in this work, is called hard attention which produces binary decisions over the input space. The hard attention function is not smooth - it has zero gradients over its entire range. As such, it cannot be directly trained with the backpropagation but rather with other methods like genetic algorithms or reinforcement learning such as the REINFORCE method (Williams, 1992).

2.3.4 Long Short Term Memory Recurrent Network

The AMNet models the memorability regression as a sequence of, disjoint, conditional predictions $p(a_t|a_{t-1})$, where a_t indicates an attention map over latent image representation $\mathbf{z}$. For this computation we employ the LSTM recurrent network (Hochreiter and Schmidhuber, 1997). Experimentally we found that the LSTM performs marginally better than the vanilla Recurrent Neural Network (RNN), particularly over longer (up to 6 steps) sequences. The best performance was, however, observed at around 3 steps $T = 3$, where the LSTM still outperforms the vanilla RNN.

The vanilla RNN cell is defined as a typical neuron with added feedback connection. The feedback carries a hidden state information $\mathbf{h}$ from step t over to $t+1$. The RNN cell is defined as:

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}), \tag{2.1}$$

where $\mathbf{W}$ and $\mathbf{U}$ are weight matrices for the input $\mathbf{x}_t$ and the hidden state from the previous step $\mathbf{h}_{t-1}$ respectively. Unlike the simple RNN cell, the LSTM introduces several gates to regulate data flow and memorization through the node. This is important to avoid vanishing/exploding gradients (Hochreiter et al., 2001a) over long range sequences, nevertheless, we found it to be beneficial even for very short sequences.

There are three gates: Forget gate (Eq. 2.2), which interrupts propagation of the cell state $\mathbf{c}_t$ over to the next step. Input gate (Eq. 2.3) allows or blocks accumulation of new inbound data $\mathbf{x}_t$ in the cell state $\mathbf{c}_t$. Output gate (Eq. 2.4) then regulates the value of the hidden state $\mathbf{h}_t$ on the LSTM output. The internal structure of the LSTM cell is shown in Figure 2.5.

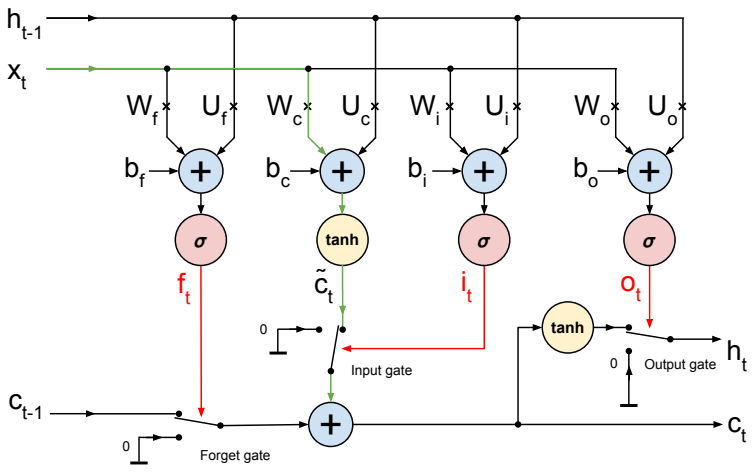


FIGURE 2.5: LSTM unit. Red color indicates the forget, input and output gate signals. The input x gets integrated with the cell state along the green path. The gate switches are implemented as elementwise multiplications.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad \text{forget gate} \tag{2.2}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad \text{input gate} \tag{2.3}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad \text{output gate} \tag{2.4}$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \quad \text{memory state activation} \tag{2.5}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t, \quad \text{memory state update} \tag{2.6}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t), \quad \text{hidden state update} \tag{2.7}$$

Each gate is controlled by its own single layer network with σ sigmoid activation function. $\mathbf{W}$ and $\mathbf{U}$ are weight matrices for the input $\mathbf{x}$ and hidden state $\mathbf{h}_{t-1}$ of their respective gates. The $\mathbf{b}$ are bias vectors, and $\circ$ denotes elementwise multiplication. The LSTM unit, as commonly being referred to is in fact a layer. The input $\mathbf{x}$, hidden state $\mathbf{h}$ and cell state $\mathbf{c}$ are vectors. The switches, shown in Figure 2.5, are only symbolic. Rather then taking the on/off states they modulate their output between zero and full input value. The only path along which the input data $\mathbf{x}$ enter the cell memory is via the input gate controlled by the $\mathbf{i}_t$ signal.

2.3.5 AMNet Model

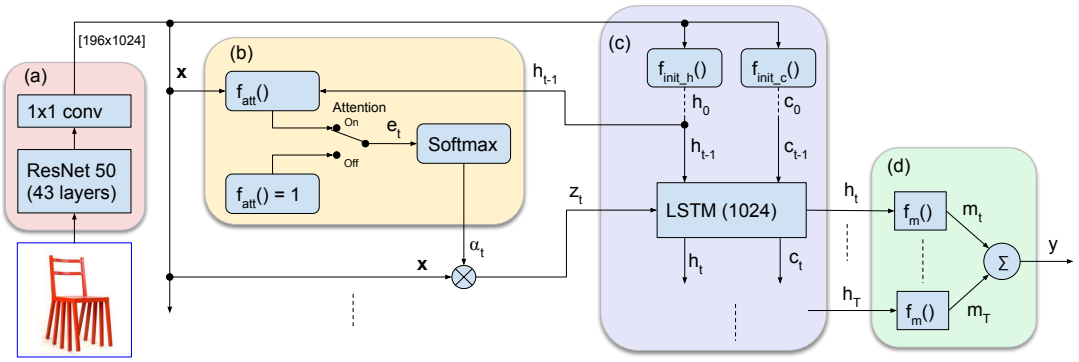


FIGURE 2.6: A pre-trained RestNet50 (a) is followed by the soft attention mechanism (b) with LSTM (c), which over a sequence of three steps $T = 3$ produces attention maps, each conditioned on the previous LSTM state $\mathbf{h}_{t-1}$ and the entire image feature vector $\mathbf{x}$. Memorability y is then calculated as a sum of discrete memorability scores in the regression network (d).

The AMNet model estimates the image memorability by taking a single image $\boldsymbol{X}$ and generating a memorability score y.

$$y = f(\boldsymbol{X}), \quad y = [0, 1]. \tag{2.8}$$

The process of memorability estimation is summarized in Algorithm 1.

Formally, the image features, extracted by the front-end CNN, from a tensor with dimensions (W, H, D), where W and H represent the spatial resolution, while D a length of feature vectors, one for each location within the (W, H) region. Specifically, in the case of AMNet the feature tensor has dimensions $14 \times 14 \times 1024$. The feature

Algorithm 1 AMNet algorithm

procedure MEMORABILITY($\boldsymbol{X}$) ▷ $y = f(\boldsymbol{X})$
  $\mathbf{x} = get\_cnn\_features(\boldsymbol{X})$ ▷ Output of the 43^{rd} layer of the ResNet50 CNN.
  $\mathbf{h}_0 = f_{init_c}(\mathbf{x})$ ▷ Eq. 2.19
  $\mathbf{c}_0 = f_{init_h}(\mathbf{x})$ ▷ Eq. 2.19
  $lstm\_init(\mathbf{h}_0, \mathbf{c}_0)$
  $y = 0$
  for $t = 0$ to T **do** ▷ at $t = 0 \rightarrow \mathbf{h}_t = \mathbf{h}_0$
    $\mathbf{e} = f_{att}(\mathbf{x}, \mathbf{h}_t)$ ▷ Eq. 2.15
    $\boldsymbol{\alpha} = softmax(\mathbf{e})$ ▷ Eq. 3.3
    $\mathbf{z} = []$
    for $i = 0$ to L **do** ▷ for all locations, Eq. 2.11
      $\mathbf{z} = \mathbf{z} + \alpha_i \mathbf{x}_i$ ▷ $\mathbf{z} \in \mathbb{R}^D$
    $\mathbf{h}_t, \mathbf{c}_t = lstm\_step(\mathbf{z}, \mathbf{h}_t, \mathbf{c}_t)$ ▷ Eq. 2.10
    $y = y + f_m(\mathbf{h}_t)$ ▷ Eq. 2.18
  return y ▷ Predicted memorability score $y \in [0, 1]$

maps are vectorized to $L = W \times H$, D-dimensional feature vectors $\mathbf{x}$:

$$\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_L\}, \ \mathbf{x}_i \in \mathbb{R}^D. \tag{2.9}$$

All vectors are column vectors, unless stated otherwise. The memorability is estimated with LSTM over a three steps long sequence $T = 3$. The LSTM updates the hidden state given $\mathbf{h}_{t-1}$ and latent image representation $\mathbf{z}_t$. It is defined as:

$$\mathbf{h}_t = \phi(\mathbf{h}_{t-1}, \mathbf{z}_t), \ t \in [0, T), \ t \in \mathbb{Z}, \ \mathbf{h}_t \in \mathbb{R}^B, \tag{2.10}$$

where $\mathbf{h}_t$ is the LSTM state at time t with size $B = 1024$. The vector $\mathbf{z}_t$ represents a new image features produced at step t, after attention weights $\boldsymbol{\alpha}^t$ are applied to the image features $\mathbf{x}$, and it is calculated as a simple weighted sum such that:

$$\mathbf{z}_t = \sum_{i=1}^{L} \alpha_{t,i} \mathbf{x}_i, \ \mathbf{z}_t \in \mathbb{R}^D, \tag{2.11}$$

where $\boldsymbol{\alpha}$ are the attention probabilities conditioned on the entire image feature vector $\mathbf{x}$ and previous LSTM hidden state $\mathbf{h}_{t-1}$.

$$\boldsymbol{\alpha}_t \sim p(\boldsymbol{\alpha}_t | \mathbf{x}, \mathbf{h}_{t-1}), \ \boldsymbol{\alpha}_t \in \mathbb{R}^L. \tag{2.12}$$

The attention weights are parameterised with neural networks. The attention is then represented as a vector of weights produced by the softmax function:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{L} \exp(e_{t,k})}. \tag{2.13}$$

The attention weights vector $\mathbf{e}_t$ is a product of the image feature vector $\mathbf{x}$ and the LSTM hidden state $\mathbf{h}_{t-1}$:

$$e_{t,i} = f_{att}(\mathbf{x}_i, \mathbf{h}_{t-1}), \tag{2.14}$$

where $f_{att}()$ is s simple sum of two affine transformations followed by logistic function

$$f_{att}(\mathbf{x}_i, \mathbf{h}_{t-1}) = \boldsymbol{M}_i \tanh(\boldsymbol{U}\mathbf{h}_{t-1} + \boldsymbol{K}\mathbf{x}_i + \mathbf{b}), \tag{2.15}$$

where $\boldsymbol{M} \in \mathbb{R}^{L\times D}, \boldsymbol{U} \in \mathbb{R}^{D\times B}, \boldsymbol{K} \in \mathbb{R}^{D\times D}$ and $\mathbf{b} \in \mathbb{R}^{D\times 1}$ are network weights and biases respectively, estimated together with other parameters of the network during optimization. The hyperblolic tangent fucntion $\tanh()$ is applied elementwise.

In order to experiment with the effects of the attention we can conditionally disable it by defining the $f_{att}()$ as a constant function with unit output such that:

$$f_{att}(\mathbf{x}_i, \mathbf{h}_{t-1}) = 1. \tag{2.16}$$

This results in all feature vectors in $\mathbf{x}$ being considered equally thus disabling the attention mechanism. At each step t the network produces one discrete memorability score m_t calculated as:

$$m_t = f_m(\mathbf{h}_t). \tag{2.17}$$

The function $f_m()$ maps the LSTM hidden state $\mathbf{h}_t$ to the memorability score $m_t = [0, 1]$. It is implemented as a two-layer neural network for regression with a single output neuron and linear activation function. Finally, the total image memorability score y is calculated as a sum of the discrete memorabilities m_t

$$y = \sum_t^T m_t, \;\; y = [0, 1], \; y \in \mathbb{R}, \tag{2.18}$$

In the first step, the LSTM hidden $\mathbf{h}_0$ and memory $\mathbf{c}_0$ states are initialized from the image feature vector $\mathbf{x}$ as follows:

$$\mathbf{c}_0 = f_{init_c}\left(\frac{1}{L}\sum_i^L \mathbf{x}_i\right), \quad \mathbf{h}_0 = f_{init_h}\left(\frac{1}{L}\sum_i^L \mathbf{x}_i\right), \tag{2.19}$$

where the $f_{init}()$ functions are single, fully connected neural networks with $\tanh()$ activation.

Dimensions of the CNN features were given by the CNN front-end model (ResNet). Dimensions and the number of hidden layers were selected empirically and verified and refined on experiments with the training and validation datasets. All hyper-parameters such as λ, the number of LSTM steps, training epochs, learning rate, dropout and ℓ_2 weights regularization, were selected experimentally, also on the training and validation datasets.

2.3.6 Training Procedure

The AMNet model is trained by minimizing the following composite loss function:

$$\mathcal{L} = (\hat{y} - y)^2 + \lambda \mathcal{L}_\alpha. \tag{2.20}$$

The first term represents a mean squared error between the ground truth $\hat{y}$ and predicted image memorability score y. In order to encourage the attention model to explore all image regions over all time steps, we add a second term $\lambda \mathcal{L}_{\boldsymbol{\alpha}}$ which performs a joint ℓ_1,ℓ_2 penalty as a function of activations of all attention maps in the LSTM sequence T. A similar method was introduced by Xu et al. (2015). The hyperparameter λ specifies the impact of the penalty

$$\mathcal{L}_\alpha = \sum_i^L s_i^2, \tag{2.21}$$

where s_i denotes the ℓ_1 penalty and L the total number of all locations in the activation map. s_i is defined as:

$$s_i = 1 - \sum_t^T \alpha_{t,i}, \tag{2.22}$$

which enforces sparsity along the sequence dimension T. In other words, it encourages a strong activation for only one of the attention maps at location i. Finally, the ℓ_2 penalty in the form of $\sum_i s_i^2$ in Eq. 2.21 further promotes an even distribution of activations over all locations. The value of the λ parameter was experimentally determined on the validation dataset as 10^{-4} for which the network achieved the highest performance.

The entire model is fully differentiable and trained end-to-end with the ADAM (Kingma and Ba, 2015) optimizer with a fixed learning rate 10^{-3}. The input image feature vector $\mathbf{x}$ is extracted from the 43^{rd} layer of the RestNet50 (He et al., 2016) with dimensions $[14 \times 14 \times 1024]$. The ResNet50 is trained for image classification on the ImageNet dataset and its weights are not updated during the AMNet training.

The AMNet network is heavily regularized with dropout and small ℓ_2 weights penalty 10^{-6}. We found that the dropout was critical to stop the network from overfitting. The training was carried out in minibatches of 256 images and terminated by early stopping when the observed Spearman's rank correlation on the validation dataset reached its maximum, which was between epoch 30 and 50 depending on the split and the training dataset (LaMem or SUN). Training and validation losses as well as the memorability rank correlation on the validation dataset in the LaMem, split 1 is showin in Figure 2.7.

2.3.7 Data Preprocessing and Augmentation

Common augmentation techniques were applied to the images during the training stage to reduce overfitting and improve generalization. These augmentations were:

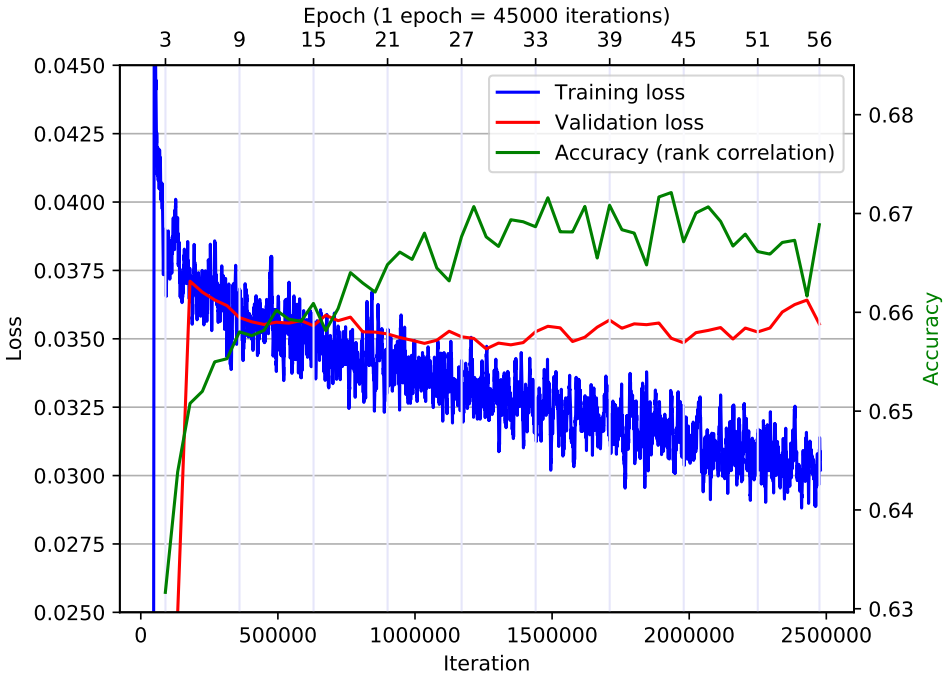


FIGURE 2.7: Training and validation losses and memorability rank correlation on the LaMem validation dataset split1.

a crop of random size between 0.08 and 1.0 of the original image, a random aspect ratio between 3/4 and 4/3 of the original aspect ratio, resize to 224×224, and random horizontal flip. For the evaluation only a center crop 224×224 was selected as the input.

Memorability scores in the LaMem dataset are in the range $[0, 1]$ with distribution shown in Figure 2.8. For the training purpose the memorability scores were zero mean centered and scaled to range $[-1, 1]$.

2.4 Experimental Results

In this sections we evaluate the AMNet on the LaMem (Khosla et al., 2015) and SUN Memorability (Isola et al., 2011b) datasets. First, we briefly describe the datasets and the evaluation metrics, and then present our qualitative and quantitative results with the comparison against the state of the art methods.

2.4.1 Datasets

Main focus of this research work is on the LaMem (Khosla et al., 2015) dataset due to its large size, making it suitable for training deep neural networks. The LaMem[2] is the largest annotated image memorability dataset to this date with total

[2] http://memorability.csail.mit.edu/download.html

of 58741 images. The images cover a wide range of indoor and outdoor environments, objects and people that were obtained from other labeled datasets such as MIR Flickr (Huiskes et al., 2010), AVA dataset (Murray et al., 2012), Affective Images dataset (Machajdik and Hanbury, 2010), image saliency datasets (Judd et al., 2009, Ramanathan et al., 2010), SUN dataset (Xiao et al., 2010), Image popularity dataset (Khosla et al., 2014), Abnormal Objects dataset (Saleh et al., 2013) and the Pascal dataset (Farhadi et al., 2009). The memorability scores were collected manually on the Amazon Mechanical Turk (AMT) by means of a memorability game introduced by Isola et al. (2011b) and improved by Khosla et al. (2015). Approximately 80 measurements (memorable=yes/no) were collected per image. There are 5 random splits, each with 45000 images for training, 3741 for evaluation and 10000 for testing.

As a second dataset for evaluation we chose the SUN Memorability dataset pioneered by Isola et al. (2011b). There are 2222 images in total, originating from the SUN (Xiao et al., 2010) dataset with memorability scores collected by method similar to the LaMem. There are 25 random splits with equal number of 1111 images for training and testing.

2.4.2 Evaluation Metrics

Following the previous work, we report on the performance in terms of rank correlation, specifically the Spearman's rank correlation coefficient (Pirie, 1988) ρ and Mean Squared Error (MSE).

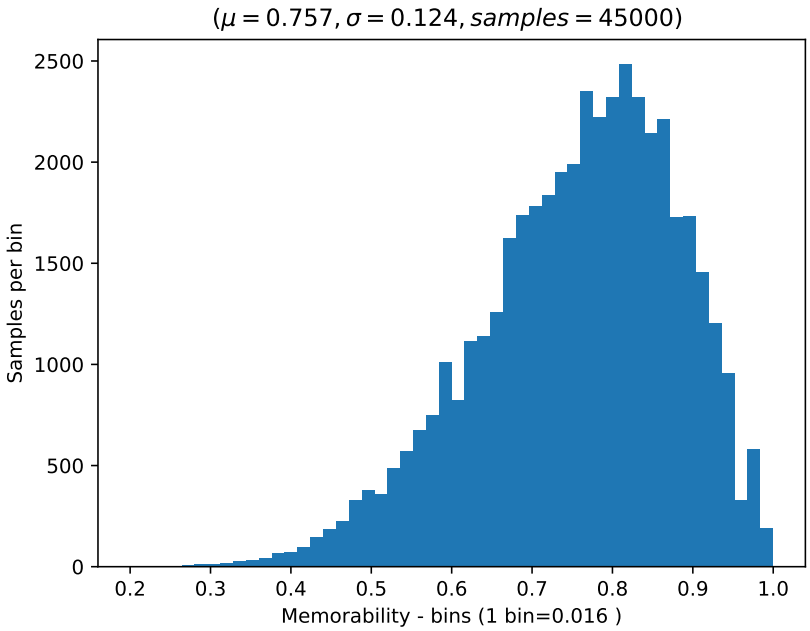


FIGURE 2.8: Histogram of ground truth memorability scores in the LaMem (Khosla et al., 2015) training dataset split 1.

The Spearman's rank correlation coefficient measures consistency between the predicted and ground truth ranking, within the range $[-1, +1]$, where zero represents no correlation. Higher ρ values indicate higher memorability prediction accuracy. The Spearman's rank correlation is calculated as follows:

$$\rho_s(\hat{r}, r) = 1 - \frac{6\sum_i^N (\hat{r}_i - r_i)^2}{N(N^2 - 1)}, \tag{2.23}$$

where N is a number of samples, $\hat{r}_i$ is a rank of the i^{th} ground truth memorability score, and r_i the i^{th} predicted value.

MSE is used as a secondary metric, not always presented in prior publications. The Spearman's rank correlation shows a monotonic relationships between the reference and observation but does not reflect the absolute numerical errors between them, which is then presented by the MSE according to:

$$\text{MSE}(\hat{y}, y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \tag{2.24}$$

where $\hat{y}_i$ is the ground truth memorability score, while y_i the memorability prediction and N the number of tested samples.

2.4.3 Performance Evaluation

In order to obtain results that are fully comparable with the previous work, we used the same training and evaluation protocol as in Khosla et al. (2015) for the LaMem dataset and in Isola et al. (2011b) for the SUN memorability dataset.

Evaluation on the LaMem dataset was performed by training five models, one on each of the five random splits. The models are then evaluated on the corresponding five test datasets and the average memorability rank correlation and average MSE are reported.

In Table 2.1 we show that the AMNet model with the active attention achieves $\rho = 0.677$, or a 5.8% improvement over the best known method MemNet (Khosla et al., 2015). Even without attention the AMNet outperforms the prior work by 3.6% which demonstrates that the pre-trained, deep CNN with our recurrent and

TABLE 2.1: Average Spearman's rank correlation ρ and MSE over 5 test splits of the LaMem dataset.

Method (on LaMem dataset)	$\rho \uparrow$	MSE $\downarrow$
AMNet	**0.677**	**0.0082**
AMNet (no attention)	0.663	0.0085
MemNet (Khosla et al., 2015)	0.64	N/A
CNN-MTLES (Jing et al., 2017) (different train/test (50/50) split)	0.5025	N/A

TABLE 2.2: Evaluation on the SUN Memorability dataset. All models were trained and tested on the 25 train/val splits.

Method (SUN Memorability dataset)	$\rho \uparrow$	MSE $\downarrow$
GIST,SIFT,HOG2x2,SSIM + SVR (Isola et al., 2011b)	0.462	0.017
SIFT,HOG,SSIM,Pixel hist. + SVR (Mancas and Le Meur, 2013)	0.479	NA
AMNet	**0.649**	**0.011**
AMNet (no attention)	0.62	0.012
MemNet (Khosla et al., 2015)	0.63	NA
MemoNet 30k (Baveye et al., 2016)	0.636	0.012
Hybrid-CNN+SVR (Zarezadeh et al., 2017)	0.6202	0.013

regression network layers still achieves high accuracy. The comparatively low performance of the CNN-MTLES (Jing et al., 2017) method can be attributed to the fact that the model uses various, specifically engineered visual features in combination with features extracted from CNN networks trained on ImageNet (Russakovsky et al., 2015) and Places (Zhou et al., 2014) datasets. Thus, it does not leverage the end-to-end deep learning procedure. The CNN-MTLES, however, uses the LaMem dataset, which indicates that even a large dataset does not significantly improve the performance of models based on engineered, visual features.

To train the deep AMNet model on the rather small SUN dataset we had to increase regularization to avoid overfitting. We found that in this specific case $\ell_2 = 10^{-4}$ weights regularization performed better than a stronger dropout or the combination of both. Table 2.2 shows that the AMNet with attention performs 2% better than the current best model. By disabling the attention the performance declined to $\rho = 0.62$, demonstrating the advantage of the visual attention for this task.

We found that during training the MSE on the validation datasets followed a similar trend with the rank correlation ρ, however the ρ peaked after the model started overfitting as apparent in Figure 2.7. It is conceivable to assume that the slightly higher variance at the maximum ρ improves generalization in terms of the predicted and ground truth monotonic relationships, even though MSE starts increasing. For example, during the training on the LaMem split 1, as shown in Figure 2.7, we attained maximum $\rho = 0.6721$ and MSE $= 0.00848$ while $\rho = 0.6676$ for minimum MSE $= 0.00844$.

Tables 2.1 and 2.2 show that the AMNet exhibits the best performance in terms of the Spearman's rank correlation as well as MSE on both, the LaMem and the SUN datasets. The best performance attains $\rho = 0.677$ on the LaMem dataset, approaching 99.6% of the human performance $\rho = 0.68$ as measured by Khosla et al. (2015). Comparison against the state of the art can be seeing in Figure 2.9.

2.4.4 Ablation Study

To see the impact of the attention mechanism and the recurrent network on the memorability estimation we conducted several ablation tests.

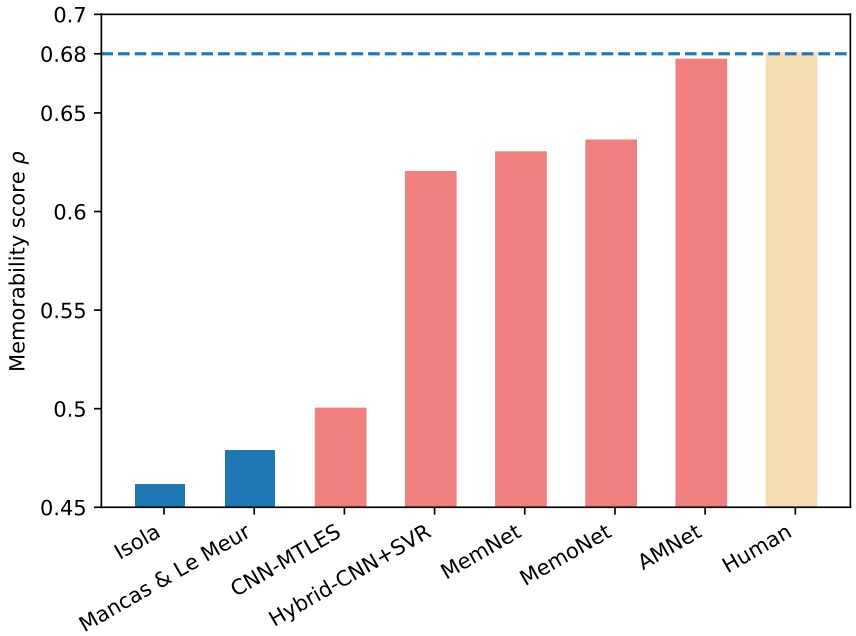


FIGURE 2.9: Comparison against the state of the art methods. Red depicts deep learning based methods. AMNet, MemNet and CNN-MTLES where trained on the LaMem, the rest on the SUN Memorability dataset.

First, we disabled the recurrent network by allowing only a single LSTM step and then trained and evaluated the network according to the protocol given above (5 models over 5 train/test splits). In Table 2.3, third row (Attention, no LSTM) we can see that the rank correlation dropped only by $\Delta\rho = -0.008$. In another experiment, we used only the final output from the LSTM at the step $t = T$ before regressing the memorability score in the layer $f_m()$ as:

$$y = f_m(\mathbf{h}_T). \tag{2.25}$$

With this setup the rank correlation declined by $\Delta\rho = -0.002$. While in both cases the network still attains comparatively high performance, we can see that the recurrent network does bring a positive contribution.

In the last row of the Table 2.3 we report results for a configuration where, in addition to the disabled LSTM, we also turn off the attention mechanism. In this case the image feature tensor $\mathbf{x}$ is passed through a single LSTM step to the regressor $f_m()$. As in the previous test, the LSTM is set to a singe iteration and the attention is disabled by setting all attention weights to one as per Eq. 2.16. The drop in the memorability rank correlation $\Delta\rho = -0.017$, indicates that a considerable performance gain can be attributed to the attention layer.

TABLE 2.3: Selected AMNet configurations evaluated on the LaMem training/test dataset split 1.

AMNet configuration	$\rho \uparrow$	MSE $\downarrow$
Attention, LSTM	**0.681**	**0.0081**
Attention, no LSTM	0.673	0.0082
Attention, LSTM, last step prediction	0.679	0.0081
No attention, no LSTM	0.664	0.0087

2.5 Impact of Feature Extractor Depth on Memorability Score

During an initial research on the transfer learning of deep models for the image memorability estimation, we found that shallower models, such as the ResNet18 (He et al., 2016), performed better than much deeper counterparts like the RestNet50. All these models were trained on the large scale image classification dataset ImageNet (Russakovsky et al., 2015) without fine tuning on the memorability datasets.

These results suggested that the feature hierarchies, learned within the very deep models, were quite specific for the composition of semantic descriptors, necessary to express the 1000 ImageNet classes. This made them less suitable to capture features underlying the image memorability. The first AMNet model was regularized by reduction of the network parameters in addition to the ℓ_2 weights and dropout regularizations. The number of trainable parameters, excluding the CNN for features extraction, was 9.5M.

Subsequently, we found that using much deeper models, such as ResNet50, but with stronger dropout regularization, resulted in a reversed effect, that is, the deeper models began performing better than the shallow ones. The number of the AMNet trainable parameters in this current model is 13M. This observation leads us to a conclusion that deep features are indeed suitable for the memorability task, however there is a need to combine them in a larger model to express the memorability effect.

Evaluation of the AMNet model with the VGG16 (Simonyan and Zisserman, 2015), ResNet18 and ResNet50 (He et al., 2016) front-ends is shown in Table 2.4. All results were obtained with visual attention and three LSTM steps.

TABLE 2.4: Comparison of the AMNet performance with feature extractors of different depths. Note that the networks were truncated at the convolution layers with feature maps with resolution 14×14.

Feature extractor	Trainable Params.	Spearman's rank correlation ρ on individual LaMem test splits					Avg. $\rho \uparrow$	Avg. MSE $\downarrow$
		1	2	3	4	5		
VGG16 (10 layers)	7.6M	0.657	0.644	0.664	0.658	0.649	0.654	0.009
ResNet18 (15 layers)	5.5M	0.663	0.650	0.667	0.666	0.653	0.660	0.009
ResNet50 (43 layers)	17M	**0.681**	**0.668**	**0.687**	**0.680**	**0.668**	**0.677**	**0.008**

2.6 Application to Image Aesthetics Estimation

To investigate how well our method generalizes to problems concerned with learning and estimating other image qualities, we conducted an experiment on the image aesthetics estimation.

As the image memorability, image aesthetics is an image property that can be measured and estimated with machine learning methods. Typically, the aesthetic quality is assigned to each image by number of human annotators and then translated to a single value range $[0..1]$ corresponding to low aesthetic on one end and high aesthetic quality on the other.

In our evaluation we focus on a challenging, large-scale Aesthetic Visual Analysis (AVA) dataset (Murray et al., 2012), commonly referenced by recent prior art methods. This dataset contains over 250k images, each being rated by at least 200 voters. The average rating is then assigned to each image as the aesthetic score. Examples of images with high, medium and low aesthetic scores are in Figure 2.10.

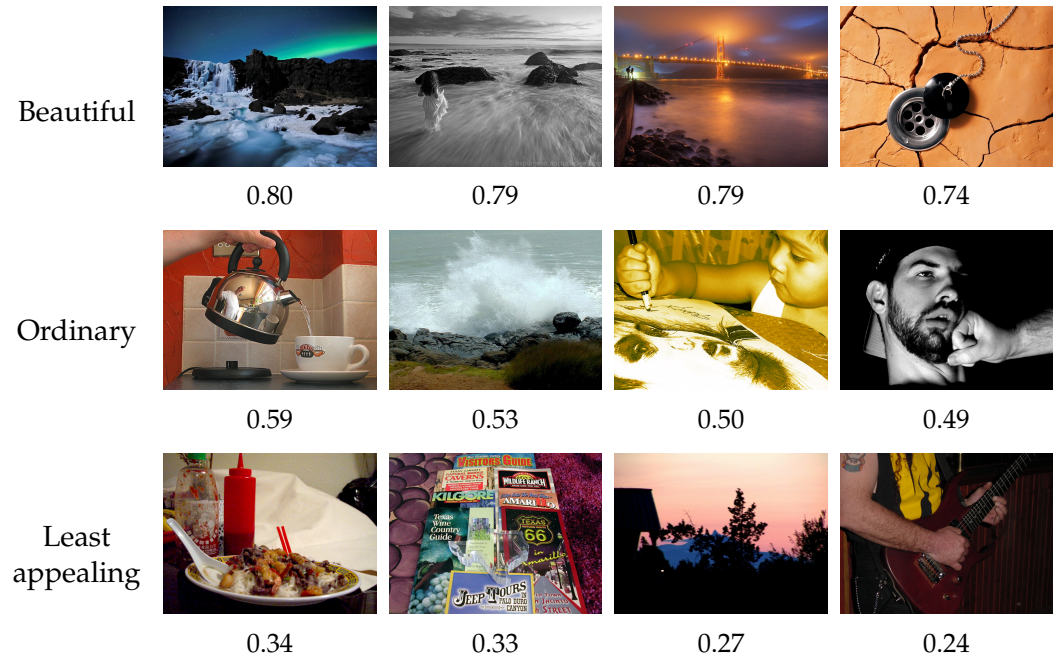


FIGURE 2.10: Examples of images from the AVA dataset with high, medium and low aesthetic scores.

To compare against the state of the art methods, we adapt a common evaluation protocol applied by Jin et al. (2016) and Kao et al. (2015). We split the total 255k images to 250k training and 5k test, randomly sampled images. This configuration is referred to as *RS-test* in Jin et al. (2016). Similar to the image memorability estimation, the model is trained with MSE loss with early stop to avoid overfitting. The AMNet model configuration is identical to the one used for the image memorability evaluation e.g. ResNet50, attention on, LSTM $T = 3$.

We compare against three state of the art methods. First method, proposed by Kao et al. (2015), leverages GIST features (Oliva and Torralba, 2001) with linear SVR

(Smola and Schölkopf, 2004). Second method, also by Kao et al. (2015), introduces a multi-task convolutional neural network (MTCNN) which performs regression for each AVA aesthetic group in a separate regression head. Finally, we compare against a method due to Jin et al. (2016) that employs the deep VGG16 (Simonyan and Zisserman, 2015) CNN network adopted for regression by replacing the last classification layer with a single output, regression layer. In Table 2.5 we compare the AMNet

TABLE 2.5: Comparison with the state of the art methods for image aesthetics prediction on the AVA dataset. AMNet trained for 97 epochs on train/test split 250k/5k without any tuning. Almost identical results were obtained when trained on 235k/20k split.

Method	MSE ↓
GIST+SVR (Kao et al., 2015)	0.522
MTCNN (Kao et al., 2015)	0.451
VGG16 for regression (Jin et al., 2016)	0.337
AMNet	**0.328**

performance against the state of the art methods. Even without any AMNet architectural modifications or training/fine-tuning the front end CNN feature extractor we obtained very high prediction scores.

2.7 The Role of Soft Attention Function on Memorability

The significant performance gain is achieved by the fact that the neural network learns to focus its attention to specific regions most relevant to memorability. The improvement is close to 2% on the LaMem and almost 5% on the SUN dataset. AMNet learns to explore the image content by producing three visual attention maps, each conditioned on the image content obtained by exploiting the previous map. We have experimented with 2,3,4,5 and 6 LSTM steps and found that three steps are sufficient to achieve the reported performance.

In order to better interpret the relation between attention maps and corresponding discrete memorability at each LSTM step, we present them as heat maps along with memorability scores. In Figure 2.11 we show selected images from the LaMem, test dataset, split 2. Images *(a)*, *(b)* and *(c)* have low memorability, image *(d)* a medium one and *(e)* and *(f)* high memorability. Images of the attention maps are produced by taking the output of the softmax function Eq. 3.3, scaling it to range $[0, 255]$ and then resizing it from 14×14 to 244×244.

As we can see in images *(a)*, *(c)* and *(d)* in Figure 2.11, most of the first attention weights gravitate towards the image center, which is most likely caused by the Center Bias, studied in Judd et al. (2009) and Zhang et al. (2008). This is typically attributed to the photographer bias (Tseng et al., 2009). In the subsequent LSTM steps, however, the attention moves mostly to the regions expected to be responsible for memorability.

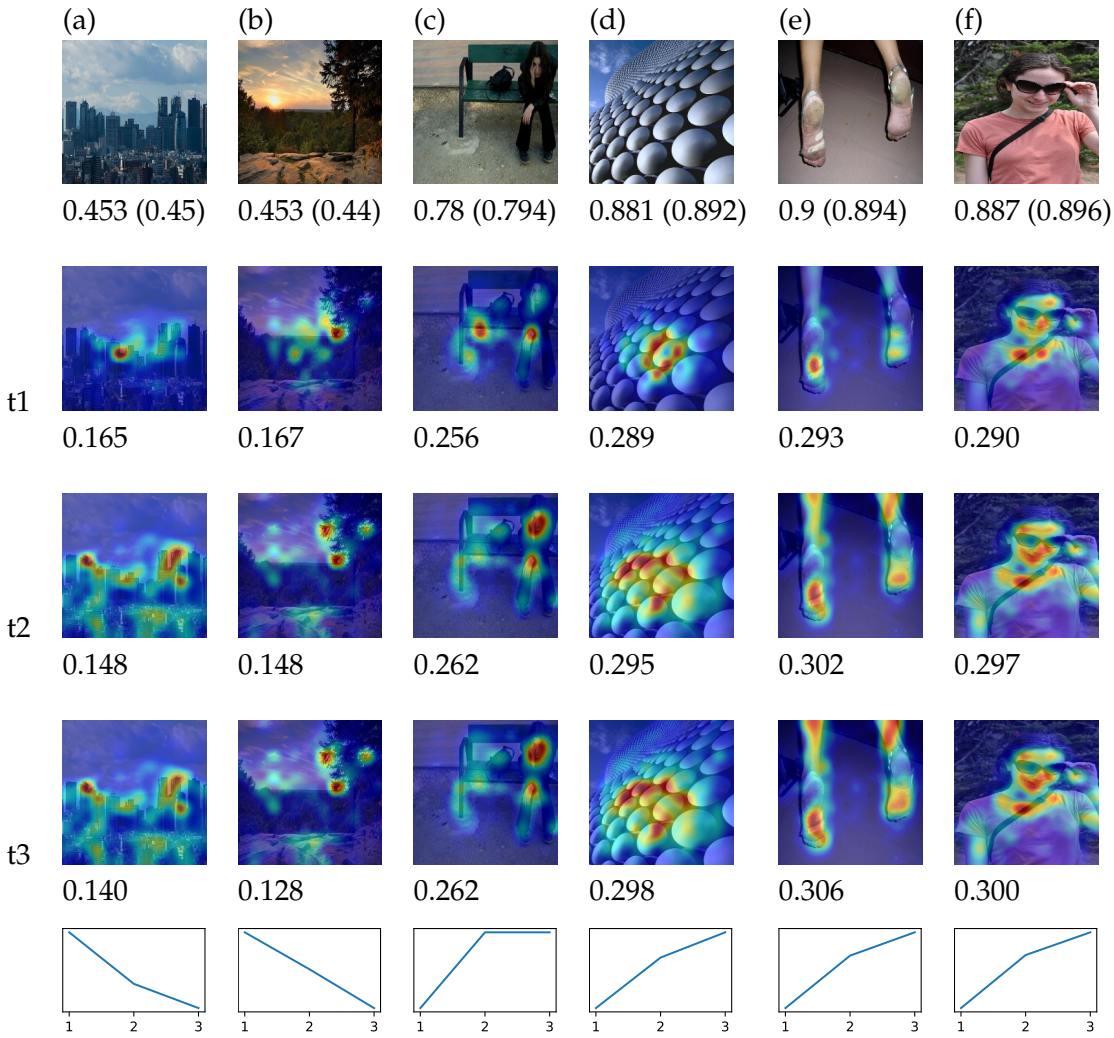


FIGURE 2.11: Examples of attention maps for low and high memorability images from LaMem test dataset split 2. Tested images, their estimated and ground truth memorabilities (in brackets) are shown in the top row. Bellow each image is a discrete memorability score estimated at the steps t_1, t_2 and t_3. Plots in the bottom row show gradients over the three LSTM steps.

After a close inspection, we found that the attention maps for low memorability images tend to be sparser with few small peaks, while for images with higher memorability, the attention maps display sharper focus spreading over larger regions around the activation peaks. The core image memorability usually originates in regions with people and human faces as is evident in images *(c)* and *(f)* shown in Figure 2.11.

Moreover, we found that the estimates of discrete memorability m_t in Eq. 2.17 decrease with each LSTM step t for low memorability images, while for high memorability images grow. This relation is shown in Figure 2.12. This effect is consistent within the LaMem test datasets across all splits and can be seen in Figure 2.11.

Initially, we experimented with an additional penalty function that encouraged

ascending or descending progression of the discrete memorabilities over the LSTM sequence. Unfortunately, in most cases this regularization led to a drop in the performance. This negative impact of the enforced memorability gradient over the regression sequence can be explained with an insight from the observed, learned gradients shown in Figure 2.12. The memorability regions that are learned at the first LSTM step appear to represent a mean image memorability. Over the following steps the model localizes regions leading to the final memorability. That is, for low final memorability, regions with lower residual memorabilities are identified while for final high memorability, regions with the above average are picked up.

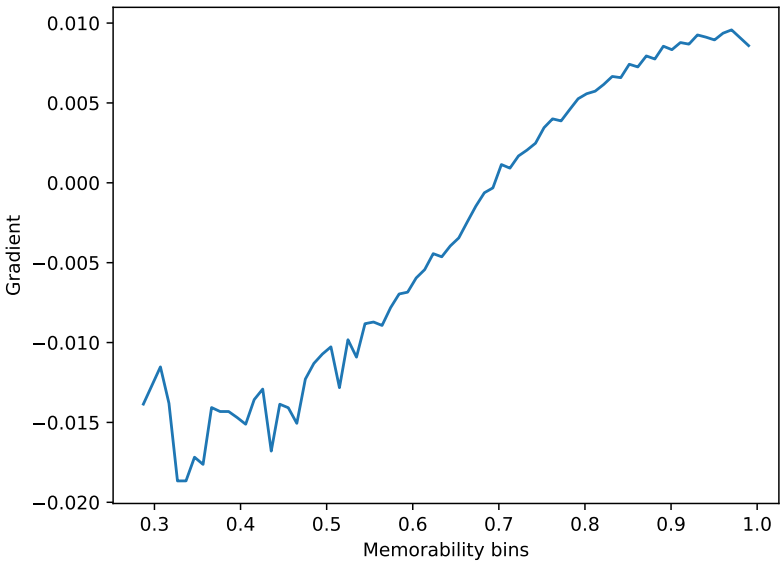


FIGURE 2.12: Histogram of gradients of discrete memorabilities over the LSTM steps. The gradient is directly proportional to the total image memorability.

To localize the memorability regions was already attempted in the prior work, most recently in Khosla et al. (2015) in the form of memorability maps. These memorability maps are produced as averaged activations of the last CNN feature maps before the regression layer. In Figure 2.13 we compare the attention maps with the MemNet memorability maps. The attention maps are sharper around the regions that we would intuitively expect to support the memorable image content. For instance, in image *(b)* attention localizes the man's face in the ball pit, which is what we would expect. The MemNet memorability map highlights only balls in the left bottom corner. Similarly, in the image *(e)*, the attention picks up a group of people on the river bank and a man on a sinking car as the regions underlying memorability of this image. The MemNet memorability map emphasises primarily regions around the trees in the background. However, our intuition behind the memorability of image regions is not entirely valid. As Isola et al. (2014) pointed out, the subjective judgement of the image memorability has a low correlation with the true values.

