DEGREE PROJECT IN COMPUTER ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2020

One-shot learning through generalized representations with neural networks

PAUL VINELL

ADAM WIKER

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

One-shot learning through generalized representations with neural networks

PAUL VINELL, ADAM WIKER

Bachelor in Computer Science
Date: June 8, 2020
Supervisor: Pawel Herman
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Swedish title: Inlärning från ett exempel med generaliserade representationer i neurala nätverk

Abstract

Despite the rapid progress in the field of machine learning and artificial neural networks, many hurdles yet remain before machines can match human capabilities. One such hurdle is the copious amount of data required for these learning machines to reach adequate performance. There have been many methods to improve learning with limited data, going so far as to only use a single example, known as one-shot learning. A common strategy to take on one-shot learning involves turning the problem into a statistical comparison of two examples. In this paper, we propose a model that attempts to combine the information from a few given examples in order to learn more about the underlying distribution. We hypothesize that the performance of a model that is not reliant on a one-to-one comparison will scale better with increasing data, as it could combine information from different examples. Our proposed model involves training a convolutional neural network (CNN) to represent images in such a way that makes it easier for a smaller artificial neural network to infer whether a specified object is present in the image. To let the system learn about a new object category, the smaller network can simply be trained from scratch on that category. Testing this method reveals that training the CNN does not result in better performance compared to an untrained CNN with random initialization. Despite this, the smaller network learns surprisingly well even when dealing with limited data. As it stands, the proposed model offers no discernible benefits compared to previous work that uses statistical comparisons, but there may be room for further testing if the training procedure for the CNN is revised and improved on.

Sammanfattning

Trots snabba framsteg i maskininlärning och artificiella neuronnät kvarstår många hinder innan maskiner kan matcha mänskliga förmågor. Ett sådant hinder är den kopiösa mängd data som krävs för att dessa lärande maskiner ska nå adekvat prestanda. Det har funnits många metoder för att förbättra inlärning med en begränsad mängd data, även med så lite som ett exempel, också känt som one-shot learning (inlärning med ett exempel). En vanlig strategi för att tackla inlärning med ett exempel går ut på att göra om problemet till en statistisk jämförelse av två exempel. I den här studien föreslår vi en modell som försöker kombinera informationen av givna exempel för att lära sig mer om den underliggande fördelningen. Hypotesen är att prestationen av en modell som inte förlitar sig på direkta jämförelser av exempel kommer fungera bättre med större mängder data eftersom den kan kombinera information från flera givna exempel. Vår föreslagna modell innefattar ett faltningsnätverk (CNN) som representerar bilder på ett sådant sätt att det är lätt för ett mindre artificiellt neuronnät att avgöra om en objekttyp är närvarande i bilden. Tester av metoden visar att träningen av faltningsnätverket inte resulterar i bättre resultat än om ett otränat faltningsnätverk används. Trots detta lär sig det mindre nätverket förvånansvärt väl även i situationer med begränsad data. Som det ser ut ger den föreslagna modellen inga märkbara fördelar över tidigare modeller som använder statistiska jämförelser mellan exempel, men det kan finnas utrymme för att se om träningsproceduren för faltningsnätverket kan göras om och förbättras.

Acknowledgement

We want to thank our supervisor Pawel Herman for all of the support and feedback he has provided. We would also like to show appreciation to LeCun, Cortes, and Burges for their work on the MNIST dataset, and to Krizhevsky for the CIFAR-10 dataset.

Contents

Chapter 1

Introduction

One characteristic that sets humans apart from learning machines is our ability to learn from a single or a few examples. A machine might need thousands of examples in order to learn how to recognize a category of objects, while humans can manage with just a few [3]. This capability to generalize from a single example is referred to as one-shot learning. We know that humans are able to do this, but that in itself raises the question of how we can recognize new objects after a mere glance or two. A common explanation revolves around the human capacity to build and combine representations of objects [4]. For example, we can recognize that a bike has two wheels, a seat, and a handle. This is an advantage that humans have but traditional machine learning approaches lack. If we train a human to recognize images of a previously unseen object, such as a centaur, they can likely learn to recognize the object with ease as it is a combination of already familiar objects, consisting of parts from both a human and a horse. If, however, traditional machine learning was to do the same, it would take many more examples and repetition. This deficient performance would not only be due to a non-existent understanding of humans and horses, but also due to the machine lacking concepts to build an understanding on, such as legs, eyes, fur, etc. These learning machines have no prior understanding of the world and need to find their own way to represent or recognize patterns that they are exposed to.

While these learning machines have the potential to be very good at certain tasks, feeding them enough data so they can learn and perform well is not always feasible. If more data cannot be gathered then the solution lies in making the machines learn faster, learn differently, or start from a more knowledgeable state. With the human capability of one-shot learning as a reference, one idea to explore is to examine whether learning machines can be primed for novel tasks by gaining prior knowledge first.

The field of one-shot learning has commonly approached the problem of learning with little available data from the perspective that prior knowledge is key. How prior knowledge manifests itself usually varies depending on the model and task at hand. Sometimes techniques that proved useful for identifying previously seen classes can be used to find patterns in unseen classes [5]. Sometimes written characters can be recognized based on assumptions on how they were drawn [4]. The general trend seems to be that performance is improved with richer representations of the underlying data structure. The importance of representational ability for one-shot learning would suggest artificial neural networks (ANN) as a good basis for one-shot models. ANNs are, as the name implies, inspired by biological neural networks. By passing through all the transformations of the ANN layers it is possible to extract more useful representations from the data [6]. Today several different models utilize ANNs for one-shot learning. Broadly speaking, these models solve the one-shot problem by transforming it into a verification problem [7, 8]. The verification problem is the problem of determining whether two things depict the same category. The verification problem is identical to true one-shot learning but differs somewhat when given more examples. For instance, imagine you have dogs Abel, Balthazar, and Cain, and you are given a creature of an unknown species called Damon and you want to determine whether Damon is a dog. In the verification problem, you ask if Damon is similar to Abel, yes or no, Balthazar, yes or no, or Cain, yes or no. If Damon is similar to any one of the dogs then Damon is likely a dog. In contrast, without reduction to the verification problem you might notice that while Damon is not very similar to either of the dogs, Damon does have the snout of Abel, paws of Balthazar, and tail of Cain. Damon is indeed a dog despite not being very similar to either one.