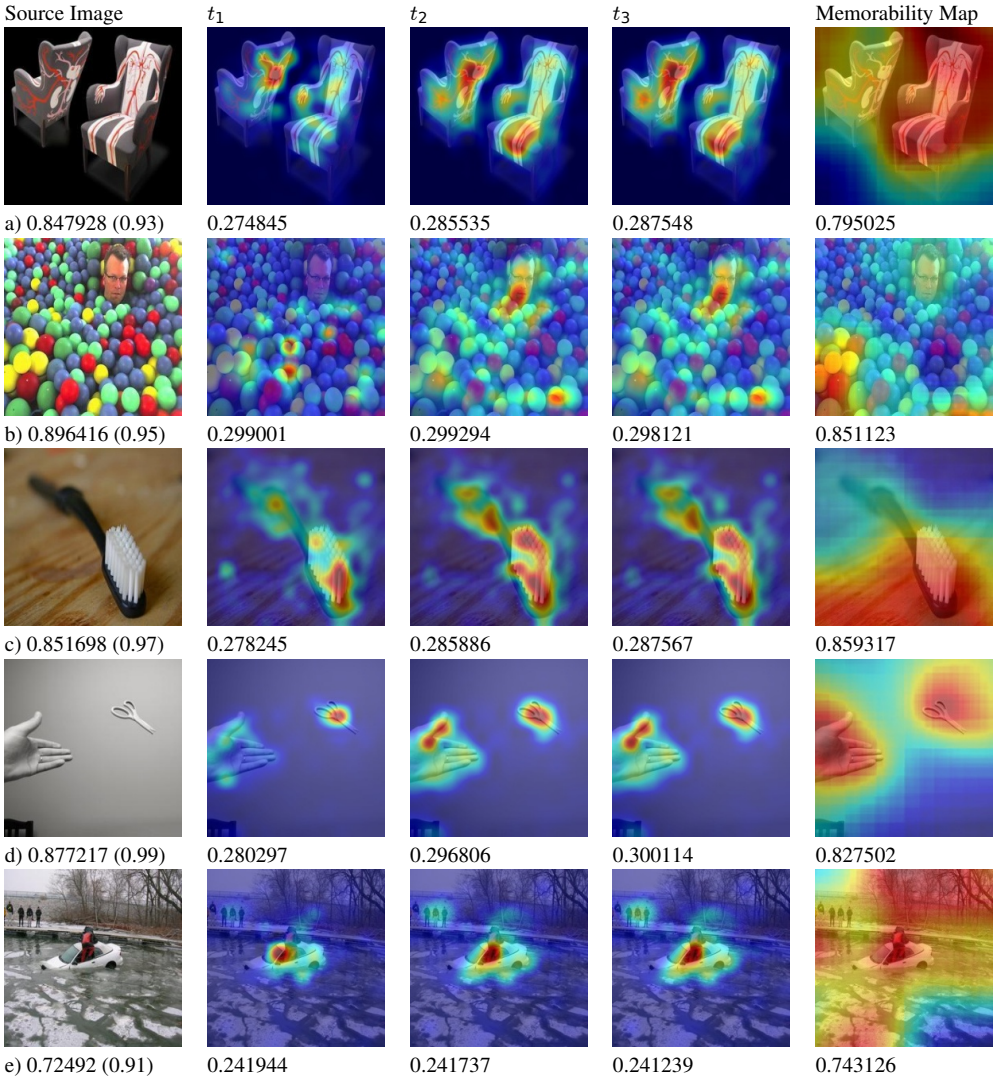


FIGURE 2.13: Qualitative comparison between the AMNet attention maps over the regression sequence and the MemNet (Khosla et al., 2015) memorability maps. Source images are shown in the left column with memorabilities estimated by AMNet, followed by ground truth in brackets. Attention maps for t_1, t_2 and t_3 LSTM steps with the corresponding discrete memorabilities are in the following columns. The last column shows memorability maps and scores estimated by the MemNet.

2.8 Implementation

The entire AMNet network is implemented in PyTorch 0.2.0. Implementation of the front-end CNNs as well as the models trained on the ImageNet are obtained from the torchvision module. The full AMNet network diagram is shown in Figure 2.14 with the LSTM sequence unrolled over the three steps. The AMNet uses visual features extracted with the pre-trained CNN. In order to obtain the feature tensor with the desired resolution 14×14, we truncate the CNN before the final classification layers. The number of preserved layers for each tested model is shown in Table 2.4.

The AMNet network with the ResNet50 (He et al., 2016) CNN has 30M parameters compared to 57M of the original MemNet (Khosla et al., 2015). However, only

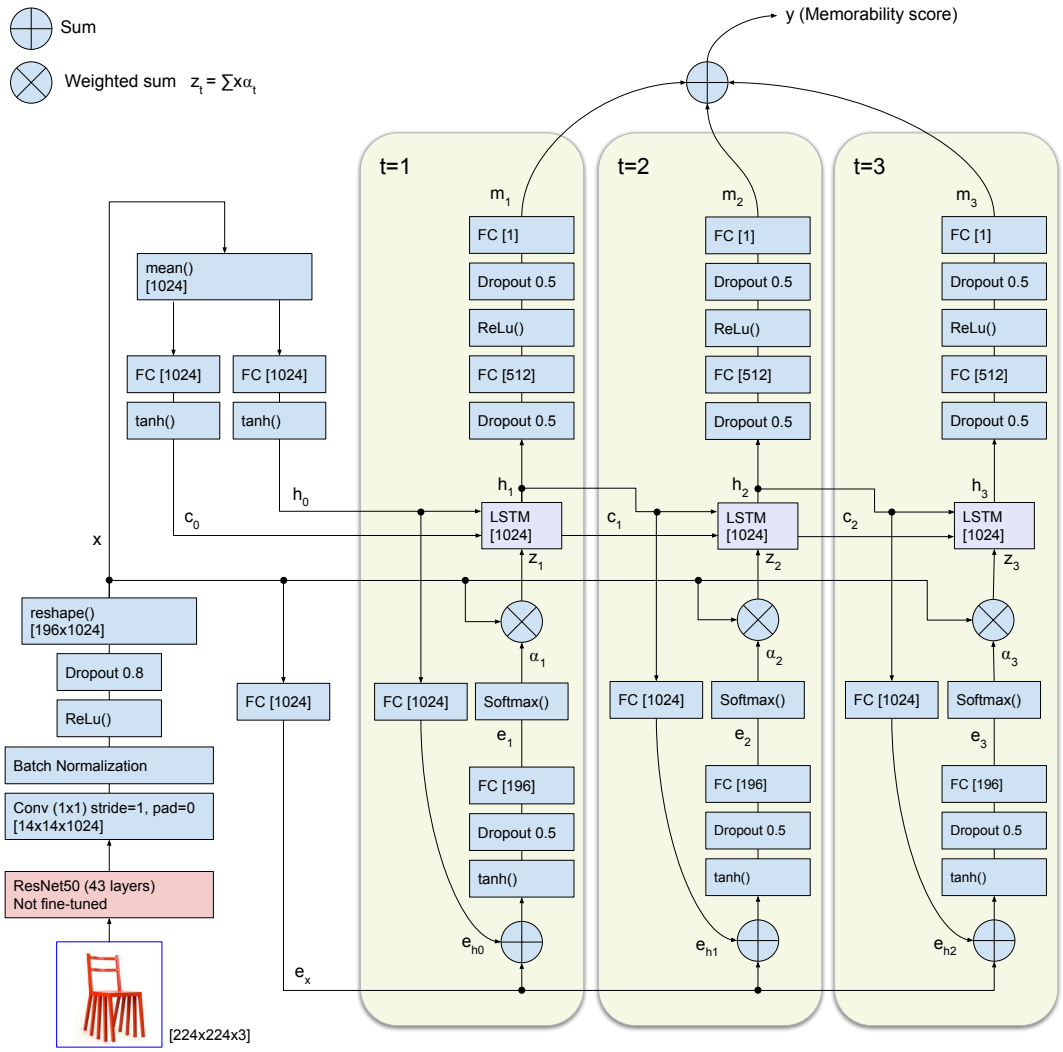


FIGURE 2.14: AMNet network diagram with LSTM unrolled over a three steps sequence. Dropout level is specified as a number of neurons to drop. Output dimensions are noted in square brackets. FC signifies a fully connected neural network.

the 13M AMNet parameters are trained during optimization. The number of parameters for all evaluated CNNs are shown in Table 2.4. Further on, we have tried to fine tune the CNN feature extractor along with the AMNet, which in majority of cases led to a drop in performance. This is very likely caused by relatively small number of training images in the LaMem compared to the ImageNet.

The source images, typically with resolution close to 256×256, are scaled to 224×224. In contrast, the MemNet estimates memorability as an average of memorability scores measured over 10 image crops (original plus left-right mirrors of four corners and center crops). It is conceivable to expected that the AMNet would achieve even higher performance with the 10 crops evaluation. We leave this for the future work.

2.9 Conclusion

Our visual memory suffers from many weaknesses, especially when seen from the perspective of silicon-based computer memory; lossy storage, slow read and write operations, nondeterministic recall and forgetting to name a few. While these memory attributes represent clear challenges in deterministic computational systems such as common computer architectures, they are unquestionably superior for applications in AGI agents. Our memory with the supporting circuits can perform complex auto and hetero-association, long and short-term predictions given new data and past knowledge, or combine the existing knowledge to form novel, artificial events, which are retained in the memory for further recall. The latter is an example of a possible source of creativity, particularly celebrated by the human race.

In our work, we focus only on visual memory. Our visual memory has a surprisingly large capacity to remember images over long periods of time. Not all images are, however, retained by our visual cortex equally well. Some, albeit not particularly appealing or engaging images, instantly stick in our memory, while on other occasions, we have a hard time memorizing images that are of high importance to us even though we thoroughly focus on them. To understand the neurological processes underpinning image retention is admittedly important to recreate similar human memory-like capabilities, hopefully also encompassing the creativity.

In this chapter, we studied image memorability learning and estimation with computational methods. Based on the prior work indicating that memorability is an intrinsic image property, we decided to implement a method relating the scene composition to the memorability scores as predicted by humans. Consequently, we proposed the AMNet method, a deep neural network with a visual attention component for image memorability estimation. This network consists of a pre-trained, deep CNN followed by a modified visual soft attention mechanism with a recurrent network completed with a network for memorability regression. The learned spatial, soft attention function localizes image regions responsible for the image retention in our memory, therefore improving the memory estimation with the machine learning method by filtering out less relevant information. By design, the AMNet is generic and could be employed for other regression, computer vision tasks. We show its successful application to image aesthetics estimation, a domain that deals with learning another perceptual image property.

We show that features extracted by a deep CNN, trained on large-scale image classification tasks, such as ImageNet, are beneficial for the memorability estimation task. This indicates that the feature hierarchies extracted for the image classification task are suitable also to express the composition underlying the memorability effect. We demonstrate that our recurrent visual attention network significantly improves performance of image memorability learning and inference. We show that the proposed method outperforms the previous state-of-the-art by 5.8% (from $\rho = 0.64$ to $\rho = 0.677$) on the Spearman's rank correlation and closely approaches the human

performance $\rho = 0.68$ with a 99.6% consistency.

Our work puts forward a novel SOTA method for image memorability estimation that also localizes image regions correlated with memorability, laying out new ground for interpretability of the memorability triggers with respect to the scene composition. However, the AMNet method has several limitations, predominantly consequential to the applied deep learning methods and the established protocol for image memorability quantification and evaluation. First, the AMNet method currently does not provide a confidence factor for the estimated memorability score - we left this functionality for future work. Next, the AMNet can process only images with fixed resolution due to the CNN architecture. This means that all images must be first scaled to these dimensions that may not always be appropriate, e.g., for images with different aspect ratios. We also leave this limitation as a subject of our future work. An inability to continually learn memorability from a stream of new images, prohibited by the catastrophic forgetting effect, is yet another CNN limitation inherited by the AMNet. A low resolution of the AMNet's memorability maps could also be seen as a limitation, particularly in the domains concerned with the model and image memorability explainability and interpretability.

In a broader sense, further limitations stem from the current, coarse memorability quantification that excludes other data such as which objects in the scene were recalled by the participants, the participants' age, gender, physiological state(rested or tired), and other parameters. Consequently, a model trained on such data cannot be customized to different users; thus, it is limited to a general, average image memorability estimation.

The main contributions of this work are:

- Proposed novel generic architecture for regression tasks with deep CNN, visual attention mechanism, and recurrent neural network.
- Application of the proposed method to image memorability estimation.
- Introduction of the incremental memorability estimation with the recurrent network and demonstration of the achieved performance gain.
- Introduction of the visual attention technique for the memorability estimation and presentation of the performance gain.
- Demonstration that transfer learning from deep models, trained for image classification, is particularly beneficial for memorability estimation.

The AMNet PyTorch implementation, including all trained models and datasets, is publicly available on https://github.com/ok1zjf. An online demo application that estimates memorability for a given image(s), including visualization of the attention maps, is accessible on https://amnet.kingston.ac.uk/.

2.9.1 Future Work

In the course of our research, we identified many avenues for future work.

For example, as a promising technique to further improve the prediction accuracy, the memorability regression could be turned into a classification problem. There is evidence presented in prior work in other domains that changing the regression to classification simplifies training and improves the model performance. This approach can be observed, for example, in work on object detection and segmentation by Liu et al. (2016), Farhadi and Redmon (2018), and He et al. (2017).

The CNN front-end for feature extraction could be trained in an unsupervised setting with methods such as Grill et al. (2020) or Chen and He (2020). It is likely that the CNN features trained on very large image datasets without supervision may be more generic (not skewed towards specific pre-training tasks, such as classification), and thus more suitable to express the image memorability.

As another future work, we consider implementing the memorability confidence predictor. Here, the AMNet model learns to predict how reliable is the current memorability estimation based on the agreement on ground truth labels among the participants annotating the data.

To enable the AMNet to process arbitrary image resolution, we propose to process the CNN feature maps (already agnostic to the input image resolution) as a sequence of $W \times H$ vectors, each with dimension $1 \times 1 \times C$ where W,H and C are the width, high and number of channels in the feature map.

In other follow-up work, we intend to compare the sequence of memorability maps to eye fixations predicted by other models. Eye fixations, either predicted or collected within the memorability game and stored alongside the memorability scores, could be used as additional input to the AMNet. It would either provide a soft prior for the attention regions refined by the model during training or hard centers for the attention regions (the eye fixation prediction would substitute AMNet attention).

Large datasets and ubiquitous computation have been critical ingredients in the success of deep learning in most machine learning areas. However, there is a considerable lack of large annotated datasets for image memorability estimation. To assemble new datasets or improve existing annotations, we are considering to use electroencephalogram (EEG) data. As recently reported by Stothart et al. (2021), EEG data can be used to detect already seen or not seen images by the participants. EEG data could also be used as direct labels during training and outputs during inference. Therefore, rather than directly predicting the memorability score, the ML model would estimate the EEG signals and then translate them to memorability in later stages of the ML pipeline. As unquestionably more expensive and time-consuming than the memorability game for data collection, this method would result in more accurate labels and new meta-data to assist neural network training.

Chapter 3

Episodic Memory Segmentation for Video Summarization

This chapter studies episodic segmentation (Zacks et al., 2007) in human memory applied to the machine learning method for video summarization. To logically split a continuous video stream into a subset of episodes, each with a specific importance level, is challenging but an equally important problem with ample applications. In this work, we propose a machine learning method for video summarization, drawing inspiration from the process of segmentation of episodic events in our brain.

The recall of past experiences is perhaps the most common cognitive process in our brain. We replay our past memories that are similar to the situation we are currently experiencing to help us improve our performance on that task (Wimmer et al., 2020). In other cases, we revisit past memories to analyze them more deeply in our thoughts, such as to create new connections with other memories, plan changes in our behavior or expand our memories for novel hypothetical cases, rare or not possible to experience in reality, but very important to us (Clewett et al., 2019). For example, what would happen if I walk closer to the cliff edge? What would I do when falling down? Would I survive? The experience replay occurs consciously at our will or spontaneously when we sleep.

Memories of our experiences were studied by Tulving (1972, 2002) in his seminal work, where he established the concept of episodic and semantic memory. Episodic memory represents our personal experiences, also referred to as autobiographical memory. In contrast, semantic memory is a memory of facts and logical relations. An example of episodic memory would be a sequence of moments to set up a campfire. A recall of a forest background, getting stones to frame the fire center, collecting dead wood and organizing it to build a fire place and lighting them. An example of semantic memory would be a recall that small deadwood burns well, that stopping the fire from spreading can be done with a stone barrier, that fire can burn us, and other facts. Episodic memories are the basis for semantic memories, or knowledge, gained by experience.

A common aspect of episodic memories is that they appear as compact segments

of real events. Each memory episode is centered around some unifying, typically semantic feature. Each episode also appears to be compressed as a succession of slices, or pictures, with considerable temporal gaps between (Jeunehomme et al., 2018, Jeunehomme and D'Argembeau, 2020). Figure 3.1 portrays a possible episodic segmentation.

Every memory episode is not retained with the same level of details and intensity. Dunsmoor et al. (2015) show that fear-conditioned events resulted in stronger memory retention compared to conceptually similar events experienced immediately afterward. It appears that such fear-conditioned episodes mask other, temporarily close events. Interestingly, even fear events that, shortly after experiencing, were portrayed as non-threatening, e.g., the snake actually was not poisonous, still exhibit retention similar to the original fearful events.

The episodic segmentation is not yet well understood, particularly which episodes are retained and how their boundaries are established. It has been shown that the integration and separation of discrete events proceed over complex interaction between hippocampus and cortex (Clewett et al., 2019). This happens proactively as experiences occur, or retroactively during the recall later on, in the wake or sleep states. Ezzyat and Davachi (2011) studied the nature of the episodic boundaries and showed that there is, indeed, a higher degree of associative memories within segments than across boundaries.

Complex events are generally summarized as key slices, likely to reduce their memory footprint and to provide fast recall and fast navigation within the events. Furthermore, the keyframes also appear to form the smallest event units that could be predicted by the brain and acted upon (Kurby and Zacks, 2008, Zacks et al., 2007). This appears to be related to information chunking in the working memory, which is also the stage for episodic replay.

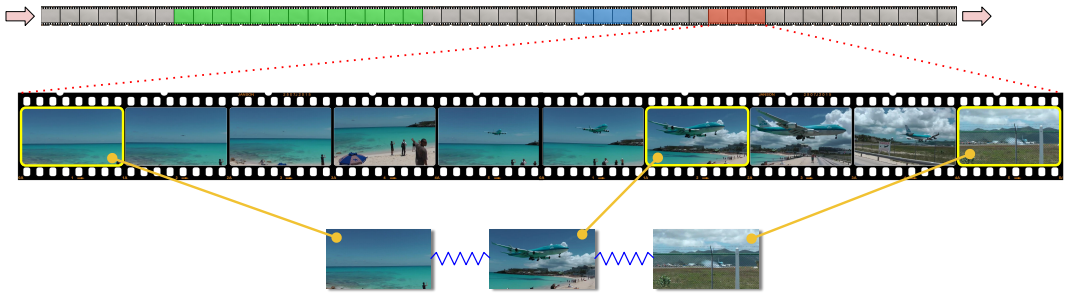


FIGURE 3.1: Illustration of episodic segmentation. Our brain segments continuous video stream into contextually coherent episodes (green, blue and red on the top strip). Each episode itself is represented by key frames (yellow). Images sourced from the TvSum dataset (Song et al., 2015).

In a famous paper, *The magical number seven plus or minus two*, Miller (1956) showed that the number of items, such as letters, digits, sentences, or images, that we can hold in our working memory is limited to 7 ± 2. This limit was later revised to about 4 in the work of Cowan (2001), who took into account more aspects such as duration of the chunks, intellectual ability, and the participant age. What constitutes

a chunk appears to be determined by our prior knowledge which, we can control to some extent. For example, to remember a long phone number, we can split it into groups of three or four digits. This is readily done when presenting phone numbers, where digits are usually arranged in groups and divided by spaces or hyphens.

The findings that episodic segmentation is importance-conditioned and that the episodes are internally represented by key events are important insights. Suppose we know how to select the episodic boundaries and the keyframes within. In that case, we could design a method for automatic video summarization that would retain information indistinguishable from what we would personally recall after experiencing that same event. To accomplish this with a machine learning method, we propose an algorithm to learn frame importance scores from annotations performed by humans that are subsequently used to select the episodic segments. To determine the frame scores, we introduce a self-attention network that learns relations between frames in the input sequence as a function of importance scores estimated by human annotators.

First, in Section 3.1, we define the video summarization problem and outline our approach. In the following Section 3.2, we review related literature and then in Section 3.3 describe details of our method. Evaluation protocol, experimental setup, and used datasets are discussed in Section 3.4. In Section 3.5, we present our results and close with the conclusion in Section 3.6.

3.1 The Video Summarization Problem

Video summarization is typically defined as a task where a video sequence is reduced to a small number of still images called keyframes, sometimes also called storyboard or thumbnails extraction. Video can be also summarized as a shorter video sequence composed of keyshots, also called video skim or dynamic summaries. The keyframes or keyshots need to convey most of the vital information contained in the original video. This task is similar to a lossy video compression, where the building block is a video frame. In this work, we focus solely on the keyshots based video summarization.

Video summarization is an inherently difficult task even for us. In order to identify the most important segments, one needs to view the entire video content and then make the selection, subject to the desired summary length. Naturally, one could define the keyshots as segments that carry mutually diverse information while also being highly representative of the video source. Some methods formulate the summarization task as clustering with cost functions based on exactly these criteria. Unfortunately, to define how well the chosen keyshots represent the video source, and the diversity between them, is extremely difficult since this needs to reflect the information level perceived by the user. Common techniques analyze motion features, measure the distance between color histograms, image entropy, or 2D/3D CNN

features (Novak and Shafer, 1992, Larkin, 2016, Athiwaratkun and Kang, 2015), reflecting semantic similarities. However, none of these approaches can truly capture the information in the video context. Therefore, in this work, we set up the video summarization as an imitation learning problem where a machine learning method learns to mimic humans on the video summarization task.

Early video summarization methods were based on unsupervised methods, leveraging low-level spatio-temporal features and dimensionality reduction with clustering techniques. The success of these methods solely stands on the ability to define distance/cost functions between keyshot frames with respect to the original video. As discussed above, this is not easy to achieve. It also introduces a strong bias in the summarization given by the type of used features such as semantic and pixel intensities. In contrast, models trained with supervision learn the transformation that produces summaries similar to those manually produced. Currently, there are two datasets with such annotations, TvSum (Song et al., 2015) and SumMe (Gygli et al., 2014), where each video is annotated by 15-20 users. The annotations vary between users, with consistency expressed by a pairwise F-score ~ 0.34. This fact reveals that video annotation is a rather subjective task. We argue that under these circumstances, it may be very challenging to craft a metric that would accurately express how to cluster video frames into keyshots, similar to human annotation. On this premise, we decided to adopt the supervised video summarization for our work.

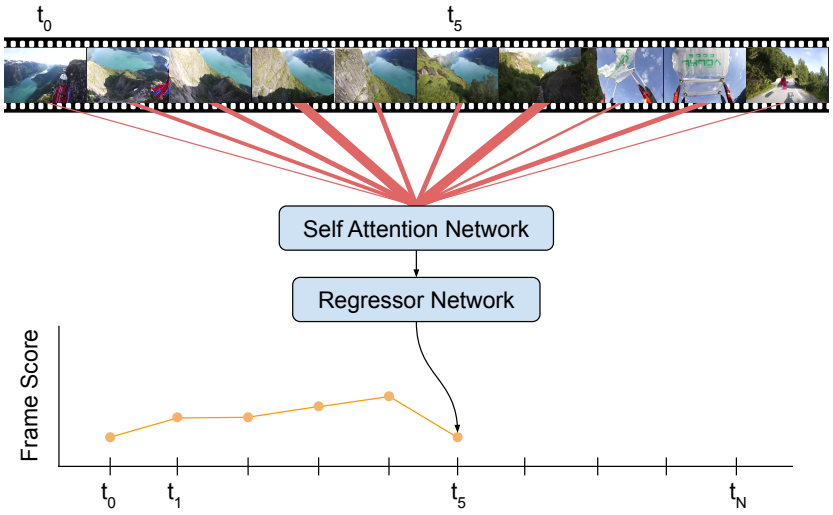


FIGURE 3.2: For each output the self-attention network generates weights for all input features. Average of the input features, weighted by this attention, is regressed by a fully connected neural network to the frame importance score.

The current state-of-the-art methods for video summarization are based on recurrent encoder-decoder architectures, usually with bi-directional LSTM (Hochreiter and Schmidhuber, 1997) or gated recurrent unit (GRU) (Cho et al., 2014) and soft attention (Bahdanau et al., 2014). While these models are remarkably powerful in many domains, such as machine translation and image/video captioning, they are

computationally demanding (see Section 3.3.4) , especially in the bi-directional configuration. Recently, Vaswani et al. (2017) demonstrated that it is possible to perform sequence to sequence transformation only with attention. Similarly, we propose attention only, sequence to sequence network VASNet for video keyshots summarization, and demonstrate its performance on TvSum and SumMe benchmarks. The architecture of this model does not employ recurrent or sequential processing and can be implemented with conventional matrix/vector operations and run in a single forward/backward pass during inference/training, even for sequences with variable length. The architecture is centered around two critical operations, attention weights calculation and frame-level score regression. An overview of this model is shown in Figure 3.2. Frame score at every step t is estimated from a weighted average of all input features. The weights are calculated with the self-attention algorithm. Given the generic architecture of our model, we believe that it could be successfully used in other domains requiring sequence to sequence transformation.

3.2 Related Work

Recent advancements in deep learning were adopted to implement video summarization, in particular the encoder-decoder method with attention for sequence to sequence transformation.

Zhang et al. (2016) pioneered the application of LSTM for supervised video summarization to model the variable-range temporal dependency among video frames to derive both representative and compact video summaries. They enhance the strength of the LSTM with the determinantal point process which is a probabilistic model for diverse subset selection. Another sequence to sequence method for supervised video summarization was introduced by Ji et al. (2017). Their deep attention-based framework uses a bi-directional LSTM to encode the contextual information among input video frames. Mahasseni et al. (2017) propose an adversarial network to summarize the video by minimizing the distance between the video and its summary. They predict video keyframes distribution with a sequential generative adversarial network. A deep summarization network in an encoder-decoder architecture via an end-to-end reinforcement learning has been put forward by Zhou et al. (2018) to achieve state of the art results in unsupervised video summarization. They design a novel reward function that jointly takes diversity and representativeness of generated summaries into account. Zhao et al. (2017) constructed a novel hierarchical LSTM to deal with the long temporal dependencies among video frames but this method fails to capture the structure of the video information, where the shots are generated by fixed length segmentation.

Some works use side semantic information associated with a video along with visual features, like surrounding text such as titles, queries, descriptions, comments, unpaired training data and similar. Rochan and Wang (2018) proposed deep learning

video summaries from unpaired training data, which means they learn from available video summaries without their corresponding raw input videos. A deep side semantic embedding model was introduced by Yuan et al. (2017) which uses both side semantic information and visual content in the video. Similarly, Wei et al. (2018) proposed a supervised, deep learning method trained with manually created text descriptions as ground truth (GT). At the heart of this method is the LSTM encoder-decoder network. Wei et al. (2018) achieves competitive results with this approach, however, more complex labels are required for the training. Fei et al. (2017) complemented visual features with video frame memorability, predicted by a separate model such as Khosla et al. (2015) or Fajtl et al. (2018).

Other approaches, like the one described in dos Santos Belo et al. (2016), use an unsupervised method to cluster features extracted from the video, delete similar frames, and then select the rest of the frames as the keyframes. In fact, they used a hierarchical clustering method to generate a weight map from the frame similarity graph in which the clusters can easily be inferred. Another clustering method is proposed by Otani et al. (2016), in which they use deep video features to encode various levels of content including objects, actions, and scenes. They extract the deep features from each segment of the original video and apply a clustering-based summarization technique on them.

3.2.1 Attention Techniques

Since the neural attention technique is a central piece of the proposed method, we dedicated this subsection to the prior work in this field.

The fundamental concept of attention mechanism for neural networks was laid by Bahdanau et al. (2014) for the task of machine translation. This attention is based on an idea that the neural network can learn how important various samples in a sequence, or image regions, are with respect to the desired output state. These importance values are defined as attention weights and are commonly estimated simultaneously with other model parameters trained for a specific objective. There are two main distinct attention algorithms, hard and soft.

Hard attention produces a binary attention mask, thus making a 'hard' decision on which samples to consider. This technique was successfully used by Xu et al. (2015) for image caption generation. Hard attention models use stochastic sampling during the training; consequently, backpropagation cannot be employed due to the non-differentiable nature of the stochastic processes. A reinforcement learning rule REINFORCE (Williams, 1992) is regularly used to train such models. This task is similar to learning an attention policy introduced by Mnih et al. (2014).

In this work we exclusively focus on soft attention. In contrast to the hard attention, soft attention generates weights as true probabilities. These weights are calculated in a deterministic fashion using a process that is differentiable. This means that we can use backpropagation and train the entire model end-to-end. Along with the LSTM, soft attention is currently employed in the majority of sequence to sequence

models used in machine translation (Luong et al., 2015), image/video caption generation (Xu et al., 2015), (Yao et al., 2015), addressing neural memory (Graves et al., 2016) and other. Soft attention weights are usually calculated as a function of the input features and the current encoder or decoder state. The attention is global if at each step t all input features are considered or local where the attention has access to only limited number of local neighbours.

If the attention model does not consider the decoder state, the model is called self-attention or intra-attention. In this case the attention reflects the relation of an input sample t with respect to other input samples given the optimization objective. Self-attention models were successfully used in tasks such as reading comprehension, summarization and in general for task-independent sequence representations (Cheng et al., 2016, Parikh et al., 2016, Lin et al., 2017). The self-attention is easy and fast to calculate with matrix multiplication in a single pass for entire sequence since at each step we do not need the result of past state.

3.2.2 Current State-of-the-Art, its Limitations and Promising Research Directions

In summary, in the field of supervised key-shot video summarization, the currently best performing method is M-AVS Ji et al. (2017) utilizing the bi-directional LSTM encoder-decoder architecture. This method is derived from the LSTM encoder-decoder with attention for natural language processing (Luong et al., 2015). Rather than translating word tokens from one language to another, in the video summarization domain, the LSTM encoder-decoder translates video frame features to frame importance scores.

In general, the major limiting factor in video summarization is the lack of large annotated datasets. Equally, the annotation quality in the current, established datasets imposes severe constraints on the summary accuracy due to diverse labeling protocols and annotation formats used. Moreover, these datasets are entirely missing context-specific summary labels. The video summarization problem is a highly perceptual and context-dependent task; people tend to produce and expect to receive different summaries based on, for example, their domain interest (e.g., general public vs. experts) or context focus (e.g., ethical, political, or domain-specific). Another drawback of current methods is their high demand for memory and computational resources for training as well as inference. However, this issue is not specific to video summarization - it affects the entire field of methods for video processing.

Video summarization is inherently a sequence to sequence transformation that would preferably digest the entire or a significant part of the video content and then label contextually significant segments. Therefore, the most promising research direction in this domain appears to be an application of high capacity encoder-decoder models such as the recently proposed family of Transformer architectures (Khan et al., 2021). The Transformer based methods GPT (Cohen and Gokaslan, 2020,

Brown et al., 2020) noticeably pushed the boundaries in natural language processing (NLP), which may be expected to happen in the video summarization, too, due to the similar nature of these domains. However, the Transformer requires significantly larger datasets than are currently available for video summarization. This holds good not only for the Transformers but also for deep network architectures that can be currently considered only for the video frames extraction but not for the summarization. Another approach likely to advance video summarization appears to be unsupervised features learning for temporal domain (Behrmann et al., 2021) and video frames prediction (Villegas et al., 2019). These approaches have not been well explored in this domain yet, however there are indicators about their benefits. Video content prediction is yet another technique applicable to video summarization. The ability to accurately predict other frames from a given video frame indicates that this frame has a high importance score thus should be included in the produced video summary. To that point, the high predictability or high associative property of frames within the episodic memories produced by our brain has also been reported (Ezzyat and Davachi, 2011). This approach could open up new directions for unsupervised video summarization, mitigating the lack of large annotated datasets in this domain.

3.3 Model Architecture

Research of the neuroscientific literature has not brought a significant insight or inspiration directly applicable to the video summarization problem. Perhaps the most insightful study was on the high intra-associative properties of frames within an episodic segment described by Ezzyat and Davachi (2011). This study reinforced our idea to propose a model that would learn relationships (relevance scores, or attention weights) among local video frames and then, considering these relations, learn the summarization importance score for each frame.

Common approach to supervised video summarization and other sequence to sequence transformations, is an application of a LSTM or GRU encoder-decoder network with attention. Forward LSTM is usually replaced with bi-directional BiLSTM since keyshots in the summary have relation to future video frames in the sequence. Unlike the RNN based networks, our method does not need to reach for special techniques, such as BiLSTM, to achieve non-causal behaviour. The vanilla attention model has equal access to all past and future inputs. This aperture can be, however, easily modified and it can even be asymmetric, dilated, or exclude the current time step t.

The hidden state passed from encoder to decoder has always fixed length, however, it needs to encode information representing sequences with variable lengths. This means that there is a higher information loss for longer sequences. The proposed attention mechanism does not suffer from such loss since it accesses the input sequence directly without an intermediate embedding.

Architecture proposed in this work replaces entirely the LSTM encoder-decoder network with the soft, self-attention and a two layer, fully connected network for regression of the frame importance score. Our model takes an input sequence $\boldsymbol{X} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_N), \quad \boldsymbol{x} \in \mathbb{R}^D$ and produces an output sequence $\boldsymbol{Y} = (y_0, \ldots, y_N), \quad y = [0, 1)$, both of length N. The input is a sequences of CNN feature vectors with dimensions D, extracted for each video frame. Figure 3.3 shows the entire network in detail.

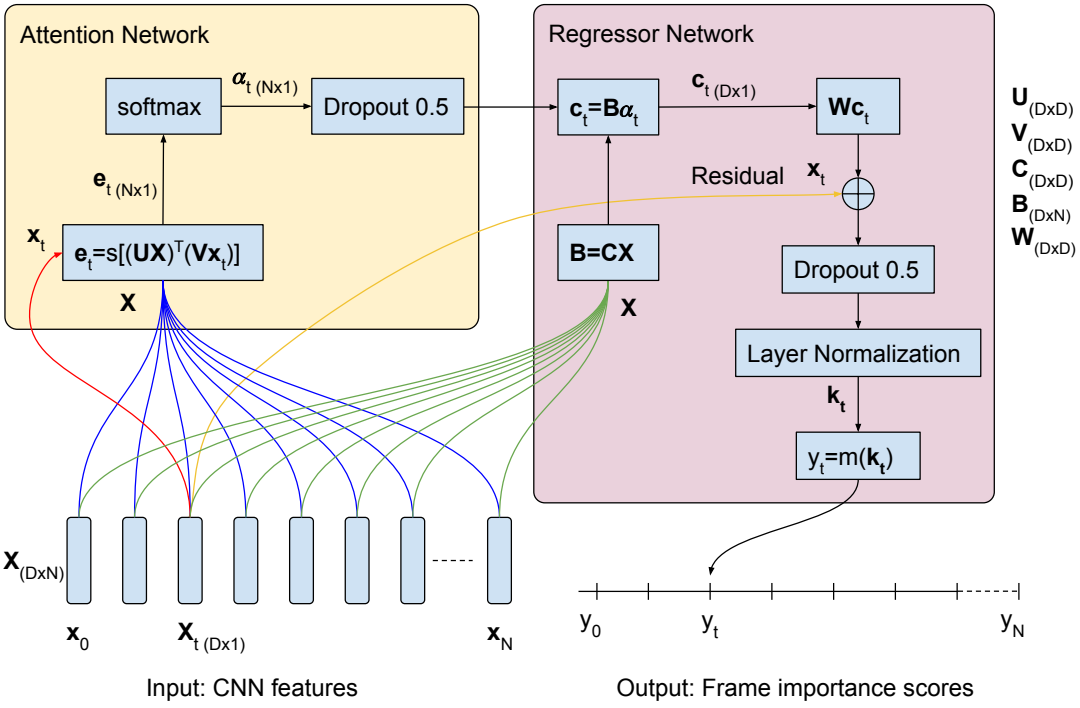


FIGURE 3.3: Diagram of VASNet network attending sample $\boldsymbol{x}_t$.

Unnormalized self-attention weight $e_{t,i}$ is calculated as an alignment between input feature $\mathbf{x}_t$ and the entire input sequence according to Luong et al. (2015).

$$e_{t,i} = s[(\boldsymbol{U}\boldsymbol{x}_i)^T(\boldsymbol{V}\boldsymbol{x}_t)] \qquad t = [0, N), \quad i = [0, N) \tag{3.1}$$

Here, N is the number of video frames, $\boldsymbol{U}$ and $\boldsymbol{V}$ are network weight matrices estimated together with other parameters of the network during optimization and s is a scale parameter that reduces values of the dot product between $\boldsymbol{U}\boldsymbol{x}_i$ and $\boldsymbol{V}\boldsymbol{x}_t$. We set the scale s to value 0.06, determined experimentally. Impact of the scale on the model performance was, however, minimal. Alternatively, the attention vector could be also realized by an additive function as shown by Bahdanau et al. (2014).

$$e_{t,i} = \boldsymbol{M} \tanh(\boldsymbol{U}\boldsymbol{x}_i + \boldsymbol{V}\boldsymbol{x}_t) \tag{3.2}$$

where $\boldsymbol{M}$ are additional network weights learned during training. Both formulas have shown similar performance, however, the multiplicative attention is easier to

parallelise since it can be entirely implemented as a matrix multiplication which can be highly optimized. The attention vector $\boldsymbol{e}_t$ is then converted to the attention weights $\boldsymbol{\alpha}_t$ with softmax.

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{N} \exp(e_{t,k})} \tag{3.3}$$

The attention weights $\boldsymbol{\alpha}_t$ are true probabilities representing the importance of input features with respect to the desired frame level score at the time t. Linear transformation $\boldsymbol{C}$ is then applied to each input and the results then weighted with attention vector $\boldsymbol{\alpha}_t$ and averaged. The output is a context vector $\boldsymbol{c}_t$ which is used for the final frame score regression.

$$\boldsymbol{b}_i = \boldsymbol{C}\boldsymbol{x}_i \tag{3.4}$$

$$\boldsymbol{c}_t = \sum_{i=1}^{N} \alpha_{t,i}\boldsymbol{b}_i \qquad \boldsymbol{c}_t \in \mathbb{R}^D \tag{3.5}$$

The context vector $\boldsymbol{c}_t$ is then projected by a single layer, fully connected network with linear activation and residual sum followed by dropout and layer normalization.

$$\boldsymbol{k}_t = norm(dropout(\boldsymbol{W}\boldsymbol{c}_t + \boldsymbol{x}_t)) \tag{3.6}$$

The $\boldsymbol{C}$ and $\boldsymbol{W}$ are network weight matrices learned during the network training. To regularize the network we also add a dropout for attention weights as shown in Figure 3.3. We found it to be beneficial, especially for small training datasets such as in the canonical setting for TvSum (40 videos) and SumMe (20 videos).

By design, the attention network discards the temporal order in the sequence. This is due to the fact that the context vector $\boldsymbol{c}_t$ is calculated as a weighted average of input features without any order information. The order of the output sequence is still preserved. The positional order for the frame score prediction is not important in the video summarization task, as has been shown in the past work utilizing clustering techniques that also discard the input frame order. For other tasks, such as machine translation or captioning, the order is essential. In these cases every prediction at time t, including attention weights, could be conditioned on state at $t-1$. Alternatively, a positional encoding could be injected to the input as proposed by Vaswani et al. (2017) and Gehring et al. (2017).

Finally, a two layer neural network performs the frame score regression $y_t = m(\boldsymbol{k}_t)$. First layer has a ReLU activation followed by dropout and layer normalization (Ba et al., 2016), while the second layer has a single hidden unit with sigmoid activation.

3.3.1 Contributions to the Field of Neural Architectures

In summary, our method builds on standard CNN methods for features extraction, on the recently proposed idea of self-attention in the Transformer architecture Vaswani et al. (2017), and typical neural network methods such as dropout and ℓ_2

weights regularization, batch normalization, tanh, sigmoid and rectified linear unit (ReLU) activations and SGD optimization methods. Our key contributions to the field of neural architectures are:

- Simplified soft self-attention network for sequences of CNN features. The key difference from the Transformer model is that our network uses only a single attention head combined with dropout regularization, with the attention being applied between fixed input-output pairs; therefore, unlike the Transformer, it does not require positional encoding.
- A regression neural network to generate the frame importance score trained in an end-to-end fashion with the self-attention layer.

3.3.2 Frame Scores to Keyshot Summaries

The model outputs frame-level scores that are then converted to keyshots. Following Zhang et al. (2016), this is done in two steps. First, we detect scene change points where each represents a potential keyshot segment. Second, we select a subset of these keyshots by maximizing the total frame score within these keyshots while constraining the total summary length to 15% of the original video length as per Gygli et al. (2014). The scene change points are detected by Kernel Temporal Segmentation (KTS) method (Potapov et al., 2014) as shown in Figure 3.4. For each detected shot

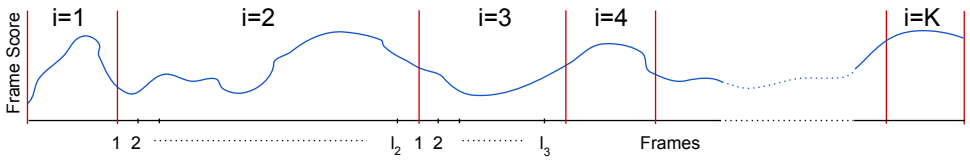


FIGURE 3.4: Temporal segmentation with KTS.

$i \in K$ we calculate score s_i.

$$s_i = \frac{1}{l_i} \sum_{a=1}^{l_i} y_{i,a} \tag{3.7}$$

where $y_{i,a}$ is score of a-th frame within shot i and l_i is the length of i-th shot. Keyshots are then selected with the Knapsack algorithm Eq. 3.8 according to Song et al. (2015).

$$\max \sum_{i=1}^{K} u_i s_i, \quad \text{s. t.} \quad \sum_{i=1}^{K} u_i l_i \leq L, u_i \in 0, 1 \tag{3.8}$$

Keyshots with $u_i = 1$ are then concatenated to produce the final video summary. For evaluation we create a binary summary vector where each frame in shot ($u_i = 1$) is set to one.

TABLE 3.1: Overview of the TvSum and SumMe properties.

Dataset	Videos	User annotations	Annotation type	Video length (sec)		
				Min	Max	Avg
SumMe	25	15-18	keyshots	32	324	146
TvSum	50	20	frame-level importance scores	83	647	235
OVP	50	5	keyframes	46	209	98
YouTube	39	5	keyframes	9	572	196

3.3.3 Model Training

To train our model we use the ADAM optimizer (Kingma and Ba, 2015) with learning rate $5 \cdot 10^{-5}$ This low learning rate is used as a result of having a batch with single sample, where the sample is an entire video sequence. We use 50% dropout and $L2 = 10^{-5}$ regularization. Training is done over 200 epochs. Model with the highest validation F-score is then selected.

All model hyperparameters, including the number of hidden layers, their sizes, and regularization parameters, were selected experimentally during training and evaluation on the training and evaluation datasets.

3.3.4 Computation Complexity

As reported by Vaswani et al. (2017) the self-attention requires a constant number of operations at each step for all input features N, each of size D. The complexity is thus $O(N^2D)$. The recurrent layer, on the other hand, requires $O(N)$ sequential operations, each of complexity $O(ND^2)$. Self-attention needs less computation when the sequence length N is shorter than the feature size D. For longer videos, a local attention would be used rather then the global one.

3.4 Evaluation

3.4.1 Datasets Overview

In order to directly compare our method with previous work we conducted all experiments on four datasets, TvSum (Song et al., 2015), SumMe (Gygli et al., 2014), OVP (De Avila et al., 2011) and YouTube (De Avila et al., 2011). OVP and YouTube were used only to augment the training dataset. TvSum and SumMe are currently the only datasets suitably labeled for keyshots video summarization, albeit still small for training deep models. Table 3.1 provides an overview of the main datasets properties.

The TvSum dataset is annotated by frame-level importance scores, while the SumMe with binary keyshot summaries. OVP and YouTube are annotated with

keyframes and need to be converted to the frame-level scores and binary keyshot summaries, following the protocol discussed in the following section 3.4.2.

3.4.2 Ground Truth Preparation

The proposed model is trained using frame-level scores, while the evaluation is performed with the binary keyshot summaries. The SumMe dataset comes with keyshot annotations, as well as frame-level scores calculated as an average of the keyshot user summaries per frame. In the case of TvSum we convert the frame-level scores to keyshots following the protocol described in section 3.3.2. Keyframe annotations in OVP and YouTube are converted to frame-level scores by temporarily segmenting the video into shots with KTS and then selecting shots that contain the keyframes. Knapsack is then used to constrain the total summary length, however in this case the keyshot score s_i (Eq. 3.8) is calculated as a ratio of number of keyframes within the keyshot and the keyshot length.