There are several reasons why looking further than the verification problem might be a good idea. One way to describe the verification problem is that it works under the assumption that images depicting the same object category lie close to each other after they have been encoded to a representative vector space. This is a reasonable assumption that empirically allows for good classification abilities in the face of little available data [7, 8]. However, firstly we ask if we are interested in how different features relate to one another, and secondly if all features are of equal importance (for an object class). To understand the importance of feature correlation, we can for instance consider the case of a square. A square is only a square when both its width and height are the same. We can not tell that an object is a square by only looking at a single edge, only when we compare the edges to each other do we know if it is a square or not. Regarding the importance of an object's features, we can think of a worm. Length is one feature that may describe a worm but has little to do with whether or not it actually is a worm. However, the cylindrical body is a very distinct feature that we expect of all worms. This example shows that length is not an important feature of the worm while the body shape is.

1.1 Problem statement

The idea behind the verification problem is to determine whether two examples are statistically similar. For one-shot learning, reduction to the verification problem is a tactic to deal with little available data [7]. If two images are similar then they likely belong to the same category. However, it seems unlikely to be the best way to utilize several examples because available information is not combined. We can compare the simplified examples in figure 1.1. The left plot represents verification based one-shot learning. Observe that a lot of examples are needed to cover the true distribution. In the plot on the right, most of the underlying distribution is easily captured because it is assumed that the space between the given examples is part of it. As a consequence of not learning the underlying distribution, we hypothesize that current one-shot models will poorly scale with increasing data compared to a model that tries to learn the underlying distribution.

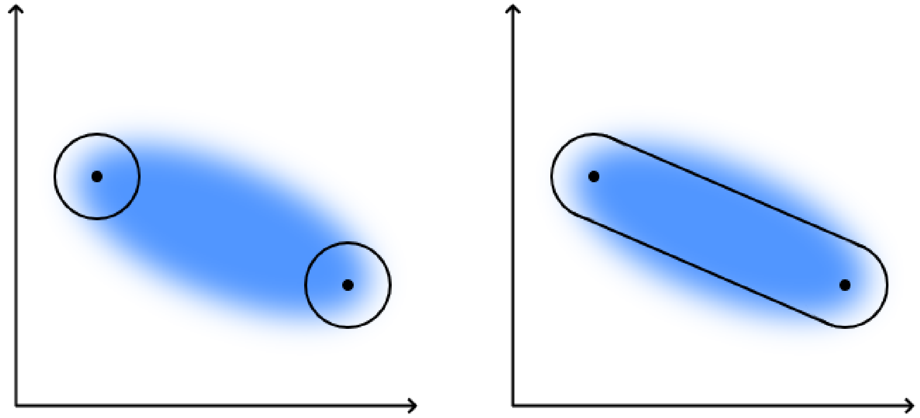

Figure 1.1: Simplified depictions of reduction to the verification problem (left) and inferring the underlying distribution (right). The black dots represent given training examples, the blue area is the underlying distribution they come from, and the black lines around the dots are the created decision boundary for deciding when something comes from the same distribution. For the verification problem (left) a decision boundary is created around each training example. In the other example (right) the distribution is assumed to be continuous between the training examples.

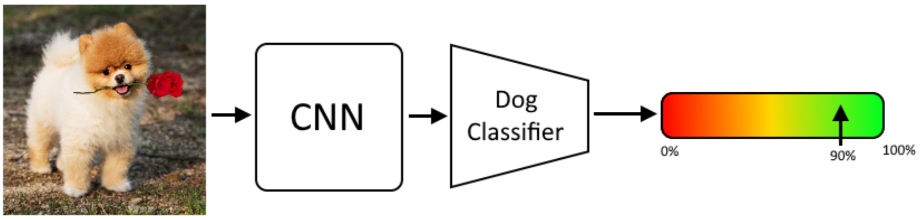

Figure 1.2: The input image is processed by a convolutional neural network, the classifier outputs its confidence (shown as 90%) that the object category (dog) is present in the image.

Feature representations that come from neural networks are often very close to each other when they represent similar objects [9]. As such there should be useful information that can be extracted from only a few examples while still being somewhat representative of an entire category. To take on the problem of one-shot learning without reducing it to the verification problem there is a need to extract useful yet generalized information from given images. As deep ANNs have the capability of creating representations of given examples and convolutional neural networks (CNN) are specialized for data with a grid-like topology [10], it is clear that deep CNNs are suitable for this problem.

But how can a CNN learn representations of something when it is only given one example? We hypothesize that this can be achieved by relying on the principles of transfer learning. In transfer learning the idea is to gain knowledge from training on one problem, then apply it to solve a different problem. According to Torrey and Shavlik, transfer learning allows a model to start with higher accuracy, increase its accuracy faster, and converge to higher final accuracy [1]. One hypothesis is that this works because some of the representations learned are useful for the subsequent task [6]. In figure 1.2 we see a model that trains on images of dogs, the hypothesis is that this model can switch the dog classifier to, for instance, a cat classifier and perform well with less training.

When only training with one or a few examples there is an increased risk of overfitting. The model may place a lot of importance on specific patterns that are present in a given example that might not be representative of the entire category. For example, take the dog image in figure 1.2 and shift and rotate the image slightly. The modified image should still clearly represent a dog, even if every single pixel is different. This is a technique called image augmentation, which helps the model capture many different ways that something can manifest itself [11]. The idea is that this combination of transfer learning and image augmentation will allow for a model to learn more about the underlying distribution instead of putting too much importance on specific details.

1.2 Research question

The previously mentioned problems lead us to the following question. Does training a system by switching binary classifiers and augmenting images offer better performance than previous approaches, which use statistical similarity,

[1] J. Brownlee. *Machine learning mastery.* Accessed on: June 8, 2020. [Online]. Available: `https://machinelearningmastery.com/transfer-learning-for-deep-learning/`

on image recognition tasks when only given one or a few examples?

1.3 Scope and objectives

Today, models for one-shot learning commonly use the strategy of comparing novel images with the reference images they have been given to classify them [7, 8]. This differs from conventional training of ANNs where the ANN tries to extract some knowledge from reference images before using this knowledge to classify novel images. The main difference between these techniques lies in verifying that two images depict the same category versus learning the characteristics of that category. The benefit of learning the characteristics is that the network can learn more complex boundaries than those created by relying on statistical similarity. This study aims to see whether a system that switches binary classifiers will learn generalized features and characteristics of the given data and if these are a good basis for the model to learn about an unseen category with only a few examples. The goal is to compare the performance of this model with that of models that rely on statistical similarity, to determine whether the proposed model has any use as an alternative.

In pursuit of this objective, the details of the chosen model and parameters were picked to be sufficient for the pattern recognition task with less importance being placed on optimal results. To test performance, two datasets, each with their own hurdles, are examined. Both datasets are meant for image classification tasks, more specifically for single object classification. The training procedure could in theory be applied to other classification tasks, but for this study, we only address image classification. This is because, while the model might be general, utilized techniques such as image augmentation do not translate directly to other types of data.

1.4 Thesis outline

The remainder of this thesis is organized in the following way. The next chapter provides relevant information on important concepts for the methods and research question. It also highlights related work that has touched upon similar or relevant topics. In chapter 3 the proposed model is explained along with the experiment structure that is used to evaluate it. Chapter 4 presents the results of those experiments, and a discussion on their meaning and importance follows in chapter 5. Final thoughts and conclusions are found in chapter 6.

Chapter 2

Background

This chapter gives the reader background information and definitions needed to understand the method and results of the report. It also presents related work to give the reader an overview of the context in the field of research.

2.1 Neural networks for image recognition

In this section general background related to neural networks and image recognition tasks is presented to give the reader an idea of the definitions used in the remainder of the report. The reader will also get an understanding of what basis the proposed ideas build upon.

2.1.1 Artificial Neural Networks

Artificial neural networks (ANN) have been proven to deal with many different tasks very well and they can even surpass human capabilities. One somewhat recent example of this is an ANN that could recognize sketches with an accuracy superior to that of humans [12]. ANNs have a wide range of possible uses, and among those, we have image classification, i.e. the ability to take an image and put a defining label on it. Using ANNs for image classification is something that has seen wide use, one commonly seen example being facial recognition. But training these networks traditionally requires many thousands of images in order for them to accurately classify any previously unseen images [3, 5]. With the development of the field, new and improved methods to train networks for these tasks emerge. Among these we have methods and techniques with the goal of allowing an ANN to train using only a few examples, even going so far as to only use one or, in the most extreme cases, zero training examples.

2.1.2 Fully connected ANN

A common, traditional, way to view ANNs is the fully connected feedforward neural network, sometimes called a multilayer perceptron. This ANN will approximate some function that takes an input and returns an output [10]. The input can for example be an image and the output can be some type of description of what is in the image. All layers in this network are alike in their input to output mapping; weighted input goes through an activation function and this will serve as input to the next layer. They are called feedforward because the data that is fed to them only goes in one direction - forward [10]. Typically for pattern recognition tasks, the output of the network will be an encoding of the different classes for the given data. This can be in the form of a vector with one node for every class, or for binary classification, a single node that ranges from zero to one.

2.1.3 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a type of ANN designed to process grid-like data, for example images [10]. These have what are called convolutional layers that are used to better capture information gained from spatial relations in the data. The name is derived from the fact that these layers use the mathematical convolution operation [10]. CNNs are very common when it comes to analyzing images as the convolutions help the network find meaningful features such as shapes and edges. Adding depth to the CNN can also help it learn the hierarchical structure of an image or pattern.

2.1.4 Representation learning

Understanding what an image contains can be a complex task. Previously, machines would require handcrafted features to interpret the content of an image. More recently, deep neural networks can interpret the content of an image from scratch by creating a self-learned representation of this content [13]. This is referred to as representation learning and can be explained as learning which underlying features of a certain image constitute its content. For example, if we have an image of a house, representation learning is about learning what parts of this image determine that the image contains a house. Training an ANN on a supervised task will naturally form a representation at each layer that helps the classification in the last layer [10]. There are however other ways to also boost the representation learning of a network. One example is called unsupervised greedy pretraining, where the network simply tries to reconstruct unlabeled data for each layer. In that process, we can learn which parts of the given data are needed to determine what is depicted. The benefit of this being that the network can use learned representations from the unsupervised phase to improve the performance of the supervised learning [10].

2.1.5 Transfer learning

Transfer learning is the ability to learn from one problem domain and use that knowledge to help solve problems in another. For example, a model that is tasked to learn categories in dataset A can have trouble doing so if dataset A does not have a lot of examples. Maybe there is another dataset with different categories, that has plenty of images, called dataset B. Training the same model to learn the categories of dataset B first can then help the generalization of dataset A, given that both tasks are similar. The idea is that even when training on a different dataset, the generalized feature representations can be useful for both tasks [10]. It is hypothesized that representation learning algorithms have an advantage in this domain because of the possibility of reusing the same representations for multiple tasks. This advantage is backed by empirical results [6].

2.1.6 Few-shot learning

As touched upon with transfer learning, it is possible for a neural network to learn a task even when there is a lower amount of training data. There is a concept called “few-shot learning” with this specific focus, learning a task with just a few data examples. For the lower bounds of few-shot learning, we of course have one-shot learning, learning from just one example, and even zero-shot learning, where no prior examples at all are given. One-shot learning can for example be about learning to recognize an object category from one example [3]. Although this does not necessarily make use of transfer learning, transfer learning is often the basis for this [14]. Methods aimed at achieving one-shot learning can make use of similarities between the unseen example and the known example, based on generalized representations learned from unrelated data. Zero-shot learning is the ability to recognize an object category never before observed. One example of how this could be done is demonstrated by Socher *et al.* [14]. In their work, zero-shot learning is achieved by having two different domains, one for text and one for images. A model is trained to map images to the text domain. This way, an unseen image can be classified by where it is mapped to in the text domain.