For objective comparison, we adopt identical training and testing ground truth data as used by Zhang et al. (2016), Zhou et al. (2018) and Mahasseni et al. (2017). This represents CNN embeddings, scene change points, and generated frame-level scores and keyshot labels for all datasets. The preprocessed data were made publicly available by Zhou et al. (2018) [1] and Zhang et al. (2016) [2]. CNN embeddings used in this preprocessed dataset have 1024 dimensions and were extracted from the pool5 layer of the GoogLeNet network (Szegedy et al., 2015) trained on ImageNet (Russakovsky et al., 2015).

We use a 5-fold cross validation for both, canonical and augmented settings as suggested by Zhang et al. (2016). In the canonical setting, we generate 5 random train/test splits for the TvSum and SumMe datasets individually. 80% samples are used for training and the rest for testing. In the augmented setting we also maintain the 5-fold cross validation with the 80/20 train/test, but add the other datasets to the training split. For example, to train the SumMe in the augmented setting we take all samples from TvSum, OVP and YouTube and 80% of the SumMe as the training dataset and the remaining 20% for evaluation.

3.4.3 Evaluation Protocol

For a fair comparison with the state of the art, we follow evaluation protocol from Zhang et al. (2016), Zhou et al. (2018) and Mahasseni et al. (2017). To asses the similarity between the machine and user summaries we use the harmonic mean of precision and recall expressed as the F-score in percentages.

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \times 100 \tag{3.9}$$

[1] http://www.eecs.qmul.ac.uk/~kz303/vsumm-reinforce/datasets.tar.gz
[2] https://www.dropbox.com/s/ynl4jsa2mxohs16/data.zip?dl=0

True and false positives and false negatives for the F-score are calculated per-frame as the overlap between the ground truth and machine summaries, as shown in Figure 3.5.

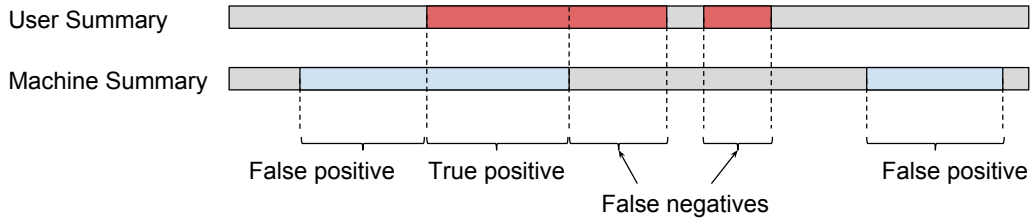


FIGURE 3.5: True positives, False positives and False negatives are calculated per-frame between the ground truth and machine binary keyshot summaries.

Following Gygli et al. (2014), the machine summary is limited to 15% of the original video length and then evaluated against multiple user summaries according to Zhang et al. (2016). Precisely, on the TvSum benchmark, for each video, the F-score is calculated as an average between the machine summary and each of the user summaries as suggested by Song et al. (2015). Average F-score over videos in the dataset is then reported. On the SumMe benchmark, for each video, a user summary most similar to the machine summary is selected. This approach is proposed by Gygli et al. (2015) and also used in the work of Lin and Chin-Yew (Lin, 2004).

3.5 Experiments and Results

Results of the VASNet evaluation on TvSum and SumMe datasets, compared with the most recent state of the art methods are presented in Table 3.3. To illustrate

TABLE 3.2: Average pairwise F-scores calculated among user summaries and between ground truth (GT) and users summaries.

Dataset	Pairwise F score	
	Among users annotations	**Training GT w.r.t. users annotations (human performance)**
SumMe	31.1	64.2
TvSum	53.8	63.7

how well the methods learned from the user annotations we show a human performance, which is calculated as pairwise F-scores between the ground truth and all user summaries. In Table 3.2 we also compare the human performance with F-scores calculated among the user summaries themselves.

We can see that the human performance is higher than the F-score among the user summaries which is likely caused by the fact that the training ground truth is calculated as an average of all user summaries and then converted to the keyshots,

TABLE 3.3: Comparison of our method VASNet with the state of the art methods for canonical and augmented settings. For a reference we add human performance measured as pairwise F-score between training ground truth and user summaries.

Method	SumMe		TvSum	
	Canonical	Augmented	Canonical	Augmented
dppLSTM (Zhang et al., 2016)	38.6	42.9	54.7	59.6
M-AVS (Ji et al., 2017)	44.4	46.1	61.0	61.8
DR-DSN$_{sup}$ (Zhou et al., 2018)	42.1	43.9	58.1	59.8
SUM-GAN$_{sup}$ (Mahasseni et al., 2017)	41.7	43.6	56.3	61.2
SASUM$_{sup}$ (Wei et al., 2018)	45.3	-	58.2	-
Human	64.2	-	63.7	-
VASNet (ours)	**49.71**	**51.09**	**61.42**	**62.37**

which are aligned on the scene change-points. These keyshots are likely to be longer than the discrete user summaries, thus having higher mutual overlap. The pairwise F-score 53.8 for TvSum dataset is higher than the F-score 36 as reported by the TvSum authors (Song et al., 2015). This is because we convert each user summary to keyshots with KTS and limit the duration to 15% of the video length and then calculate the pairwise F-scores. The TvSum authors calculate the F-score from *gold standard labels*, that is, from keyshots of length 2 seconds, a length used by users during the frame-level score annotation. We chose to follow the former procedure which is maintained in all evaluations in this work to make the results directly comparable.

In Table 3.3 we can see that our method outperforms all previous work in both canonical and augmented settings. On the TvSum benchmark the improvement is by 0.7% and 1% in the canonical and augmented settings respectively and 2% lower than the human performance. On the SumMe this is 12% and 11% in the canonical and augmented settings respectively and 21% below the human performance. In Figure 3.6 we show this improvements visually.

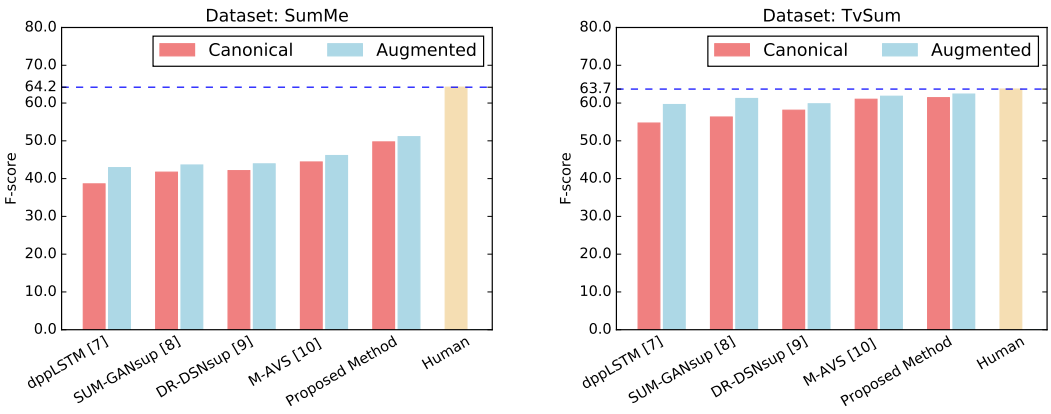


FIGURE 3.6: VASNet performance gain compared to the state-of-the-art and human performance.

The higher performance gain on the SumMe dataset is very likely caused by the

fact that our attention model can extract more information from the ground truth compared to the TvSum, where most methods already closely approach the human performance. It is conceivable to assume that the small gain on the TvSum is caused by the negative effect of the global attention on long sequences. TvSum videos are comparatively longer than the SumMe as seen in Table 3.1. At every prediction step the global attention 'looks' at all video frames. For long video sequences frames from temporally distant scenes are likely less relevant than the local ones, but the global attention still needs to explore them. We believe that this increases variance in the attention weights, which negatively impacts the prediction accuracy. We hypothesize that this could be mitigated by the introduction of local attention.

3.5.1 Correlation with Ground Truth

To show correlation between the machine summaries produced by our method and the ground truth, we plot the ground truth and predicted scores for two videos from TvSum in Figure 3.7. For a direct comparison with prior work we selected videos 10 and 11 as they are used in Zhou et al. (2018). We can see a clear correla-

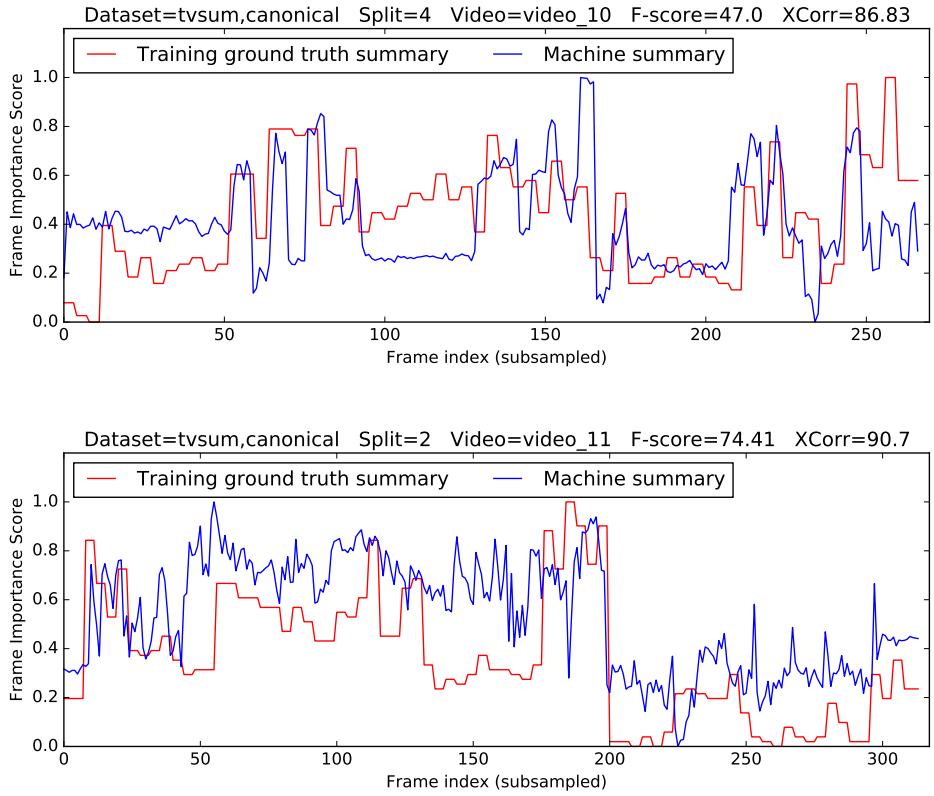


FIGURE 3.7: Correlation between ground truth and machine summaries produced by VASNet for test videos 10 and 11 from TvSum dataset, also evaluated in Zhou et al. (2018).

tion between the ground truth and machine summary, confirming the quality of our method. Original videos and their summaries are available on YouTube. [3]

We also compare the final, binary keyshot summary with the ground truth. In Figure 3.8 we show machine generated keyshots in light blue color over the ground truth importance scores shown in gray. We can see that the selected keyshots align with most of the peaks in the ground truth and that they cover the entire length of the video.

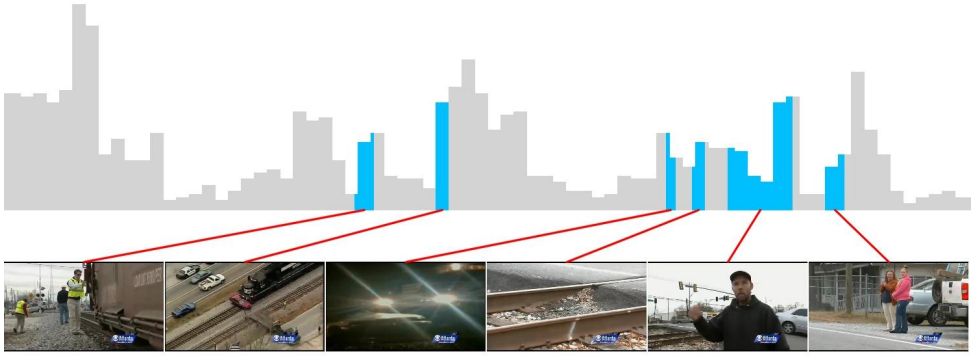


FIGURE 3.8: Ground truth frame scores (gray), machine summary (blue) and corresponding keyframes for test video 7 from TvSum dataset. video was also evaluated by Zhou et al. (2018).

The confusion matrix in Figure 3.9 shows attention weights produced during evaluation of TvSum video 7. We can see that the attention strongly focuses on frames either correlated with low frame scores (top and bottom image in Figure 3.9, attention weights for frames ∼80 and ∼190) or high scores (second and third image, frames ∼95 and ∼150). It is conceivable to assume that the network learns to associate every video frame with other frames of similar score levels.

Another interesting observation to make is that the transitions between the high and low attention weights in the confusion matrix highly correlate with the scene change points, shown as green and red horizontal and vertical lines. It is important to note that the change points, detected with KTS algorithm, were not provided to the model during learning or inference, nor were used to process the training GT. Thus, we believe that this model could be also applied to scene segmentation, removing the need for the KTS post-processing step. We will explore this possibility in our future work.

3.6 Conclusion

This chapter proposed a novel deep neural network for keyshot video summarization based on standalone soft self-attention. This network performs a sequence to sequence transformation without recurrent networks such as LSTM based encoder-decoder models. The self-attention function learns mutual relations of frames in the

[3] https://www.youtube.com/playlist?list=PLEdpjt8KmmQMfQEat4HvuIxORwiO9q9DB

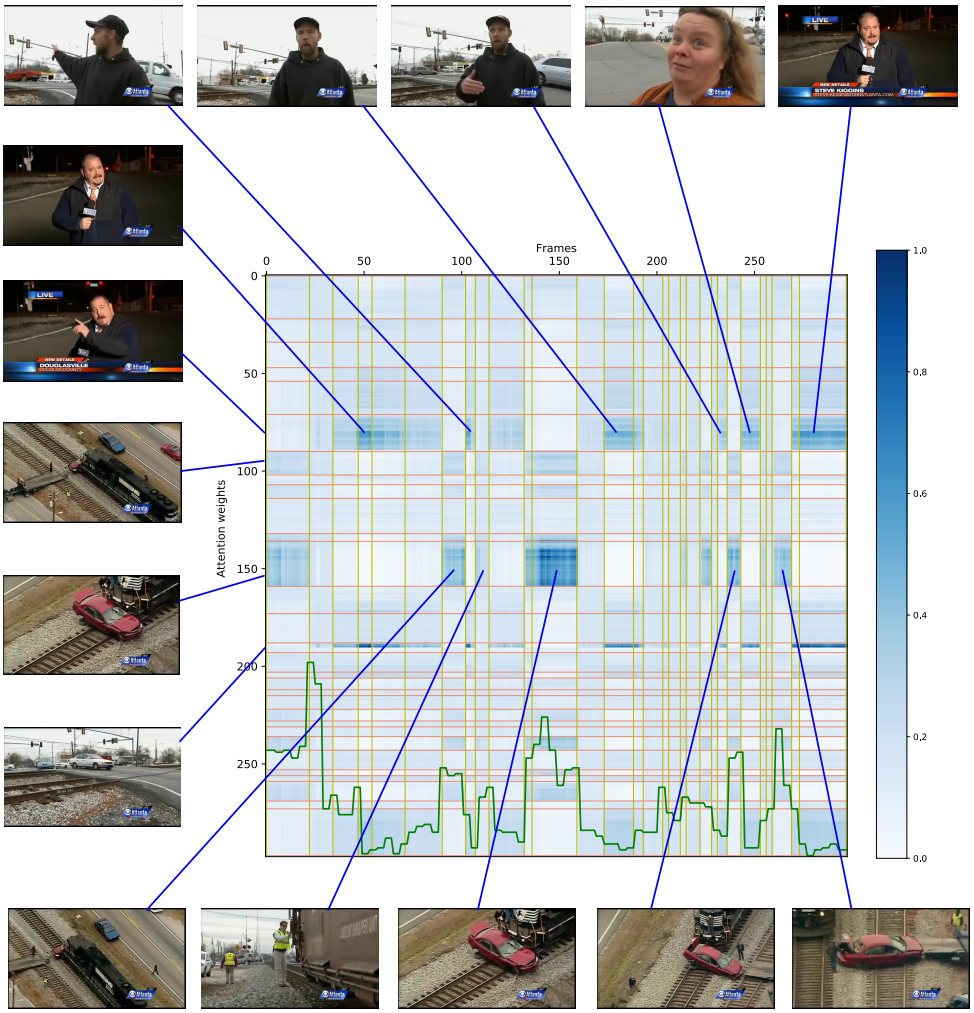


FIGURE 3.9: Confusion matrix of attention weights for TvSum video 7, test split 2. Green plot at the bottom shows the GT frame scores. Green and red horizontal and vertical lines show scene change points. Values were normalized to range 0-1 across the matrix. Frames are sub-sampled to 2fps.

video sequence as a function of the importance score of each frame. We show that on the supervised, keyshot video summarization task, our model outperforms the existing state-of-the-art methods on the TvSum and SumMe benchmarks. Given the simplicity of our model, it is easier to implement and less resource-demanding to run than LSTM encoder-decoder based methods (see Section 3.3.4), making it suitable for applications on embedded or low-power platforms.

The proposed model is based on a single, global, self-attention layer followed by two, fully connected network layers. We intentionally designed and tested the simplest architecture with global attention without positional encoding to establish a baseline method for this type of architecture. Limiting the aperture of the attention to a local region, introducing strictly causal attention (only past samples can be considered), or adding the positional encoding are simple modifications likely to further improve the performance.

Perhaps the biggest drawback of our method is the processing of all video frames

in a single batch. As mentioned earlier, limiting the attention aperture to local frames (either symmetric or process only already seen samples) would fully address this limitation. Another limitation of the VASNet method is its dependency on the external kernel temporal segmentation (KTS) and knapsack algorithms. This is not unique to our method but rather a common approach taken by most recent work, including the state-of-the-art method M-AVS Ji et al. (2017). Our method, however, appears to have the potential to mitigate this shortcoming, which we discuss in the Future Work Section 3.6.1.

Our contributions are:

- Proposed novel method for a sequence to sequence transformation for video summarization based on soft, self-attention mechanism. In contrast, the current state-of-the-art algorithms rely on complex LSTM/GRU encoder-decoder methods (as defined in Sections 1 and 3.3.4).
- Demonstration of the suitability of the proposed method to replace recurrent networks on the video summarization tasks.

The complete PyTorch 0.4 source code to train and evaluate our model, as well as trained weights to reproduce results in this paper, is publicly available on `https://github.com/ok1zjf/VASNet`.

3.6.1 Future Work

This work opened more research opportunities than expected. The VASNet method processes all frames simultaneously in a single batch, limiting the source video length. As follow-up work, we propose to add a limit for the number of past and future video frames included in the attention matrix with respect to the current frame. The attention would still be calculated for each video frame. We plan to evaluate two primary configurations: the limits admitting only past frames and the limits centered around the current frame.

To eliminate the need for the Kernel Temporal Segmentation (KTS) algorithm, we propose to leverage the sharp transitions in the attention matrix occurring between sets of frames with diverse internal content consistency. A comparison of these changes with the scene change points produced by KTS (green color) can be seen in Figure 3.9.

The VASNet evaluation presented in this chapter was conducted according to an established protocol followed by the majority of recent work. This included adopting CNN features for the video frame of the used datasets. These features were produced by the GoogLeNet network (Szegedy et al., 2015) trained on ImageNet (Russakovsky et al., 2015). Many modern CNN architectures have since superseded this model with higher performance in all ML domains (ResNet 152 top 1 accuracy on ImageNet 81.6% vs. GoogLeNet 66.5%). In light of this advancement, it would be valuable to evaluate the VASNet and the other benchmarked methods with CNN

features produced by a high-performance CNN model.

Along similar lines, we are considering further improving the performance of the CNN features by utilizing recent deep architectures for self-supervised representation learning (Grill et al., 2020, Chen and He, 2020, Caron et al., 2020). trained on very large datasets such as the Tencent ML-Images (Wu et al., 2019a) containing 17M images.

We believe that a good indicator of the importance score of the current video frame is the level with which other frames can be predicted from its content. Methods for video frame prediction (Villegas et al., 2019) could be used to learn and then measure the prediction error as an additional input to the frame importance score regression model.

While significantly diverging from the presented VASNet method, we believe that another promising future work direction would be an application of the Transformer method (Vaswani et al., 2017, Khan et al., 2021) to video summarization. The transformer would be applied to CNN features rather than the pixel space in this method. The summarization would then be conducted in two passes over the entire video length. First, the transformer learns its internal attention states over the full video sequence. Then, the model generates the key-frame summarization in the second pass. This type of few-shots learning, referred to as "in-context learning", has been recently proposed and successfully implemented and deployed in NLP by Brown et al. (2020).

Chapter 4

Learning Latent Discrete Representations

In this chapter, we study the feasibility of modeling and learning discrete latent representations with an unsupervised training method on datasets of images. Equally, we investigate methods enabling operations in the discrete latent space, such as sampling novel images, interpolating and modifying existing data sample attributes.

We mainly focus on learning binary latent representations since they appear to be attractive for many applications, for example the realization of an encoder for sparse distributed representations for methods such as the Hierarchical Temporal Memory (HTM) (Hawkins and Ahmad, 2016), modeling neurobiological processes (Bethge and Berens, 2008), data compression, memory addressing (Rae et al., 2016), for gating (hard attention)(Xu et al., 2015) or general representation learning (Bengio et al., 2012, 2013a). Binary features also have great potential in applications to energy-based memory models such as Hopfield networks (Hopfield, 1982) or Sparse Distributed Memory (SDM) (Kanerva, 1988).

In the search for better representation encoding for artificial neural network (ANN), we first looked at the elementary activation mechanism performed by the biological neuron and its encoding. Then, we explored encoding schemas for populations of neurons that can be loosely equated to the layer activations in ANN.

This chapter is structured as follows: First, in Section 4.1, we outline basic neurological data processing steps from neurons to the neural codes and then, in the subsections, detail the elementary activation function of the biological neuron, rate and temporal coding and also compare its computational power with the artificial neuron. Review of the neuroscientific work closes with two subsections describing the sparse and convolutional coding. In Section 4.2, we lay out the proposition behind our method. Related work in Section 4.3 is followed by Section 4.4, where we formalize our method for learning Bernoulli latent representations and describe the operations in latent space. Conducted experiments and results are presented in Section 4.5. In Sections 4.6 and 4.7, we explore similarities of our method with the Vector

Quantised-Variational Autoencoder (VQ-VAE) and then extend our view on the latent space as a space of discrete embeddings implicitly learned in a dictionary. The network diagram of our model in Section 4.8, followed by a conclusion in Section 4.9, ends this chapter.

4.1 Neural Processing Pathway

Data processing, whether in humans or machines, undertakes several phases. In a very simplistic view, for learning agents, it is the projection of sensory data to latent representations, binding with other sensory data and auto or hetero associations with already retained knowledge and consolidation in memory. For agents interacting with their environment, then also a motor output. In this work, we focus exclusively on the process of passive knowledge formation and accumulation for vision tasks. That is, we use snapshots of the environment for training and performance evaluation. We do not require the agent to interact with its environment.

Sensory input is first transformed from the raw sensory data, such as light intensities or audio signal frequencies, into an internal representation. For vision tasks, this would entail a transformation of the light intensities from the retina or a machine camera to representations expressing the presence and motion of edges and simple textures (Hubel and Wiesel, 1962). The nature of these features is learned during the early development stages in the primary visual cortex. These representations are believed to stabilize over the pruning period where the number of synaptic connections dramatically drop (Blakemore and Cooper, 1970) and then stay mostly fixed over the lifetime.

The neural coding in the primate visual cortex is distributed, sparse but highly dimensional, with strong signatures of minimum entropy encoding (Barlow et al., 1961, 1989, Olshausen and Field, 1996). This is not surprising, considering that functions of the vast majority of biological forms gravitate towards energy minimization, such as non-exercised muscles shrinking and synaptic connections without reinforcements weakening. Similarly, a population of neurons activated upon a perception of common events is reduced, leading to lower energy consumption. In particular, empirically observed cases, the neural activity is reduced to just a few active neurons known as concept neurons or grandmother neurons (Barlow, 1972, Quiroga, 2012). The function of procedural memory could also be attributed to energy minimization, where the more demanding, cognitive tasks, initially solved in the working memory, are transferred to the subconscious, implicit memory. Procedural memory does not require interaction with higher cognitive layers, which leads to lower neural activity and, consequently, lower energy consumption and faster response time. Finally, the data are associated with other sensory inputs, such as vision with somatosensory data, and then with the prior knowledge recalled from memory (Quiroga, 2016). The completed patterns of new experiences are retained in

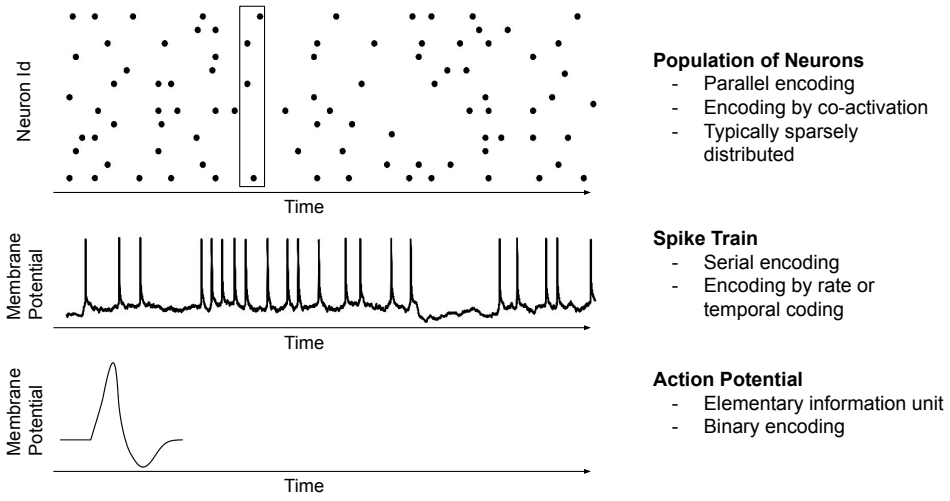


FIGURE 4.1: A simplified, bottom-up overview of the elementary neural codings.

short-term memory or trigger a behavioral response which can be an implicit action recalled from the procedural memory.

Neural coding is a central pillar of brain function, as is the feature encoding in the machine learning domain. There are many machine learning methods ranging from basic such as K-means or Gaussian Mixture Models (GMM) to deep learning methods that stay and fall with the encoding quality of the feature representations.

We will now review a number of neurological concepts relating to the Action Potential (AP), Spike Train, and Population Coding, as shown in Figure 4.1.

4.1.1 Biological and Artificial Neurons

While artificial neurons were inspired by their biological counterparts, they fundamentally differ in a number of ways. The role of a neuron is to receive, process, and transform information. Here, the most notable contrast between the artificial and biological neurons is that the former operates with continuous input and output, whereas the latter with trains of spikes. The spike, typically called Action Potential

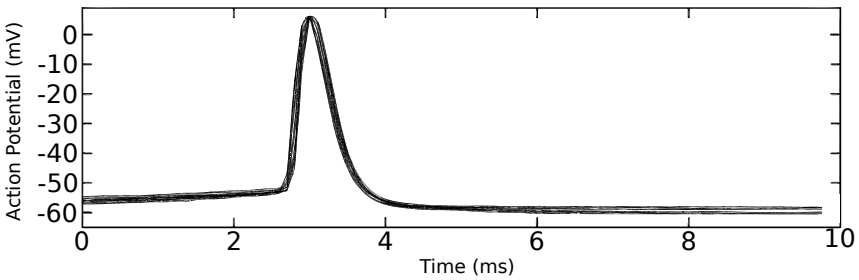


FIGURE 4.2: Action Potential (spike) travels down the axon to the axon terminals where it signals to other neurons over synaptic connections. Based on data from Toledo-Rodriguez et al. (2004).

(AP), is a narrow, approximately 1ms wide, uniform electrical pulse about 80mV, generated by neuron on its axon when the membrane potential (electrical gradient

between the interior and exterior of the biological cell) reaches a threshold value. An example of the spike is pictured in Figure 4.2.

The AP propagates down the axon and via synaptic connections to dendrites of postsynaptic neurons, where it triggers postsynaptic current. The current, carried by the dendrites, boosts or inhibits the Excitatory Postsynaptic Potential (EPSP). Contribution from all dendrites (spatial) and within a time window (temporal) is integrated. When the EPSP is reached, the cell generates AP. The absence of the postsynaptic current causes rapid decay of the EPSP.

The APs are combined into spike trains encoded either by rate coding (frequency coding) or spike time coding (temporal) (Perkel and Bullock, 1968). Durations of the stimulus and response are typically comparable, although some neurons generate a sustained response to a short stimulus (Robinson, 2015). An example of spike trains produced by the visual cortex as a response to visual stimuli is shown in Figure 4.3.

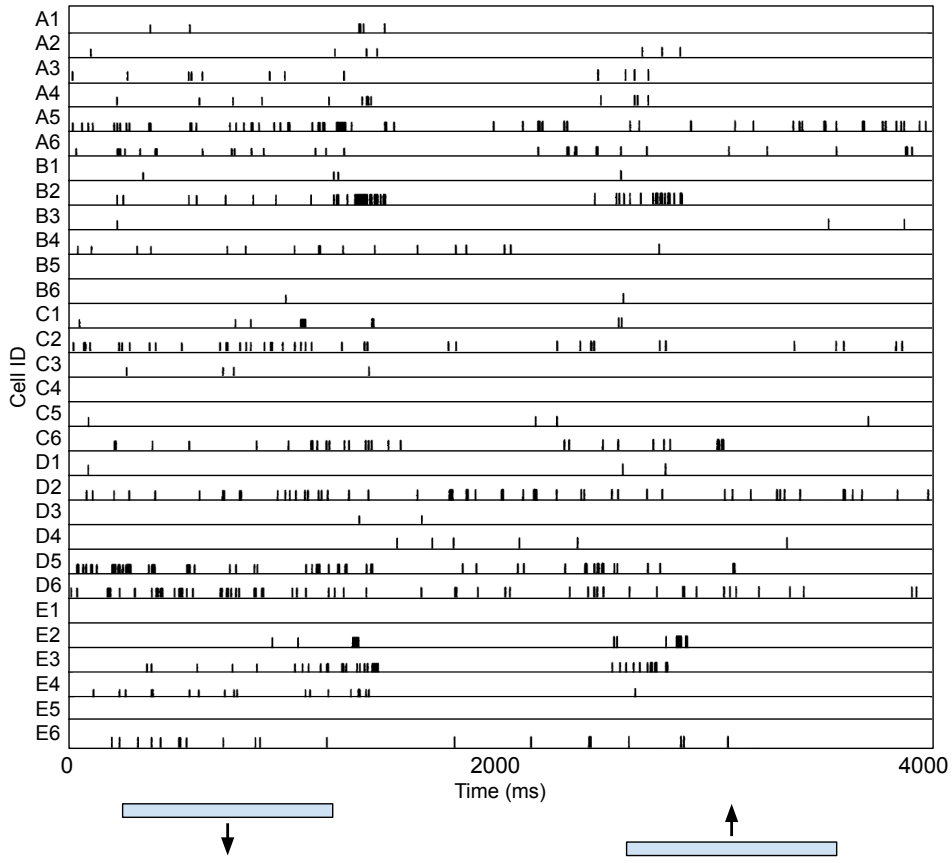


FIGURE 4.3: 4 seconds long spike trains from 30 neurons from a macake monkey cortex. These trains are responses to horizontal bars moving up and down in the monkey's visual field (shown at the bottom). Based on data from Kruger and Aiple (1988).

Historically, it has been assumed that the rate codes are the primary information carriers (Borst and Theunissen, 1999, Rolls and Treves, 2011). For example, it has been shown that a muscle contraction is solely proportional to the neural firing rate. This has been a convincing argument supporting the rate-coding-only hypothesis,

however, it may be a specific case of communication in the motor system. It has been argued recently that the rate coding may be too simplistic and that there is likely information encoded in the precise spike timing (Stein et al., 2005, Brette, 2015).

With respect to our work, we are interested in whether the biological neurons operate mainly in a continuous or discrete domain. Internally, the neuron is typically characterized as a thresholded, analog integrate-and-fire model. The thresholded EPSP indicates a discrete nature of the AP on the neuron's output (Hodgkin and Huxley, 1952). On its input, the postsynaptic neuron also receives binary APs; nevertheless, a spike train on the input could still be interpreted as a continuous value over a time window.

4.1.2 Sparse Distributed Representation

A sensory stimulus activates a number of neurons that, in the response, fire spike trains (Figure 4.3). This activity is distributed across a wide population of neurons that manifest considerable signs of sparsity in both the temporal activations (0.5% to 2%) and connectivity (1% to 10%) (Willmore and Tolhurst, 2001, Attwell and Laughlin, 2001, Lennie, 2003). The Sparse Distributed Representation (SDR) appears to be the primary encoding across the majority of brain regions, ranging from sensory inputs (Olshausen and Field, 2004, Hromádka et al., 2008, Weliky et al., 2003) to the neocortex (Barth and Poulet, 2012, Hulme et al., 2014, Foldiak, 2003), pre-motor areas (Graziano et al., 2002), and the primary motor system (Graziano and Aflalo, 2007). As expected, the encoding abstraction from the sensory input and the observed motor activity increases the deeper to the neocortex we move (Kiani et al., 2007).

SDR has been shown to be very robust to noise and also energy efficient, particularly with the minimum entropy coding (Olshausen and Field, 1996, Attwell and Laughlin, 2001, Marr, 1969). From the computational standpoint, the SDR has been a cornerstone of many methods (Kanerva, 1988, Golomb et al., 1990, Hawkins et al., 2019) precisely due to its robustness to the noise and a solid mathematical framework that validates these properties. In this work, we do not directly target the sparsity, however, we do design our method as such it can be easily enabled. In Section 4.4.4, we propose a regularization method to enforce it.

4.1.3 Correlation Coding

As information in the digital domain is expressed by bits, bytes, words, vectors of words, tensors, and higher data structures, in the brain, this would correspond to Action Potential (AP), spike trains, rate and temporal coding, and then population coding. We look at the population coding as a form of latent neural representation. There are several hypothesized neural coding schemas (Averbeck et al., 2006, Panzeri et al., 2015). Our work draws main inspiration from the correlation coding due to its resiliency to noise and computational efficiency The correlation coding also allows a

relatively straightforward parametrization of the underlying data space distribution by the maximum entropy model.

Substantial evidence of the correlation coding has been demonstrated by Schneidman et al. (2006). Over many experiments, the researchers documented that even weak, pairwise correlations of the retinal ganglion cells (RGC) in the vertebrate retina could explain 90% of the network interactions. These results were achieved with a pairwise (second-order) maximum entropy model, further indicating that higher interaction orders, necessary to express problems of the XOR complexity and higher, would perform better. The authors used the maximum entropy model to verify that, indeed, the second-order model was sufficient to express the recorded neural code. An example of correlated neural activations and comparison with their simulation by the maximum entropy model is shown in Figure 4.4. Shlens et al. (2006) improved on the Schneidman et al. (2006) experimental methodology, reporting a substantially higher proportion of the network interactions (98%-99%) that the correlation coding could explain.

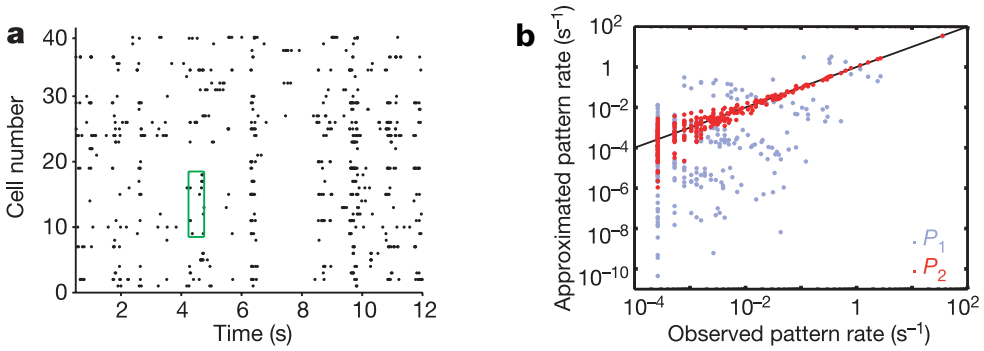


FIGURE 4.4: (**a**) A segment of the simultaneous responses of 40 retinal ganglion cells in the salamander to a natural movie clip. Each dot represents the time of an action potential. (**b**) Rate of occurrence of each firing pattern of cells from the green box in **a**, predicted by the maximum entropy model P_2 is plotted against the measured rate. Sourced with permission from Schneidman et al. (2006).

Ruda et al. (2020), building on similar work by Franke et al. (2016), investigated the impact of correlated noise on the population coding in RGC of rats under day and moonlight conditions. This study revealed that ignoring the pairwise interactions during the more noisy moonlight stimuli resulted in significantly less information up to the point when a single RGC performed better than the entire population.

Due to the ease of experimental setup, most of the work on correlation coding is typically conducted on the RGC population. Consequently, the results may not be entirely applicable to other brain regions such as the cortical neurons, which may exhibit specific interactions, depending on layer and cell type, as shown by Yoshimura and Callaway (2005). Therefore, while the correlation codding served as important inspiration in our work, we took it cautiously, merely as a hint rather than a solid method for replication in ANN.

4.2 Unsupervised Discrete Representations Learning

Our insights from the information encoding in the brain, spanning from an individual neuron to their population, solidified our initial intuition that the latent encoding should take discrete form. Particularly influential was the work on information representation in the brain by Tee and Taylor (2020) that provides theoretical support for the discrete neural communication in the brain as the only feasible encoding. Equally, our work on the latent binary representation learning and sampling was motivated by the neural correlation coding due to its resiliency to noise and relatively straightforward parametrization of the underlying data space distribution by a pairwise (second-order) correlation model. Furthermore, we are going to leverage the unsupervised learning due to the ease of application of our method to any unlabelled data, with a plethora of publicly available datasets readily available for training and validation.

There are many machine learning algorithms with a potential to benefit from low dimensional, highly expressive features, whether for object detection, classification, reinforcement learning, or as generative models for compression, super-resolution, or novel sample generation. This direction has been successfully pursued with autoencoder models and particularly the Variational Autoencoder (VAE) (Kingma and Welling, 2014) and its derivatives. Recently, fully deterministic, regularized (Ghosh et al., 2020) and discrete, VQ-VAE (van den Oord et al., 2017) have been proposed, demonstrating performance comparable to theirs stochastic counterparts. In this work, we focus on the deterministic class of autoencoders, learning a discrete latent representation, specifically the multivariate Bernoulli distribution. Our model is trained with a gradient-based method in end-to-end fashion without enforcing any prior on the latent space.

Current neural network learning algorithms are almost exclusively based on very successful gradient-based learning methods. However, the need for the differentiability of each layer represents a challenge if one desires to train stochastic neurons or other non-differentiable functions such as quantization. A number of techniques have been proposed, allowing gradient propagation through such neurons such as re-parametrization (Kingma and Welling, 2014), surrogate gradient functions (Bengio et al., 2013b), or continuous relaxation of non-differentiable nodes (Jang et al., 2017). In our method, we follow the approach behind the straight-through estimator (Bengio et al., 2013b, Hinton, 2012) due to its conceptually simple setup.

Sampling from and interpolating in the discrete latent space is equally challenging. Unlike multimodal, Gaussian, and many other real-valued distributions, the multivariate Bernoulli distribution concentrates most of the information on the second and higher moments, since the marginals are strictly unimodal and entirely described by the mean $p = \mathbb{E}[\mathrm{b_i}]$, directly giving rise to variance $Var[b_i] = p(1-p)$

for Bernoulli variable b_i at dimension i. This also seems to play a vital role in biological neurons, where the binary, pairwise correlations provide strikingly accurate encoding for neuronal firing patterns in the primate retina (Schneidman et al., 2006, Shlens et al., 2006, Nirenberg and Victor, 2007).

Given that our model learns a distribution with unknown prior, and based on the aforementioned premise, we propose to parametrize the learned distribution by its first two moments, also motivated by the cross moment model by Mishra et al. (2012). These parameters are learned from latents encoded on the training data. To sample and interpolate in the multivariate Bernoulli latent space, we propose a novel method based on a random hyperplane rounding technique derived from the MAX-CUT algorithm (Goemans and Williamson, 1995). Within this work, we abbreviate the Latent Bernoulli Autoencoder as LBAE.

We evaluate our method on the CelebA and CIFAR-10 datasets and, for completeness, the MNIST. We show that our method is competitive with the current state-of-the-art variational and deterministic autoencoders. Our model shows high performance, particularly on the interpolation task, which is remarkable, considering we are operating in the discrete latent space. To the best of our knowledge, none of the existing discrete autoencoders can perform sensible interpolation in the latent space. For example, a state-of-the-art method VQ-VAE (van den Oord et al., 2017), does not suggest how to do so, and even explicit methods, as in Berthelot et al. (2019), admit difficulties in accomplishing this task. Finally, we present a simple method for attribute modification in the latent space, also showing competitive results.

4.3 Related Work

Unsupervised representation learning has been successfully pursued with autoencoder models, particularly the VAE model (Kingma and Welling, 2014) due to its simplicity and well defined probabilistic framework. VAE unfortunately suffers from number of issues, most notably producing blurred images (Dumoulin et al., 2017) and posterior collapse (Razavi et al., 2019a). A number of methods have been proposed to improve the image quality with reconstruction loss based on perceptual similarity in the feature space of an external CNN (Dosovitskiy and Brox, 2016, Hou et al., 2017) or in its own latent space (Zhang et al., 2019). Success of the Generative Adversarial Networks (GAN) to learn image distribution motivated application of the adversarial training to the latent space distribution in the Adversarial Autoencoders (AAE) (Makhzani et al., 2016) and its generalization in Wasserstein Autoencoders (WAE) (Tolstikhin et al., 2018). More recently Dai and Wipf (2019) introduced a 2 stage VAE where the second stage learns the latent space distribution, in principle, performs a density estimation. From the work of Ghosh et al. (2020) it is apparent that deterministic autoencoders are competitive with the VAE and its derivatives, only for the price of ex-post density estimation.

Most of the methods learn real-valued latent space owning to the established gradient-based optimizations. VQ-VAE (van den Oord et al., 2017) is perhaps the first competitive deterministic, autoencoder that learns discrete representations. As in Ghosh et al. (2020), this method does not impose any prior on the learned latent distribution, thus it requires some form of external post density estimation. Authors propose the PixelCNN (Van den Oord et al., 2016), an autoregressive density estimator which learns a categorical prior over the stored latents encoded from the training dataset.

Learning discrete representations with a gradient-based optimization is not straightforward. Bengio et al. (2013b) proposed four methods, addressing the learning through stochastic neurons, most notably the straight-through gradient estimator, originally described by Hinton (2012). The straight-through estimator is also used in the VQ-VAE model to allow gradient flow over non differentiable, nearest neighbour operation in the forward pass. Chung et al. (2017) then introduces a straight-through estimator with the slope annealing extension. Over the training period, this method gradually reduces the difference between the non-differentiable function in the forward pass and the surrogate in the backward pass to converge to the discrete distribution in the limit. This method is somewhat similar to the ST Gumbel-Softmax (Jang et al., 2017). The Gumbel-Softmax was also applied to the autoencoder model in JointVAE by Dupont (2018).

4.3.1 Current State-of-the-Art, its Limitations and Promising Research Directions

The current state of the art in the domain of deterministic discrete representation learning is the VQ-VAE model. To sample from its discrete latent distribution and generate new samples, an additional autoregressive model, PixelCNN, needs to be trained on the discrete latent space. This method is not an end-to-end trainable generative model but rather a composition of two methods: discrete latent space encoder and generator, both independently trained.

The core limitations manifested by the current representation learning methods could be summarized as:

- Learned latent representations have lower utility across diverse domains (classification, segmentation, image captioning, visual question answering) compared to latents learned with supervision on a specific task.
- Inability to produce disentangled representations. (values of the disentangled representations are easily interpretable with respect to the content of corresponding images).
- A prior distribution imposed on the latent space during learning, such as the multivariate Gaussian in the case of VAE (Kingma and Welling, 2014), cannot sufficiently express the underlying data distribution. Such approaches require

a compromise between the divergence of the prior and learned distributions and the reconstruction loss typically (Matthey et al., 2017) leading to high reconstruction error or high divergence from the prior distribution and consequently high generative and interpolation error (high FID score).

- Difficulty to parametrize and sample from an unknown latent distribution learned by models without a known prior distribution enforced on the latent space, such as the vanilla autoencoder (AE). Moreover, such methods typically restrict access to the latent space for operation, such as the attributes modification and interpolation.

Recent advancements in non-autoencoding unsupervised learning methods are likely to pave a path to the future of methods for discrete representation learning. This is due to the fact that unsupervised methods such as Grill et al. (2020), Chen and He (2020), Caron et al. (2020), Zbontar et al. (2021) learn representations that encode semantic information while autoencoding methods tend to develop an internal encoding sufficient to reconstruct the source data points. These methods would still need to be coupled to a quantization network to produce the discrete representations. Also, an external generator would have to be trained to learn the latent distribution for sampling. As with the VQ-VAE, these approaches still do not enable direct operations in the latent space, such as attributes modification and interpolation. This is due to the unknown latent space distribution.