2.1.7 Leave-one-out

When the goal is to perform one-shot learning on one of multiple categories, there is a need to properly handle the data for all of these categories. One way to go about this is the leave-one-out method. Leave-one-out is more commonly used as an effective error estimator for pattern recognition [15]. The method is about as simple as the name implies, all images are included for training except for those of the chosen category. After training on those categories the network will have no knowledge of the final category and therefore all images of that category will remain as an unseen example. In this specific case, all the data for the category that one-shot learning will be attempted on is left out (the leave-out category) while all the data for the remaining categories is used for training.

2.1.8 Data augmentation

As has been previously highlighted, more training data usually leads to better performance in an ANN. This is also true even if the training data is of lower quality, as long as useful information can be extracted from it [16]. However, large amounts of data are not always readily available. One way to work around a small number of training examples is data augmentation, the idea of which is that by doing small modifications to an image from the actual dataset, more images of the same category can be created. It is thought that it may be possible to extract more information from a given image using these techniques [11]. Examples of simple image modifications include mirroring the image, rotating the image, or cropping the image.

2.2 Related work

To make an apt comparison to determine whether our chosen model can rival state-of-the-art methods we will first need to familiarize ourselves with some of those methods. This section will, therefore, look into previous work centered around few-shot learning with ANNs.

One model that resembles the model proposed in the thesis is that of Torralba *et al.* [17], who in 2004 investigated a boosting model for multiclass object detection. The concept is to split the model into a shared part and a specific part. The shared part looks for general image features and the specific parts use these results to draw conclusions about the contents of an image. There is one shared part and one specific part per object class. The training of all the classifiers is done jointly instead of independently to make the shared part look for patterns useful to all classifiers. One might expect general models to be outperformed by specialized models. Instead, Torralba et al. (2004) find that all classifiers benefit from being trained jointly, and a subset of those classifiers benefit greatly. This suggests potential benefits to favoring a general model over a more specific one.

In 2005, Bart and Ullman introduced a concept called cross-generalization [5]. They used this to train a classification model for one-shot learning, with positive results to show. By analyzing images of an object category for features it is possible to use the features to determine whether other images contain the same object category or not. Some features prove more useful than others when trying to classify new images. The idea behind cross-generalization is that important features for classifying one category will resemble the important features for a different but similar category. As an example, we can observe that both horses and dogs usually have four legs and a tail. The experiments were done on the Caltech 101 dataset [18], a dataset with 101 object categories, supplemented with six additional categories and 400 non-category images for negative examples. Using the leave-one-out method, all but one of the classes was used for training classifiers. Lastly, a new classifier was constructed based on one single positive example of an unseen category using cross-generalization. The classifier for this unseen category was then tested. This resulted in a test accuracy of 75%, where 50% is expected for a random guess. This rose to 90% when given 15 examples. When the same experiment was done on a subset consisting of eleven classes, the test accuracy was 66.5%, where chance again is 50%. With these results, Bart and Ullman successfully show the value of reusing prior knowledge of related classes in one-shot learning tasks. In another experiment in 2006 conducted on the Caltech dataset, Li Fei-Fei *et al.* demonstrated their Bayesian approach to one-shot learning [3]. In contrast to that of Bart and Ullman [5], their model was founded on the idea that knowledge acquired from previous categories can be utilized to classify future categories regardless of how past and future categories are related. Instead of supervised training, training before the one-shot task is done weakly supervised, meaning that the model is simply presented with "foreground images" that are guaranteed to contain an object. The model is then presented with a few examples of a novel category to be learned. The test of accuracy achieved for one-shot learning was around 78% for one example, and 90% after 15 examples, where guessing would result in around 50%. Through the success of their methods Li Fei-Fei *et al.* demonstrate that prior knowledge does not necessarily need to be related to the following task.

In a study by Lake *et al.* in 2011, a generative model for one-shot learning of handwritten characters from various languages was presented based on the idea of the importance of understanding [4]. They theorized that knowing the underlying structure of the characters, or in other words how they are drawn, would lay better grounds for classification than the image alone. By gathering a dataset of characters along with examples of people drawing them, the model could learn to guess how characters are most often drawn (in terms of the number of strokes, shape of strokes, order of strokes, etc). This dataset is known as the Omniglot dataset and features 1600 different characters with 20 examples per character [4]. Through the stroke abstraction, the model could make assumptions of how characters in images were drawn and use that to classify novel characters as the same (or different). Their model was evaluated and compared to other models on 20-way classification. This was done by picking 20 different, random, novel characters from their dataset. Accuracy was then measured by performance on novel images from these sets. Their method achieved greater results than all other models tried, getting 54.9% correct where 5% is chance performance. When given the correct strokes instead of inferring them they got 63.7% correct. This goes further than just demonstrating the power of prior knowledge. It also demonstrates the power of abstraction for classification tasks.

Koch introduced the idea of using siamese neural networks for one-shot learning in 2015 [7]. The foundation for this is that given two images, they can be compared by running them through two instances of the same CNNs resulting in two feature vectors. Ideally, a CNN will represent similar images as feature vectors lying close together in feature space. Classifying whether two images depict the same object category is done by observing their distance in this feature space, also known as the verification task. In the paper, this model was evaluated on the Omniglot dataset [4]. Test accuracy for 20-way classification on the Omniglot dataset was 90.61% after training with 30k training images. After training with 150k original images, with an addition of eight times as many generated images, they reached a test accuracy of 93.42%. Guessing would result in a performance of 5% on average. With this performance, the convolutional siamese neural network beats most models despite it not using the Omniglot stroke information. As a result, Koch demonstrates convincingly that one-shot learning capability is built upon a strong representational ability.