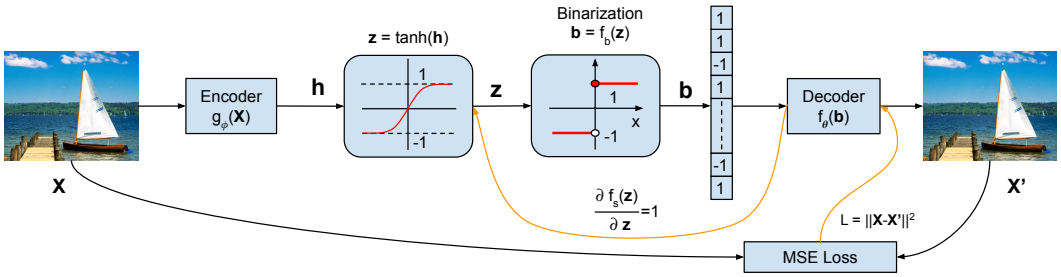


FIGURE 4.5: For N dimensional latent space the information bottleneck of a typical autoencoder is in LBAE replaced with tanh() followed by binarization $f_b() \in \{-1, 1\}^N$ with unit gradient surrogate function $f_s()$ for backward pass.

4.4 Bernoulli Latent Space

4.4.1 Learning the Bernoulli Latent Space

The base of our method is a deterministic autoencoder with encoder $\mathbf{z} = g_\phi(\mathbf{X})$, parametrized by ϕ, that produces typically real-valued latent representation $\mathbf{z}$ for input $\mathbf{X}$. Decoder $\mathbf{X}' = f_\theta(\mathbf{z})$, parametrized by θ, attempts to reconstructs $\mathbf{X}$ from $\mathbf{z}$. Our model is trained with a single, common objective function $\mathcal{L}(\theta, \phi) = \mathbb{E}[L(\mathbf{X}, \mathbf{X}')]$, where L is the reconstruction loss function. To discretize an N dimensional latent

$\mathbf{z} \in \mathbb{R}^N$ into the binary range $\mathbf{b} \in \{-1, 1\}^N$ we threshold $\mathbf{z}$ at zero as follows:

$$b_i = f_b(z_i) = \begin{cases} 1, & \text{if} \quad z_i \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{4.1}$$

We choose to represent the binary values by states $\{-1, 1\}$ rather than $\{0, 1\}$ due to its computational benefits such as zero being the threshold level and $b_i^2 = 1 \, \forall i$. The latents can be easily converted between these two ranges without loss of information.

Since $f_b()$ is not differentiable, we define a surrogate differentiable function $f_s(\mathbf{z}) = \mathbf{z}$ with unit gradient $\nabla_{\mathbf{z}} f_s = 1$ operating in the same domain as $f_b()$; $f_s()$ is then used in the backward pass. During the backpropagation this allows the gradient to flow through the binarization operation and lets the encoder correct its output in the direction of the binarized quantities read by the decoder. The rounding during the binarization brings an additional error that is not corrected during the backpropagation and manifests as noise. This noise can be reduced by lowering the learning rate but it slows down the training or hinders the convergence altogether. To alleviate this weakness we add $\tanh()$ before the binarization, which limits the gradient flow from the decoder and minimizes the optimization overshoot during the gradient descent.

4.4.2 Sampling Correlated Multivariate Bernoulli Latents

Our goal is to implement a generative model of the form

$$\mathbf{b} \sim p(\mathbf{b}), \quad \mathbf{b} \in \{-1, 1\}^N \tag{4.2}$$

$$\mathbf{X} \sim p(\mathbf{X} \mid \mathbf{b}; \theta), \tag{4.3}$$

where $\mathbf{X}$ is generated image, $\mathbf{b}$ an N-dimensional binary latent vector and θ parameters of the generator. Unlike VAE, we do not enforce any prior on the latent space during the training, thus the learned distribution $p(\mathbf{b})$ is unknown. Therefore, to efficiently sample novel latents we first learn $p(\mathbf{b})$ from the distribution of the training dataset in the latent space and parametrize it by its first two moments.

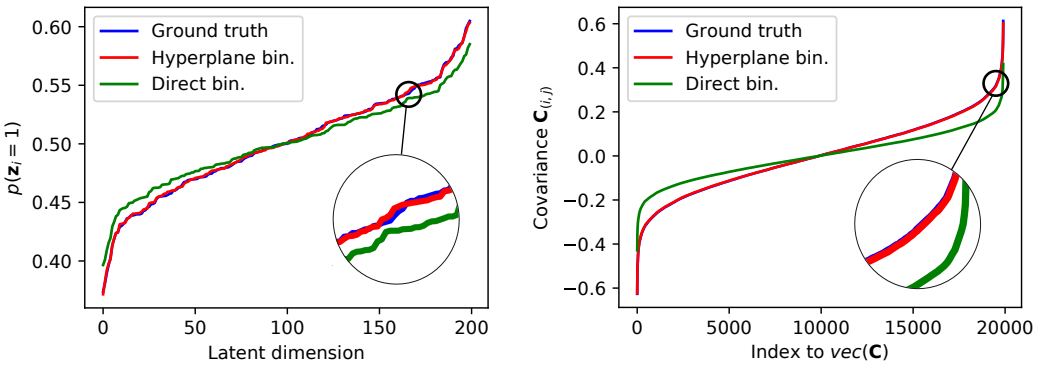
The direct way to learn and sample from the correlated Bernoulli distribution would be to approximate it as a Gaussian distribution with the binarization step. Let us consider a matrix $\mathbf{Y} \in \{-1, 1\}^{(N \times K)}$ of K N-dimensional latent vectors encoded on training data. Given expected value $\mathbb{E}[\mathbf{Y}] \in R^N$ and covariance Σ we can sample latent $\mathbf{b}$ from the distribution as:

$$\Sigma = \mathbb{E}[\mathbf{Y}\mathbf{Y}^{\mathrm{T}}] - \mathbb{E}[\mathbf{Y}]\,\mathbb{E}[\mathbf{Y}]^T \tag{4.4}$$

$$\mathbf{z} \sim \mathcal{N}_N(0, \mathbf{I}_N) \tag{4.5}$$

$$\mathbf{b} = f_b(\mathbf{L}\mathbf{z} + \mathbb{E}[\mathbf{Y}]), \quad \mathbf{b} \in \{-1, 1\}^N, \tag{4.6}$$

where $\Sigma = \mathbf{L}\mathbf{L}^T$ is a lower triangular Cholesky decomposition. This approach, however, does not produce Bernoulli samples with the correct distribution. To mitigate this issue, we propose a method inspired by the cross moment model method (Mishra et al., 2012) and random hyperplane rounding technique for MAX-CUT (Goemans and Williamson, 1995). In Figure 4.6 we can see that a distribution generated by the direct binarization (Eq. 4.6) (green) exhibits noticeable error compared to the ground truth (blue). The red plot shows distribution generated with the proposed random hyperplane method.



Sorted marginal probabilities

Vectorized, sorted covariances

FIGURE 4.6: Ground truth (200bits latents, MNIST train data) and the distribution sampled with the random hyperplane method appear identical while the direct rounding method exhibits a clear error. Note the ground truth (blue) is mostly hidden behind the red.

Our method can be summarized in the following three steps: (**1**) parametrize distribution of the training dataset in latent space by first two moments, (**2**) relax each latent dimension to an unit vector on a hypersphere with a position corresponding to its correlation with other dimensions, (**3**) sample latent $\mathbf{b}$ by randomly splitting the sphere through the centre with a hyperplane normal $\mathbf{r}$ and assigning binary state 1 to dimensions corresponding to vectors on one side of the plane and -1 to the rest.

The distribution of $\mathbf{Y}$ is parametrized by first moments and second non-central moments, similar to Mishra et al. 2012, in matrix $\mathbf{M}$ as:

$$\mathbf{M} = \begin{bmatrix} \mathbb{E}[\mathbf{Y}\mathbf{Y}^{\mathrm{T}}] & \mathbb{E}[\mathbf{Y}] \\ \mathbb{E}[\mathbf{Y}]^T & 1 \end{bmatrix}, \mathbf{M} \in [-1,1]^{(N+1)\times(N+1)}. \tag{4.7}$$

For N dimensional latent space we generate $N+1$ unit length vectors on sphere $S^{(N+1)}$. These vectors are organized as rows in matrix

$$\mathbf{V} \in \mathbb{R}^{(N+1)\times(N+1)}, \forall i \in [1,..,N+1], \|\mathbf{V}_i\| = 1, \tag{4.8}$$

where $\mathbf{V}_i$ is an i^{th} row of $\mathbf{V}$. Each vector $\mathbf{V}_i$ represents one dimension in the latent space. This is graphically shown in Figure 4.7. We express the covariances $\mathbf{M}$ as probabilities of vectors $\mathbf{V}_i, \mathbf{V}_j$ pointing in the same or opposite direction. For

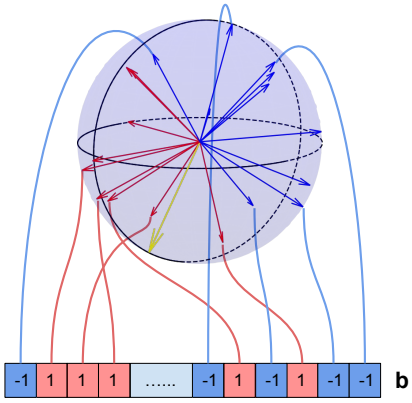


FIGURE 4.7: Each dimension in the latent space is represented by an unit vector on a hypersphere. Pairwise correlations are given by angle between vectors; the smaller angle the higher correlation between corresponding dimensions.

positive, high covariance between dimensions i and j the angle $\alpha_{i,j}$ between corresponding vectors $\mathbf{V}_i$ and $\mathbf{V}_j$ will be small and $P(\mathbf{V}_i, \mathbf{V}_j) \to 1$, while for negative covariance $\alpha_{i,j} \to \pi$ with $P(\mathbf{V}_i, \mathbf{V}_j) \to 0$. For non correlated dimensions $\mathbf{V}_i \perp \mathbf{V}_j$ with $P(\mathbf{V}_i, \mathbf{V}_j) \approx \frac{1}{2}$. Bits of positively correlated dimensions share the same state (-1 or 1) while negatively correlated take opposite states. We set the probabilities as:

$$P(\mathbf{V}_i, \mathbf{V}_j) = \frac{M_{i,j} + 1}{2}, \forall(i,j), P(\mathbf{V}_i, \mathbf{V}_j) \in [0,1] \tag{4.9}$$

and express them as a function of the angle $\alpha_{i,j}$ or dot product $\langle \mathbf{V}_i, \mathbf{V}_j \rangle$.

$$P(\mathbf{V}_i, \mathbf{V}_j) = 1 - \frac{\alpha_{i,j}}{\pi}, \forall(i,j),\ \alpha_{i,j} \in [0,\pi], \tag{4.10}$$

$$= 1 - \frac{\cos^{-1}(\langle \mathbf{V}_i, \mathbf{V}_j \rangle)}{\pi}. \tag{4.11}$$

We define the dot products as Gram matrix

$$H_{i,j} = \langle \mathbf{V}_i, \mathbf{V}_j \rangle,\ \mathbf{H} \in \mathbb{R}^{(N+1)\times(N+1)} \tag{4.12}$$

which, as a function of $\mathbf{M}$ is

$$H_{i,j} = \cos\left(\left(1 - \frac{1}{2}(M_{i,j}+1)\right)\pi\right) \tag{4.13}$$

$$= \cos\left(\frac{\pi}{2}(1 - M_{i,j})\right). \tag{4.14}$$

To obtain $\mathbf{V}$ we perform a square root of $\mathbf{H}$ by lower triangular Cholesky decomposition

$$\mathbf{H} = \mathbf{V}\mathbf{V}^T \quad s.t. \quad \mathbf{H} \succcurlyeq 0, \tag{4.15}$$

where $\mathbf{V}$ is a row-normal lower triangular matrix with rows being the desired unit

vectors on $S^{(N+1)}$. The $\mathbf{V}_{(N+1)}$ represents the boundary conditions for the first moments $\mathbb{E}[\mathbf{Y}]$. Concretely, it defines the positive hemisphere in S where all vectors receive positive binary state. In other words, this boundary vector orientates the hypersphere space according to the marginals $\mathbb{E}[\mathbf{Y}]$. Finally, to generate a novel la-

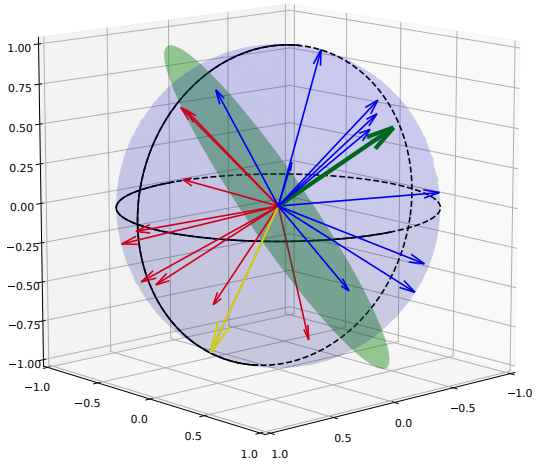


FIGURE 4.8: New samples are generated by splitting the sphere with a random plane (green) and assigning positive states to dimensions (red) on the side of the plane shared by the boundary vector (yellow) and negative to the rest (blue).

tent $\mathbf{b}$ we split the sphere S with a random plane through the center and then assign positive binary states to latent dimensions represented by vectors $\mathbf{V}_i$ in one hemisphere and negative to the rest. Vectors sharing hemisphere with $\mathbf{V}_{(N+1)}$ (yellow in Figure 4.8) will receive positive values. For a random hyperplane given by normal $\mathbf{r} \sim \mathcal{N}_{(N+1)}(0, \mathbf{I}_{(N+1)})$ (green in Figure 4.8) we generate the latent $\mathbf{b}$ with bits at each dimension as:

$$b_i = \begin{cases} 1, \text{ if } f_b(\langle \mathbf{V}_i, \mathbf{r} \rangle) = f_b(\langle \mathbf{V}_{(N+1)}, \mathbf{r} \rangle) \\ -1, \text{ otherwise} \end{cases}, \tag{4.16}$$
$$\forall i \in [1, .., N], \mathbf{r} \in \mathbb{R}^{(N+1)\times 1}.$$

In vector form the Eq. 4.16 is then:

$$\begin{aligned} &\mathbf{b} = f_r(\mathbf{r}) = f_b(\mathbf{V}\mathbf{r})_{-(N+1)} f_b(\mathbf{V}_{(N+1)}\,\mathbf{r}), \\ &\mathbf{b} \in \{-1, 1\}^N, \end{aligned} \tag{4.17}$$

where subscript $_{-(N+1)}$ denotes all but $(N+1)$ dimension. The expression $f_b(\langle \mathbf{V}_{(N+1)}, \mathbf{r} \rangle)$, and its vectorized form $f_b(\mathbf{V}_{(N+1)}\,\mathbf{r})$, returns the boundary decision bit. If positive, the hyperplane normal $\mathbf{r}$ is located in the same hemisphere as the boundary vector $\mathbf{V}_{(N+1)}$. Finally, an image $\mathbf{X}'$ is decoded from the binary latent $\mathbf{b}$ as $\mathbf{X}' = f_\theta(\mathbf{b})$. The generative process flow is illustrated in Figure 4.9.

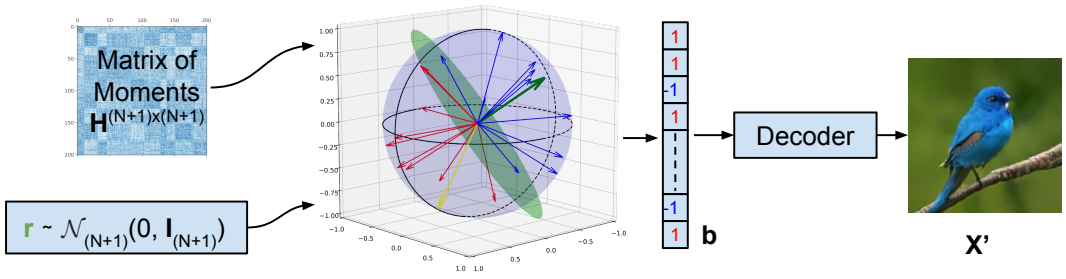


FIGURE 4.9: The latent space is parametrized by matrix **H**. where each dimension is represented by an unit vector on a hypersphere.

4.4.3 Interpolation in the Bernoulli Latent Space

To interpolate between two images we first generate their latent representations with the encoder. For each latent we lookup a hyperplane normal responsible for generating that latent vector according to Section 4.4.2. Intermediate latents are then interpolated between endpoints on the $S^{(N+1)}$ sphere with spherical linear interpolation (SLERP) (Shoemake, 1985). Interpolation from image with latent $\mathbf{s}$ and hyperplane vector $\mathbf{r}_s$ to image with latent $\mathbf{t}$ and hyperplane $\mathbf{r}_t$ is shown in Figure 4.10.

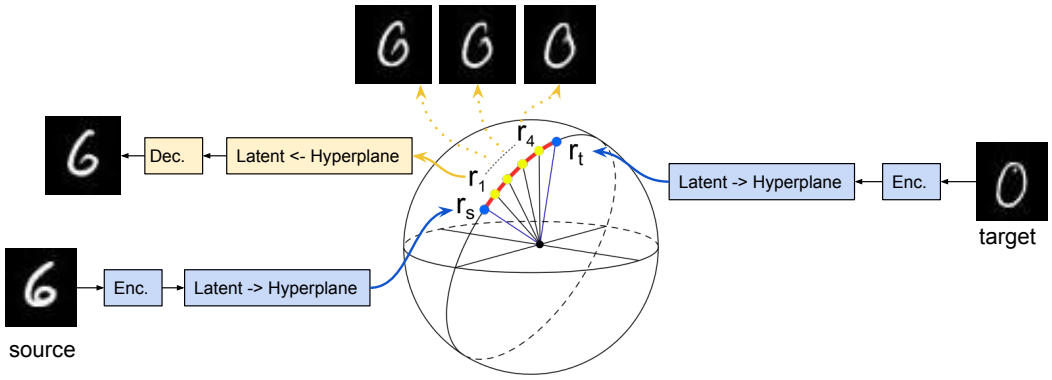


FIGURE 4.10: Spherical interpolation on sphere between source and target images represented by hyperplane vectors $\mathbf{r}_s$ and $\mathbf{r}_t$.

Let us consider $\mathbf{s} \in \{-1, 1\}^N$ to be our latent vector for which we desire to find a hyperplane normal $\mathbf{r} \in \mathbb{R}^{(N+1)}$ that would generate back the latent $\mathbf{s}$ as per Eq. 4.16. Intuitively, one could attempt to find the solution as $\mathbf{r} = \mathbf{V}^{-1}[\mathbf{s}, 1]$ which, indeed, recovers $\mathbf{s}$ back as: $\mathbf{s} = f_r(\mathbf{r})_{-(N+1)}$, where $[\mathbf{s}, 1]$ denotes $\mathbf{s}$ concatenated with 1. By setting the boundary decision bit to positive state $[\mathbf{s}, 1]$ we will get a hyperplane normal in the same hemisphere as the boundary vector $\mathbf{V}_{(N+1)}$, consequently this hemisphere represents positive binary states.

The hyperplane found this way is, however, not suitable for interpolation. Interpolating between such hyperplanes produces exact copies of the source latent until the midpoint where the latent vector instantly flips to the target latent and stays there until the end of interpolation. In our experiments we found that the

source/target flip happens over less than $1/10^6$ degrees step. It appears that the hyperplanes found this way are degenerate in some sense. They produce latents very far from the main distribution manifold. To find the nature of this behaviour is a subject of ongoing research.

Instead, we found that the most suitable latent-hyperplane inversion can be carried out by placing the hyperplane normal close to the centroids of the positive and negative vectors in $\mathbf{V}$. First, we get the centroid for all positive vectors in $\mathbf{s}$.

$$\mathbf{r}_p = \sum_i^N \left(u(s_i)\,\mathbf{V}_i\right)^T,\ \mathbf{r}_p \in \mathbb{R}^{(N+1)\times 1}, \tag{4.18}$$

where $u(s_i) = \frac{1}{2}(1+s_i)$ changes the range of its argument from $\{-1,1\}$ to $\{0,1\}$. $\mathbf{r}_p$ is a prototype of the hyperplane normal but it typically does not reproduce $\mathbf{s}$ accurately, causing reconstruction error in the pixel space when decoded. To mitigate this, we propose an iterative process that tilts the normal $\mathbf{r}_p$ towards the vectors incorrectly placed behind the hyperplane. The process stops when the Hamming distance between $\mathbf{s}$ and $f_r(\mathbf{r}_p)$ does not decrease, typically within < 4 steps. Similarly, we create a normal for the negative vectors $\mathbf{r}_n = \sum_i^N \left(u(-s_i)\,\mathbf{V}_i\right)^T$. The target normal is then:

$$\mathbf{r} = \frac{\mathbf{r}_p}{\parallel \mathbf{r}_p \parallel} - \frac{\mathbf{r}_n}{\parallel \mathbf{r}_n \parallel}. \tag{4.19}$$

A vector of error bits between $\mathbf{s}$ and its reconstruction with hyperplane normal $\mathbf{r}$ is calculated as:

$$E_b(\mathbf{s},\mathbf{r}) = u(-f_r(\mathbf{r}) \odot \mathbf{s}), \tag{4.20}$$

where $\odot$ is the Hadamard product. The Hamming distance d is then:

$$d = \sum_{i=0}^{N} E_b(\mathbf{s},\mathbf{r})_i,\ d \in \{0,...,N\}. \tag{4.21}$$

Algorithm 2 summarizes the process of looking up a hyperplane normal for a given Bernoulli latent vector and $\mathbf{V}$.

Algorithm 2 Latent **s** to hyperplane normal **r** inversion

procedure LATENT_TO_HYPERPLANE(**s**, **V**)
  $\mathbf{r} = \sum_i^N (u(s_i)\,\mathbf{V}_i)^T$ ▷ **r** is a mean vector of rows in **V** at $s_i = 1$ (Eq. 4.18)
  $d_{best} = N$ ▷ Start with the maximum Hamming distance for N dimensional vector.)
  repeat
    $\mathbf{e} = E_b(\mathbf{s}, \mathbf{r})$ ▷ Error between **s** and its reconstruction with hyperplane **r** (Eq. 4.20)
    $d = \sum_i^N e_i$ ▷ Hamming distance
    if $d \geq d_{best}$ **then**
      return r ▷ If the distance does not improve, return the hyperplane normal **r**
    $\mathbf{r} = \mathbf{r} + \sum_i^N (e_i\,\mathbf{V}_i)^T$ ▷ Add vectors at the error bits positions
    $\mathbf{r} = \mathbf{r} \,/\, \| \mathbf{r} \|$
    $d_{best} = d$
  until *True*

We then interpolate T normals between source $\mathbf{r}_s$ and target $\mathbf{r}_t$ on the hypersphere, and for each generate a latent vector according to Eq. 4.17 and decode it as an image $\mathbf{X'}_i = f_\theta(\mathbf{r}_i)$.

4.4.4 Enforcing Sparsity by Regularization

Sparsity reaches 50% across the latent dimensions without any regularization as apparent in the Figure 4.6. This is expected since the encoder and decoder try to use full channel capacity. Enforcing sparsity with the proposed method is accomplished by adding a loss term that restricts the channel capacity by penalizing activations of a given number of bits in the latent space.

The final training loss $\mathcal{L}$ is a composition of the image reconstruction loss $\mathcal{L}_R$ and a sparsity loss $\mathcal{L}_S$ defined as:

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_S, \tag{4.22}$$

where λ weighs the importance of the sparsity loss with respect to the reconstruction loss. The image reconstruction loss $\mathcal{L}_R$ is a mean square error between the original **X** and reconstructed images **X′** over set of K images in the training batch.

$$\mathcal{L}_R = \frac{1}{K} \sum_{i=0}^{K} (\mathbf{X}_i - \mathbf{X'}_i)^2 \tag{4.23}$$

The sparsity loss $\mathcal{L}_S$ across the training batch is a mean square loss of sparsity losses L_i of all K latents in the batch. This ℓ_2 penalty ensures similar sparsity in each latent vector. The loss is defined as:

$$\mathcal{L}_S = \frac{1}{K} \sum_{i=0}^{K} L_i^2. \tag{4.24}$$

Sparsity loss of each latent L_i is setup as a modified sigmoid function with parametrized range shift and the transition slope

$$L_i = \frac{1}{1 + e^{g(p\sqrt{2} - a_i)}} \tag{4.25}$$

where g defines a loss gradient between the number of active and inactive bits, specified as a ratio p. The $\sqrt{2}$ term shifts the exponential elbow to the position of desired sparsity ratio. The a_i term is a number of active bits in the latent i calculated as

$$a_i = \frac{1}{2N} \sum_{j=0}^{N} \mathbf{B}_{i,j} + 1, \tag{4.26}$$

where $\mathbf{B} \in \{-1, 1\}^{(K,N)}$ is a training batch of K latents, each of dimension N. Here, the sum over the active bits servers as a ℓ_1 regularization, contributing to the sparsity promotion. In our experiments on CIFAR-10 we set the batch size $K = 512$, gradient $g = 100$ and sparsity limit $p = 0.1$, that is 10% of the latent dimensions. Figure 4.11 shows the sparsity loss function L_i for a latent space with dimension $N = 1000$ and parameters $g = 100$ and $p = 0.1$. We have conducted several cursory experiments

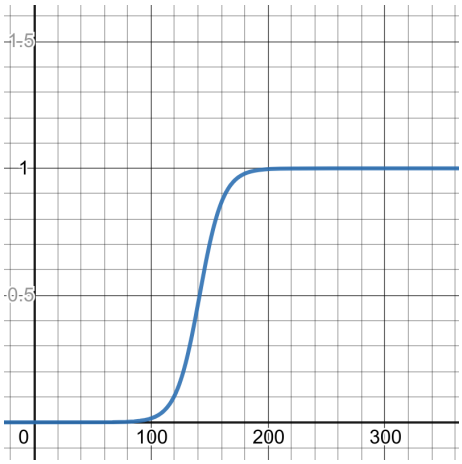


FIGURE 4.11: Loss function with transition gradient $g = 100$ enforcing sparsity bellow 10% in latent space with 1000 dimensions

with this loss on the CIFAR-10 dataset where we enforced 10% sparsity while maintaining the original number of dimensions $N = 600$. As expected, the reconstruction quality deteriorated by about 10 Fréchet Inception Distance (FID) compared to the dense latents. Increasing the latent dimensionality to $N = 3000$ improved the performance comparable to the dense latent representations.

4.4.5 Contributions to the Field of Neural Architectures

LBAE leverages a typical vanilla convolutional autoencoder with a novel binarization layer trained end-to-end with stochastic gradient descent. A second key contribution, albeit not directly utilizing neural architecture, is a novel algorithm for

parametrizing and sampling from the latent binary distribution. This method enables elementary operations in the latent space such as novel samples generation, interpolation, and attributes modification.

4.5 Evaluation

In this section we evaluate how well our method reconstructs images from latents, generates new images, interpolates between existing images and modifies image attributes in latent space. We finish with a brief look at the compression capabilities.

TABLE 4.1: Image resolutions, latent sizes and training epochs.

	IMAGE RESOLUTION	LATENT SIZE		EPOCHS
		LBAE (bits)	VAE(ours) (float32)	
MNIST	32x32x1, zero padded from 28x28	200	16	2000
CIFAR-10	32x32x3	600	128	2000
CELEBA	64x64x3, cropped to 1:1 and scaled	1500	64	500

We trained and tested our model on the CelebA (Liu et al., 2015), CIFAR-10 (Krizhevsky and Hinton, 2009) and MNIST (LeCun et al., 2010) datasets with the default train/test splits and image resolutions shown in Table 4.1. To evaluate LBAE against VAE with the LBAE identical architecture, we modified the LBAE encoder to output (μ, σ) and trained it in the VAE setup. We call this model VAE(ours).

The architectures of the encoder and decoder networks were adapted from typical autoencoder models for images with resolutions 64x64 and 32x32.

Dimensions of the binary latent space were devised experimentally over several tries on each dataset (the datasets differ in number of samples, image resolutions and data distributions). The latent dimensions are not sensitive within the given dataset and image resolution. For example, changing the CelebA latent dimensions to 1500 ± 500 resulted in an identical training profile and final performance.

All other hyperparameters were determined experimentally on a few trial epochs on training datasets. Grid search over the hyperparameters would likely improve the model performance. We leave this for future work.

For all datasets we use almost identical models, varying in the latent dimensions. Encoder and decoder are CNN networks with residual connections, where the decoder mirrors the encoder with transposed convolutions. The model was trained using ADAM (Kingma and Ba, 2015) with learning rate 10^{-3}, no weight decay and 512 batch size. Mean squared error is used as the reconstruction loss except for MNIST where we use the binary cross entropy. The model architecture is shown in more details in Section 4.8. The training is slower compared to VAE due to the gradient propagation through the tanh() and binarization, nevertheless comparable to other methods such as the 2 stage VAE (Dai and Wipf, 2019) which requires 420 epochs on CelebA, 3000 on CIFAR-10 and 1200 on MNIST.

We use the FID (Lucic et al., 2018), Kernel Inception Distance (KID) (Bińkowski et al., 2018) and Precision/recall (Sajjadi et al., 2018) as evaluation metrics. For consistency, we use reference implementations for all metrics [1][2][3]. To compute FID and KID we use 10k reference and evaluation images.

4.5.1 Reconstruction and Random Samples Generation

In Tables 4.2,4.3 and 4.4 we show that our model achieves the lowest reconstruction FID and KID scores.

TABLE 4.2: FID scores for reconstruction and interpolation tests. Results are taken from the corresponding publications for VAE, WAE-MMD, RAE-L2 and RAE-SN (Ghosh et al., 2020). VAE(ours) architecture is identical to LBAE. Lower FID values indicate better-quality images.

	MNIST		CIFAR-10		CELEBA	
	RECO.	INTERP.	RECO.	INTERP.	RECO.	INTERP.
VAE	18.26	18.21	57.94	88.62	39.12	44.49
VAE (OURS)	8.77	15.01	37.94	82.34	34.96	42.03
WAE-MMD	10.03	14.34	35.97	76.89	34.81	40.93
RAE-L2	10.53	14.54	32.24	62.54	43.52	45.98
RAE-SN	15.65	15.15	27.61	63.62	36.01	39.53
LBAE (OURS)	**8.11**	**9.80**	**19.37**	**34.41**	**7.71**	**14.87**

TABLE 4.3: FID scores for random image generation. Results are taken from the corresponding publications for VPGA,LPGA (Zhang et al., 2019), VAE, WAE-MMD, RAE-L2, RAE-SN (Ghosh et al., 2020) and Best GAN, 2 Stage VAE (Dai and Wipf, 2019). For fair comparison, VAE, WAE-MMD, RAE-L2 and RAE-SN results are split into $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu,\Sigma)$ columns. VAE(ours) architecture is identical to LBAE. Lower FID values indicate better-quality images.

	MNIST		CIFAR-10		CELEBA	
	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$
BEST GAN	10		70		49	
VAE	19.21		106.37		48.12	
VAE (OURS)	18.52		68.43		56.08	
2 STAGE VAE	12.6		72.9		44.4	
WAE-MMD	20.42		117.44		53.67	
RAE-L2		22.22		80.8		51.13
RAE-SN		19.67		84.25		44.74
LPGA	12.06		55.87		**14.53**	
VPGA	**11.67**		**51.51**		24.73	
LBAE (OURS)	88.13	**11.36**	71.48	**53.55**	64.65	**34.95**

This can be attributed to the prior-free training, where the model is not constrained to approximate any prior, which is believed to produce blurry images in the case of VAE. From Figure 4.12 it is apparent that LBAE reconstructions are sharper than typical VAE outputs. We can also see that, on the generative task, LBAE

[1] https://github.com/bioinf-jku/TTUR
[2] https://github.com/mbinkowski/MMD-GAN
[3] https://github.com/msmsajjadi/precision-recall-distributions

TABLE 4.4: KID scores scaled by 10^3 as in Dai and Wipf 2019.Lower KID values indicate better-quality images.

	MNIST			CIFAR-10			CELEBA		
	RECO.	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	RECO.	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	RECO.	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$
VAE (OURS)	6.43	12.41		30.87	74.1		30.49	58.83	
2 STAGE VAE		**6.7**			59.3			**40.9**	
WAE-MMD		137.8			**58.7**			59.7	
LBAE (OURS)	**5.39**	84.48	**6.34**	**13.01**	74.4	**51.9**	**6.15**	75.29	**30.33**

FIGURE 4.12: Reconstruction on the MNIST, CIFAR-10 and CelebA test datasets with the LBAE method. The ground truth image on the left is followed by the reconstruction on right.

outperforms all except the VPGA method, when sampled with the proposed hyperplane rounding method. Examples of novel images generate with our LBAE method are shown in Figure 4.13. When sampled from the binarized normal distribution

FIGURE 4.13: Novel samples generated with the LBAE method.

$f_b(\sim \mathcal{N}_N(0, \mathbf{I}_N))$, our scores are worse. This can be also seen perceptually in Figure 4.14 where the generated images are sharp but composed of features with wrong consistency.

FIGURE 4.14: MNIST and CelebA images generated by LBAE from latents $\mathbf{b} = f_b(\sim \mathcal{N}_N(0, \mathbf{I}_N))$

Note that the very high performance of the 2 Stage VAE (Dai and Wipf, 2019) and the VPGA, LPGA (Zhang et al., 2019) on the CelebA can be, in large part, attributed

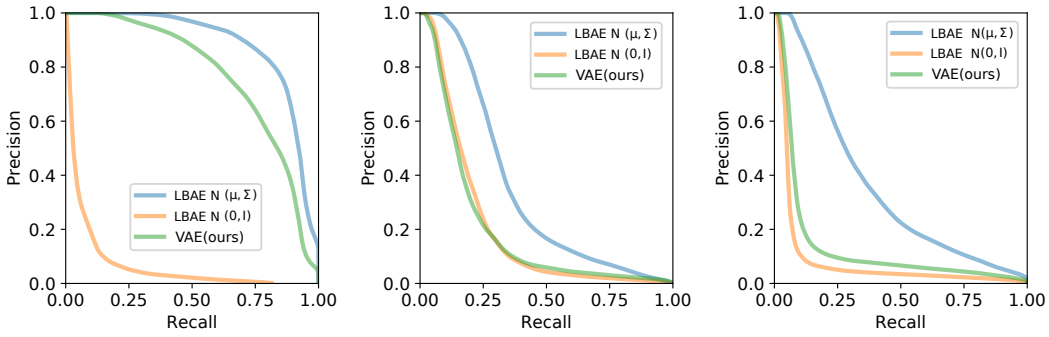


FIGURE 4.15: Precision / recall curves.

to the image preprocessing. For example, the Dai and Wipf 2019 authors center-crop 108×108 patch and resize it to 64×64. This augmentation removes most of the background which simplifies the generative task.

While FID and KID metrics indicate a similarity between the quality of generated and reference images, they do not explain other important attributes such as the coverage of the generated distribution. To disentangle the FID/KID 1D quality measure we evaluate our method by the precision/recall metric (Sajjadi et al., 2018). Precision measures qualitative distance between the generated and reference images, and recall how well the entire reference distribution (e.g. all classes) is represented by the randomly generated images. We set the entire test datasets of respective benchmarks as the reference distributions. In Figure 4.15 we show the Precision/recall curves for random images generated by sampling from normal, binarized distribution $\mathcal{N}(0, \mathbf{I})$, the LBAE method, noted as $\mathcal{N}(\mu, \Sigma)$, and VAE(ours) - VAE model with the LBAE architecture. We can see that our method shows consistently higher, balanced precision and recall with the exception of sampling from $\mathcal{N}(0, \mathbf{I})$. In Table 4.5 we then compare our method with Ghosh et al. 2020. Here, again, the LBAE achieves relatively high precision as well as recall. This signifies that the generated images represent the entire distribution equally well and that the image quality is close to the reference distribution.

TABLE 4.5: Precision / Recall evaluation between LBAE and methods VAE, WAE-MMD, RAE-L2, RAE-SN from Ghosh et al. 2020.

	MNIST		CIFAR-10		CELEBA	
	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$	$\mathcal{N}(0,1)$	$\mathcal{N}(\mu,\Sigma)$
VAE	**0.96 / 0.92**		0.25 / 0.55		0.54 / 0.66	
VAE (OURS)	0.88 / 0.93		**0.55 / 0.74**		**0.62 / 0.64**	
WAE-MMD	0.93 / 0.88		0.38 / 0.68		0.59 / 0.68	
RAE-L2		0.92 / 0.87		0.41 / 0.77		0.36 / 0.64
RAE-SN		0.89 / 0.95		0.36 / 0.73		0.54 / 0.68
LBAE (OURS)	0.37 / 0.44	**0.92 / 0.97**	0.48 / 0.76	**0.66 / 0.87**	0.50 / 0.57	**0.73 / 0.82**

In Table 4.6 we compare our model with results from Ghosh et al. 2020, obtained by sampling from a GMM (10 Gaussians) trained on latents encoded on the training

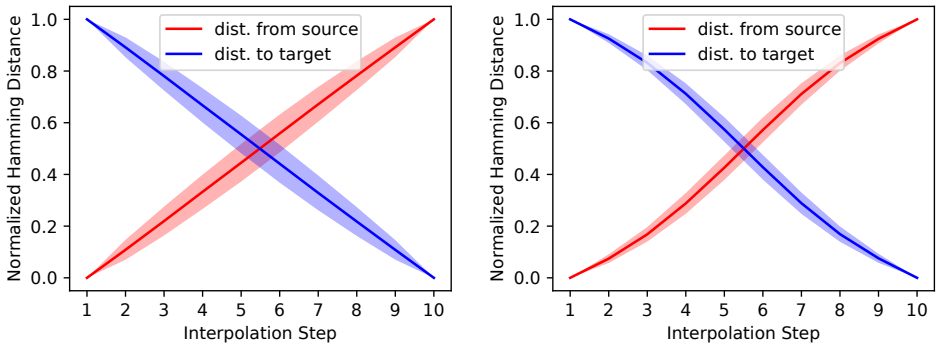
TABLE 4.6: Precision/recall and FID scores for sampling from GMM, except our method LBAE where we sample from the matrix of moments with the random hyperplane method.

	MNIST		CIFAR-10		CELEBA	
	FID ↓	PRECISION /RECALL ↑	FID ↓	PRECISION /RECALL ↑	FID ↓	PRECISION /RECALL ↑
VAE	17.66	0.95 / 0.96	103.78	0.37 / 0.56	45.52	0.50 / 0.66
WAE-MMD	9.39	0.98 / 0.95	93.53	0.51 / 0.81	42.73	0.69 / 0.77
RAE-L2	**8.69**	**0.98 / 0.98**	74.16	0.57 / 0.81	47.97	0.44 / 0.65
RAE-SN	11.74	0.98 / 0.97	75.3	0.52 / 0.81	40.95	0.55 / 0.74
LBAE (OURS)	11.36	0.92 / 0.97	**53.55**	**0.66 / 0.87**	**34.95**	**0.73 / 0.82**

data. With the exception of MNIST, our model outperforms the GMM sampling on both FID and Precision/recall scales. Note that the RAE-L2 method with GMM sampling shows lower FID score than on the reconstruction on the test dataset. It is conceivable that latents sampled from GMM, fitted to the training data, are decoded by the RAE-L2 model that overfits on the training data.

4.5.2 Interpolation in Latent Space

In Figure 4.17, we show interpolation between two images over $T = 10$ steps. We can see the interpolation is smooth between the endpoints; there are no abrupt changes in the context nor the image intensities. The composition of the intermediate samples also seems to lie on the path between the endpoints as we intuitively expect. The FID and KID scores for interpolation in Tables 4.2 and 4.4 support this observation.



MNIST CIFAR-10

FIGURE 4.16: μ and σ of Hamming distance between interpolated latent at step k and source and target latents.

SLERP interpolation given by two endpoints follows the shortest path on the sphere. To understand what path the latents follow in the binary space, we measure Hamming distance between the interpolated latent $\mathbf{b}_k, k \in [1, .., T]$ at step k and the source and target latents. Plot of these distances over 1k interpolations is shown in Figure 4.16. The Hamming distance is normalized between the source and target latents. We can see that the interpolation in the binary space is almost linear which

indicates that the feature manifolds in this space are continuous between the end-points and that our interpolation method provides a suitable mapping between the binary latent space and the continual space on the sphere.

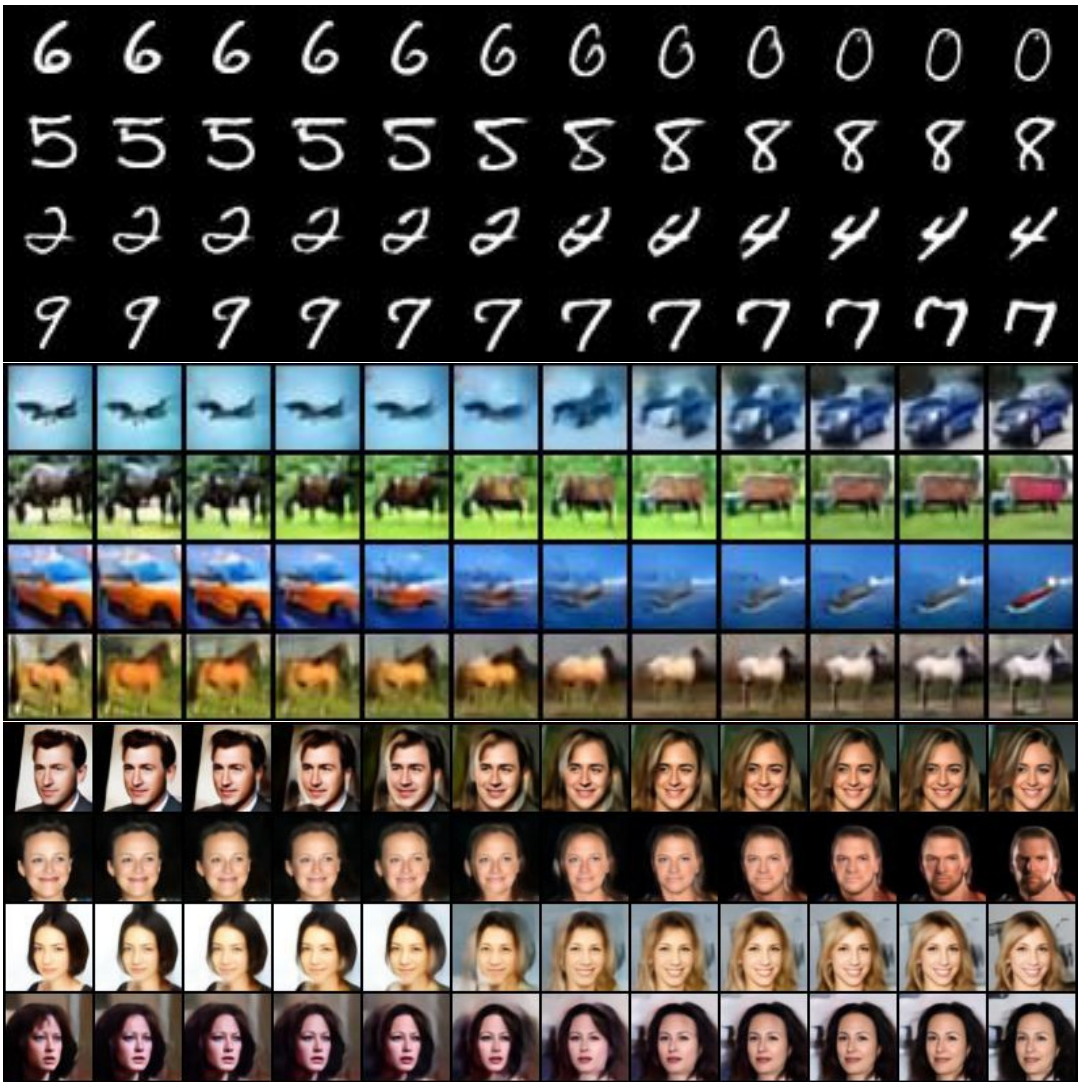


FIGURE 4.17: Interpolations between test images from MNIST, CIFAR-10 and CelebA.

4.5.3 Attribute Manipulation in Latent Space

Attributes of the generated samples can be directly modified in the latent space with very simple method. We demonstrate this on two examples where we add *eyeglasses* or *goatee* CelebA attributes to random test images. This operation does not require the model to be conditionally trained with the attribute labels.

For example, we want to add attribute $a = \textit{eyeglasses}$ to a random target image. First, we collect K images with the attribute a from an image dataset (training dataset in our experiment, but it can be any set of images). Second, we locate bits in the latent space that encode the attribute a. We do this by encoding the K images to latents $\mathbf{Y}^a \in \{-1, 1\}^{(N \times K)}$ and obtaining the expected value $\mathbf{p} = \mathbb{E}[\mathbf{Y}^a], \mathbf{p} \in \mathbb{R}^N$. To change the attribute a in an image represented by latent $\mathbf{b}$ we set its bits b_i whose expected value p_i is outside a threshold D as:

$$b_i = \begin{cases} 1, & \text{if} \quad p_i > D \\ -1, & \text{if} \quad p_i < -D \\ b_i, & \text{otherwise.} \end{cases} \tag{4.27}$$

The modified latent b is then decoded to a new image. This is graphically depicted in Figure 4.18.

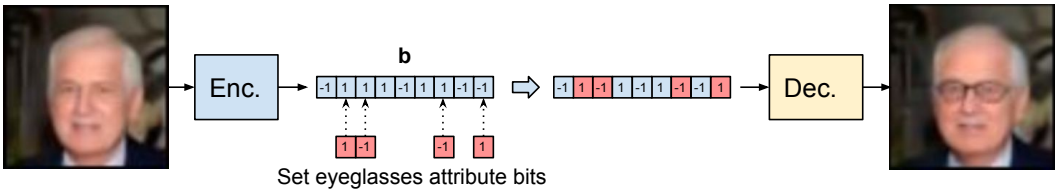


FIGURE 4.18: To modify an image attribute in the binary latent space we first identify bits with the highest activation in other images with the targeted attribute and then set these bits in the latent $\mathbf{b}$ of the image to be modified.

The threshold D determines how many bits will be modified, consequently how strongly the source image will be altered. Experimentally we found that $D = 0.1$ provides satisfactory results and used this value for all our experiments. Interpolation is then performed by the method described in the Section 4.4.3. Examples of two attribute alterations are shown in Figure 4.19.

4.5.4 Compression

While not a comprehensive evaluation, Table 4.7 shows the compression performance of our method LBAE compared with VAE and, only on the CIFAR-10 dataset, to the VQ-VAE (van den Oord et al., 2017). The paper introducing VQ-VAE presents the compression ratio on ImageNet images with resolution $128 \times 128 \times 3$, not yet explored with our method.

Setting the CelebA *eyeglasses* attribute.

Setting the CelebA *goatee* attribute.

FIGURE 4.19: Interpolation between test images (left) and the same images (right) with modified attributes.

The compression is reported as the input sample (image) size over the compressed latent representation both in bits, similar to what is used the in the VQ-VAE publication. Additionally, we relate the compression ratio to the reconstruction quality reported as FID. On the CIFAR-10, the VQ-VAE discrete latent code indices $8 \times 8 \times 10$ embeddings in a dictionary with 512 items. Therefore, each index requires 9 bits and together the code consumes $8 \times 8 \times 10 \times 9 = 5760$ bits. Size of the real-valued VAE latents is estimated in bits for 32bits floats per dimension. Arguably, the latents do not saturate all 32 bits at each dimension, thus the reported values are just informative.

In Table 4.7, we can observe that LBAE shows significantly higher compression compared to VAE as well as higher quality in FID. LBAE offers also higher compression than the VQ-VAE, although we could not compare the reconstruction quality, thus this result can not be considered conclusive.

4.6 Relation to VQ-VAE

VQ-VAE was introduced by van den Oord et al. 2017 as a discrete, deterministic autoencoder. In a multi-scale, hierarchical organization (Razavi et al., 2019b) it achieves generative performance comparable to GANs.

TABLE 4.7: Comparion of input/latent size compression ration and corresponding FIDs. VQ-VAE compression is based on data from the publication van den Oord et al. (2017), available only for, CIFAR-10.

Method	MNIST Image size: (32x32x1) 1024 bits			CIFAR-10 Image size: (32x32x3x8) 24576 bits			CelebA Image size: (64x64x3x8) 98304 bits		
	Latent size(bits)	Comp. ratio	FID (reco)	Latent size(bits)	Comp. ratio	FID (reco)	Latent size(bits)	Comp. ratio	FID (reco)
LBAE (OURS)	200	5.12	8.112	600	40.96	19.37	1500	65.54	7.71
VAE (OURS)	512	2	8.767	4096	6	37.9	2048	48	34.96
VQ-VAE				5760 (8x8x10x9)	4.27				

VQ-VAE learns the discrete representations indirectly, as indices to a codebook with continual-value embeddings that are then passed to the decoder for reconstruction. The indices are looked up as nearest codebook neighbours of the encoder output. During training the indexed embeddings in the codebook are moved closer to the encoder output. The non differentiable nearest neighbour operation is replaced with the straight-through gradient estimator in the backward pass. To sample new images authors propose to learn a categorical prior over latents encoded on the training data with PixelCNN (Oord et al., 2016).

LBAE learns the latent codes directly, although we could think of the first fully connected layer in the decoder as a dictionary of embeddings that is implicitly learned. The binary latents, in fact, work as row selectors of the weight matrix, where each row can be considered an embedding vector. Row vectors corresponding to ones in the input are summed together and sent down to the following layers in the decoder. The possibility of training an autoregressive model on this dictionary, in the VQ-VAE fashion, is left for future research.

Unlike LBAE, the VQ-VAE cannot be easily used for interpolation and other operations in the latent space as shown by Berthelot et al. 2019. While VQ-VAE needs to train an external autoregressive model, LBAE can perform the generative tasks in the discrete latent space with its decoder.

4.7 Implicitly Learned Dictionary of Embeddings

In this section we propose an alternative view of the LBAE as an autoencder learning a dictionary of embeddings in a similar fashion as in the VQ-VAE (van den Oord et al., 2017).

The binary latent is decoded by a transposed CNN with the first layer being fully connected. We can view the binary latent as a set of indices to to the weight matrix of this fully connected layer. In this section we call the weight matrix a dictionary $\mathbf{W} \in \mathbb{R}^{(NxL)}$ and its rows the embeddings $\mathbf{W}_i \in \mathbb{R}^{(1xL)}$. Embeddings at the positions of bits that are set in the latent are added together and then decoded by the following transposed CNN. In our implementation, the latent vector is encoded as

$\mathbf{b} \in \{-1, 1\}^N$ which is then converted to the range $\widehat{\mathbf{b}} \in \{0, 1\}^N$ by passing through the ReLU. The zero bits mask out the undesired embeddings. The transformation takes a form of a simple matrix-vector multiplication $\mathbf{z} = relu(\mathbf{b})\,\mathbf{W}$. A diagram of this method is shown in Figure 4.20.

The sum of embeddings from the dictionary $\mathbf{z} \in \mathbb{R}^{(1xL)}$ is not averaged which results in a non stationary distribution shift. The ill effects of this internal covariate shift is then removed by batch normalization that whitens the combined embeddings, therefore stabilizes the distribution on the input of the CNN decoder.

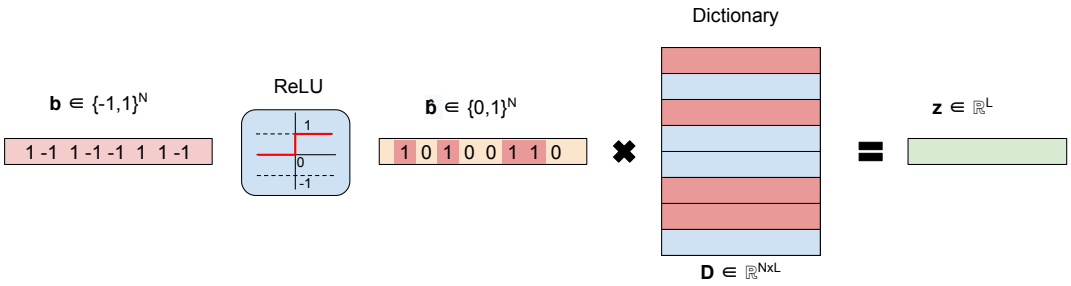


FIGURE 4.20: Weight matrix of the input, fully connected layer of the LBAE decoder can be treated as a dictionary of embeddings. The latent vector $\widehat{\mathbf{b}}$ functions as a selector of the embeddings. Embeddings (row vectors in the dictionary) at the positions of bits that are set in the latent $\widehat{\mathbf{b}}$ are added together to be decoded by the transposed CNN.

4.8 Implementation

The LBAE decoder and encoder (Figure 4.21 and 4.22) are common CNN networks with residual blocks with batch normalization. Decoder uses transposed convolutions to increase spatial resolution and sigmoid() function on the output.

4.9 Conclusion

In this chapter, we set to propose a method for learning discrete latent representations. We reviewed some basic findings from neurobiology, framing the landscape that we searched for inspirations.

Motivated by a multitude of applications and supported by insights from biology, we formed a proposition for a novel, closed-form method for sampling from the Bernoulli latent space and performing a smooth interpolation and attribute modification in this space. To our knowledge, this is the first successful method that directly learns binary representations of images and allows for smooth interpolation in the discrete latent space.

We show that a simple deterministic, discrete latent autoencoder, trained with the straight-through gradient estimator performs on a par with the VAE model, its derivatives and the latest regularized, deterministic autoencoders, on all common

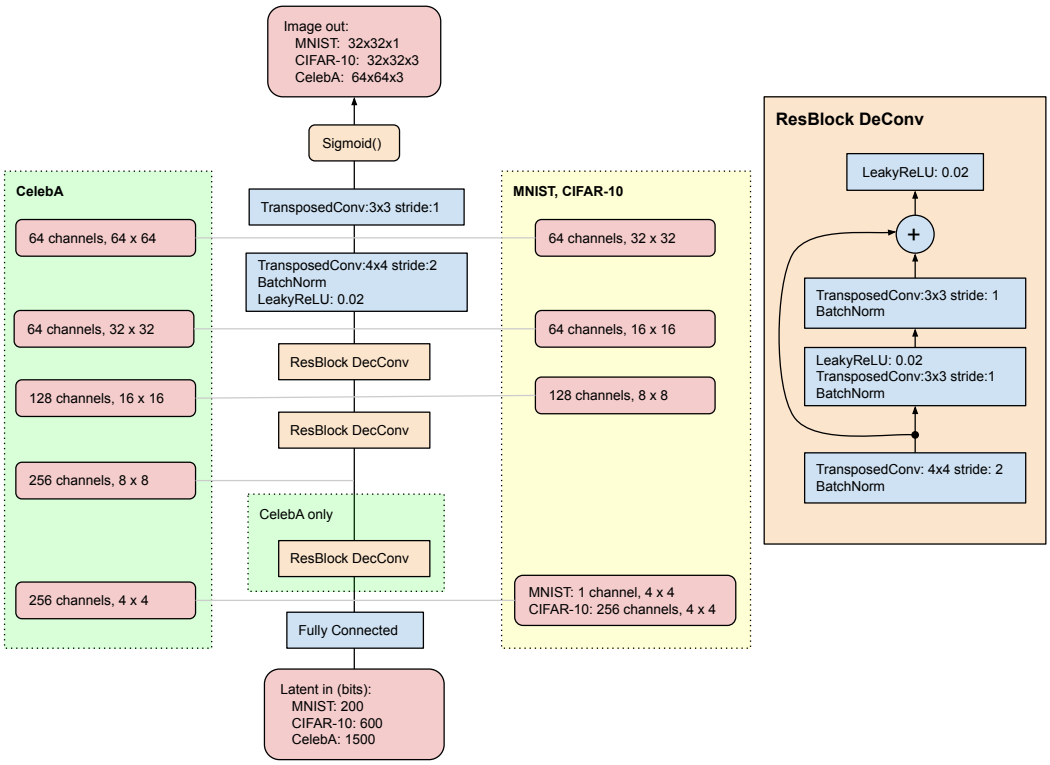


FIGURE 4.21: LBAE Decoder

tasks such as reconstruction, novel samples generation, interpolation and attribute modification on the CelebA, CIFAR-10 and MNIST benchmarks.

Our model achieves higher reconstruction as well as generative image quality compared to VAE. Furthermore, our method for random sampling from the latent space covers the entire distribution without over or under-representation of any classes, indicating resilience to mode collapse.

Equally, on a simple experiment of modifying image attributes, we show the potential of the representation power of the Bernoulli latent space.

Finally, we related our method to the VQ-VAE, a high-performance discrete autoencoder with a dictionary of latent embeddings. We proposed to view the weight matrix of the first fully connected layer in the decoder as an implicitly learned dictionary of embeddings with the binary latent serving as their selector. It is plausible that such a dictionary would display a similar characteristic as the one learned by the VQ-VAE method. Furthermore, it would be insightful to conduct a detailed analysis of the embeddings. For instance, to learn relations between the embeddings and the class labels or their compositionality, e.g. which embeddings correlate with vertical, horizontal, or curved features of the MNIST digits.

The perceptual quality of novel images generated with our method, including interpolation, are still far from the state-of-the-art GAN models (Karras et al., 2018) or Diffusion models (Song et al., 2020), however better than that of AE, VAE, and even VQ-VAE without the autoregressive generator. Another limitation of our method is

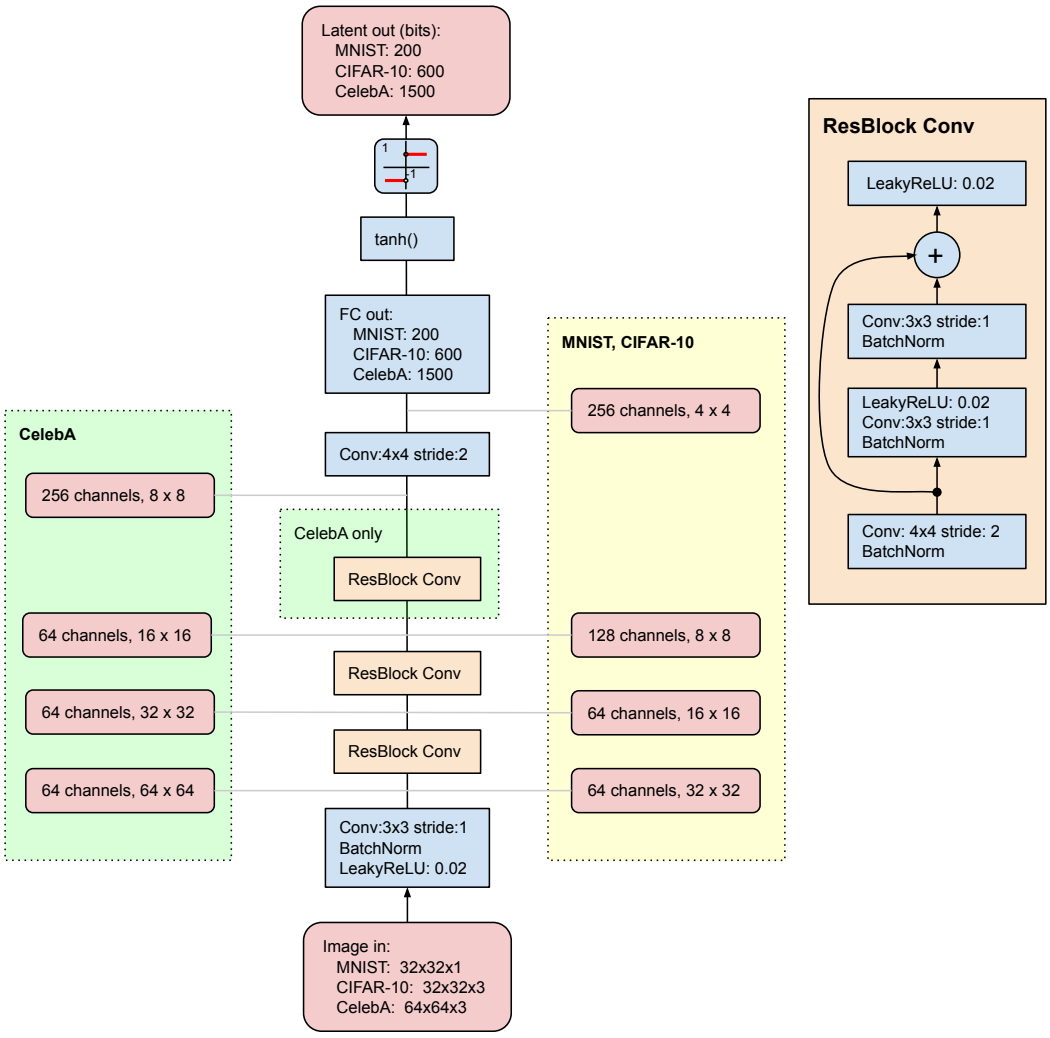


FIGURE 4.22: LBAE Encoder

the need to use the external algorithm (described in Section 4.4.2) to enable operations in the latent space. This increases the method complexity (as defined in Section 1.1).

Our work brings the following main contributions:

- Demonstration that a tanh function followed by a straight-through estimator with a unity surrogate function in the backward pass can be used to efficiently train an autoencoder with state-of-the-art performance.
- Proposed a novel method to generate correlated Bernoulli samples, perform smooth interpolation, and modify sample attributes in the discrete latent space.
- Showed that, albeit its simplicity, our method performs equally well or better than the state-of-the-art using the FID, KID, and Precision/recall metrics.

The LBAE PyTorch implementation, including trained models, is publicly available on `https://github.com/ok1zjf/lbae`.

4.9.1 Future Work

To evaluate the LBAE performance in a setting similar to the VQ-VAE, we consider applying the PixelCNN (Van den Oord et al., 2016) to the implicitly learned dictionary (see Section 4.7).

Another promising avenue for future work is integrating the hyperplane rounding method with the encoder. In this configuration, the encoder would not learn binary values directly but rather as unit vectors on an n-sphere one for each latent bit (dimension). Binary values would then be generated by the random hyperplane method and enter the decoder as binary vectors. During the backpropagation, a covariance matrix of the errors on the binary input of the decoder would be calculated and used to update unit vectors on the n-sphere learned by the encoder. This method would not be deterministic since the input to the decoder would be stochastic, similar to the VAE.

There are many additional evaluations to conduct. For example, to map the activations of the latent bits with respect to a set of input features such as lines, corners, curves, and circles of the MNIST model. In another evaluation, we plan to visualize activations of discrete bits by the decoder. Here, a list of binary latent representations would be built with bits sets in descending order of their expected value. That is, the first representation would include only a few bits with the highest expected value followed by representations with more bits set but lower expected value and so on. Each latent would be subjected to a distribution alignment, similar to the procedure for random sampling from the latent space described in Section 4.4.2.

An experiment of high practical utility is to test the performance of the binary latents in established computer vision domains such as image classification, object detection, image captioning, and others.

Autoencoders with an information bottleneck, trained with mini-batch gradient descent, learn low entropy codes; a small number of bits encode the most common features across the training data while a large number of bits encode less frequent features. This is a desirable encoding for compression schemas, however not necessarily appropriate to produce semantically meaningful and interpretable latent representations; the entropy coding does not encourage learning semantic features. As demonstrated in recent work on self-supervised representation learning (Grill et al., 2020, Chen and He, 2020, Caron et al., 2020, Zbontar et al., 2021) it is beneficial to encourage the ANN to learn the latent space by minimizing variance within representations of augmented images and maximizing between different images. We propose to add the data augmentation coupled with the contrastive loss on the latent space binarized by our method. This new loss would also help reduce overfitting, thus enabling expansion of the latent space and increasing its sparsity. It is conceivable that this method may learn more semantically focused latent representations that would further improve sampling from this space. It is also likely that such a latent space may also be transferable to other ML domains, e.g., for image classification.

Chapter 5

Continual Learning and Memorization with Sparse Representations

An essential aspect of human intelligence is the ability to continually learn from a non-stationary environment, incrementally building upon past knowledge. More formally, Continual Learning (CL) is the ability to extract knowledge from the environment, combine it with past knowledge and retain it for future applications. Knowledge can originate from entirely new domains or extend or refine existing ones. The main property of a continual learning system is to minimize forgetting of already stored information while acquiring a new one. Other important properties are the forward and backward transfers allowing the CL system to progress rapidly through learning more and more complex tasks. During the forward transfer, we leverage old knowledge to learn new, more complex tasks. For example, for us, humans it is easier and faster to learn how to ride a bicycle if we already know how to ride a two-wheel scooter. This is due to a robust forward transfer in our learning system. Backward transfer, on the other hand, affects already retained knowledge in positive or negative ways. Positive backward transfer helps to improve the old knowledge upon acquisition of a new one. For instance, learning tightrope walking improves ones existing ability to ride a bicycle despite the seemingly different nature of these tasks. On the other hand, negative backward transfer causes a decrease in performance on old tasks, more commonly known as catastrophic forgetting, catastrophic interference, or retroactive interference.

Catastrophic Forgetting (CF) does not appear significant to us, as we do not forget old knowledge when learning a new one. It is, however, the main challenge for artificial neural networks that suffer from significant catastrophic forgetting, that is, the performance of an ANN on past tasks rapidly declines upon updating the same network on new tasks without including the old samples. To avoid CF, all current deep neural models need to be retrained with interleaved all past and new data, an energy, time, and resource demanding process. It also poses a security risks since all

past data, such as medical images or legal documents, need to be stored for future model updates.

In this work, we address the continual learning problem in machine learning through the lens of the memory circuits of our brain. We study the systems enabling continual learning in the primate brain as a source of inspiration applicable to the computational methods. We then review the CF problem in more detail, particularly its causes and mitigations in typical ANN for CL applications.

Building on the latest CL research in machine learning and our insights from the neural processing in our brain, we propose a novel, biologically inspired computational method for continual learning based on high dimensional, sparse binary representations and growing neural memory. We show that, in large part, the sparse binary representations stay behind a significant CL performance boost while also reducing computational load. Our method addresses the most challenging task-free, class-incremental learning scenario with a replay-free strategy, utilizing fixed representations learned in a supervised fashion.

This chapter has the following structure. In Section 5.1 we detail the CL in the brain and the neurological mechanisms avoiding the CF. In Section 5.2 we review literature related to the CL in ANN along with common CL scenarios, methodological strategies, and benchmarks. Our new CL method is proposed in Sections 5.2.4 and 5.3. Section 5.4 then presents conducted experiments and results. Finally, the chapter closes with a conclusion in Section 5.10.

5.1 Continual Learning in the Brain

5.1.1 Complementary Learning Systems

The ability of our brain to continually learn without forgetting is attributed to the Complementary Learning Systems (CLS), a dual memory framework pioneered by McClelland et al. (1995) and recently reviewed by O'Reilly and Norman (2002) and O'Reilly et al. (2014a).

In the CLS theory, new experiences are rapidly accumulated in the hippocampus's short-term memory and later, slowly over several sleep cycles, transferred to the neocortex for long-term storage. This transfer happens during a hippocampal replay (Ji and Wilson, 2007). A study by Richards et al. (2014) points out that the episodic memories in the hippocampus are encoded as instance-based, non-parametric representations while the neocortex operates with parametric representations. Integration of new information in such a parametric system cannot be done directly by updating the synaptic connections since each encodes a number of different distributed representations. Doing so would result in catastrophic forgetting, as observed in ANN (McCloskey and Cohen, 1989, Ratcliff, 1990, French, 1999).

The integration of the new experiences in the neocortex, also called memory consolidation, is performed by the hippocampal replay, which is believed to be the

core mechanism mitigating the catastrophic forgetting in the CLS theory. The hippocampus accumulates a larger number of episodic events to extract the underlying statistic and then integrates them with old experiences during the replay. It has been observed that the replay is time-compressed by about factor of 20, allowing each experience to be replayed multiple times (Káli and Dayan, 2004, Wilson and McNaughton, 1994, Buzsáki, 1989). In fact, new memories are retained in the hippocampus for up to one week. During this time they are periodically replayed into the neocortex, integrating increasingly new information and establishing more associations. This is called a hippocampal-dependent stage (Frankland and Bontempi, 2005, Dudai, 2004). After this period, the new experiences become hippocampus independent. The replay is believed to also enhance generalization, establish inter-experience links and generate novel representations from their recombinations (Wu and Foster, 2014, Gupta et al., 2010).

The CLS, particularly its replay mechanism, inspired a number of machine learning methods targeting catastrophic forgetting. Retraining ANN on batches of new data interleaved with old data samples, either stored in a replay buffer (Rebuffi et al., 2017) (sometimes also called episodic memory or rehearsal) or generated by sampling from the trained model (pseudo-rehearsal) (Robins, 1995), has shown remarkable resilience to forgetting. While the replay mechanism is a good stepping stone towards the CL, the underlying representation encoding may be equally important for this task. Therefore, in this work, we study how the representation encoding impacts catastrophic forgetting.

5.1.2 Pattern Separation and Sparsity in the Hippocampus

The encoding of new and old memories in the hippocampus proceeds via a number of transformations from sensory data acquisition to memory consolidation in the neocortex. The most notable transformations are pattern separation and pattern completion with characteristic variability in the representation sparsity. Hippocampal regions participating in the memory retrieval, completion, and consolidation, along with the information flow directions, are shown in Figure 5.1.

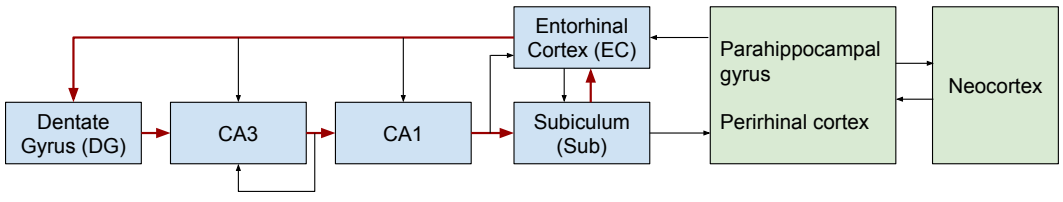


FIGURE 5.1: Schematic diagram of the main regions of the hippocampus. The red path signifies the main feedforward path of the autoassociative circuitry. Adopted from O'Reilly and McClelland (1994).

According to O'Reilly and McClelland (1994) and O'Reilly and Rudy (2001) the memory encoding in the hippocampus is formed by a projection of representations

from number of regions in the neocortex to the Entorhinal Cortex (EC). Dense, overlapping patterns are then separated in the DG with the help of the Cornu Ammonis (CA)3 region. These representations are characteristic by high dimensionality and sparsity. CA3 is considered to be the center where the separated representations are stored. In order to perform the pattern completion, the EC needs to receive patterns from the CA3 with similar encoding. CA1 serves as a translation between the CA3 separated patterns and the denser EC representations. The subiculum with the EC then binds together the completed patterns and projects them back to neocortical regions. Most of the connections shown in Figure 5.1 are partially bidirectional except the projection from EC to DG and CA3, which are strictly feedforward (McNaughton and Nadel, 1990).

The pattern separation reduces or eliminates the overlap of similar representations stored by the neocortex (Knierim and Neunuebel, 2016)(Figure 5.2). This is

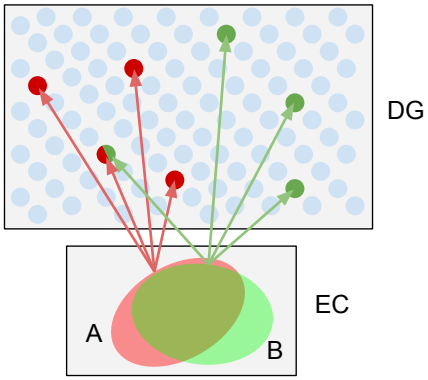


FIGURE 5.2: Illustration of pattern separation of the neocortical representations by sparsification between the Entorhinal (EC) and the Dentate Gyrus (DG) in hippocampus. A small difference in similar patterns results in a large difference in the projected, sparse representations.

important for one's ability to relate a specific new experience, such as an encounter with an unusual bicycle type, to the representation of a bicycle encoded in the cortex. Studies by Neunuebel and Knierim (2014), Leutgeb et al. (2007) and Lee et al. (2015) show that representations in DG are highly sensitive to even small changes in the experienced environment but not the representations encoded by EC, supporting the pattern separation theory. The impact of reduced pattern separation in DG was documented in the work of McHugh et al. (2007) where DG lesions reduced an animal's ability to react differently in similar environments but left intact the ability to react differently in distinct environments. In a similar study Gilbert et al. (2001) compared impact of damaged DG and CA1 regions on performance in spatial and temporal tasks. They found that the damage to the DG reduced the rat's ability to differentiate between similar spatial patterns, but it did not impact the temporal tasks. The damaged CA1 region, on the other hand, reduced the temporal pattern separation but not spatial. A study on humans (Brock Kirwan et al., 2012) showed that patients with damaged hippocampal regions performing the pattern separation,

including the downstream (CA3/CA1) regions, were less likely to identify presented patterns as similar compared to the control group.

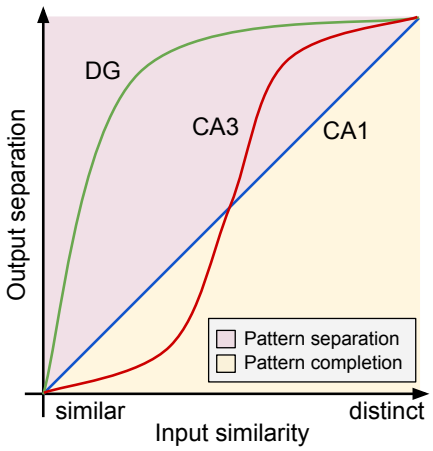


FIGURE 5.3: Illustration of the pattern separation transfer function in Dentate Gyrus (DG), CA3 and CA1 regions. Similar patterns are pushed apart by the DG while distinct patterns are left intact. Plotted according to Yassa and Stark (2011).

The pattern separation can be seen as a process where a dense representation of a scene gets split into a high dimensional sparse representation where each object in the scene is encoded by a single or few bits. Yassa and Stark (2011) shows that the intensity of the separation in DG is a function of the similarities within the patterns. Representations of highly similar objects will be separated more in the DG than distinct objects, as illustrated in Figure 5.3.

During the pattern completion, the representations of similar features are compressed into denser representations where similar features overlap. In Figure 5.3 we can see how the CA3 region completes (reduces separation and increases overlap) similar features but separates or leaves intact more distinct features.

The highest sparsity was observed in the DG region, although other hippocampal regions also exhibit higher sparsity than the neocortex. In Figure 5.4 (left) we show how the sparsity progresses from the neocortical input from the bottom up to the CA3 region. Sparsity in each region is quantified in the graph in Figure 5.4.

Sparsity appears to be characteristic to a number of other neural regions, particularly in the primary vision cortex (Olshausen and Field, 1996, 1997, Serre et al., 2006) and in other sensory inputs (Olshausen and Field, 2004, Babadi and Sompolinsky, 2014), reaching the neocortex (Quiroga et al., 2008).

The sparsity has been observed in all animals, most notably in the Drosophila fruit fly olfactory system (Turner et al., 2008). Approximately 50 projection neurons send their activities to about 2500 Kenyon cells (neurons in higher brain regions of insects called mushroom bodies responsible for olfactory learning and memory (Aso et al., 2014)). An input stimulus activates approximately 50% of projection neurons and less than 10% Kenyon cells. In our model for the CIFAR-100 dataset, we adopt

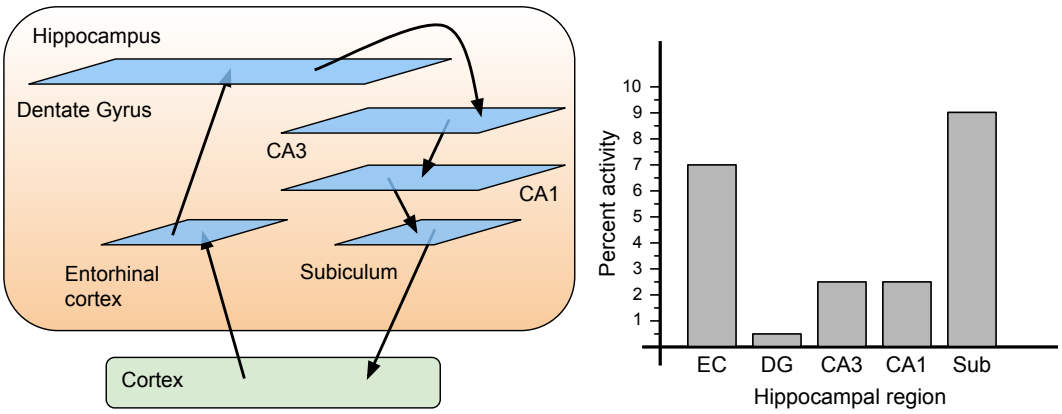


FIGURE 5.4: The diagram on the left shows both cortical and hippocampal components. The activation sparsity progresses from the cortex to the Dentate Gyrus (DG). The hippocampus can reinstate a pattern of activity over the cortex via the entorhinal cortex (EC). The graph on the right shows activation sparsity per region in percent. Based on data from O'Reilly and Rudy (2001).

similar architecture. We first reduce the dimensionality of dense features from 1028 down to 256 and then expand to sparse binary representation with 8192 bits.

Sparse representations are metabolically efficient, statistically less prone to interference, and more resilient to noise. Extreme sparsity, however, results in complete pattern separation and consequently reducing generalization and any forward or backward transfer since each experience then becomes unique (Sharkey and Sharkey, 1995, French, 1994).

In our work, we adopt sparsity as the primary tool to alleviate the catastrophic forgetting while still maintaining sufficient pattern overlap allowing generalization within-class categories.

5.1.3 Early Development of the Primary Visual Cortex

In addition to the CLS dual memory with replay and pattern separation, catastrophic forgetting highly depends on the level of plasticity in neural regions that are updated. Plastic regions exhibit fast learning abilities but high levels of catastrophic forgetting. Conversely, rigid regions retain new information slowly but with low interference with the already encoded patterns. In the context of continual learning we would like to identify brain regions susceptible to catastrophic forgetting, more specifically, the level of plasticity in regions responsible for learning elementary vision features.

Early research by Wiesel and Hubel (1965) on the effect of visual deprivation in kittens suggests that the neural wiring supporting vision in the primary visual cortex (V1) develops very early in life with minimal changes later on. Kittens deprived of visual stimuli during the first few months of life showed no changes in the physiology of retinal cells up to the lateral geniculate nucleus (LGN). The sight deprivation mainly affected the visual cortex with irreversible changes even after the

sight recovery. Timney et al. (1978) and Mitchell (1988) showed that only minimal visual recovery was observed in animals deprived of vision beyond one year of age.

In humans, the early visual cortex development stages are similar. Synaptic density in the visual cortex increases to a maximum during the first year of life. Ten years after that, about 40% of synaptic connections are gradually pruned to the adult level. The first 2 to 4 months is the most critical period of the visual cortex development, where the synaptogenesis is the most rapid (Huttenlocher and De Courten, 1987). This is also the period where the majority of stereo perception develops (Sacks, 2006). The importance of the critical period on the visual cortex development is supported by a body of research on patients undergoing treatment of cataracts. Surgery on children older than eight years born with cataracts often results in permanent vision deficit throughout the life, but infants fully recover (Taylor et al., 1979).

A very insightful study on the primary visual cortex development was recently conducted by Jiang et al. (2009). This work shows that the visual cortices of an early blind remain significantly thicker compared to individuals with healthy sight. It is conceivable to assume that the reduced pruning due to the lack of visual stimuli is caused by the inability of the visual cortex to develop feature filters, typically sparsely populated. In the context of modern CNN, we could relate this phenomenon to a network of fully connected layers able to learn the convolutional filters from exposure to natural visual stimuli by pruning connections. In the case of the missing visual input, such a network would not learn the filters and remain dense.

5.2 Continual Learning in Artificial Neural Networks

The difficulty to continually learn in artificial neural networks was already studies three decades ago by McCloskey and Cohen (1989) and more notably later by French (1999, 1992). The problem was identified to stem from Catastrophic Forgetting (CF), where ANN performance on old tasks rapidly declines after being trained exclusively on new tasks. A number of methods followed, trying to solve the CF problem (Hinton and Plaut, 1987, Goodrich and Arel, 2014, Kirkpatrick et al., 2017) and more, which we discuss below.

5.2.1 Continual Learning Scenarios

The most concise and practical CL scenarios were proposed by van de Ven and Tolias (2018) and, since its introduction, applied in several studies such as Hsu et al. (2018), Rao et al. (2019), von Oswald et al. (2019) and others. The scenarios are:

- Task-incremental Learning (TIL)
  In this scenario, the method learns sequentially new tasks with their identities known during training and inference. This is the simplest setup.

- Domain-incremental Learning (DIL)
  The method learns several different representations (domains) of a single task

e.g., different appearances of numbers 1,2, and 3 in a single task. The domain identity during the inference is not known, and the algorithm needs to classify only the digits without inferring the distribution. This scenario, in large part, amounts to learning new instances of the same classes.

- Class-incremental Learning (CIL)
  Here the method continually learns individual classes, which then need to be identified during the inference. If the classes come grouped in tasks, the task identity also needs to be inferred. This is the most complex scenario.

In this context, the *domain* is a set of patterns with similar underlying statistics. The *task* is typically a group of patterns. For example, task A can be a group of digits 1,2, and 3, task B digits 4,5, and 6, and so on. What precisely constitutes the task is usually established by the evaluation procedure. The *class* is then a unique identifier of a pattern with a specific characteristic. While the TIL and DIL scenarios are very common, however, from the standpoint of practical applications are of little use. The CIL scenario is the closest to the natural learning desirable for an intelligent agent. There are many modifications of these scenarios, typically targeting the performance boost of the proposed methods. For example, it is common to split large number of classes into a substantially smaller number of tasks and then train the method with all task samples interleaved in batches. Consequently, the task-independent and identically distributed (IID) batches considerably reduce forgetting within each task. It is also common to set up a multi-head classifier for each task. In this configuration, classes within each task are learned and inferred in isolation sharing only the front-end part of the network. This approach also reduces forgetting. Splitting the given class group into tasks or providing the task or domain information during the training or inference time boosts the performance, but it is not representative of the ultimate goal, that is, continually learn new knowledge as it becomes available.

Our work focuses on the most challenging subset of the CIL scenario: a replay-free, task-free, class-incremental learning over a large number of classes without revisiting prior samples. Update of already presented classes in the future is allowed but not required. This scenario is somewhat similar to the proposition from Maltoni and Lomonaco (2019) but with strict adherence to a single class CL.

5.2.2 Continual Learning Strategies

The most common methods targeting CL were categorized by Maltoni and Lomonaco (2019) along with Kemker and Kanan (2018), Kemker et al. (2018), and Zenke et al. (2017) as *Regularization*, *Replay and Rehearsal*, and *Architectural* based methods. We also add *Sparsity based* strategy, also discussed by Kemker et al. (2018), which is of our particular interest. We will now summarize the main work within these categories.

Regularization Strategies

The most popular regularization approach to CL method was proposed by Kirkpatrick et al. (2017) as the Elastic Weight Consolidation (EWC) technique. EWC uses Fisher's information matrix to identify parameters encoding high information content of the previously learned tasks and applies structural regularization to discourage new tasks from using these parameters. Similarly, Synaptic Intelligence (SI) method by Zenke et al. (2017) regulates the synaptic strength to minimize changes in parameters encoding past tasks. Unlike EWC, the SI computes the synaptic importance online and considers the entire learning trajectory, not only the final weight values. Chaudhry et al. (2018a) relate the EWC and SI weight importance metrics as distances in the Riemannian manifold and proposes a better performing method called Riemannian Walk (RWalk), which is based on the modified EWC and SI. A knowledge distillation method by Hinton et al. (2015) is leveraged to slow down weight changes in the Learning without Forgetting (LwF) method (Li and Hoiem, 2017). The LwF method uses distillation to approximate outputs of a network learning new tasks with a network trained on old tasks. Unfortunately, regularization methods do not work well in the class-incremental learning scenarios (Hsu et al., 2018, Farquhar and Gal, 2018).

Replay and Rehearsal

Catastrophic forgetting can be successfully reduced by interleaving already learned and new samples during the network training. In an extreme case, this reduces to mini-batch learning where IID samples of the entire distribution are interleaved. The replay (also referred to as episodic replay, rehearsal, or recall) is a very successful method to combat catastrophic forgetting and is believed to be the central mechanism of learning in our brain in the CLS theory. Many machine learning methods use the replay by keeping a subset of old samples in a replay buffer, sometimes called episodic memories, and mixing them with new data for the model updates.

A typical representative of a replay method is the Incremental Classifier and Representation Learning (iCaRL) Rebuffi et al. (2017) method. Furthermore, the iCaRL also introduces a Nearest Mean Exemplars (NME) method for inference. The NME creates a class prototype from training samples that is then used by the nearest neighbour classifier to predict the class category. The NME method has been frequently used in recent methods such as Douillard et al. (2020) and Hou et al. (2019).

Gradient Episodic Memory (GEM) method (Lopez-Paz and Ranzato, 2017) stores parameter gradients of past tasks for the replay. GEM uses the replay to stabilize gradients of model parameters over old samples while allowing gradient progression for new samples. A refined GEM method called Averaged Gradient Episodic Memory (A-GEM) was put forward by Chaudhry et al. (2018b), focusing on learning by a single pass through all the training data. A mathematically similar method to

the GEM, leveraging episodic replay with a meta-learning update, was proposed in Meta-Experience Replay (MER) method (Riemer et al., 2018).

To avoid storing a potentially infinite number of past samples, a pseudo rehearsal technique was proposed by Robins (1995). The pseudo rehearsal, frequently called generative replay, generates samples representative of the knowledge already stored in the model. These samples then can be interleaved with new samples to retrain the model. A dual-memory network with a recurrent network and pseudo rehearsal for transfer between short and long-term memory was already proposed by French (1997). Thanks to the relative simplicity of the pseudo rehearsal, many methods recently emerged, utilizing various generative algorithms such as AE (Kramer, 1991), VAE (Kingma and Welling, 2014) and GAN(Goodfellow et al., 2014a).

In Shin et al. (2017), a GAN model is used to produce samples of old data points, which are then interleaved in batches with samples of new classes to train a new generative model. Similarly, Ostapenko et al. (2019) apply a conditional GAN for the generative replay with added, learned synaptic plasticity expressed by means of binary masks. An innovative method by van de Ven et al. (2020) performs an internal replay in a Conditional Variational Autoencoder (CVAE) where the old tasks are represented by intermediate latent vectors in the model rather than images. The van de Ven et al. (2020) method also adds a conditional gating of neurons in the network for specific tasks. Another example of VAE based generative replay is studied by Kang and Zhang (2018). The authors modify the CVAE variational objective to follow the posterior distributions of already stored knowledge while updating the model with new data. The Meta-Consolidation for Continual Learning method (MERLIN) (KJ and Nallure Balasubramanian, 2020) uses meta-learning to consolidate model parameters of old and new tasks through parameters replay with VAE. This method then generates an ensemble of models for each task for the inference.

Architectural Strategies

An early successful attempt to reduce catastrophic forgetting with a specific neural network architecture is the Progressive Neural Networks method (Rusu et al., 2016). This method learns to freeze subsets of network parameters that encode old tasks and expand to learn new tasks. This approach has shown remarkable resilience to CF but at the cost of extensive memory consumption, especially for long sequences.

Another example of a similar approach is the PathNet (Fernando et al., 2017) which learns a path through the network for each task that is then frozen for subsequent learning updates. This reduces catastrophic forgetting but leads to network capacity exhaustion. A biologically inspired method in this category is the context-dependent gating (XdG) by Masse et al. (2018). The interference between tasks is avoided by gating randomly selected subsets of neurons assigned to each task. This method uses the identity of the tasks during learning and inference, thus it is not directly applicable in the class-incremental learning scenario. Another biologically inspired method is the FearNet by Kemker and Kanan (2018), which follows the

CLS dual memory concept. The FearNet is based on short and long-term memory with generative replay for consolidation. In contrast to other dual memory methods, FearNet introduces a memory commit circuit, loosely based on the fear response of the amygdala, that reinforces the consolidation of strong experiences to the long-term memory. In neuroscience-inspired research, Parisi et al. (2018) argue for dynamic memory expansion through neurogenesis as a mechanism to reduce catastrophic forgetting.

The Bias Correction (BiC) method (Wu et al., 2019b) targets CL for large datasets. This method measures and corrects bias differences between old and new tasks with distillation (Hinton et al., 2015) in a small linear layer following the classification layer. The BiC method stores two sets of old samples, one for model update and the second for the bias correction. Another recent, CL-specific architecture is the UCIR (Learning a Unified Classifier Incrementally via Rebalancing) method (Hou et al., 2019). Using episodic replay, this method operates as a metric-learning model with cosine similarity and NME classification (Rebuffi et al., 2017). A combination of episodic replay, bias compensation, and distillation in a specific CNN architecture was also proposed by Douillard et al. (2020) under the name PODNet (Pooled Outputs Distillation for Small-Tasks Incremental Learning). This method uses replay of a limited number of old samples with distillation applied across CNN layers combined with a cosine loss function for bias minimization and NME for class category inference. The PODNet stores 20 images per every learned class. The authors evaluated this method in the task-free class-incremental learning scenario over 50 CIFAR-100 classes with state-of-the-art results. Despite the PODNet being a relatively complex method requiring episodic memory and an elaborate training regime, we compare its performance with our, substantially simpler method.