Following the Siamese neural network Vinyals *et al.* built another ANN-based construction in 2016 that they called matching networks [8]. In comparison to previous models, matching networks are not trained to recognize a specific set of object classes. Instead, they are given a support set, which is a set filled with labeled images of different object classes. When given an image to classify, the image is compared to members of the support set. The image inherits the label of the best match. The model is trained to improve performance on this task rather than on specific object classes. To compare images, each image in the support set is encoded by the same function. The image to be classified is encoded by a different function. Comparisons are done by distance in feature space. To make these encodings more useful, the encoding functions are conditioned on the support set as a whole. This means that the network is not purely based on verification but allows for some form of combination of different sets of information from the support set. They posted impressive results on ImageNet [19] and in language modeling tasks and improved performance on the Omniglot dataset [4] to 93.8% for one-shot and 98.7% for five-shot, besting competing approaches. This approach is interesting for several reasons, one of them being that it blends the verification problem with combining information from the given examples.

Chapter 3

Method

This chapter presents the proposed model along with the used data. How the model was evaluated is also described.

3.1 Data

For the one-shot experiments, the datasets CIFAR-10 [2] and MNIST [1] were used. Both CIFAR-10 and MNIST contain ten object categories. CIFAR-10 contains object categories such as dogs, frogs, automobiles, airplanes, etc. MNIST contains images of handwritten digits ranging from zero to nine. There is a big difference in the complexity of these two datasets. CIFAR-10 is the more complex dataset as two images of the same category, such as dogs, can vary a lot in appearance. MNIST is the less complex dataset as each digit is supposed to look a certain way, with differences being more in the style of handwriting.

For the training of the network the data was split into positive and negative examples. When training a certain classifier on a certain category 50% of the data from that category was used as positive examples labeled as the correct category. An equal amount of examples were taken from the remaining categories, this data was labeled as not being the correct category. In figure 3.1 we see a simplified illustration of how the data could be split between different classes. 10% of the training data was also used as validation data to monitor the training. When attempting one-shot learning on the last category, a few examples (between 1 and 15) were taken at random from the correct category. These few images were then used as the base to generate about 100 or 300 images through data augmentation, the original images did not remain in this training set. Each augmented set of images was of size $n_{augmented} = m * n_{original}$ such that $m * n_{original} \geq 100$ (or ≥ 300) and $m \in N$. The image generator applies modifications to copies of the images such as minor horizontal or vertical shifts, image rotations, horizontal mirroring, brightness, or zooming.

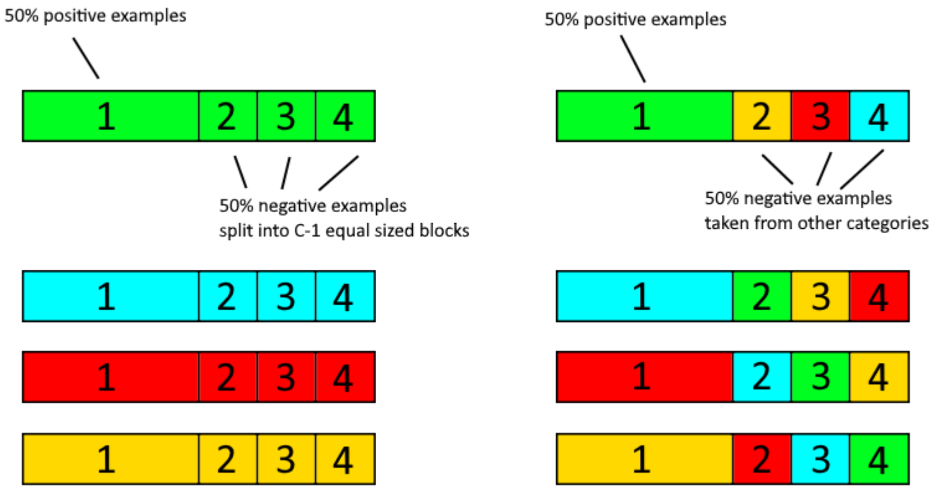

Figure 3.1: Example of how data of four classes would be split and shared to create a set for each class. The left side depicts how the original data is split, right side how the sets are structured.

3.2 Model

The model used consisted of two parts. The first part was a deep CNN tasked with representation learning, to extract meaningful features from the image that was to be classified. The second part was a shallow, fully connected ANN with the task of answering the simple question of whether or not the given image belonged to a certain class. The second part of the model was changed with every new label that the network attempted to classify.

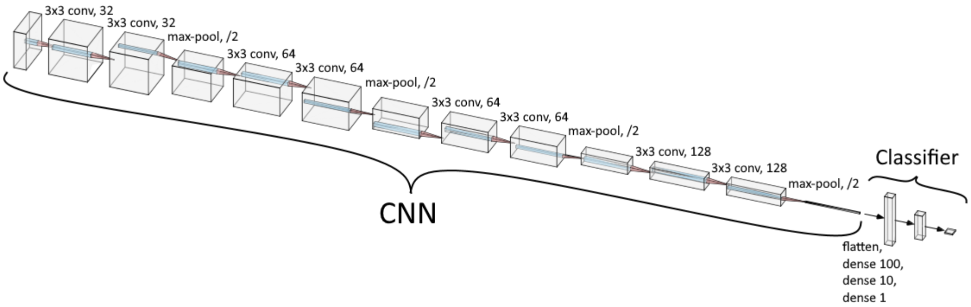

Figure 3.2: Detailed view of the model architecture depicting the shared CNN and a classifier.

The initial model was designed to get a good result with the classical approach for CIFAR-10 image classification with a 10-way classifier. A good result here is considered one that is not easy to make significant improvements upon. This model trained the same on all training data and had 10 output nodes, one for each category. The model consisted of eight convolutional layers and three fully connected layers, not counting the output layer (as seen in figure 3.2). To prevent overfitting, dropout and batch normalization were both used (as described by Chen *et al.* [20]). The classifier used L2-regularization (set to 0.001) for all layers for the same reason. All hyperparameters for this model were chosen from trial and error by hand, based on the validation loss and accuracy they yielded. This initial model was then used as the base for the proposed one-shot learning model. The one-shot learning model used almost the same structure and the same hyperparameters. The exception being the 10-way classification layer that was removed and replaced with a binary output, to create a binary classifier. The last few fully connected layers were considered the binary classifier and switched for each category it trained on.

3.3 Evaluation

Using the leave-one-out method training was done on all but one of the categories of images with the remaining one being left out. The larger set of categories will be referred to as the training categories and the remaining set as the leave-out category. Every category was chosen as the leave-out category once. In this stage, no training set was generated for the leave-out category, but validation and test sets were generated for all categories. For each training category, we trained a classifier. Performance was measured by accuracy on the test set.

The model was trained in rounds, each category trained for a single epoch at a time. One round is defined as an epoch of training for each training category. Validation data showed that training should be stopped after 16 rounds. After this, the accuracy of each binary classifier was tested on a set of test data. This is used as grounds for a relative measure of how the one-shot performance compares to training with a lot of data.

Lastly, one or a few images from the leave-out category were used to train a one-shot classifier. The number of images supplied were $n \in \{0, 1, 2, 3, 5, 7, 10, 15\}$. This is similar to the values used by Li Fei-Fei *et al.* [3]. The experiment was repeated 10 times in total for each value of n (except for $n = 0$ where only one trial was run because no training occurs).

When training a one-shot classifier, three augmented sets of images were created. Each set had different sets of negative examples for training. The proportion of positive examples was again half. Negative examples were chosen at random from the training categories. The training used early stopping and returned to the best set of weights after stopping. Only the shallow classifier was trained for this step, and the weights in the convolutional layers were fixed.

To account for the effects of random weight initialization and shuffling of the data, the experiment as a whole was run three times. Everything was run initially with 100 generated images to train with, and again with 300 generated images. This second experiment with 300 generated images was performed in order to see if the number of images would affect the result consistency and variance. It was also of interest to see if the training procedure actually leads to improvements in the CNNs representations, therefore an additional experiment was run. This time the convolutional layers remained fixed from the start and never trained. This allowed for a comparison to see how much of an impact there was from training the CNN.

After getting all the results it was possible to analyze if this model can rival the performance of methods using statistical comparison. The results were evaluated based on the test data accuracy. This was compared to the reported test data accuracy of previous methods.

Chapter 4

Results

The results in this section are ordered by dataset. The experiments were run on datasets CIFAR-10 and MNIST. The goal of the experiments was to test whether the model and training procedure have a positive effect on one-shot learning and to compare the final results with previous methods to see if this method has any benefit over them. One-shot performance is evaluated based on binary classification accuracy, this means that the classifiers are evaluated individually on their ability to discriminate their assigned object category from all other object categories. For interpretability, the performance of the one-shot classifiers is compared to that of normally trained classifiers and one-shot classifiers attached to randomly initialized, untrained CNNs. The comparison between one-shot classifiers and normally trained classifiers sheds light on to what degree the lack of data affects accuracy. The comparison between one-shot classifiers and one-shot classifiers attached to an untrained CNN quantifies the impact of repeated switching during training on the classification accuracy. All values in the graphs are depicted as a calculated mean and standard deviation from the data in all result batches, which means the results across all classifiers are taken into account. "Leave-out classifier" refers to the classifier that was left out during the initial training and instead trained with limited examples. "Other classifiers" refers to the classifiers that were part of the initial training and therefore trained with all available data.

4.1 CIFAR-10

When evaluating the model on CIFAR-10, the existing distinction between training and testing data was preserved. However, the data from each set was reorganized and relabeled so that it worked as a binary classification task instead. The results of the experiments on CIFAR-10 with 100 generated images can be seen in figure 4.1.

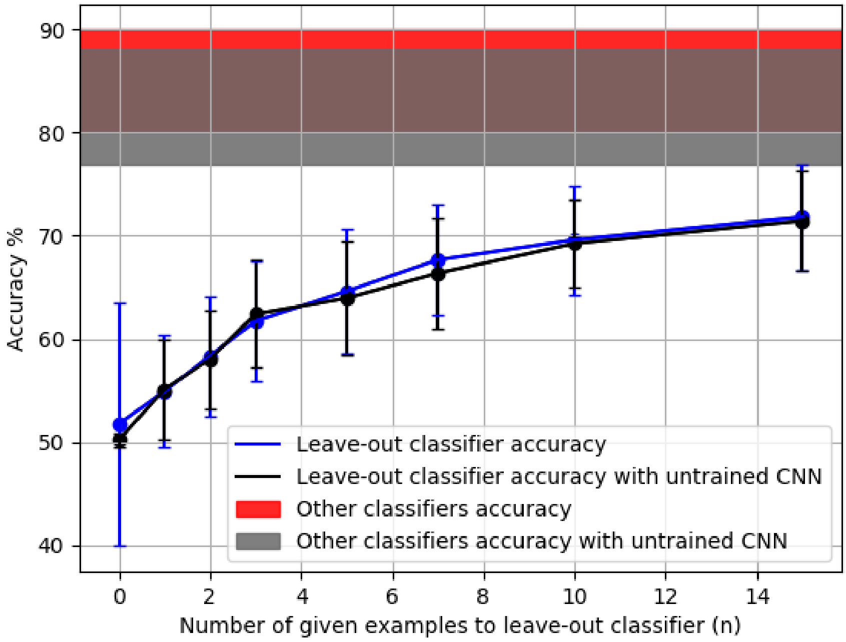

Figure 4.1: The results of the experiments performed on CIFAR-10 using 100 generated sample images. The black and blue lines show the mean accuracy of the test data and the standard deviation. The blue line depicts accuracy for the one-shot classifier with a trained CNN. The black line depicts accuracy for the one-shot classifier with a randomly initialized CNN. The red area depicts the mean accuracy of all normally trained classifiers across all experiments combined (±1 standard deviation). The gray area depicts the same as the red area but for an untrained CNN.