Sparse Coding and Sparse Representation Learning

Sparsity as a tool to reduce catastrophic forgetting has not received nearly as much attention over recent years as other strategies. Particularly binary sparse representations, frequently studied in the past (Hopfield, 1982, Kanerva, 1988, Willshaw et al., 1969), stand in the shadow of deep learning methods, arguably due to the complexity associated with learning the non-differentiable binary functions with gradient-based optimizers.

In early research on catastrophic forgetting in connectionist networks, Kruschke (1992) proposed the ALCOVE method, which uses orthogonality of sparse representations to reduce changes in weights of old tasks when learning new. The benefits of the sparse representations to minimize forgetting in recurrent networks for image classification was studied by Coop and Arel (2013) on Elman recurrent network (Elman, 1990) enhanced by sparse coding and expansion layers. Method CALM, proposed by Murre (2014), develops sparsity by competition among neural nodes not occupied by other representations.

It has been recently argued that ideal continual learning performance is achievable only with perfect memory of past experiences (Knoblauch et al., 2020). Under the lens of this study, a large body of research on memory models become immediately applicable as another powerful CL strategy.

Significant, biologically-inspired research on the topic of continual learning with SDRs is carried out in the work on HTM (Hawkins and Ahmad, 2016, Hawkins et al., 2019, Cui et al., 2017, Lewis et al., 2019, Ahmad and Hawkins, 2016). The HTM work has shown remarkable properties of the SDR, which we apply in our work.

Binary, and frequently sparse, representations lie in the center of many classical methods such as the Willshaw associative memory (Golomb et al., 1990, Willshaw et al., 1969) and Hopfield network (Hopfield, 1982) with recent updates to continuous representations by Krotov and Hopfield (2016) and even more recent work (Krotov and Hopfield, 2021) aligning the basic principles behind the Hopfield memory with the latest state-of-the-art deep learning methods on transformers (Vaswani et al., 2017). Notably, the sparse binary representations are the cornerstone of the SDM (Kanerva, 1988). The SDM was recently rephrased under the deep learning methodology in the works of Wu et al. (2018a), Wu et al. (2018b), Marblestone et al. (2020), and Ramapuram et al. (2021), all leaving the SDR concept and adopting Bayesian or deep learning methods with continuous representations and gradient-based learning.

The most compelling recent work on the utility of sparse representations in the CL domain was introduced by Javed and White (2019). With a modified meta-learning algorithm, Javed and White (2019) propose the OML method (Online aware Meta-learning) to learn representations that reduce interference among tasks over subsequent continual learning steps. To learn the representations, OML uses a modified MAML (Model-agnostic meta-learning) method (Finn et al., 2017) and a specific training regime where the meta-learning inner loop incrementally learns new tasks, while the outer loop performs a recall of past tasks. In follow-up work, Beaulieu et al. (2020) put forward a method ANML (A Neuromodulated Meta-Learning Algorithm) that extends the OML method for a neuromodulatory network with synaptic gating. The ANML method holds the latest state-of-the-art performance on learning 600 classes in the task-free class-incremental learning scenario on the Omniglot dataset.

5.2.3 Common Datasets for Continual Learning

The majority of CL experiments are conducted on modified datasets for image classification. The most common Split MNIST (Zenke et al., 2017) and permuted MNIST (Goodfellow et al., 2014b) datasets are modifications of the original MNIST (LeCun et al., 2010). The split MNIST introduces tasks where each is typically made of two-digit classes. Permuted MNIST creates several tasks or domains where each is a random pixel-wise permutation of the MNIST images. Despite its unrealistic nature, this dataset is still commonly used.

Initially proposed for few shots learning, the Omniglot (Lake et al., 2015) dataset is also used in the CL domain. Few shots learning is similar to the CL in that the few shots methods need to rapidly learn new classes with minimal forgetting. However, the few shots learners typically learn only a limited number of classes with an emphasis on the learning of a low number of train instances, in extreme cases a single class instance. The Omniglot consists of training and test images of disjoint class categories with high number of test classes (600). Omniglot is considered to be one of the most challenging, low resolution, grayscale datasets due to its high statistical variations among the image instances, large number of classes, and small number of train and test instances. We use this dataset for our experiments. We discuss this dataset in more detail in Section 5.4.1.

Split CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton, 2009) are other image classification datasets commonly used in CL domain. These datasets are more difficult in that they feature the natural world, color images in relatively low resolution 32x32. This resolution is higher than that of MNIST but small for discrimination of natural images. Unfortunately, the vast majority of the experiments with CL targets the Split CIFAR-10/100 with a small number, typically 10, tasks. Another CIFAR variation for CL is the iCIFAR-100 (Rebuffi et al., 2017). Here the authors split the 100 classes into tasks with 2,5,10,20 and 50 classes each. We evaluate our method on the CIFAR-100, where we learn half (50) classes in the true task-free, class-incremental scenario. More details on this dataset are in Section 5.4.1.

Other less common datasets for CL are the SVHN (Sermanet et al., 2012), Fashion-MNIST (Xiao et al., 2017), almost exclusively in the task split settings. Datasets Split CUB200 (Caltech-UCSD Birds-200) (Welinder et al., 2010) and Split MiniImageNet (Vinyals et al., 2016) (a 100 classes subset of the ImageNet (Russakovsky et al., 2015), are typically split into 20 tasks, each with 5 classes) and used for higher image resolution CL. The original 1000 classes ImageNet is also sometimes used but mainly in the task split setup.

Recently, new datasets Core50 (Lomonaco and Maltoni, 2017) and Stream51 (Roady et al., 2020) were assembled. These datasets target perhaps the most natural CL scenarios where each class is presented as a short video sequence. This allows the CL method to leverage temporarily correlated class instances.

5.2.4 Mitigating Catastrophic Forgetting with Meta-learning

Deep learning has shown remarkable performance in a vast majority of machine learning domains. This success can, in large part, be attributed to countless architectural modifications, extensive hyperparameters turning, and careful training and evaluation setups. Equally, applying a trained model to a different task requires retraining or fine-tuning on the target problem and often hyperparameters adaptation. Meta-learning methods address this problem by learning how to learn. Historically, meta-learning methods, typically under different names, can be traced decades back (Schmidhuber, 1987, Thrun and Pratt, 1998, Hochreiter et al., 2001b, Schmidhuber,

1993). A vast number of meta-learning methods is generally categorized (Vinyals, 2017) as Model-Based (Santoro et al., 2016, Munkhdalai and Yu, 2017), Metric-Based (Vinyals et al., 2017, Snell et al., 2017), and Optimization-Based (Finn et al., 2017, Li and Malik, 2017) methods.

The recent state-of-the-art (SOTA) in CL has been achieved with models OML (Javed and White, 2019) and ANML (Beaulieu et al., 2020) based on the meta-learning optimization method MAML (Model-Agnostic Meta-Learning) (Finn et al., 2017). We will now review the MAML method, followed by a review of its application to the CL domain.

Model-Agnostic Meta-Learning

The core of the MAML (Finn et al., 2017) method is a two-cycle training procedure with an outer loop performing typical iteration over batches of training samples and an inner loop that, over a small number of cycles, updates the main model parameters on task-specific training samples. The inner loop starts by copying the main model θ_0 parameters, which are then updated at each inner cycle, typically by stochastic gradient descent (SGD). All inner parameter updates are recorded as a sequence and presented to the outer loop as a single differentiable operation. All inner iterations, precisely all weight update operations, are then backpropagated through the outer loop and all weights updated. The objective within each inner iteration is to maximize performance on each task, while the objective of the outer loop is to maximize performance across all tasks, thus learning how to learn the individual tasks. We can also view this method as learning model weights for fast transfer learning to other tasks.

During the inference, the model parameters θ_0 are fine-tuned to a new task, not necessarily originating from the training dataset. This meta-learning method enables fast learning of new classes and is commonly applied in the few-shots learning domain. Figure 5.5 illustrates the model loss space across all tasks with a trajectory depicting the meta-learned model parameters θ_0. In the inner loop and then during the inference, the model parameters θ_0 are rapidly updated to improve performance on the specific tasks. In the example in Figure 5.5, it would be task 1 or 2 with final model parameters θ_1 and θ_2, respectively. In summary, the main goal of the MAML method is to learn the model parameters θ_0 to be closest to all target tasks.

While conceptually simple, the MAML method is rather difficult to configure and resource-demanding at training and inference due the need to compute higher-order derivatives to backpropagate through all inner loop updates. During the inference, then, to perform gradient descent steps. The MAML authors addressed the speed shortcomings in a model called First-Order MAML (FOMAML) (Finn et al., 2017), which avoids the higher derivatives by backpropagating only through the last update of the inner loop. This improved the training speed and memory utilization for a price of slightly lower accuracy compared to MAML. In another work, Nichol et al. (2018) proposed a first-order meta-learning algorithm called Reptile with the

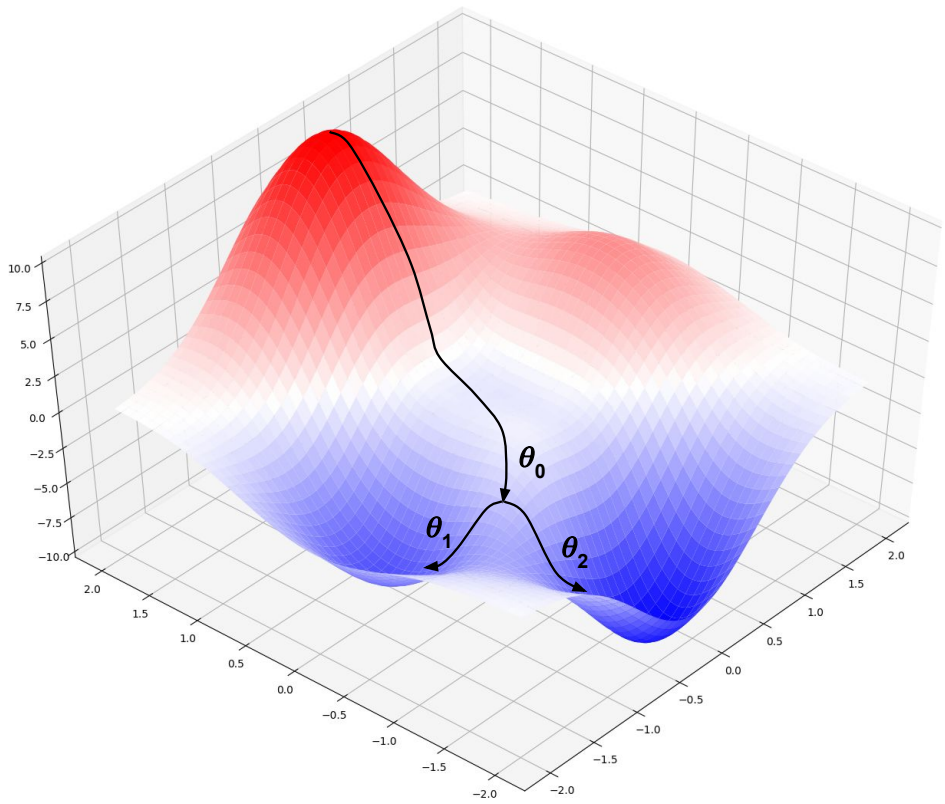


FIGURE 5.5: Illustration of the meta-learned main model parameters θ_0 which can be rapidly adapted to θ_1 or θ_2 for best performance on tasks 1 and 2 respectively.

FOMAML simplicity and accuracy comparable to MAML and even better on some specific tasks.

Meta-Learning Representations for Continual Learning

It is only very recently that the meta-learning methods have been applied to the CL problem in an OML (Meta-Learning Representations for Continual Learning) method (Javed and White, 2019) and later extended with neuromodulation in an ANML (A Neuromodulated Meta-Learning Algorithm) method (Beaulieu et al., 2020), setting a new state of the art performance on a large number of continually learned tasks on the Omniglot benchmark (Lake et al., 2015).

The OML model is a conventional CNN network split into a Representation Learning Layer (RLN) followed by few fully connected layers terminated by a classification layer called Prediction Learning Layer (PLN) (Figure 5.6). In each outer loop iteration, the OML method learns a new class in the inner loop over 20 iterations, each updating the model with SGD on a unique instance of the learned class. Upon exit of the inner loop, a loss is calculated over a number of recall samples. A batch of the recall samples includes 20 instances of the currently trained class but different from the training samples plus 64 other classes, one instance each. The loss

is then backpropagated through the chain of all inner updates. Unlike MAML, the OML updates only the PLN network. The CL evaluation is then conducted over two stages:

- **test-training**, where the model continually learns instances of new classes not seen during the main training stage.
- **test-testing**, where the model, trained over the test-training stage, is evaluated on the classification of the same classes used for the test-training but different instances.

During the evaluation, the PLN network is updated on classes from a disjoint meta-test-train dataset. Each new class is trained by a single SGD step on each of the 15 meta-test-train instances. The meta-test-test is then conducted on the same meta-test-train classes but different instances.

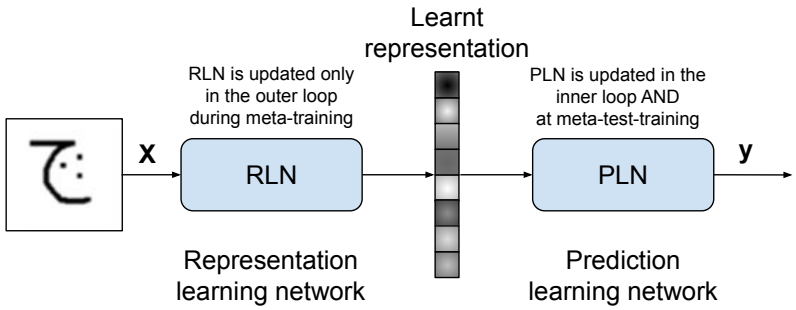


FIGURE 5.6: Network diagram of the OML model (Javed and White, 2019).

The ANML method (Beaulieu et al., 2020) introduces a neuromodulatory network (Figure 5.7) that individually gates each dimension of the learned representation via a sigmoid function. The ANML preserves the OML training regime. Only the final classification layer is updated in the inner loop during the meta-training and meta-test-train stages. The Prediction and Neuromodulatory networks are updated in the outer loop and never during the inference - the meta-test-training stage.

Over numerous experiments with the OML and ANML, we reached several observations leading to several intuitive modifications resulting in a performance gain.

First, updating only the last classification layer during the meta update in the OML, as done in the ANML, improved the performance, matching the ANML. This is quite obvious since learning a new class during the meta-training and meta-test-training steps updates primarily only weights connected to the specific target class in the classification layer, therefore minimizing interference with other classes.

Second, the cross-entropy loss function, used in the meta-test-train update in OML and ANML, updates weights of all classes in the classification layer with every new class learned. This is the correct and desirable behavior of the cross-entropy loss function for mini-batch gradient learning with IID data. It pushes the correct target class weights in the classifier towards the classifier's input and away weights

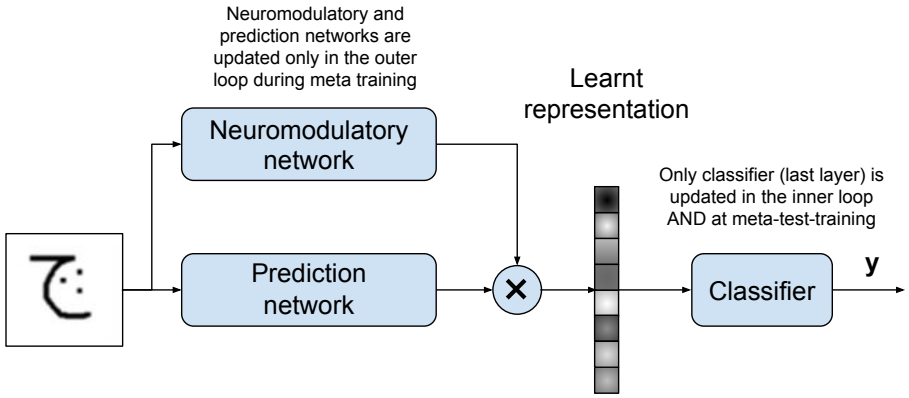


FIGURE 5.7: Network diagram of the ANML model (Beaulieu et al., 2020).

of all other classes. During the continual learning, this meta-test-training updated causes interference with classes learned in the past. Therefore, we modified the meta-test-train loss function to update only weights of the currently learned class, which further improved the performance. These modifications were implemented in our OML-S, OML-W, and OML-WB models discussed in Sections 5.5 and 5.6.

Finally, we modified the meta-training and meta-test-training procedures to dynamically expand the classification layer as new classes are being learned. This modification enables these methods to learn a potentially unlimited number of classes bounded only by the available computational resources.

Our experiments with the ANML showed that the neuromodulation does not improve the performance over the OML with a single adaptive layer and modified loss function, and thus in our further experiments we focus on the OML architecture. In summary, the meta-learning is capable of learning representations highly suitable for CL irrespective of the final classification method, whether with multiple layers as in the original OML, neuromodulation in ANML, or a single classification layer with class-isolated meta-test-training updates.

5.2.5 Current State-of-the-Art, its Limitations and Promising Research Directions

The current state-of-the-art method in the task-free, class-incremental continual learning scenario (as discussed in Section 5.2.1) is the ANML (reviewed in Section 5.2.4). This method addresses the catastrophic forgetting with the sparse coding strategy (see Section 5.2.2) by meta-learning sparse representations.

The major drawbacks of most CL methods could be summarized as lack of performance compared to the training of similar models with IID batches, still suffers from a high level of Catastrophic Forgetting (CF), and lastly, memory and computational demanding training and inference. The high memory load is due to their need to either maintain very large models (typically not optimally utilized by dedicating the network sub-paths to specific tasks only - isolating tasks by not sharing weights,

Section 5.2.2) or to maintain a replay buffer (rehearsal buffer or episodic memory) for a large number of past training samples (Section 5.2.2). The high computational load is mainly caused by the need to retrain the entire model on the new data interleaved with all past training data or some subset from the replay buffer or to generate samples representative of the already stored knowledge. Additionally, some methods use memory and computationally intensive algorithms such as meta-learning.

Following the evidence from neuroscience and current machine learning methods for CL, it appears that a large part of the CL problem will likely be solved by learning high-dimensional, discrete sparse representations. We follow this approach in our work. Sparse representations alleviate catastrophic forgetting among learned tasks but do not implicitly improve forward and positive backward transfers associated with the short to long-term memory consolidation and long-term memory storage. Therefore, a combination of sparse representations with memory replay (rehearsal), akin to the CLS (see Sections 5.1.1, 5.2.2), appears as a promising approach to address both the continual learning in short-term memory and the subsequent consolidation in long-term memory that promotes the forward and positive backward transfers.

5.3 Continual Learning with Sparse Binary Representations

As discussed in the previous Section 5.2.4 the meta-learning can produce representations with good performance on CL tasks. The meta-learning models are, however, difficult to configure and train and resource-demanding. There are many sensitive hyperparameters to tune. For example, the OML requires to configure the number of samples in the training trajectory structure, that is, the number of instances necessary to learn a new class, instances to recall the learned class, and the number of past classes to recall. There are the outer and inner learning rates to set and also the meta test-train learning rate. The meta-test-train learning rate significantly affects the final meta-test-test results and in both OML and ANML studies, the authors perform a computationally demanding grid search to find settings for best results.

Exploration of neural coding in the brain and the hippocampal mechanisms, believed to be consequential to our ability to form new memories without forgetting the past, inspired two aspects of our method. First, we followed the fast, early, and permanent stabilization of the primary visual cortex and proposed a representation learning CNN that is trained only once on the target visual domain (in the ideal case, the visual domain of the entire world), and then fixed for the lifetime of the model. The stable visual features considerably reduce the plasticity stability dilemma rising during the incremental learning updates. The center of the continual learning problem then shifts to the latter layers. Second, for the continual learning stage, we leveraged the high dimensional, sparse binary representations, identified as the dominant neural coding format in our brain (Khaligh-Razavi et al., 2016, Tee and

Taylor, 2020, DiCarlo et al., 2012, Rigotti et al., 2013, Chung et al., 2018). We particularly drew inspiration from the pattern separation observed in the Dentate Gyrus (DG) in the hippocampus that was experimentally shown to avoid interference between similar patterns (Neunuebel and Knierim, 2014, Leutgeb et al., 2007, Lee et al., 2015). To simplify referencing, we call our method SBRCL, which stands for Sparse Binary Representations for Continual Learning.

The core idea behind our method rests on the function of pattern separation observed in the hippocampus (Section 5.1.2). Rather than using meta-learning to learn model weights close to the targets tasks, that still need to be refined during the meta-test-training steps, our model learns a high dimensional sparse binary latent space that already encapsulates all target tasks.

5.3.1 Properties of High Dimensional Spare Representations

Sensory data are typically high dimensional and continuous; however, they lie on low dimensional structures. Elhamifar and Vidal (2013) reported that these structures do not occupy a single continuous subspace but rather the union of many small, discrete subspaces. These low dimensional subspaces are likely disconnected. For example, image or audio features cannot be expected to continuously interpolate and still lie in a domain of natural data. Continuous, low dimensional representations can be learned by clustering common data features by enforcing some prior on the latent space, for example, the Gaussian distribution as is the case of VAE (Kingma and Welling, 2014) models. However, sampling new images, interpolation, or classification with these features is far from optimal. Discrete autoencoders such as VQ-VAE (van den Oord et al., 2017) or LBAE (Fajtl et al., 2020) exhibit higher performance, which indicates that the discrete and sparse features may be a better way to represent the image data structure.

Benefits of high dimensional sparse representations have been extensively discussed and applied in many seminal studies, most notably by Willshaw et al. (1969), Hopfield (1982), and Kanerva (1988) and applied in the computer vision domain (Wright et al., 2008, 2010). Our work primarily concentrates on the orthogonality, union, and overlap properties, as discussed in Ahmad and Hawkins (2016).

Despite the small number of active bits, sparse binary representations can still reference a vast number of patterns. The number of patterns for n dimensions with w active bits is given by the binomial coefficient:

$$\binom{n}{w} = \frac{n!}{w!(n-w)!} \tag{5.1}$$

In the case of the SBRCL latent space with $n = 4096$ and sparsity $5\%(w = 200$ active bits$)$, the number of patterns is more than 10^{345}.

The most crucial property of sparse, high dimensional spaces is the tendency towards orthogonality of random vectors, which becomes more pronounced as the

dimensionality grows. The orthogonality underpins operations such as the union set and overlap similarity.

The union of binary vectors is a result of binary OR operation between bits at corresponding dimensions. The probability of exact match of two random vectors is:

$$p_{fp} = \binom{n}{w}^{-1} \tag{5.2}$$

In our example $p_{fp} \approx 0$ for $n = 4096, w = 200$. For $w = 100, p_{fp} \approx 10^{-200}$ and for $w = 10, p_{fp} \approx 10^{-30}$. The match probability increases with the increase in sparsity but is still near zero. Consequently, a union of random vectors always expands the number of active bits. With more patterns in the union, the bits will saturate, which results in a large number of false positives. This behavior is similar to the Bloom filter (Bloom, 1970), where the false-positive rate grows with the number of stored patterns, but it never results in false negatives.

Union of M random Sparse Binary Representation (SBR) patterns with exactly w active bits leads to probability of false positives p_{fpu} defined as:

$$p_0 = \left(1 - \frac{w}{n}\right)^M \tag{5.3}$$

$$p_{fpu} = (1 - p_0)^w, \tag{5.4}$$

where p_0 is the probability of any bit being zero in the union. The p_{fpu} is the probability that a random vector overlaps by w bits with the union set of M random patterns. In our example, $p_{fpu} = 0.991$ for $n = 4096, w = 200$ and $M = 200$ patterns. For $M = 100, p_{fpu} = 0.26$ and for $M = 50, p_{fpu} = 4 \times 10^{-8}$. We can see that the more random patterns we add the faster the union space gets exhausted.

Since we do not store random but rather correlated patterns, the sparsity of the union will decay much slower, depending on the SBR encoder. We can calculate the probability of false positives as a probability of random vector y, with sparsity $\binom{n}{w}$, overlapping by exactly b bits with union U of M correlated patterns with sparsity $\binom{n}{w_u}$ (Ahmad and Hawkins, 2016, O'reilly and McClelland, 1994). The union U may, for example, represent patterns constituting a single image class. Assuming, subject to the SBR encoding, that representations of single image class are highly correlated, the sparsity of the corresponding union would be substantially lower than the union of random vectors. To calculate the probability of a random vector y overlapping with the union U, we first calculate the total number of all overlapping vectors as:

$$\Omega_u(n, w, b) = \binom{w_u}{b}\binom{n - w_u}{w - b}, \tag{5.5}$$

where Ω_u is the number of all possible vectors with sparsity $\binom{n}{w}$ that have exactly b bits overlap with union U with sparsity $\binom{n}{w_u}$. Here, we assume the number of active bits in the union U is w_u. The probability of the y random vector overlapping with

the union is

$$p_{fpo} = \frac{\Omega_u(n, w, b)}{\binom{n}{w}}. \tag{5.6}$$

For sparsity of the overlapping set that equals to the desired number of matching bits $b = w$, the equation reduces to

$$p_{fpo} = \frac{\binom{w_u}{w}}{\binom{n}{w}}. \tag{5.7}$$

For example, for $n = 4096, w = 200, w_u = 400$ probability of false positives is $p_{fpo} = 4 \times 10^{-227}$. We can observe that if the union is an assembly of vectors with correlated dimensions, the probability of a false positive match is extremely low.

There is an apparent trade-off between the sparsity and generalization. In the case of extreme sparsity, the representations of class instances are orthogonal to each other. There is either an exact match between identical instances or no match between similar but not exact instances; thus, the generalization property is lost. We can only check the presence of the exact training samples in the union. Extreme sparsity, leading to the lack of generalization, was already studied by Sharkey and Sharkey (1995) and French (1994). On the other hand, low sparsity reduces the discriminative property of the SBRs between tasks due to the considerable overlap. Therefore, the performance of the union set as a pattern attractor for classification as well as the overlap similarity rests on the quality of the SBR, particularly the level of sparsity and the within and between-class correlations.

In our method, we propose to learn the sparse binary representation with ℓ_0 regularization over activations and only in the last Layer. The motivation to learn the sparsity in the latter layers originates from the work of Hoefler et al. (2021), who showed in a number of experiments that sparsity, learned by pruning, naturally occurs mostly in the last layers of deep neural networks. This also coincides with the progression of catastrophic interference through the ANN layers (Ramasesh et al., 2021), where the latter, higher layers, exhibit the highest forgetting.

Finally, despite the high dimensionality, the SBR representations are memory efficient. The SBR can be encoded on common computer architectures, without a loss of information, by storing indices of active bits or directly mapping bits to machine variable types such as int32. To encode the SBR by the active bits indexing, the number of bytes b required to store n dimensional representation with sparsity s in percentage can be calculated as follows:

$$w = \left\lceil \frac{s}{100} \right\rceil \tag{5.8}$$

$$b = \left\lceil \frac{log_2(n)}{8} \right\rceil w, \tag{5.9}$$

where b is the number of bytes needed and w is the number of active bits in the SBR. For possible compression of the SBRCL representations, see Section 5.9. Sparse representation is also known to be an energy-efficient neural coding in the brain with

low metabolic cost (Graham and Field, 2006).

In addition to the SBRCL that directly learns the binary representation, we also propose a meta-learned SBRCL, referred to as MLSBRCL. This model uses the meta-learning method only to learn the binary representations, but the continual learning and inference are performed by the union set and the overlap similarity.

5.3.2 SBRCL Method

The core of the SBRCL model is a common multilayer CNN network split into three logical modules. A front-end feature extractor CNN is followed by a single or a few layers network learning the sparse binary representations. During the representations learning, the binary representations enter a common linear classification layer (Figure 5.8). During the continual test-training stage, the binary representations are combined by the binary OR operation and added to the memory matrix $\mathbf{M}$. We will discuss details of the continual learning and inference processes in Section 5.3.2.

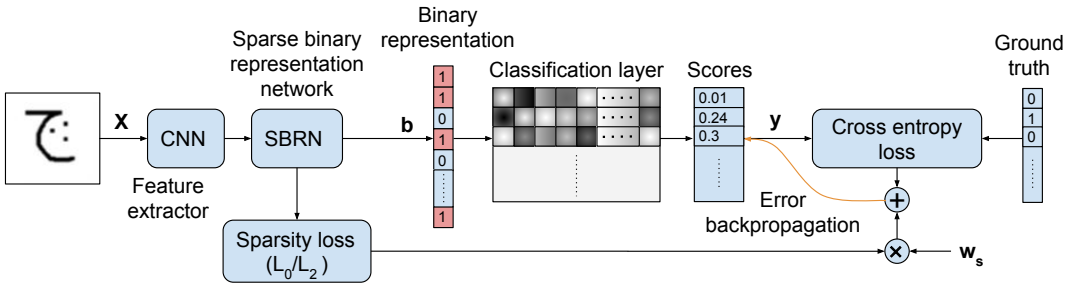


FIGURE 5.8: SBRCL is a conventional CNN with binarization before the final classification layer, used only during the representations learning. During the continual learning the classification layer is replaced with a matrix $\mathbf{M}$ of binary pattern attractors.

Learning Binary Latent Space with Backpropagation

The binary vectors $\mathbf{b} \in \{-1, 1\}^N$ are produced by rounding the continuous N dimensional latent $\mathbf{z} \in \mathbb{R}^N$ with the sign() function followed by the max operation.

$$b_i = f_b(z_i) = \begin{cases} 1, & \text{if} \quad z_i \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.10}$$

A diagram of this operation is shown in Figure 5.9. The binarization function $f_b()$ is a non differentiable operation, which cannot be directly backpropagated. To avoid this issue, we substitute the otherwise zero gradient function $f_b()$ with a surrogate, unit function $f_s()$, which lets the gradient flow unchanged through the binarization.

We accomplish this but implementing the binarization function as follows:

$$\mathbf{b} = max(0, sign(\mathbf{z})) + \mathbf{z} - sg(\mathbf{z}) \tag{5.11}$$

where the $sg()$ is a stop gradient operation that zeroes gradient for every input. During the forward pass, the $\mathbf{z} - sg(\mathbf{z})$ cancels out. During the backward pass, the gradient through $max(0, sign(\mathbf{z}))$ and $sg(\mathbf{z})$ stops, but it will flow unchanged directly through $\mathbf{z}$.

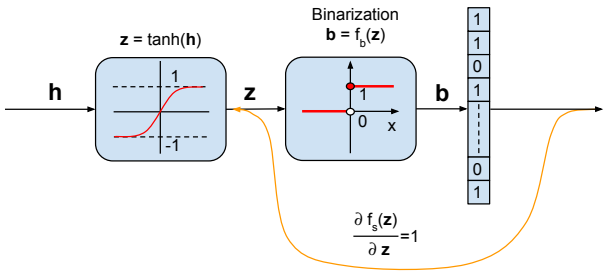


FIGURE 5.9: Binarization layer

Enforcing Sparsity

Sparsity in the representations learned by the OML and ANML networks naturally arises during the meta-learning process. Our experiments showed that this phenomenon is not unique to these methods. This is, in part, caused by the competitive process established by the cross-entropy loss function in the outer loop. In fact, the sparsity in the OML and ANML methods can be controlled by introducing a temperature T to the softmax in the cross-entropy loss in the outer loop as follows:

$$p_i = \frac{e^{\frac{\mathbf{h}_i}{T}}}{\sum_{k=1}^{C} e^{\frac{\mathbf{h}_k}{T}}} \tag{5.12}$$

$$L = -\sum_i y_i log(p_i), \tag{5.13}$$

where $\mathbf{h}$ are the logits of the classification layer, C number of classes, $\mathbf{y}$ ground truth labels and L the cross entropy loss scalar.

In addition to the sparsity induced by the learning process with the cross-entropy loss, we introduce a ℓ_0,ℓ_2 regularization on the activations of the binary latent vector $\mathbf{b}$ as follows:

$$S^j = \frac{1}{N}\sum_{i=1}^{N} \mathbf{b}_i^j \tag{5.14}$$

$$L_s = \frac{1}{B}\sum_{j=1}^{B} (S^j)^2, \tag{5.15}$$

where S^j is the ℓ_0 sparsity penalty of the j^{th} sample in the batch, calculated as a number of non zero elements in $\mathbf{b}^j$. L_s equalizes the ℓ_0 losses across the training batch of B samples. The final loss is then calculated as the cross-entropy loss L_c plus sparsity penalty L_s weighted by w_s.

$$L = L_c + w_s L_s \tag{5.16}$$

We set the $w_s = 1$ for the duration of the main training stage and $w_s = 10$ for few final training epochs. We found this weight experimentally, however, its exact value is not critical; values up 50 resulted in the same training profile and final performance. Higher values then resulted in a performance decline.

Continual Learning and Inference

As with the meta-learning methods, the continual learning and inference in SBRCL are split into the *test-training* and *test-testing* stages. During the *test-training* stage, instances of novel classes, not used during the representation learning, are incrementally learned by the model. During the *test-testing* stage, new instances of the *test-train* classes are classified.

We follow a simple, intuitive model to form a template characterizing a single learned task (class). Let us consider an instance of a given class as a pattern represented by the SBR whose small subset of bits encodes all features of this pattern. To ensure that these known patterns are always correctly detected during the inference, we set all active SBR bits of all training instances in the template. We create a union of single class training instances by performing a binary *OR* operation along each SBR dimension. We call this union vector a *pattern attractor*. Pattern attractors of all learned classes are stacked in a memory matrix $\mathbf{M}$ for inference. This process is illustrated in Figure 5.10.

Formally, the union a_c over representations $\mathbf{b}^j$, $j \forall \in [1, ..., I_c]$ of all instances I_c of class c is calculated as

$$\mathbf{a_c} = f_b \left(\sum_{j=1}^{I_c} \mathbf{b}^j \right). \tag{5.17}$$

The function $f_b()$ (Eq. 5.10) binarizes the vector sum into range $b_i^j \in \{0, 1\}$.

The inference during the test-test stage is carried out by performing binary AND operation between binary latent of the test sample and all attractors in the memory matrix $\mathbf{M}$. It is calculated as a dot product between binary vectors.

$$y_c = \mathbf{M}_c \, \mathbf{b} \tag{5.18}$$

The results is a number of matching active bits between $\mathbf{M}_c$ and $\mathbf{b}$. The overlap operation over the memory matrix $\mathbf{M}$ is shown in Figure 5.11. The test sample is categorized by a class whose attractor shows the highest overlap wit the sample's

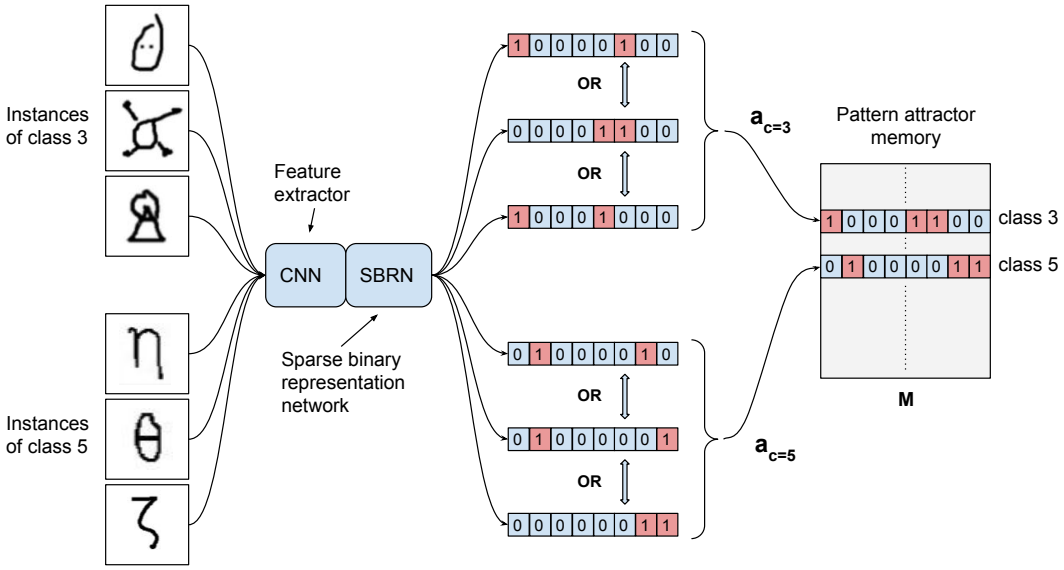


FIGURE 5.10: With trained CNN and SBRN (a single CNN for Omniglot) a new class c is learned by calculating union set $\mathbf{a}_c$ of all training instances of the new class (binary OR operation over the representations) and adding it into the pattern attractor memory matrix $\mathbf{M}$. A class of an unknown image is inferred by overlapping (binary AND) its binary representation with all attractor patterns in $\mathbf{M}$ and selecting the one with the highest overlap.

binary representation.

$$c = \underset{c}{\operatorname{argmax}}(y_c) \tag{5.19}$$

The union set and overlap metric may not appear as the best candidates for the learning and inference. However, by computing the overlap of the test representation with a pattern attractor (a row in the memory $\mathbf{M}$), we are checking how many discrete semantic features in the test sample agree with the specific pattern attractor while ignoring all other semantic features. We assume here that the bits in the binary representation have semantic meaning. The Euclidean or Hamming distance, on the other hand, tell us how many semantic components differ between the template and the test sample, which is not a valid measure to categorize the pattern. Intuitively, a class category of a pattern is given by the number of semantic features it shares with the template rather than the number of features it is missing. The difference between the overlap set and the Hamming distance calculations is illustrated in Figure 5.11. As pointed out in the work on the sparse distributed representations by Hawkins and Ahmad (2016), Ahmad and Hawkins (2016), and Cui et al. (2017), the overlap metric is also a biologically plausible method since it does not require full connectivity among all neurons in the layer. The full connectivity would be necessary to calculate the Euclidean, Hamming, and other p-norm similarities.

Finally, the union and overlap operations are extremely fast to execute and simple to implement on most computational platforms.

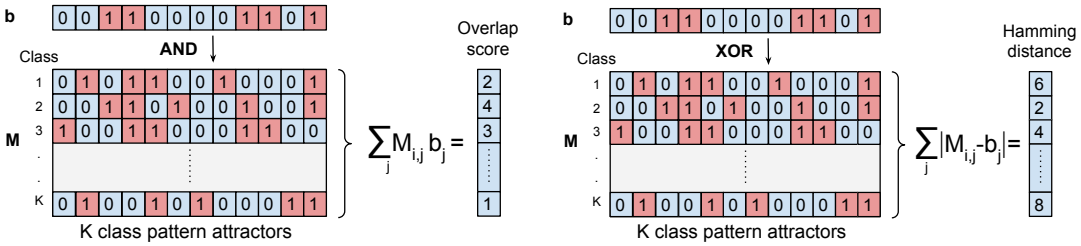


FIGURE 5.11: Overlap similarity calculation between test representation **b** and the memory of pattern attractors **M** (left). Hamming distances for the identical setup are shown on the right.

5.3.3 Meta-Learned Sparse Binary Representations

To compare the binary and the meta-learned representations, we introduced a number of modifications to the OML. First, we replaced the last layer of the representation learning network (RLN) with the binary layer of the exact dimensions (4096) as in the SBRCL. The prediction network (PLN) was set up with only a single classification layer updated only during the inner meta-training loop and the test-training stage. We call this *OML-WB* model as the OML Wide Binary. Second, we modified the *OML-WB* by replacing the entire meta-test-training and meta-test-testing process with the union set and overlap similarity methods, identical to SBRCL. We call this *MLSBRCL* method as the Meta-Learned SBRCL.

5.3.4 Contributions to the Field of Neural Architectures

The SBRCL model builds on existing neural network methods, particularly CNN and the meta-learning method. Additionally, we leverage established activation, regularization and optimization methods, namely: ReLU and $tanh()$ activations, ℓ_0, ℓ_1, ℓ_2 and dropout regularizations and ADAM (Kingma and Ba, 2015) optimization. Our method introduces a novel, binarization layer, end-to-end trainable with gradient-based methods and a single layer architecture for continual learning of binary representations centered around binary *OR* and *AND* operations. Further on, for very high dimensional latent space, necessary to express a sufficient amount of information in complex data such as natural images of the CIFAR-100 dataset, we propose a binarization layer with information bottleneck as discussed in Section 5.6.

5.4 Evaluation Protocol

We adopt the evaluation protocol from Javed and White (2019) and Beaulieu et al. (2020). The model is first trained on a dataset of training classes disjoint with the test dataset. The trained model is then evaluated on the continual learning tasks on the test dataset.

The SBRCL and MLSBRCL methods for Omniglot utilize an exact copy of the representation learning network introduced by the OML (Javed and White, 2019)

method. Since the original OML study, no other follow-up work has evaluated the OML on the CIFAR or other datasets of natural images; we adopt ResNet (He et al., 2016) architecture for the representation learning network. The dimensions of the binary latent representations and the information bottleneck for the CIFAR model have been devised empirically. In particular, the latent representation for the Omniglot model has been chosen to have twice the dimensionality of the OML, continuous-valued representation, and for the CIFAR model twice the dimensionality of the Omniglot SBRCL model. This was to increase the latent dimensionality compared to the continuous-valued representations that would still maintain a reasonable information bottleneck. Moreover, the higher dimensionality of the CIFAR-100 SBR, compared to the Omniglot, reflects higher CIFAR data complexity. The ℓ_0 regularization controlling the SBR sparsity and the w_s weight balancing the classification accuracy with the sparsity were set to produce in average $\approx 5\%$ of active latent bits during training. Other hyperparameters were chosen experimentally in line with the prior work OML and ANML to maximize classification accuracy and reduce the overfitting of the representation learning network. There are no hyperparameters in the SBRCL and MLSBRCL continual learning networks.

The test is conducted in two stages; test-training and test-testing. During the test-training stage, the model learns a sequence of test class instances, where each is presented to the model only once and then discarded. Only after all instances of one class have been presented, next class can be learned. When the entire sequence of all classes is learned, the model is evaluated on the test-test instances.

The performance is reported as an average accuracy over all tasks learned up to a checkpoint. For example, after learning 100 Omniglot test classes, we report an average accuracy over all test instances of the learned 100 classes. We also report the *average incremental accuracy* metric introduced by Rebuffi et al. (2017) and adopted in a number of recent studies (Douillard et al., 2020, Hou et al., 2019). This metric was proposed to be used in cases where a single value is preferable.

5.4.1 Datasets

We conduct all experiments on the Omniglot (Lake et al., 2015) and CIFAR-100 (Krizhevsky and Hinton, 2009) datasets. A CL test on an out-of-distribution data is carried out on the MNIST (LeCun et al., 2010) dataset.

The Omniglot dataset contains 1623 images of handwritten characters originating from 50 diverse alphabets. Each character was drawn by 20 different people, thus there are 20 instances of each character. The dataset is further split into 963 training and 660 test images with disjoint classes. Examples of three Omniglot alphabets are shown in Figure 5.12. In all our experiments, the Omniglot images were scaled down to a grayscale 28x28 resolution.

MNIST is an image classification dataset with 10 classes, each with 6000 training and 1000 test images of handwritten digits. MNIST images are grayscale with white digits on black background with 28x28 pixels resolution. For our experiments, we

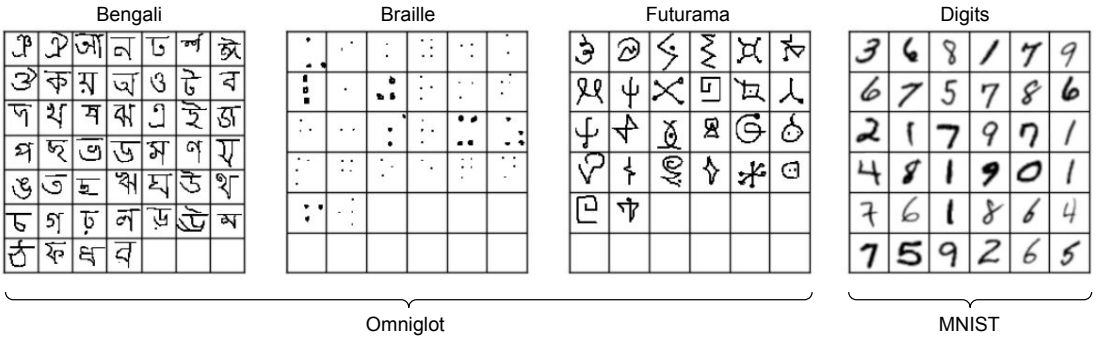


FIGURE 5.12: Examples of three Omniglot alphabets and the MNIST digits.

inverted the images to black foreground and white background as are the Omniglot images. Examples of some MNIST digits are shown on the right in Figure 5.12.

CIFAR-100 dataset contains 100 classes, each with 500 training and 100 test, RGB images with resolution 32x32 pixels. Some examples are shown in Figure 5.14. We conducted all experiments with original 32x32 RGB images. For the continual learning scenario, the datasets were split into the training and test datasets with disjoint classes and then further into test-train and test-test datasets of the same classes but disjoint instances. The split is illustrated in Figure 5.13.