Based on the data shown in figure 4.1 there are multiple interesting takeaways. Most strikingly we can see that the performance of one-shot classifiers that are attached to a trained CNN is very similar to that of classifiers that are attached to an untrained CNN. We can also see that the one-shot classifier is capable of correctly identifying more than half the images, even with just a single training example, thus always performing better than a random choice. The average test accuracy continuously increases as more training examples are provided. The rise is fairly steady but pans out somewhat towards the end. A few examples of training data allows the one-shot classifier to confidently outperform pure chance but not to match the other classifiers. Next we have the results on CIFAR-10 with 300 generated images in figure 4.2.

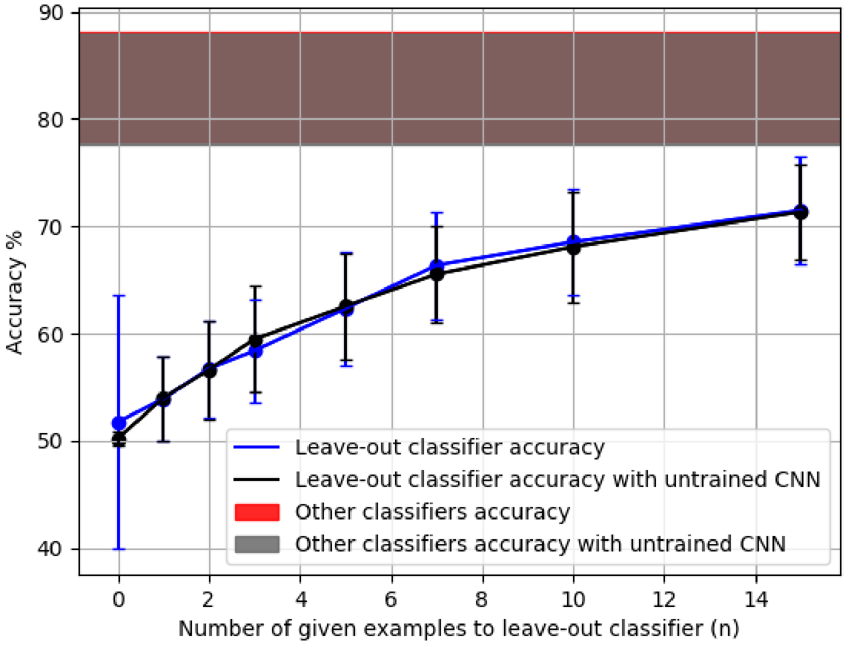

Figure 4.2: The results of the experiments performed on CIFAR-10 using 300 generated sample images.

Unlike the one-shot classifiers which appear the same in figure 4.1 and figure 4.2, the accuracy of classifiers attached to a trained CNN versus an untrained one differ. In figure 4.1 we see that the red and grey areas do not completely overlap, as they do in figure 4.2. Both of these results stem from the same conditions, which implies that the data in one of these figures is unreliable, or that there simply is some variance.

We can also make a comparison between figure 4.1 and figure 4.2 to observe that increasing the number of generated images, based on the one given example, does not seem to make much difference. In this case, we can in fact see that the experiments with more examples appear to have given slightly worse results than the experiments with fewer examples. This could be attributed to the network likely overfitting more with too many examples that look very similar to each other.

4.2 MNIST

Like previously with CIFAR-10, in the MNIST experiment, the existing split between training and testing data remained the same. The data was relabeled and reorganized into sets suited for binary classification. The MNIST dataset is very different from CIFAR-10 in that the images are more focused on the specific category, there is a single digit in each image and nothing else. Digits themselves also have less complicated features than the categories of CIFAR-10. As with CIFAR-10 we first have the results with 100 generated images in figure 4.3.

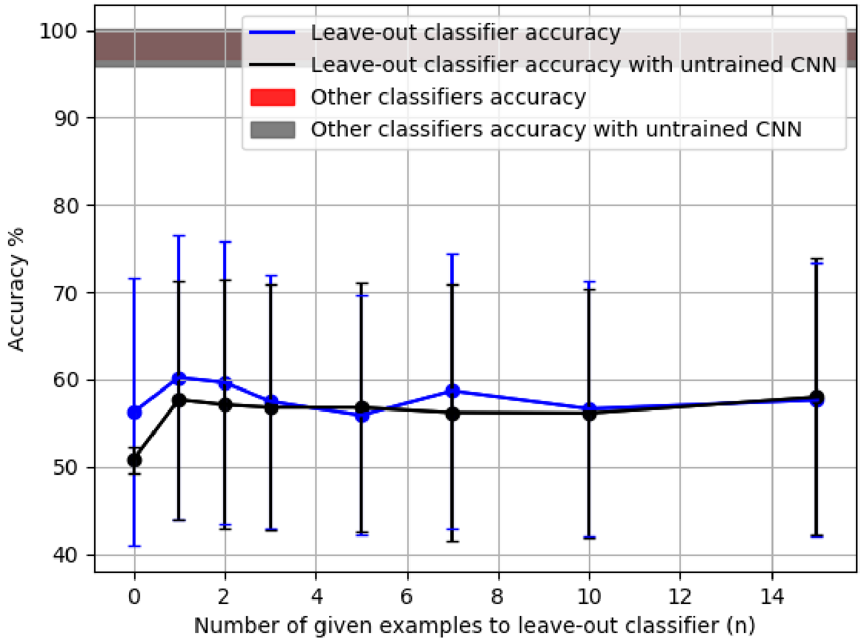

Figure 4.3: The results of the experiments performed on MNIST with 100 generated samples. The blue line represents one-shot accuracy on test data with a trained CNN, the black line with an untrained CNN. The red and grey areas describe the average performance of normally trained classifiers with trained and untrained CNNs (±1 standard deviation).

Just as in the experiments on CIFAR-10, we see again in figure 4.3 the results of a trained and untrained CNN performing about the same. However, compared to CIFAR-10 we see that test accuracy for the leave-out classifiers starts higher and scales worse. The normally trained classifiers also perform significantly better on MNIST than on CIFAR-10, but here there is no real difference when they are attached to a trained CNN versus when they are not. One notable difference from the CIFAR-10 results is the variance of the data points. As an example for n=15, the range goes from around 42% up to around 75%, which is a very big range as it covers around a third of all possible outcomes. We do however get a different point of view from the next results in figure 4.4.