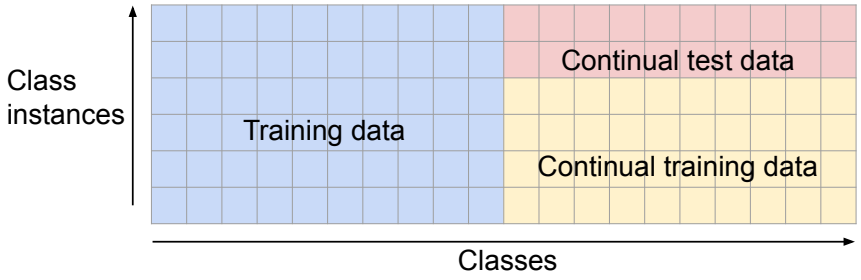


FIGURE 5.13: Illustration of the train, test-train and test-test dataset splits.

5.5 Experiments on Omniglot Dataset

The SBRCL model for the Omniglot dataset learns 4096 dimensional binary representations. The model is trained on batches of 64 unique classes, identical in size to the OML remember sequence. In addition to the remember sequence, the OML training sequence also includes 20 instances of the learned class plus another 20 instances of the same class concatenated to the remember sequence. Before each batch update, we reset weights in the classification layer belonging to a randomly selected class from the 64 classes in the training batch. We found that this technique significantly helps with the model regularization. The value of $w_s = 10$ was established experimentally over several trial runs, however, the exact value is not critical (see Section 5.3.2). We train the model for 42 epochs where it appears to reach the best

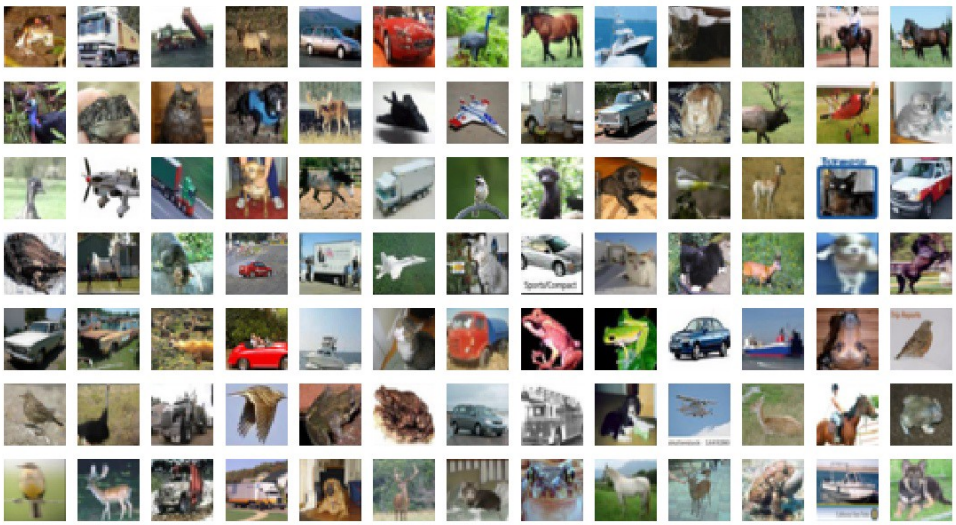


FIGURE 5.14: Examples from the CIFAR-100 dataset.

performance. We have not carried out any extensive optimization or hyperparameters tuning. It is likely that such tuning may further improve the performance.

For the evaluation, we use test-train and test-test split, identical to the OML and ANML. That is, for the test-training, we use 15 random instances of the 660 test classes. For the test-testing, then the remaining 5 instances of the same classes. All reported results are averaged over results on 5 random splits of the test dataset.

The OML-W model is the OML model with a wider, 4096 dimensional representation with float32 values. The OML-WB is identical to OML-W, but it adds the binarization layer identical to the SBRCL, thus learning 4096 dimensional binary representations. Finally, the MLSBRCL model is the OML-WB with the union and overlap continual learning and inference method.

TABLE 5.1: Test-test accuracy on the Omniglot dataset. Each column shows average accuracy over all learnt classes up to that point. Last column shows mean incremental accuracy in percent as proposed by Rebuffi et al. (2017). All results are averaged over five models trained on five random test dataset splits. Our methods are higlighted in bold.

	Sequence lengths. Mean accuracy over all classes learnt in each sequence are bellow in %							
Method	10	100	200	300	400	500	600	Inc. Acc.
ANML	96.00	83.00	79.00	75.00	71.00	66.00	63.80	76.26
OML	92.00	69.00	55.00	41.00	32.00	23.00	18.20	47.17
OML-W	98.00	92.72	89.42	86.88	85.71	83.50	81.59	88.26
OML-WB	98.00	92.32	88.34	86.23	85.17	82.53	80.68	87.61
MLSBRCL	98.40	**95.12**	**92.88**	90.41	**90.05**	**88.14**	**87.15**	**91.74**
SBRCL	**99.20**	94.40	92.44	**90.67**	88.14	86.42	85.00	90.89

All models were evaluated on sequences of continually learned 10, 100, 200, 300, 400, 500, and 600 tasks where each task is one class. Each sequence was evaluated individually over five test dataset splits. Accuracy averaged over all tasks learned in

the sequence over the five splits is reported in Table 5.1. Our methods are compared to the ANML (Beaulieu et al., 2020) and OML (Javed and White, 2019) methods as the current SOTA in this domain. In Figure 5.15, we show the results on the test-test data.

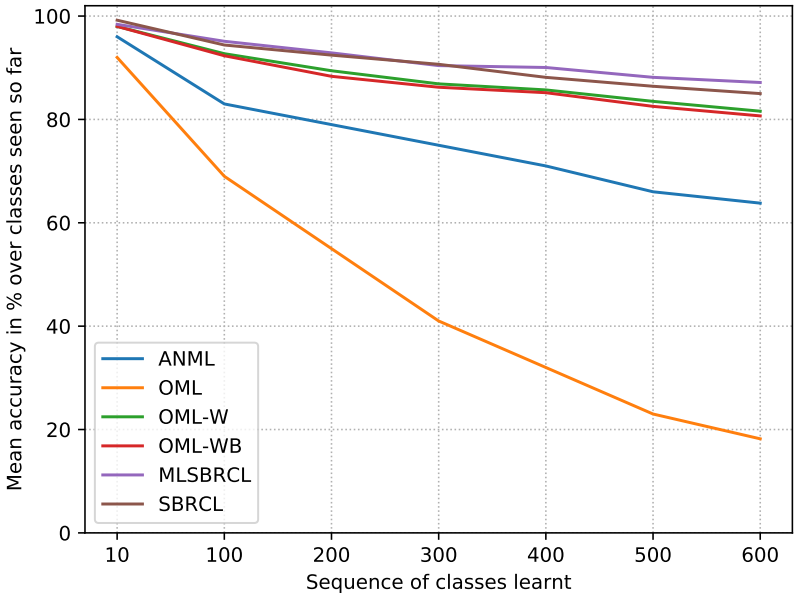


FIGURE 5.15: Continual learning performance on the Omniglot test-test dataset.

The MLSBRCL method performs marginally better than the SBRCL, which indicates that the meta-learning produces semantically more meaningful binary representations but only in combination with the union/overlap CL learning and inference method. The OMNI-WB method, identical to MLSBRCL but with the meta-training and meta-testing for inference, shows a decline in performance.

The OMNI-W and OMN-WB show very similar performance on the Omniglot, which indicates that our binarization method performs well even in the setup where the features are evaluated in the meta test region as continuous-valued features. In fact, the binary features significantly reduce the information flow through the representation layer compared to continuous features of identical dimensions. Consequently, bits in the binary features must correlate more with specific semantic features, unlike the continuous values that can encode many representations into a single dimension.

The performance of our methods is substantially better than that of the OML and ANML. We can attribute this gain to the higher dimensionality of the learned representations (2304 vs. 4096) for the meta-learned methods, but more importantly to the sparse binary representations and the union/overlap technique in the simple SBRCL method.

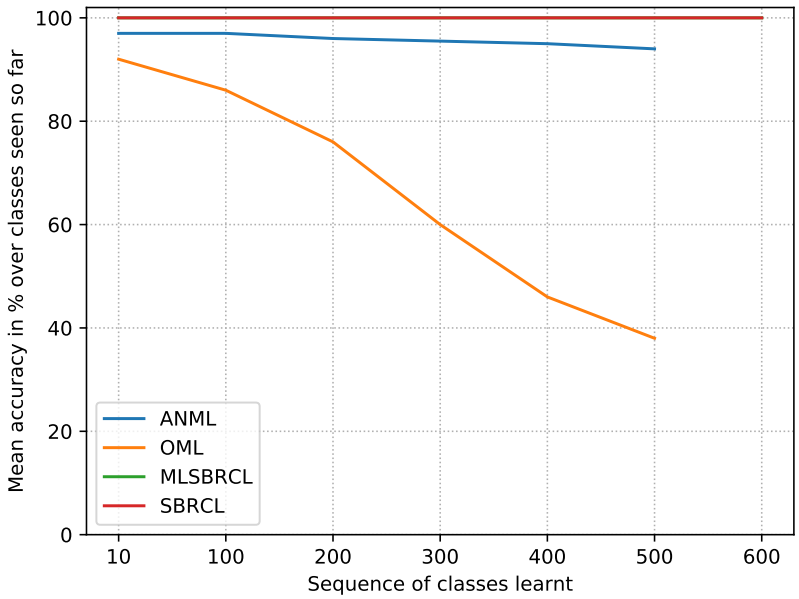


FIGURE 5.16: Continual learning performance on the Omniglot test-train dataset.

In Figure 5.16, we report accuracy on the test-training samples; we classify the same images used during the CL training. Both models MLSBRCL and SBRCL, show 100% accuracy for all CL sequences. Representations of the training samples are combined with the binary OR operation and compared to the overlap with the binary AND operation. Therefore, as with the Bloom filter (Bloom, 1970), the inserted patterns never result in false negatives in the test. Equally, our method always detects the training samples in the memory $\mathbf{M}$, but if the patterns are not well separated among the classes, false positives can be generated.

A correlation between binary representations across 30 random classes from the test dataset is shown in Figure 5.17. The figure was produced by correlating the binary features of ordered 15 instances of each class. That is, 15 instances of class A are followed by 15 instances of class B and so. We can see a strong correlation within the classes instances but weak between different classes. More detailed analysis with quantification follows in Section 5.8.

5.6 Experiments on CIFAR-100 Dataset

The SBRCL model for the CIFAR-100 model is composed of a CNN feature extractor front-end followed by two layers network for binary representation learning. The CNN network is trained on the first half of the 100 CIFAR-100 classes and stays fixed over the duration of the CL learning and inference.

The reason for this architecture is twofold. First, the CIFAR-100 dataset is considerably more challenging due to its color, natural 32x32 images, thus it requires

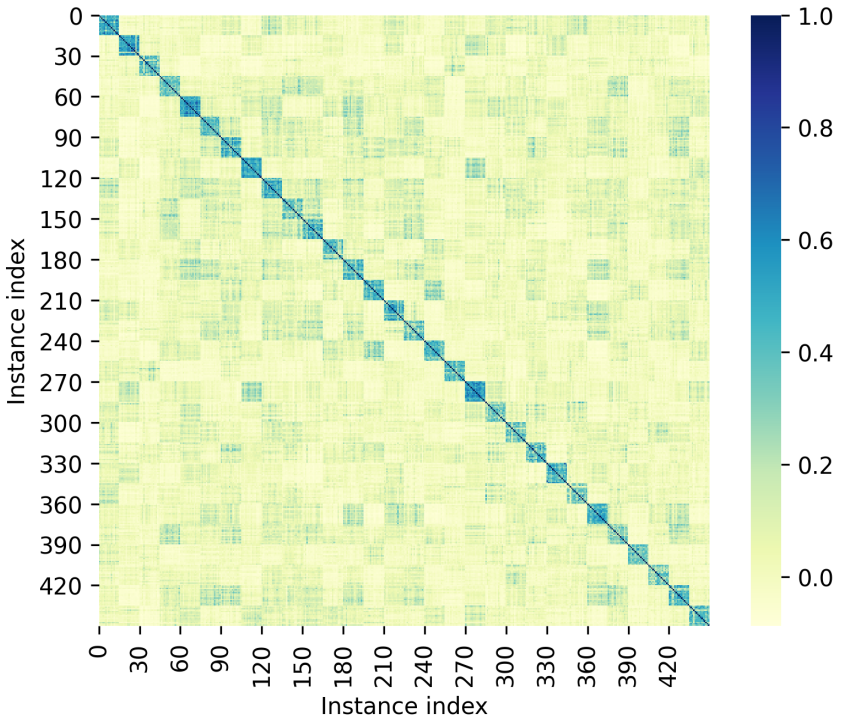


FIGURE 5.17: Correlation matrix between the SBRCL binary representations of random 30 Omniglot test classes with 15 instances each.

a deeper network to learn good features compared to the Omniglot. Training the entire network together with the binary part is a time and resource-demanding task. Second, we wanted to demonstrate that even fixed features can be used to train a high-performance CL model. From the neuroscience perspective, the fixed feature extractor could be contextualized with the primary cortex (V1) of an adult individual, which has been observed to exhibit very low plasticity.

The CNN front-end for feature extraction is a conventional ResNet152 (He et al., 2016) model with a modified last layer to output 50 classes. This model is trained from scratch on the first 50 CIFAR-100 classes. We refer to this model as CF50. For the CL training, the ResNet classification layer is discarded, and its input, a 1028 dimensional vector, is used as input to the binary representation learning network (BRLN). Unlike the Omniglot model, the BRLN has an additional information bottleneck before the binarization layer. To express more complex datasets than the Omniglot, such as the CIFAR-100, we need to increase the SBR dimensionality further. However, learning high-dimensional codes is problematic due to their tendency to overfit. One way to alleviate this problem is to reduce the amount of information flow through the network by reducing the dimensionality of one or several hidden layers. This approach is readily applied in ANN for classification; however, we hypothesize a distinction between the information capacity required for a typical one-hot ANN classifier and learning the SBR binarization layer.

Learning the binary representation can be seen as learning a multi-class classifier where each class is a low-level data feature. In this configuration, a class is not represented by a single continuous-valued vector, a typical input to a common classification layer, but rather by a set of several continuous-valued vectors, each activating one bit in the binary SBR. In the case of the typical one-hot classification, the latent representation must be able to express all intra-class variations (for good generalization) as well as to discriminate between classes. The generalization property (the space of intra-class variations) of the SBR is given by combining the active SBR bits. Each bit represents a low-level data feature with low intra-class variations, e.g., a specific feature is present in the input or not. Therefore, for the typical one-hot classification task, the latent representation needs to generalize as well as to discriminate, while for the SBR, it needs only to discriminate.

From another perspective, during a typical one-hot classification, a latent representation is matched (by dot product) with a template vector (of the same dimensionality) of each class stored in the classifier's matrix. This similarity needs to capture the intra-class variations. In the case of the SBR binarization (akin to the multi-class classification of low-level features), the latent representation needs to only discriminate between features encoded by SBR bits. Consequently, we hypothesise that the information carried by the template vector of each SBR bit in the matrix of the binarization layer needs to carry less information (can have lower dimensionality) than the template vector of the typical one-hot classifier. However, it is important to note that the latent representation, regardless of its format, entering both the one-hot classifier and the binarization layer must carry the same amount of information for its upper bound performance considering an ideal classifier model.

The inspiration behind this architecture originates from the configuration observed in the olfactory circuits of Drosophila Fly (Turner et al., 2008), which performs dimensionality reduction followed by expansion to sparse representations. Empirically we found this architecture to considerably reduce overfitting and boost the CL performance while reducing the number of learned parameters.

The bottleneck is implemented as a neural network with two fully connected layers with ReLU activations. The first layer reduces the 1028 dimensional CNN features to 256 dimensions, and the second layer then expands this compressed representation to 8192 dimensions, followed by binarization. This architecture is illustrated in Figure 5.18.

The ResNet152 was trained on the first 50 CIFAR-100 classes with images augmented by random horizontal flip followed by 4 pixels padding on each side and random crop to 32x32. The performance of the pre-trained CNN is reported in Table 5.2.

We call the modified ResNet152 as CF100 or CF50 according to how many classes it was trained on. The CF100 was trained only to establish the upper bound of the highest possible accuracy over all 100 classes trained with the mini-batch gradient descent.

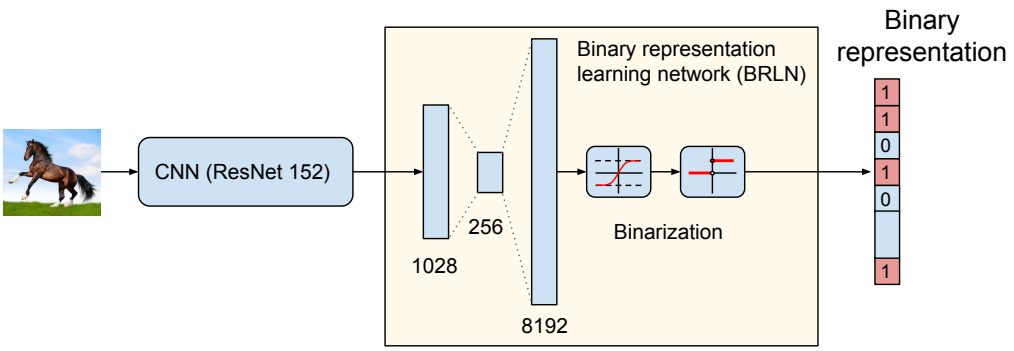


FIGURE 5.18: SBRCL CIFAR-100 model.

TABLE 5.2: Performance of the pre-trained CIFAR-100 ResNet152 models. TC stands for trained classifier and TBC for trained binary classifier.

Model	Model Architecture	Training Dataset	Test Accuracy
CF100	ResNet152, last layer set to output 100 logits.	All 100 classes	72%
CF50	ResNet152, last layer set to output 50 logits.	First 50 classes	75.70%
CF50+TC	Last CF50 conv layer followed by a single linear layer 1028->50. CF50 trained on first 50 classes. Only last new layer trained.	Second 50 classes	64.26%
CF50+TBC	Last CF50 conv layer followed by layers 1028->4096->tanh->bin->50 CF50 trained on first 50 classes. Only added two layers trained.	Second 50 classes	69.55%

The BRLN part of the SBRCL model was again trained on the first 50 classes in batches of 512 images while keeping the CF50 network fixed. The first 10 epochs were trained with $w_s = 1$ and then with $w_s = 10$ until convergence, approximately for the next 150 epochs. All reported results were averaged over 5 test splits. For the continual learning evaluation, we use the remaining 50 classes. 300 instances of these classes were used for the test-training and the remaining 200 for test-testing.

The task-free, class-incremental continual learning that we consider in our work is the most challenging scenario. Consequently, there is a substantial lack of published work, notably on the more challenging benchmarks such as the CIFAR-100. To our knowledge, there is no work on the CIFAR-100 targeting the task-free, class-incremental CL scenario with a comparable method that we could use for a fair evaluation. Therefore, we compare against considerably more complex methods utilizing extra memory to store past samples for replay and perform various regularization techniques. The OML and ANML studies do not include tests on large datasets with natural images, nor are reported in other literature. We compare our methods with the following recent methods: BiC (Wu et al., 2019b), UCIR (Hou et al.,

2019), PODNet (Douillard et al., 2020), and iCaRL (Rebuffi et al., 2017).

TABLE 5.3: Test-test accuracy on the CIFAR-100 dataset. Each collumn shows average accuracy over all learnt classes up to that point. Last column shows mean incremental accuracy as proposed by Rebuffi et al. (2017). All results are averaged over five models trained on five random test dataset splits. Our models are higlighted in bold.

	Sequence lengths. Mean accuracy over all classes learnt in each sequence are bellow in %							
Method	2	5	10	20	30	40	50	Inc. Acc.
PODNet (NME)	76.00	71.00	67.00	61.00	55.00	52.00	48.00	61.43
UCIR (CNN)	74.00	56.00	53.00	46.00	42.00	39.00	34.00	49.14
iCaRL	72.00	47.00	47.00	42.00	37.00	33.00	29.00	43.86
BiC	74.00	53.00	48.00	43.00	41.00	36.00	35.00	47.14
OML-S	87.80	64.40	59.82	47.27	40.64	33.58	30.63	52.02
OML-WB	89.30	71.72	63.00	45.38	34.79	31.11	27.16	51.78
MLSBRCL	88.90	78.00	**70.02**	56.07	47.70	41.81	39.48	60.28
SBRCL	**95.10**	**81.32**	67.78	**67.58**	**57.86**	**54.30**	**51.05**	**67.86**

In addition to the SBRCL model, we also train MLSBRCL in the meta-learning setting with the union and overlap inference. Furthermore, we compare against an OML-S model, which is our implementation of the OML with 2038 dimensional representations but with the same CNN front-end as the SBRCL - the CF50 trained on the first 50 CIFAR-100 classes. We attempted to train the original OML implementations on CIFAR-100 but failed to achieve meaningful results. The OML-S provides more objective results for comparison against other methods. Other our models OML-W and OML-WB follow the same design as the Omniglot models. That is, the OML-W is the OML-S but with a wider representation with 8192 dimensions. The OML-WB then extends the OML-W by the binarization layer.

In Table 5.3 and Figure 5.19, we compare our results with the recently published work. We can see that our method SBRCL performs statistically on a par with the PODNet despite the PODNet being a complex method (as defined in Section 1.1), utilizing a replay memory buffer with distillation across ResNet layers and bias balancing.

Surprisingly, the OML-S performs better than the OML-WB and OML-W. We attribute this difference to the regularization effect of the smaller representation size. Along the same line, we explain the performance gap between OML-WB and OML-W. It is conceivable that the higher performance of the OML-WB over the OML-W could be explained by the information bottleneck introduced by the binarization function.

In Figure 5.20, we show the correlation of the binary representations of 10 random classes with 40 grouped instances each. The within-class correlations are noticeably lower compared to the Omniglot (Figure 5.17) experiments, nevertheless clearly observable. This lower correlation can be explained by the higher diversity of features constituting the CIFAR-100 classes. While the Omniglot images are composed

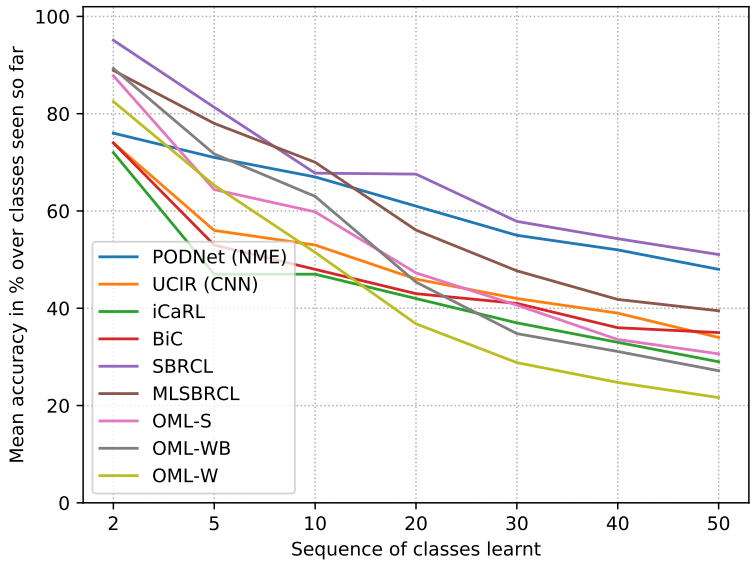


FIGURE 5.19: Continual learning performance on the CIFAR-100 test-test dataset.

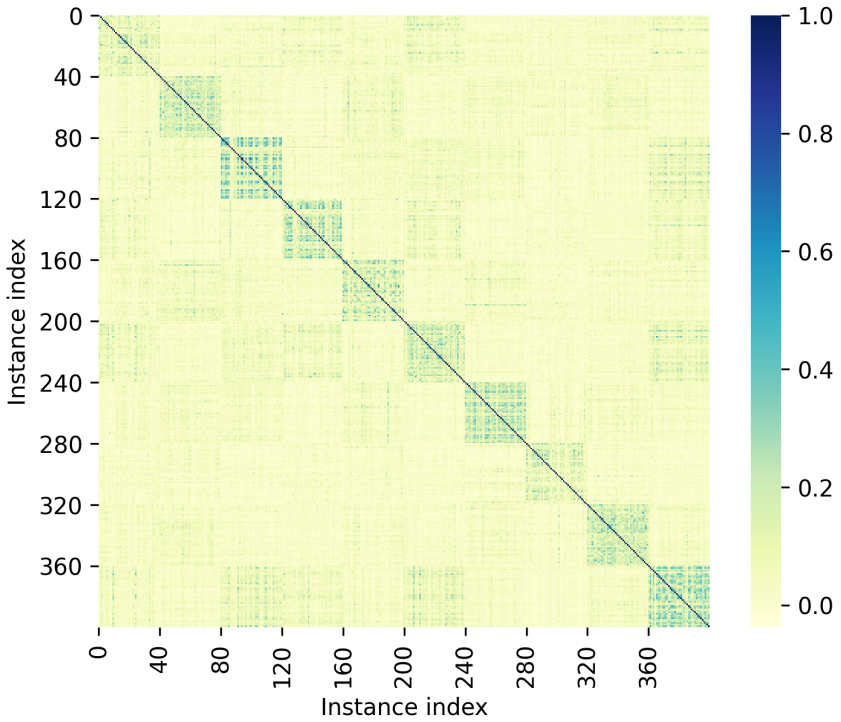


FIGURE 5.20: Correlation matrix between the SBRCL binary representations of random 10 CIFAR-100 test classes with 40 instances each.

of a limited number of binary line and curve feature primitives, the natural CIFAR-100 images require a much richer feature set, likely also having a higher intra-class variance.

5.7 Out-of-Distribution Continual Learning

The ability to continually learn and then recognize new tasks with distribution far from the distribution of the training dataset is a fundamental requirement for machine learning models deployed in real-world applications.

The out-of-distribution (ood) experiment is conducted with the SBRCL model trained on Omniglot, which we then use to continually learn all 10 MNIST (LeCun et al., 2010) classes. The MNIST digits or similar symbols do not appear in the Omniglot, thus the MNIST is a suitable dataset for the ood test.

The ood performance of the SBRCL model was compared against state-of-the-art methods evaluated in the class-incremental learning scenario: MERLIN (KJ and Nallure Balasubramanian, 2020), iCaRL (Rebuffi et al., 2017), GSS (Aljundi et al., 2019), and FRCL and BASELINE from Titsias et al. (2019). Results for the GSS, iCaRL and MERLIN methods originate from the KJ and Nallure Balasubramanian (2020) publication while FRCL and BASELINE from Titsias et al. (2019). Experiments in these publications were carried out in simpler CL settings (noted in Table 5.4) on the Split MNIST dataset and not in the ood test.

The results are reported in Table 5.4, along with notes highlighting the differences in the evaluation protocol. While not reaching the state-of-the-art results, our method still shows comparable CL performance despite its simple architecture and the evaluation in the ood setting.

In Figure 5.21, we present a correlation between 40 sorted instances of all 10 MNIST classes. As in the Omniglot and CIFAR-100 cases, we can see a clear within-classes correlation and between-classes separation.

TABLE 5.4: Out-of-distribution continual learning of 10 MNIST classes with SBRCL model trained on Omniglot. Average accuracy over all classes is reported. Other methods are trained and evaluated on Split MNIST with 5 two-digits tasks.

Method	Accuracy (%)	Note
GSS	88.30	Replay buffer 200 samples/task
iCaRL	89.90	Replay buffer 200 samples/task
MERLIN	90.70	Generative replay in parameter space, inference over ensemble of 30 models per task
BASELINE	95.80	Replay buffer 40 samples/task
FRCL (TRACE)	**97.80**	Approx. Bayesian inference with Gaussian Processes. Replay buffer 40 samples/task.
SBRCL	95.78	Sample is a latent and dist. (μ, σ) per task.

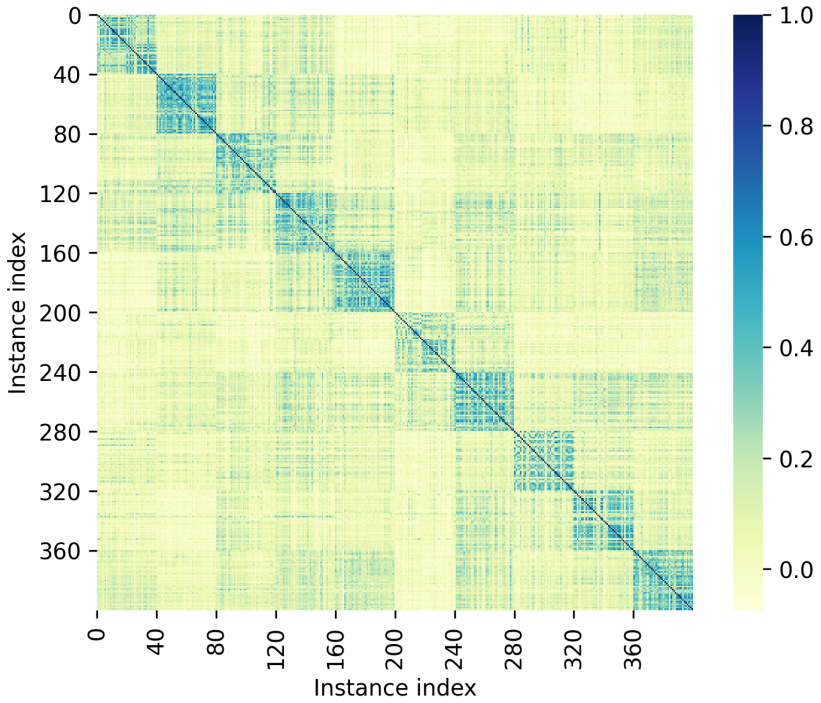


FIGURE 5.21: Correlation matrix between the SBRCL binary representations of random 10 MNIST test classes with 40 instances each.

5.8 Clustering Performance

To gain some insights into the high performance of SBRCL method, we analyzed the clustering performance of the binary features. We calculate the within and between-cluster Hamming distances and the overlap similarities. The results are presented in Table 5.5. We do not use the union sets as the cluster centers here but instead randomly sampled several thousand pairs of within and between-classes representations, calculated their distances, and averaged them. We can observe that the ratio

TABLE 5.5: Clustering performance of the CIFAR-100 and Omniglot binary representations calculated over all instances of 50 CIFAR-100 test classes, and all instances of the 600 Omniglot classes. The cluster distances are Hamming distances. The overlap refers to the similarity by binary AND operation (number of matching active dimensions). The arrows show the direction of the desired performance.

	CIFAR-100		Omniglot	
	SBRCL	MLSBRCL	SBRCL	MLSBRCL
Latent dimensions	8192	8192	4069	4069
Accuracy ↑	51.05%	39.48%	85%	87.15%
Sparsity	2.70%	2.60%	6.50%	6.30%
Between clusters distance ↑	422.54	414.86	482.24	478.69
Within clusters distance ↓	387.13	345.63	297.16	327.73
Between clusters overlap ↓	6.68	12.56	12.74	9.59
Within clusters overlap ↑	15.62	28.55	55.11	44.21

of the within-classes and between-classes overlap similarities is significantly higher compared to the ratio of Hamming distances. On the Omniglot, the within-classes overlap similarity is more than four times higher than the between-classes overlap

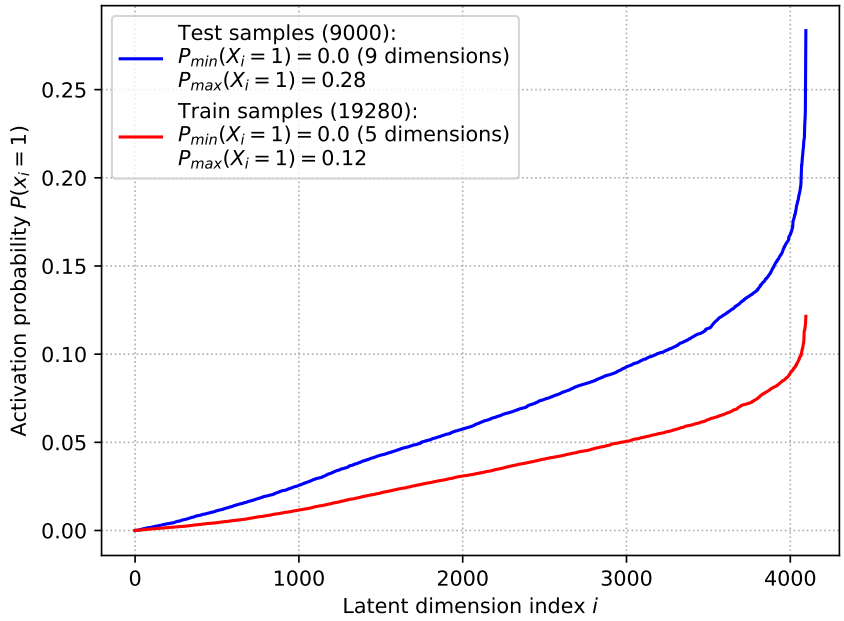


FIGURE 5.22: Sorted activation probabilities of the latent dimensions calculated on the training and test Omniglot data.

on both SBRCL and MSBRLC methods. This ratio directly correlates with the observed method performance (accuracy). The CIFAR models follow the same trend. The Hamming distances also do correlate with the observed classification accuracy, although the between/within distance ratios are smaller.

In Figure 5.22 we plot the probability of each bit in the binary representation being active, calculated over all instances of test and train classes. We can notice that

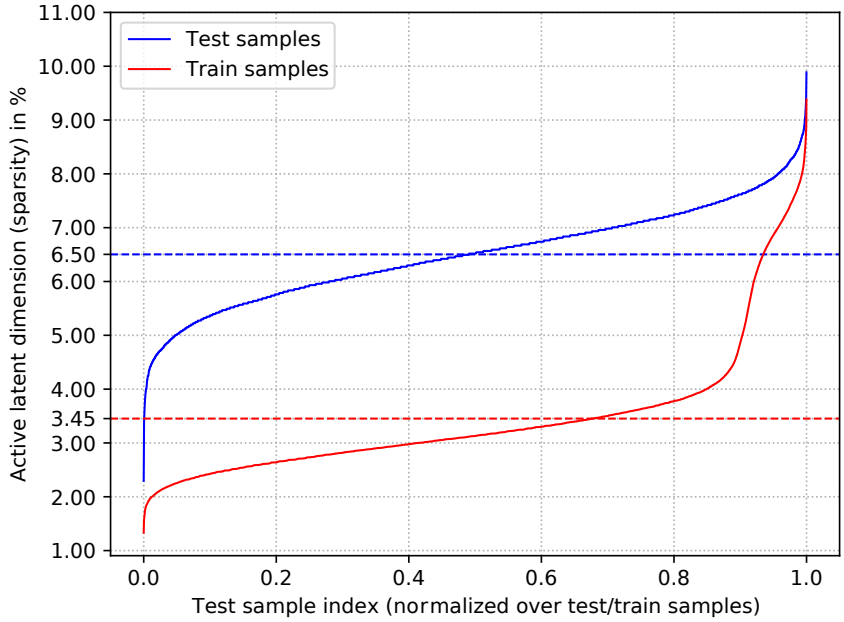


FIGURE 5.23: Activation sparsity on the training and test Omniglot samples. Training and test representations are sorted by the sparsity and their normalized indices shown along the x axis.

the test data shows lower sparsity (higher activation probabilities) than the training one. This is even more apparent in Figure 5.23 that shows the sparsity per training and test samples. The difference between the train and test sparsity is caused by the model overfitting on the training data. Without regularization, the sparsity decreases over the training period until the activations in the binary representation start to correlate significantly with the class categories. These representations become degenerate since now the bits do not encode semantic attributes of the image but rather its entire category. Such defective representations lose the ability to generalize. A decline in the generalization due to the high sparsity was already reported by Sharkey and Sharkey (1995) and French (1994).

In Figures 5.24 and 5.25, we visualize activations of the binary representations along with few cases of the union set and overlap similarity. For the visualization purpose, the 4096 bits vectors are padded with zeros are reshaped to images with 57x73 resolution. Figure 5.24 shows binary activations of five instances of three random Omniglot test classes and their union sets. Each column in Figure 5.24 shows

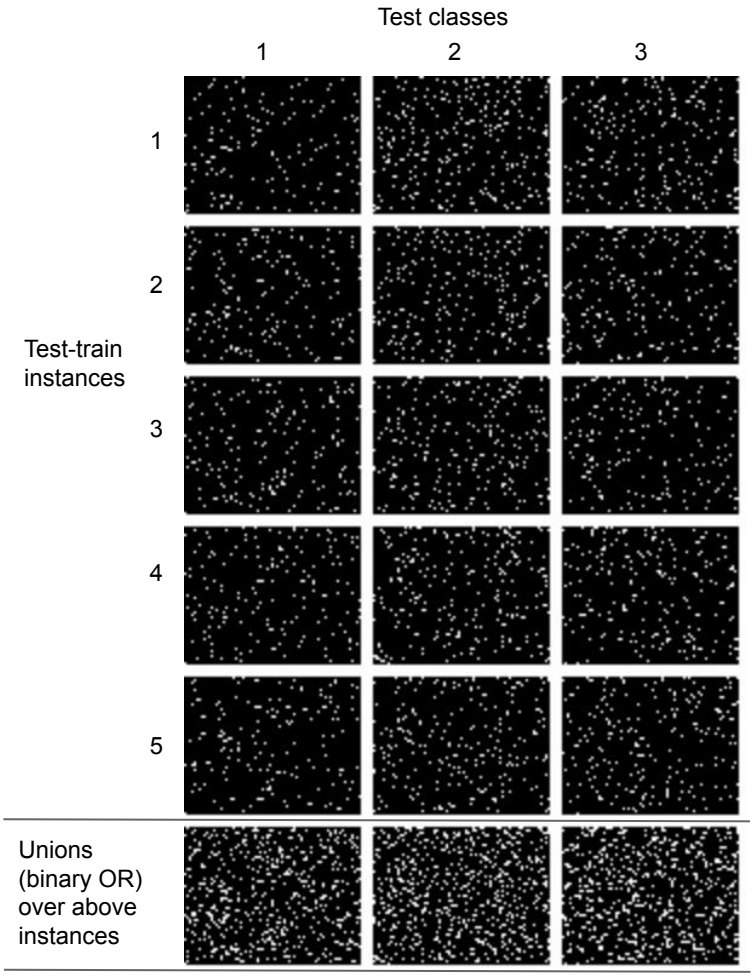


FIGURE 5.24: Examples of union sets of binary representations of five class instances. These unions form the pattern attractors in the memory **M**. The representations were produced by the SBRCL model on Omniglot test-train data. For visualization the 4096 bits vectors are zero-padded and reshaped to 57x73 images.

SBRCL binary representations of five test instances of the same Omniglot class. The bottom image in each column shows the union set (binary OR operation) of the above representations. The number of active bits in the union set is higher than in each representation. This activity increases with added instances but quickly stabilizes with the density shown in Figure 5.24. Adding representations of different classes results in the rapid exhaustion of all bits in the union set. This highlights that each class category is a composition of specific features expressed by a subset of bits in the binary representation. Then all bits in the binary representation make up all possible compositional features of the training dataset distribution.

Figure 5.25 illustrates two cases of the overlap similarities: between the union set (center image) and a binary representation of the same class (class in the top left), and a union set (center image) and binary representations of different classes (class 2 and 3). Note that the representation of class 1 originates from a different image instance that is not part of the union set. Images in the bottom row show results of the overlap (binary AND operation). The number below or next to each image shows a sum of active bits in each representation. We can observe that the overlap

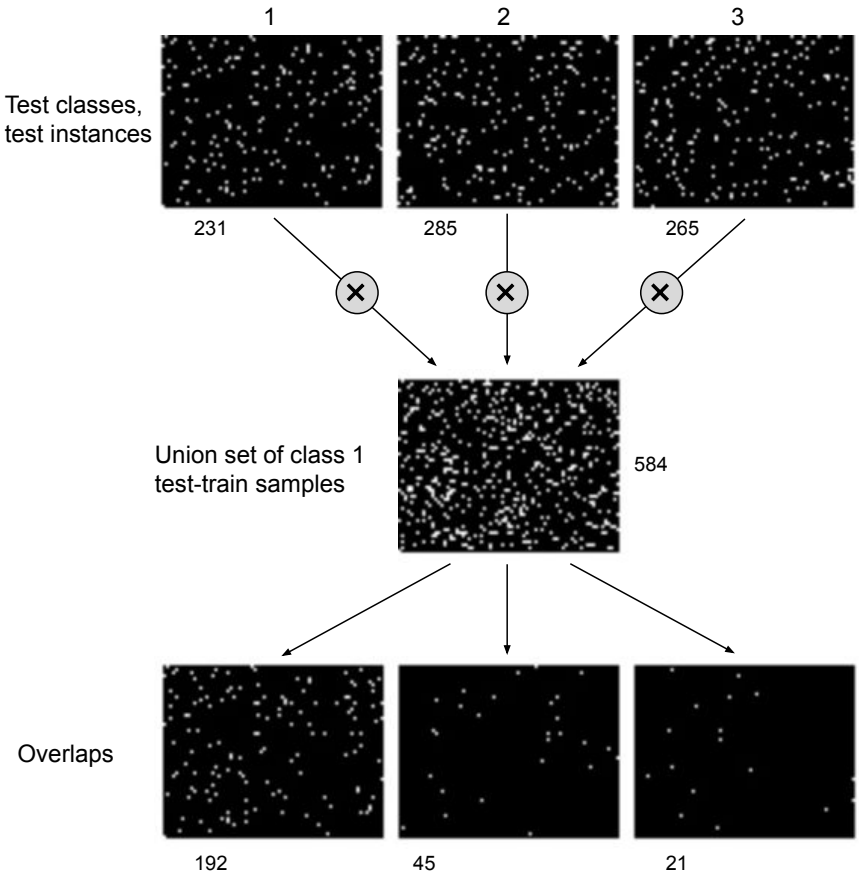


FIGURE 5.25: Visualization of the overlap (binary AND or dot product) of the sparse binary representations with a union set of class 1 (in the center). The union set is produced from test-train instances of class 1. The representation of class 1 in the top left comes from a disjoint group of test-test instances not used to assemble the class 1 union set. Numbers next to images indicate number of active bits.

between a union set and a representation of the same class exhibits a higher overlap than those of different classes.

5.9 Compression

A high-dimensional binary representation can be efficiently encoded by an integer vector of indices of active bits. Similarly, sparse continuous-valued representation can be encoded by an integer vector of indices of active bits and their values. However, for binary representations with lower sparsity (higher number of active bits), it is more efficient to map the binary sequence directly to fixed or floating-point variables, such as common int32 or float32 types.

For the direct bit-mapping of n-dimensional SBR, we need $\frac{n}{8}$ fractional bytes. To encode the same SBR with sparsity s with bit-indexing, we need $\frac{2ns}{100}$ bytes, assuming two bytes per index (since the bit-width of the index variable must be $> \log_2(n)$, which is 12 bits for $n = 4096$). These encoding schemas produce identical compression for sparsity s_e given by:

$$\frac{n}{8} = \frac{2ns_e}{100} \tag{5.20}$$

$$s_e = \frac{100}{16} = 6.25\% \tag{5.21}$$

As evident from Eq. 5.20, the higher the sparsity (fewer active bits), the more efficient it is to use the bit-indexing compression. For the SBRCL 4096 bit binary vectors, it becomes more efficient to directly map the bits for sparsity above 6.2%. For example, for sparsity 6.2%, the bit indexing compression results in 508 bytes long vector while the direct bit mapping produces $4096/8 = 512$ bytes vector. This method is only applicable to binary representations, not the sparse continuous-valued representation.

Table 5.6 shows the size and sparsity of the OML,ANML, and SBRCL representations along with the indices-based compression. It is apparent that even high, 4096 dimensional sparse representations occupy less memory than the sparse continuous-valued representations of lower dimensionality.

5.10 Conclusion

This work studied selected neurological processes in the primate's brain related to our ability to continually learn and their application to computational methods.

With these insights, we proposed a novel machine learning method SBRCL for continual learning of image classes with low catastrophic interference. SBRCL is a conceptually very simple method that requires very low computational resources compared to the latest state-of-the-art methods. SBRCL is a non-iterative method

TABLE 5.6: Comparison of representation dimensionality and sparsity produced by methods evaluated on Omniglot and their possible compression. The OML and ANML sparsity was taken from the corresponding publications Javed and White (2019) and Beaulieu et al. (2020).

Method	Sparsity %	Representation Size	Compressed Size (bytes)	Compression Method
OML	3.8	2304 (float32)	701	Indices of active dimensions and their values. 2 bytes per index plus 4 bytes for the value.
ANML	5.9	2304 (float32)	1088	
SBRCL	6.5	4096 (binary)	533	Indices of active bits. 2 bytes per index.
SBRCL	6.5	4096 (binary)	512	Direct mapping to bytes.

that requires only a single forward pass through the feature extraction CNN followed by binary *OR* operation for learning or binary *AND* and max operations for inference. The recent SOTA methods, on the other hand, require either iterative CL learning and inference (Javed and White, 2019, Beaulieu et al., 2020), maintaining a replay buffer in combination with the model fine-tuning for CL training (Rebuffi et al., 2017, Douillard et al., 2020, Riemer et al., 2018), generative models with pseudo rehearsal (Shin et al., 2017, Ostapenko et al., 2019, van de Ven et al., 2020) or internally partitioned ANN (Rusu et al., 2016, Fernando et al., 2017).

We evaluated our method on Omniglot and CIFAR-100. It significantly outperforms the current state-of-the-art methods on Omniglot and stays on a par with other much more complex and resource-demanding methods on CIFAR-100. We also tested SBRCL on the out-of-distribution CL task with comparatively high performance.

Our meta-learning adaptation of the SBRCL method (MLSBRCL) showed marginally better performance than the SBRCL but for the cost of a more complex and resource-intensive training regime of the meta-learning. The meta-learning requires more training iterations due to the extra inner loop updates and backpropagation through the unrolled iterations of the inner loop. The continual learning and inference algorithms of the MLSBRCL are identical to the SBRCL method, thus equally fast.

While our method shows significant progress on the CL benchmark for continual learning of a large number of classes, we do not claim it to be a general solution to the CL problem. Our method's main limitations are that it does not allow forward and positive backward transfers across tasks due to the separation of the tasks during the CL learning, and it requires supervision to train the CNN feature extractor.

In our future work, we propose training the CNN front-end with a self-supervised method that addresses the latter drawback; however, enabling the forward and backward transfer is more challenging. Forward transfer happens when performance on newly learned tasks is enhanced by tasks learned in the past. Positive backward transfer indicates a performance boost of old tasks upon learning new ones. Both transfers require representation sharing during learning. This is believed

to occur in the CLS (O'Reilly et al., 2014b) of our brain during autoassociation of new experiences with past ones in the hippocampus, then a projection of short-term memories to the long-term memory in the neocortex. Along a similar direction, in our future work, we propose a replay-based method that may enable the forward and backward transfers with the sparse binary representations.

Our model can be very loosely contextualized to the memorization of the visual stimuli up to the CA3 region in the hippocampus, potentially also with the CA1 for pattern completion. The CNN front-end could be positioned as the primary visual cortex, the SBRN as the entorhinal region, along with the Dentate Gyrus as the pattern separator. The binary representations are then memories residing in the CA3 region. The CA1 region could be seen as associating the memories to the class categories represented by pattern attractors in the memory $\mathbf{M}$ (Figure 5.1). Output of the CA1 would provide reduced representations (class categories) as also observed in the hippocampus.

This work resulted in the following contributions:

- Proposed novel SBRCL method for continual learning with sparse binary representations with state-of-the-art results, learning 600 classes in the challenging task-free class-incremental CL scenario. This method learns directly binary representations and uses fast union set and overlap similarity operations to continually learn and infer new class categories.
- Proposed novel MLSBRCL method for continual learning with meta-learned binary representations combined with the union set and overlap similarity, showing equally high performance.
- Demonstrated the SBRCL capability to perform well in the out-of-distribution setting.

The SBRCL PyTorch implementation, including trained models, is publicly available on `https://github.com/ok1zjf/SBRCL`.

5.10.1 Future Work

This work barely opened the door to the possibilities of high dimensional, binary, sparse representations on the path to the continual learning solution. In our upcoming research, we will conduct more experiments on large datasets with more classes starting with the ImageNet with 1000 classes learned in the true task-free class-incremental regime.

Equally, we will conduct more extensive, objective out-of-distribution (ood) evaluation by training all competitive methods in the identical CL scenarios and dataset configurations. The biggest challenge in this evaluation, that we have already encountered, is training other models to perform continual learning even on the same data distribution in the target task-free, class-incremental learning scenario on the Omniglot dataset.

The SBRCL and most CL methods rely on splitting the target dataset into a disjoint set, with one being utilized for the CL model pre-training which is sometimes called bootstrapping. Alternatively, this can be accomplished by leveraging other datasets, where particular care is taken not to include the same or similar classes in the dataset for the CL model pre-training. We plan to address this issue by taking advantage of recent developments in self-supervised training, specifically by contrastive or momentum-based methods such as BYOL, SimSiam, SwAV or Barlow Twins (Grill et al., 2020, Chen and He, 2020, Caron et al., 2020, Zbontar et al., 2021). Within the currently conducted follow up work, we are implementing the Barlow Twins (Zbontar et al., 2021) method modified to directly learn the SBR with the self-supervision on generic large image datasets ImageNet (Russakovsky et al., 2015) and Tencent ML-Images (Wu et al., 2019a) with 17M images.

In other future work, we are considering to extend our model for Sparse Distributed Memory (SDM) (Kanerva, 1988), working as an autoassociative memory. Before forming new or updating existing pattern attractors, the SBR of the input instance is first autoassociated with all known patterns in the SDM and then subjected to the union or overlap operations. Alternatively, the SDM could be replaced with an energy-based ANN model of an autoassociative memory, similar to Muezzinoglu et al. (2005). Since the SDM and the energy-based neural network model store overlapping representations, it is likely to enable the forward and positive backward transfers. Evaluation of these transfers is a part of this upcoming research.

A large part of our future work focuses on interpreting the learned SBRs and understanding how the sparsity level and SBR dimensionality impact the model performance. For example we would like to understand what type of features in the pixel space is encoded by each SBR bit. We are considering visualizing the function of the SBR bits by setting the desired SBR bits and backpropagating them to the pixel space. Alternatively, we can generate images with patterns of elementary shapes (e.g., shapes common in the Omniglot images) and map their appearance with the SBR bits activations.

Chapter 6

Conclusions

In this thesis, we conducted research in four machine learning areas: image memorability, episodic segmentation, representation learning, and continual learning.

Our primary goal was to propose novel, low complexity (as defined in Section 1.1), and computationally efficient methods competitive with the latest research work. To ascertain the performance of our methods, we evaluated our work on established benchmarks in each area.

In the search for inspiration, we focused on our brain's functions, the only known example of an intelligent system. Our goal was not to replicate the neurological processes underlying the functions of interest or mimic the exact brain behavior. To do so would arguably require fundamental changes in the computation architecture, such as a transition to the neuromorphic processors or to simulate the spiking neural network, which is outside the scope of our work. Instead, we explored the brain functions to gain insights applicable in the context of artificial neural networks. In the following sections, we summarize our accomplishments in each domain and present our contributions.

In Chapter 2, we analyzed what makes us remember some images more than others. Based on prior research work, indicating that image memorability is an intrinsic property of an image, we proposed a learned, spatial attention module that is capable of learning and predicting image regions correlated with the retention level of a given image in our memory. We evaluated our method on the latest image memorability datasets and compared it against the current state-of-the-art methods that our algorithm outperformed by 5.8%, closely approaching the human performance with 99.6% consistency. We also successfully evaluated our method on the image aesthetics estimation task. We showed that our model could also learn and predict other perceptual image attributes with high performance.

In work on episodic segmentation in Chapter 3, we explored how our brain forms episodic memories and their characteristics in an attempt to apply our findings in the design of a less complex (as defined in Section 1.1) and more accurate

method for video summarization, compared to the latest research work. Our research converged towards an elementary yet powerful method based on the observation that boundaries of the memory episodes are likely determined over a longer sequence of events or retroactively during recall. Therefore, we proposed a novel method with a soft, self-attention function to learn relations among frames in the input sequence as a function of the frame importance scores as estimated by manual annotation. We compared our method with the latest research in this domain, confirming considerable performance gain on standard benchmarks for video summarization.

Learning discrete latent representations is the core topic of Chapter 4. In this work, we set to develop a method to learn binary latent representations in an unsupervised way with gradient-based techniques and demonstrate an ability to navigate this latent space and modify its attributes. The biological inspiration behind this method originates from the nature of the neural coding in our brain. The neural coding has been reported to exhibit a discrete, binary form and a high degree of sparsity across all brain regions. Over several experiments, we developed a straightforward method to efficiently learn binary representations with the backpropagation method. The binarization in our method is accomplished by setting up a tanh() function followed by thresholding around zero. This non-differentiable function is then treated as a unit function during backpropagation. That is, a gradient flowing back to the binarization function is passed through to its input unchanged. Furthermore, we developed new algorithms for generating novel images, interpolating between given images, and changing image attributes, all in the binary latent space. Here, we proposed a method where the binary state of each latent dimension is relaxed to a unit length, continuous-valued vector with the same dimensionality as the latent space. These vectors, each representing one bit, are positioned in an n-sphere according to their mutual correlations. Operations in the binary latent space are then conducted on the surface of this n-sphere. We evaluated our method on the MNIST, CIFAR-10 and CelebA benchmarks and compared against generative, representation learning methods with our method exhibiting superior performance.

In the final Chapter 5, we present our method on continual learning with sparse binary representations. In this work, we explored brain processes mitigating catastrophic forgetting, particularly the complementary learning systems and the pattern separation in the hippocampus. Furthermore, we studied the plasticity of the primary visual cortex (V1) in the context of catastrophic forgetting due to learning new visual features. The research literature on this topic revealed that the primary visual cortex is highly plastic over the critical developmental period when it likely learns most of the visual features. This period is then followed by pruning, when the visual cortex stabilizes to remain primarily as a fixed feature extractor for the rest of our lives. In light of this observation, we focused on the later layers of our neural network, as the location experiencing the highest interference between already

learnt and new data points. In experiments with natural images from the CIFAR-100 dataset, we set up a pre-trained CNN network as the fixed feature extractor, similar to the function of the V1 cortex. More importantly, our research on the representation sparsification in the Dentate Gyrus region in the hippocampus brought the most compelling inspirations. Particularly, it highlighted the low interference between random sparse vectors and, consequently, a high overlap of similar vectors.

The main contributions of our work are: (1) learning sparse binary representations by novel binarization method with sparsity enforced by ℓ_0/ℓ_2 regularization, (2) methods to continually learn new and infer old classes with the sparse binary representations. To continually learn new class categories, our method calculates a union set of all instances within the same class category by performing a binary *or* operation along each latent dimension. The union is then added to a dynamically expanded memory as a pattern attractor for the specific class category. A class of an unknown test sample is then established by comparing how many bits of the test sparse representation overlap with each pattern attractor in the memory and selecting the category with the highest overlap.

Our method significantly outperforms the latest state-of-the-art methods on continually learnt long class sequences in the most challenging task-free, class incremental learning scenario. Moreover, despite the high dimensionality of the latent space, our sparse binary representations occupy less memory than the low dimensional, continuous-valued latents of the prior art methods.

While architecturally quite different, the SBRCL model presented in this work evolved from a meta-learning method. Over a number of experiments, we established that learning the high dimensional sparse binary representations outperformed the meta-learning method in the term of the computational complexity (for each CL training update the meta-learning requires several gradient descent updates, while SBRCL only a binary *OR* operation over training SBRs) as well as the performance to adapt to novel tasks. These experimental results indicate that our method could be likely applied in other domains as an alternative to the meta-learning methods. The meta-learning method learns a parameter space bordering with all targeted tasks. The parameter space is then fine-tuned to a specific task during the meta-test-training stage. Rather than learning a parameter space that could be updated towards individual tasks, our method SBRCL appears to reserve regions in the latent space for the targeted tasks. As this method helps avoid catastrophic forgetting in the continual learning scenario, it is conceivable to believe that such a method would also avoid interference among the meta-learnt tasks in other domains.

Bibliography

Aditya Khosla, Akhil S. Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 2390–2398, 2015.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015.

Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Maria Toledo-Rodriguez, Barak Blumenfeld, Caizhi Wu, Junyi Luo, Bernard Attali, Philip Goodman, and Henry Markram. Correlation maps allow neuronal electrical properties to be predicted from single-cell gene expression profiles in rat neocortex. *Cerebral cortex*, 14(12):1310–1327, 2004.

J Kruger and F Aiple. Multimicroelectrode investigation of monkey striate cortex: spike train correlations in the infragranular layers. *Journal of neurophysiology*, 60 (2):798–828, 1988.

Elad Schneidman, Michael J Berry, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007–1012, 2006.

Randall C O'Reilly and James L McClelland. Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off. *Hippocampus*, 4(6):661–682, 1994.

Michael A Yassa and Craig EL Stark. Pattern separation in the hippocampus. *Trends in neurosciences*, 34(10):515–525, 2011.

Randall C O'Reilly and Jerry W Rudy. Conjunctive representations in learning and memory: principles of cortical and hippocampal function. *Psychological review*, 108 (2):311, 2001.

Khurram Javed and Martha White. Meta-learning representations for continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O. Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 992–1001. IOS Press, 2020.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020.

Zijun Zhang, Ruixiang Zhang, Zongpeng Li, Yoshua Bengio, and Liam Paull. Perceptual generative autoencoders. In *International Conference on Learning Representations, Workshop DeepGenStruct*, 2019.

Bin Dai and David Wipf. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*, 2019.

Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H Lampert. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*, 2017.

Graeme S Halford, Nelson Cowan, and Glenda Andrews. Separating cognitive capacity from knowledge: A new hypothesis. *Trends in cognitive sciences*, 11(6):236–242, 2007.

D.L. Schacter, D. T. Gilbert, and D. M. Wegner. *Psychology (2nd Edition)*. Worth, New York, 2011.

Jost B Jonas, Andreas M Schmidt, JA Müller-Bergh, UM Schlötzer-Schrehardt, and GO Naumann. Human optic nerve fiber count and optic disc size. *Investigative ophthalmology & visual science*, 33(6):2012–2018, 1992.

Genevieve Leuba and Rudolf Kraftsik. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anatomy and embryology*, 190(4):351–366, 1994.

Ronald J MacGregor. Theoretical mechanics of biological neural networks. 1993.

Mary C Potter, Brad Wyble, Carl Erick Hagmann, and Emily S McCourt. Detecting meaning in rsvp at 13 ms per picture. *Attention, Perception, & Psychophysics*, 76(2): 270–279, 2014.

Rodrigo Quian Quiroga. Neuronal codes for visual perception and memory. *Neuropsychologia*, 83:227–241, 2016.

Szabolcs Káli and Peter Dayan. Off-line replay maintains declarative memories in a model of hippocampal-neocortical interactions. *Nature neuroscience*, 7(3):286–294, 2004.

Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the intrinsic memorability of images. In *Advances in Neural Information Processing Systems*, pages 2429–2437, 2011a.

Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and AlexanderC Berg. SSD: Single shot MultiBox detector. In , *Computer Vision - ECCV 2016*, volume 9905 of *Lecture Notes in Computer Science*, chapter 2, pages 21–37. Springer International Publishing, Cham, December 2016.

Soodabeh Zarezadeh, Mehdi Rezaeian, and Mohammad T. Sadeghi. Image memorability prediction using deep features. In *Iranian Conference on Electrical Engineering (ICEE)*, pages 2176–2181. IEEE, 2017.

Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018.

Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020.

Koen V Haak and Christian F Beckmann. Understanding brain organisation in the face of functional heterogeneity and functional multiplicity. *NeuroImage*, 220: 117061, 2020.

Sujatha R Upadhyaya. Parallel approaches to machine learning—a comprehensive survey. *Journal of Parallel and Distributed Computing*, 73(3):284–292, 2013.

Udo Seiffert. Artificial neural networks on massively parallel computer hardware. *Neurocomputing*, 57:135–150, 2004.

Sebastian B Gaigg, John M Gardiner, and Dermot M Bowler. Free recall in autism spectrum disorder: The role of relational and item-specific encoding. *Neuropsychologia*, 46(4):983–992, 2008.

Conor F Underwood and Louise C Parr-Brownlie. Primary motor cortex in parkinson's disease: Functional changes and opportunities for neurostimulation. *Neurobiology of Disease*, 147:105159, 2021.

Mariese A Hely, Wayne GJ Reid, Michael A Adena, Glenda M Halliday, and John GL Morris. The sydney multicenter study of parkinson's disease: the inevitability of dementia at 20 years. *Movement disorders*, 23(6):837–844, 2008.

P. Isola, J. Xiao, D. Parikh, A. Torralba, and A. Oliva. What makes a photograph memorable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1469–1482, July 2014.

Wilma A. Bainbridge. The resiliency of image memorability: A predictor of memory separate from attention and priming. *Neuropsychologia*, 141:107408, 2020.

Seyed-Mahdi Khaligh-Razavi, Wilma A Bainbridge, Dimitrios Pantazis, and Aude Oliva. From what we perceive to what we remember: Characterizing representational dynamics of visual memorability. *BioRxiv*, page 049700, 2016.

Wilma A Bainbridge, Phillip Isola, and Aude Oliva. The intrinsic memorability of face photographs. *Journal of Experimental Psychology: General*, 142(4):1323, 2013.

Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 145–152. IEEE, 2011b.

Lionel Standing. Learning 10000 pictures. *The Quarterly journal of experimental psychology*, 25(2):207–222, 1973.

Timothy F. Brady, Talia Konkle, George A. Alvarez, and Aude Oliva. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*, 105(38):14325–14329, 2008.

Thomas K. Landauer. How much do people remember? some estimates of the quantity of learned information in long-term memory. *Cognitive Science*, 10(4):477–493, 1986.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114, 2001.

Rachit Dubey, Joshua Peterson, Aditya Khosla, Ming-Hsuan Yang, and Bernard Ghanem. What makes an object memorable? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1089–1097, 2015.

Kevin S LaBar. Beyond fear: Emotional memory mechanisms in the human brain. *Current directions in psychological science*, 16(4):173–177, 2007.

Matei Mancas and Olivier Le Meur. Memorability of natural scenes: The role of attention. In *International Conference on Image Processing*, pages 196–200. IEEE, 2013.

Aditya Khosla, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Memorability of image regions. In *Advances in Neural Information Processing Systems*, pages 296–304, 2012.

Talia Konkle, Timothy F Brady, George A Alvarez, and Aude Oliva. Conceptual distinctiveness supports detailed visual long-term memory for real-world objects. *Journal of Experimental Psychology: General*, 139(3):558, 2010.

Timothy F Brady, Talia Konkle, George A Alvarez, and Aude Oliva. Real-world objects are not represented as bound units: independent forgetting of different object details from visual memory. *Journal of Experimental Psychology: General*, 142 (3):791, 2013.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010.

Lore Goetschalckx, Pieter Moors, and Johan Wagemans. Image memorability across longer time intervals. *Memory*, 26(5):581–588, 2018.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3): 145–175, 2001.

David G. Lowe. Distinctive image features from Scale-Invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

Zoya Bylinskii, Phillip Isola, Constance Bainbridge, Antonio Torralba, and Aude Oliva. Intrinsic and extrinsic effects on image memorability. *Vision Research*, 116 (Part B):165–178, 2015. Computational Models of Visual Attention.

Yoann Baveye, Romain Cohendet, Matthieu Perreira Da Silva, and Patrick Le Callet. Deep learning for image memorability prediction: The emotional bias. In *Proceedings of the ACM International Conference on Multimedia*, pages 491–495, New York, NY, USA, 2016. ACM.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE CVPR*, pages 1–9, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and Others. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press, 1997.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 487–495, Cambridge, MA, USA, 2014. MIT Press.

Peiguang Jing, Yuting Su, Liqiang Nie, and Huimin Gu. Predicting image memorability through adaptive transfer learning from external sources. *IEEE Transactions on Multimedia*, 19(5):1050–1062, 2017.

Houwen Peng, Kai Li, Bing Li, Haibin Ling, Weihua Xiong, and Weiming Hu. Predicting image memorability by multi-view adaptive regression. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1147–1150. ACM, 2015.

Bora Celikkale, Aykut Erdem, and Erkut Erdem. Visual attention-driven spatial pooling for image memorability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 976–983, 2013.

Thorsten Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

George Stothart, Laura J Smith, Alexander Milton, and Elizabeth Coulthard. A passive and objective measure of recognition memory in Alzheimer's disease using Fastball memory assessment. *Brain*, 144(9):2812–2825, 09 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pages 806–813, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jurgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In , *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001a.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*, pages 2048–2057, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Mark J Huiskes, Bart Thomee, and Michael S Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *Proceedings of the international conference on Multimedia information retrieval*, pages 527–536, 2010.

Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A large-scale database for aesthetic visual analysis. In *Computer Vision and Pattern Recognition*, pages 2408–2415. IEEE, 2012.

Jana Machajdik and Allan Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 83–92. ACM, 2010.

Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *Proceedings of the International Conference on Computer Vision*, pages 2106–2113, 2009.

Subramanian Ramanathan, Harish Katti, Nicu Sebe, Mohan Kankanhalli, and Tat-Seng Chua. An eye fixation database for saliency detection in images. *European Conference on Computer Vision*, pages 30–43, 2010.

Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *Proceedings of the international conference on World wide Web*, pages 867–876. ACM, 2014.

Babak Saleh, Ali Farhadi, and Ahmed Elgammal. Object-centric anomaly detection by attribute-based reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–794, 2013.

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.

W. Pirie. Spearman rank correlation coefficient. *Encyclopedia of statistical sciences*, 1988.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Bin Jin, Maria V. Segovia, and Sabine Süsstrunk. Image aesthetic predictors based on weighted CNNs. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2291–2295. Ieee, 2016.

Yueying Kao, Chong Wang, and Kaiqi Huang. Visual aesthetic quality assessment with a regression model. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 1583–1587. IEEE, 2015.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Lingyun Zhang, Matthew H. Tong, Tim K. Marks, Honghao Shan, and Garrison W. Cottrell. SUN: A bayesian framework for saliency using natural statistics. *Journal of vision*, 8(7):32, 2008.

Po-He Tseng, Ran Carmi, Ian GM Cameron, Douglas P Munoz, and Laurent Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of vision*, 9(7):4–4, 2009.

Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. In *Computer Vision and Pattern Recognition*, pages 1804–2767. Springer Berlin/Heidelberg, Germany, 2018.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. corr abs/2011.10566 (2020). *arXiv preprint arXiv:2011.10566*, 2020.

Jeffrey M Zacks, Nicole K Speer, Khena M Swallow, Todd S Braver, and Jeremy R Reynolds. Event perception: a mind-brain perspective. *Psychological bulletin*, 133 (2):273, 2007.

G Elliott Wimmer, Yunzhe Liu, Neža Vehar, Timothy EJ Behrens, and Raymond J Dolan. Episodic memory retrieval success is associated with rapid replay of episode content. *Nature Neuroscience*, 23(8):1025–1033, 2020.

David Clewett, Sarah DuBrow, and Lila Davachi. Transcending time in the brain: How event memories are constructed from experience. *Hippocampus*, 29(3):162–183, 2019.

Endel Tulving. Episodic and semantic memory. *Organization of memory/Eds E. Tulving, W. Donaldson, NY: Academic Press*, pages 381–403, 1972.

Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53:1–25, 02 2002.

Olivier Jeunehomme, Adrien Folville, David Stawarczyk, Martial Van der Linden, and Arnaud D'Argembeau. Temporal compression in episodic memory for real-life events. *Memory*, 26(6):759–770, 2018. PMID: 29173013.

Olivier Jeunehomme and Arnaud D'Argembeau. Event segmentation and the temporal compression of experience in episodic memory. *Psychological research*, 84(2): 481–490, 2020.

Joseph E Dunsmoor, Vishnu P Murty, Lila Davachi, and Elizabeth A Phelps. Emotional learning selectively and retroactively strengthens memories for related events. *Nature*, 520(7547):345–348, 2015.

Youssef Ezzyat and Lila Davachi. What constitutes an episode in episodic memory? *Psychological science*, 22(2):243–252, 2011.

George A Miller. The magical number seven plus minus two: some limits on our capacity for processing information. *Psychological Review*, 29:106–112, 1956.

Carol L Novak and Steven A Shafer. Anatomy of a color histogram. In *Proceedings of the IEEE CVPR*, pages 599–605. IEEE, 1992.

Kieran G. Larkin. Reflections on shannon information: In search of a natural information-entropy for images. *CoRR*, abs/1609.01117, 2016.

Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks. *arXiv preprint arXiv:1507.02313*, 2015.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 505–520. Springer, 2014.

Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the EMNLP*, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the NIPS*, pages 5998–6008. Curran Associates, Inc., 2017.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 766–782. Springer, 2016.

Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder-decoder networks. *arXiv preprint arXiv:1708.09545*, 2017.

Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. *Proceedings of the IEEE CVPR*, pages 2982–2991, 2017.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the ACM Multimedia Conference*, pages 863–871, 2017.

Mrigank Rochan and Yang Wang. Learning video summarization using unpaired data. *arXiv preprint arXiv:1805.12174*, 2018.

Yitian Yuan, Tao Mei, Peng Cui, and Wenwu Zhu. Video summarization by learning deep side semantic embedding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang, and Chen Yao. Video summarization via semantic attended networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Mengjuan Fei, Wei Jiang, and Weijie Mao. Memorable and rich video summarization. *J. Vis. Comun. Image Represent.*, 42(C):207–217, January 2017.

Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Amnet: Memorability estimation with attention. In *Proceedings of the IEEE CVPR*, pages 6363–6372, 2018.

Luciana dos Santos Belo, Carlos Antônio Caetano Jr, Zenilton Kleber Gonçalves do Patrocínio Jr, and Silvio Jamil Ferzoli Guimarães. Summarizing video sequence using a graph-based hierarchical approach. *Neurocomputing*, 173:1001–1016, 2016.

Mayu Otani et al. Video summarization using deep semantic features. In *Proceedings of the Asian Conference on Computer Vision*, pages 361–377, 2016.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Proceedings of the NIPS*, pages 2204–2212, 2014.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the EMNLP*, 2015.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE ICCV*, pages 4507–4515, 2015.

Alex Graves et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471, 2016.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the EMNLP*, pages 551–561, 2016.

Ankur Parikh et al. A decomposable attention model for natural language inference. In *Proceedings of the EMNLP*, pages 2249–2255, 2016.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *Proceedings of the ICLR*, 2017.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

Vanya Cohen and Aaron Gokaslan. Opengpt-2: Open language models and implications of generated text. *XRDS*, 27(1):26–30, sep 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Nadine Behrmann, Jurgen Gall, and Mehdi Noroozi. Unsupervised video representation learning by bidirectional feature prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1670–1679, 2021.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. *Advances in Neural Information Processing Systems*, 32:81–91, 2019.

Jonas Gehring et al. Convolutional sequence to sequence learning. In *Proceedings of the ICML*, pages 1243–1252, 06–11 Aug 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Proceedings of the ECCV*, pages 540–555. Springer, 2014.

Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

Michael Gygli et al. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE CVPR*, pages 3090–3098, 2015.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Baoyuan Wu, Weidong Chen, Yanbo Fan, Yong Zhang, Jinlong Hou, Jie Liu, and Tong Zhang. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *IEEE Access*, 7, 2019a.

Jeff Hawkins and Subutai Ahmad. Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Frontiers in Neural Circuits*, 10:23, 2016.

Matthias Bethge and Philipp Berens. Near-maximum entropy models for binary neural representations of natural images. In , *Advances in Neural Information Processing Systems 20*, pages 97–104. Curran Associates, Inc., 2008.

Jack W Rae, Jonathan J Hunt, Tim Harley, Ivo Danihelka, Andrew Senior, Greg Wayne, Alex Graves, and Timothy P Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3628–3636, Red Hook, NY, USA, 2016. Curran Associates Inc.

Yoshua Bengio, Aaron C Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1: 2012, 2012.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013a.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, apr 1982.

Pentti Kanerva. *Sparse Distributed Memory*. MIT Press, Cambridge, MA, USA, 1988.

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160:106–154, January 1962.

Colin Blakemore and Grahame F Cooper. Development of the brain depends on the visual environment. *Nature*, 228(5270):477–478, 1970.

Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01), 1961.

Horace B Barlow, Tej P Kaushal, and Graeme J Mitchison. Finding minimum entropy codes. *Neural Computation*, 1(3):412–423, 1989.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Horace B Barlow. Single units and sensation: a neuron doctrine for perceptual psychology? *Perception*, 1(4):371–394, 1972.

Rodrigo Quian Quiroga. Concept cells: the building blocks of declarative memory functions. *Nature Reviews Neuroscience*, 13(8):587–597, 2012.

D.H. Perkel and T.H. Bullock. *Neural Coding: By Donald H. Perkel and Theodore Holmes Bullock*. Neurosciences Research Program bulletin. Nature Publishing Group, 1968.

Richard Robinson. Short stimulus, long response: sodium and calcium dynamics explain persistent neuronal firing. *PLoS biology*, 13(12):e1002320, 2015.

Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.

Edmund T Rolls and Alessandro Treves. The neuronal encoding of information in the brain. *Progress in neurobiology*, 95(3):448–490, 2011.

Richard B Stein, E Roderich Gossen, and Kelvin E Jones. Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6(5):389–397, 2005.

Romain Brette. Philosophy of the spike: Rate-based vs. spike-based theories of the brain. *Frontiers in Systems Neuroscience*, 9:151, 2015.

Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

Benjamin Willmore and David J Tolhurst. Characterizing the sparseness of neural codes. *Network: Computation in Neural Systems*, 12(3):255–270, 2001.

David Attwell and Simon B Laughlin. An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow & Metabolism*, 21(10):1133–1145, 2001.

Peter Lennie. The cost of cortical computation. *Current biology*, 13(6):493–497, 2003.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

Tomáš Hromádka, Michael R DeWeese, and Anthony M Zador. Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS Biol*, 6(1):e16, 2008.

Michael Weliky, József Fiser, Ruskin H Hunt, and David N Wagner. Coding of natural scenes in primary visual cortex. *Neuron*, 37(4):703–718, 2003.

Alison L Barth and James FA Poulet. Experimental evidence for sparse firing in the neocortex. *Trends in neurosciences*, 35(6):345–355, 2012.

OJ Hulme, M Skov, MJ Chadwick, HR Siebner, and TZ Ramsøy. Sparse encoding of automatic visual association in hippocampal networks. *NeuroImage*, 102(2):458–464, 2014.

Peter Foldiak. Sparse coding in the primate cortex. *The handbook of brain theory and neural networks*, 2003.

Michael SA Graziano, Charlotte SR Taylor, and Tirin Moore. Complex movements evoked by microstimulation of precentral cortex. *Neuron*, 34(5):841–851, 2002.

Michael SA Graziano and Tyson N Aflalo. Mapping behavioral repertoire onto the cortex. *Neuron*, 56(2):239–251, 2007.

Roozbeh Kiani, Hossein Esteky, Koorosh Mirpour, and Keiji Tanaka. Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *Journal of neurophysiology*, 97(6):4296–4309, 2007.

David Marr. A theory of cerebellar cortex. *The Journal of Physiology*, 202(2):437, 1969.

D Golomb, N Rubin, and Haim Sompolinsky. Willshaw model: Associative memory with sparse coding and low firing rates. *Physical Review A*, 41(4):1843–1854, 1990.

Jeff Hawkins, Marcus Lewis, Mirko Klukas, Scott Purdy, and Subutai Ahmad. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, 12, jan 2019.

Bruno B Averbeck, Peter E Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature reviews neuroscience*, 7(5):358–366, 2006.

Stefano Panzeri, Jakob H Macke, Joachim Gross, and Christoph Kayser. Neural population coding: combining insights from microscopic and mass signals. *Trends in cognitive sciences*, 19(3):162–172, 2015.

Jonathon Shlens, Greg D Field, Jeffrey L Gauthier, Matthew I Grivich, Dumitru Petrusca, Alexander Sher, Alan M Litke, and EJ Chichilnisky. The structure of multi-neuron firing patterns in primate retina. *Journal of Neuroscience*, 26(32):8254–8266, 2006.

Kiersten Ruda, Joel Zylberberg, and Greg D Field. Ignoring correlated activity causes a failure of retinal population codes. *Nature communications*, 11(1):1–15, 2020.

Felix Franke, Michele Fiscella, Maksim Sevelev, Botond Roska, Andreas Hierlemann, and Rava Azeredo da Silveira. Structures of neural correlation and how they favor coding. *Neuron*, 89(2):409–422, 2016.

Yumiko Yoshimura and Edward M Callaway. Fine-scale specificity of cortical networks depends on inhibitory cell type and connectivity. *Nature neuroscience*, 8(11): 1552–1559, 2005.

James Tee and Desmond P Taylor. Is information in the brain represented in continuous or discrete form? *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 6(3):199–209, 2020.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In , *International Conference on Learning Representations*, 2014.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013b.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Geoffrey Hinton. Neural networks for machine learning, coursera video lectures. *Coursera*, 2012.

Sheila H Nirenberg and Jonathan D Victor. Analyzing the activity of large populations of neurons: how tractable is the problem? *Current opinion in neurobiology*, 17 (4):397–400, 2007.

Vinit Kumar Mishra, Karthik Natarajan, Hua Tao, and Chung-Piaw Teo. Choice prediction with semidefinite optimization when utilities are correlated. *IEEE Transactions on Automatic Control*, 57(10):2450–2463, 2012.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations*, 2019.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. In *International Conference on Learning Representations*, 2017.

Ali Razavi, Aaron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-VAEs. In *International Conference on Learning Representations*, 2019a.

Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In , *Advances in Neural Information Processing Systems 29*, pages 658–666. Curran Associates, Inc., 2016.

Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision*, pages 1133–1141. IEEE, 2017.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *International Conference on Learning Representations*, 2017.

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 710–720, 2018.

Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR 2017*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In , *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 2021.

Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, December 2015.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. In *Master's thesis, Department of Computer Science, University of Toronto*. Technical Report,University of Toronto, 2009.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are GANs Created Equal? A Large-Scale Study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 698–707, Red Hook, NY, USA, 2018. Curran Associates Inc.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, pages 5228–5237, 2018.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14866–14876, 2019b.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In , *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016. PMLR.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Randall C. C. O'Reilly and Kenneth A. A. Norman. Hippocampal and neocortical contributions to memory: advances in the complementary learning systems framework. *Trends in cognitive sciences*, 6(12):505–510, December 2002.

Randall C O'Reilly, Rajan Bhattacharyya, Michael D Howard, and Nicholas Ketz. Complementary learning systems. *Cognitive science*, 38(6):1229–1248, 2014a.

Daoyun Ji and Matthew A Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107, 2007.

Blake A Richards, Frances Xia, Adam Santoro, Jana Husse, Melanie A Woodin, Sheena A Josselyn, and Paul W Frankland. Patterns across multiple memories are identified over time. *Nature neuroscience*, 17(7):981–986, 2014.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Matthew A Wilson and Bruce L McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172):676–679, 1994.

György Buzsáki. Two-stage model of memory trace formation: a role for "noisy" brain states. *Neuroscience*, 31(3):551–570, 1989.

Paul W Frankland and Bruno Bontempi. The organization of recent and remote memories. *Nature reviews neuroscience*, 6(2):119–130, 2005.

Yadin Dudai. The neurobiology of consolidations, or, how stable is the engram? *Annu. Rev. Psychol.*, 55:51–86, 2004.

Xiaojing Wu and David J Foster. Hippocampal replay captures the unique topological structure of a novel environment. *Journal of Neuroscience*, 34(19):6459–6469, 2014.

Anoopum S Gupta, Matthijs AA van der Meer, David S Touretzky, and A David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5): 695–705, 2010.

Anthony V. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

Bruce L McNaughton and Lynn Nadel. Hebb-marr networks and the neurobiological representation of action in space. *Neuroscience and connectionist theory*, pages 1–63, 1990.

James J Knierim and Joshua P Neunuebel. Tracking the flow of hippocampal computation: Pattern separation, pattern completion, and attractor dynamics. *Neurobiology of learning and memory*, 129:38–49, 2016.

Joshua P. Neunuebel and James J. Knierim. CA3 Retrieves Coherent Representations from Degraded Input: Direct Evidence for CA3 Pattern Completion and Dentate Gyrus Pattern Separation. *Neuron*, 81(2):416–427, 2014.

Jill K. Leutgeb, Stefan Leutgeb, May-Britt Moser, and Edvard I. Moser. Pattern Separation in the Dentate Gyrus and CA3 of the Hippocampus. *Science*, 315(5814): 961–966, 2007.

Heekyung Lee, Cheng Wang, Sachin S. Deshmukh, and James J. Knierim. Neural Population Evidence of Functional Heterogeneity along the CA3 Transverse Axis: Pattern Completion versus Pattern Separation. *Neuron*, 87(5):1093–1105, 2015.

Thomas J. McHugh, Matthew W. Jones, Jennifer J. Quinn, Nina Balthasar, Roberto Coppari, Joel K. Elmquist, Bradford B. Lowell, Michael S. Fanselow, Matthew A. Wilson, and Susumu Tonegawa. Dentate gyrus nmda receptors mediate rapid pattern separation in the hippocampal network. *Science*, 317(5834):94–99, 2007.

Paul E Gilbert, Raymond P Kesner, and Inah Lee. Dissociating hippocampal subregions: A double dissociation between dentate gyrus and ca1. *Hippocampus*, 11(6): 626–636, 2001.

C. Brock Kirwan, Andrew Hartshorn, Shauna M. Stark, Naomi J. Goodrich-Hunsaker, Ramona O. Hopkins, and Craig E.L. Stark. Pattern separation deficits following damage to the hippocampus. *Neuropsychologia*, 50(10):2408–2414, 2012.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward theory of visual cortex accounts for human performance in rapid categorization. *CBCL Paper# MMVI-02. Cambridge, MA, Massachusetts Institute of Technology*, 2006.

Baktash Babadi and Haim Sompolinsky. Sparseness and expansion in sensory representations. *Neuron*, 83(5):1213–1226, 2014.

R Quian Quiroga, Gabriel Kreiman, Christof Koch, and Itzhak Fried. Sparse but not 'grandmother-cell'coding in the medial temporal lobe. *Trends in cognitive sciences*, 12(3):87–91, 2008.

Glenn C Turner, Maxim Bazhenov, and Gilles Laurent. Olfactory representations by drosophila mushroom body neurons. *Journal of neurophysiology*, 99(2):734–746, 2008.

Yoshinori Aso, Daisuke Hattori, Yang Yu, Rebecca M Johnston, Nirmala A Iyer, Teri-TB Ngo, Heather Dionne, LF Abbott, Richard Axel, Hiromu Tanimoto, et al. The neuronal architecture of the mushroom body provides a logic for associative learning. *elife*, 3:e04577, 2014.

Noel E Sharkey and Amanda JC Sharkey. An analysis of catastrophic interference. *Connection Science*, 1995.

Robert M French. Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference. In *Proceedings of the 16th Annual Cognitive Science Society Conference*, pages 335–340. Earlbaum, 1994.

Torsten N Wiesel and David H Hubel. Extent of recovery from the effects of visual deprivation in kittens. *Journal of neurophysiology*, 28(6):1060–1072, 1965.

B Timney, DE Mitchell, and F Giffin. The development of vision in cats after extended periods of dark-rearing. *Experimental Brain Research*, 31(4):547–560, 1978.

Donald E Mitchell. The extent of visual recovery from early monocular or binocular visual deprivation in kittens. *The Journal of physiology*, 395(1):639–660, 1988.

Peter R Huttenlocher and Chr De Courten. The development of synapses in striate cortex of man. *Human neurobiology*, 6(1):1–9, 1987.

Oliver Sacks. Stereo sue. *New Yorker*, 82(18):64, 2006.

David Taylor et al. Critical period for deprivation amblyopia in children. *Transactions of the ophthalmological societies of the United Kingdom*, 99(3):432–439, 1979.

Jiefeng Jiang, Wanlin Zhu, Feng Shi, Yong Liu, Jun Li, Wen Qin, Kuncheng Li, Chunshui Yu, and Tianzi Jiang. Thick visual cortex in the early blind. *Journal of Neuroscience*, 29(7):2205–2211, 2009.

Robert M French. Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, 4(3-4):365–377, 1992.

Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the ninth annual conference of the Cognitive Science Society*, pages 177–186, 1987.

Ben Goodrich and Itamar Arel. Unsupervised neuron selection for mitigating catastrophic forgetting in neural networks. In *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 997–1000. IEEE, 2014.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Gido M van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *CoRR*, abs/1810.12488, 2018.

Dushyant Rao, Francesco Visin, Andrei A Rusu, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Continual unsupervised representation learning. In *Conference on Neural Information Processing Systems*. Conference on Neural Information Processing Systems, 2019.

Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2019.

Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, 2017.

Arslan Chaudhry, Puneet Kumar Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 2018a.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12365, pages 86–102. Springer, 2020.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NIPS*, 2017.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *CoRR*, abs/1812.00420, 2018b.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2018.

Robert M French. Pseudo-recurrent connectionist networks: An approach to the'sensitivity-stability'dilemma. *Connection Science*, 9(4):353–380, 1997.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014a.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NIPS*, pages 2994–3003, 2017.

Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019.

Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11 (1):1–14, 2020.

Woo-Young Kang and BT Zhang. Continual learning with generative replay via discriminative variational autoencoder. In *NIPS, Continual Learning Workshop*, 2018.

Joseph KJ and Vineeth Nallure Balasubramanian. Meta-consolidation for continual learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Andrei A. Rusu, Neil C. Rabinowitz, G. Desjardins, Hubert Soyer, J. Kirkpatrick, K. Kavukcuoglu, Razvan Pascanu, and R. Hadsell. Progressive neural networks. *ArXiv*, abs/1606.04671, 2016.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734, 2017.

Nicolas Y Masse, Gregory D Grant, and David J Freedman. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, 115(44):E10467–E10475, 2018.

German Ignacio Parisi, Xu Ji, and Stefan Wermter. On the role of neurogenesis in overcoming catastrophic forgetting. *CoRR*, abs/1811.02113, 2018.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019b.

David J Willshaw, O Peter Buneman, and Hugh Christopher Longuet-Higgins. Non-holographic associative memory. *Nature*, 222(5197):960–962, 1969.

John K Kruschke. ALCOVE: an exemplar-based connectionist model of category learning. *Psychological Review*, 99:22–44, 1992.

Robert Coop and Itamar Arel. Mitigation of catastrophic forgetting in recurrent neural networks using a fixed expansion layer. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2013.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Jacob MJ Murre. *Learning and categorization in modular neural networks*. Psychology Press, 2014.

Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pages 5327–5337. PMLR, 2020.

Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. The htm spatial pooler-a neocortical algorithm for online sparse distributed coding. *Frontiers in computational neuroscience*, 11:111, 2017.

Marcus Lewis, Scott Purdy, Subutai Ahmad, and Jeff Hawkins. Locations in the neocortex: a theory of sensorimotor object recognition using cortical grid cells. *Frontiers in neural circuits*, 13:22, 2019.

Subutai Ahmad and Jeff Hawkins. How do neurons operate on sparse distributed representations? a mathematical theory of sparsity, neurons and active dendrites. *arXiv preprint arXiv:1601.00720*, 2016.

Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *NIPS*, 2016.

Dmitry Krotov and John J. Hopfield. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*, 2021.

Yan Wu, Greg Wayne, Alex Graves, and Timothy Lillicrap. The kanerva machine: A generative distributed memory. In *ICLR*, 2018a.

Yan Wu, Gregory Wayne, Karol Gregor, and Timothy P. Lillicrap. Learning attractor dynamics for generative memory. In *NeurIPS*, 2018b.

Adam Marblestone, Yan Wu, and Greg Wayne. Product kanerva machines: Factorized bayesian memory. *arXiv preprint arXiv:2002.02385*, 2020.

Jason Ramapuram, Yan Wu, and Alexandros Kalousis. Kanerva++: Extending the kanerva machine with differentiable, locally block allocated latent memory. In *International Conference on Learning Representations*, 2021.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradientbased neural networks. In *In Proceedings of International Conference on Learning Representations (ICLR)*, 2014b.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266): 1332–1338, 2015.

Pierre Sermanet, Soumith Chintala, and Yann LeCun. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3288–3291. IEEE, 2012.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638, 2016.

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In , *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 17–26. PMLR, 13–15 Nov 2017.

Ryne Roady, Tyler L. Hayes, Hitesh Vaidya, and Christopher Kanan. Stream-51: Streaming classification and novelty detection from videos. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn*. PhD thesis, Technische Universität München, 1987.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pages 3–17. Springer, 1998.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001b.

Jürgen Schmidhuber. A 'self-referential'weight matrix. In *International Conference on Artificial Neural Networks*, pages 446–450. Springer, 1993.

Oriol Vinyals. Model vs optimization meta learning. In *NIPS 2017 Metalearning Symposium, 2017*, 2017.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, December 2017.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

Ke Li and Jitendra Malik. Learning to optimize. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

James J DiCarlo, Davide Zoccolan, and Nicole C Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. Classification and geometry of general perceptual manifolds. *Physical Review X*, 8(3):031003, 2018.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35 (11):2765–2781, 2013.

Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Latent bernoulli autoencoder. In *International Conference on Machine Learning*, pages 2964–2974. PMLR, 2020.

John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

Randall C O'reilly and James L McClelland. Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off. *Hippocampus*, 4(6):661–682, 1994.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *arXiv preprint arXiv:2102.00554*, 2021.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021.

Daniel J Graham and David J Field. Sparse coding in the neocortex. *Evolution of nervous systems*, 3:181–187, 2006.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019.

Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

Randall C O'Reilly, Rajan Bhattacharyya, Michael D Howard, and Nicholas Ketz. Complementary learning systems. *Cognitive science*, 38(6):1229–1248, 2014b.

Mehmet Kerem Muezzinoglu, C Guzelis, and Jacek M Zurada. An energy function-based design method for discrete hopfield associative memory with attractive fixed points. *IEEE Transactions on Neural Networks*, 16(2):370–378, 2005.