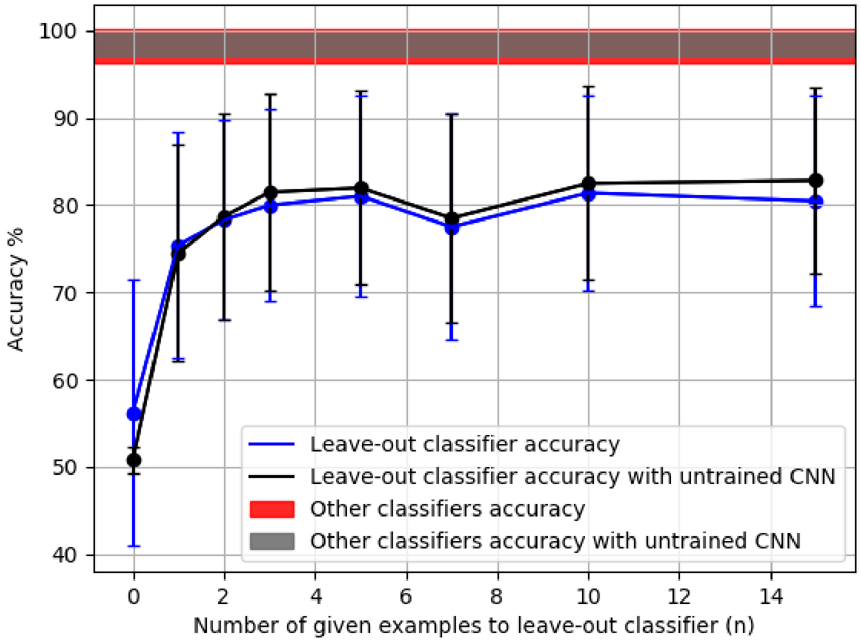

Figure 4.4: The results of the experiments performed on MNIST using 300 generated sample images.

When we take a look at the results on MNIST in figure 4.4 with 300 generated sample images we can see right away that the performance is a lot better than it was with only 100 generated sample images. This shows a clear difference from the CIFAR-10 experiments, where the increase in generated images did not have a positive effect. However, we still see the same pattern of the trained CNN performing about the same as the untrained CNN.

Chapter 5

Discussion

In general, the results support two key findings. Firstly, the output of the CNN trained with repeated switching of classifiers is of no more use than a randomly initialized CNN to a newly created classifier. Secondly, a low amount of positive examples could still be leveraged to achieve results well distinguished from chance. This means that the CNN part of the model has to be redesigned or trained differently for the proposed model to be useful.

The model fails to outperform older models that also use binary classification performance. For instance, in binary classification trials on the Caltech101 dataset Li Fei-Fei *et al.* [3] achieved results such as 78% accuracy from one example, increasing to 90% when given 15 examples. In comparison, the model in this paper got 55% from one example and 72% from 15 examples on CIFAR-10. The Caltech101 dataset [3] like CIFAR-10 contains images of different object categories. However, unlike CIFAR-10, the dimensions of images are around 300x200 pixels compared to 32x32. To what degree the discrepancy in image resolution contributes is not clear.

Another relevant comparison is to Matching Networks [8]. Here two barriers are preventing direct comparison - differing datasets, and differing evaluations. Nonetheless, a comparison can be made. When evaluating 20-way classification on the Omniglot dataset [4], Vinyals *et al.* [8] had an accuracy of 93.8% for one-shot and 98.7% for five-shot learning. The Omniglot dataset like MNIST depicts handwritten characters, but in comparison to MNIST, the amount of categories is vast and the amount of examples per character is limited. This outperforms the results of our proposed model that had 72% accuracy for one-shot and 75% for five-shot learning on a binary classification task on MNIST.

Based on these comparisons it seems that the proposed model does not improve on previous models when it comes to one-shot accuracy or scaling of accuracy with an increased amount of examples.

To explain why the network with a trained CNN consistently fails to outperform the network with an untrained CNN we propose a few hypotheses. One is that repeated switching might not do anything but create noise due to the different wants of the different classifiers, each epoch undoing any progress made in the last on the CNN. Maybe this supposed behavior could be somewhat counteracted by training only a single example of any one classifier at a time. Another hypothesis is that the technique encourages classifiers to off-load object category-specific calculations into the CNN meaning that instead of forcing general adaptations to occur in the CNN, some calculations specific to each object category enter the CNN during training. These adaptations are beneficial to each normally trained classifier but useless to new classifiers.

The results on CIFAR-10 with 100 generated images supports the hypothesis of off-loading object category-specific calculations into the CNN due to the higher performance of normally trained classifiers attached to a trained CNN. However, the results on CIFAR-10 with 300 generated images, along with the results on MNIST support the "no more than noise"-theory since normally trained classifiers are of the same average regardless of whether they are attached to a trained or untrained CNN. More experiments would need to be run in order to get a clear view on this, as it is hard to tell why the results on the different CIFAR-10 experiments do not align without any additional information.

There is also a factor that we did not expect to be of importance before the experiments started - how the number of generated examples improved the performance on the MNIST dataset but not on the CIFAR-10 dataset. The likely reason for this is the difference in complexity for the two datasets. For CIFAR-10 two images of different dogs can look rather different while still being in the same category. Therefore, training a lot on images that look like only one of the dogs is unlikely to help much. However, the digits in the MNIST dataset will all look somewhat the same, even though they can be drawn differently. One way to look at it is that the digits in MNIST already are distorted versions of each other, training on augmented versions of only a single example is therefore a lot more useful. An alternative explanation lies not in the trained category, but rather in the other categories. As the training data needs to remain balanced, adding more positive examples also means there will be more negative examples, which is images from other categories. It could be that the network simply learns more about what is not in the current category, rather than learning more about what is in it.

5.1 Limitations

When training the one-shot classifier, the validation accuracy was used as a measure for the early stopping of training. This prevents overfitting which easily happens with the few given training examples. The validation sets used contain plenty of positive examples, which in some sense means that training is done on more data than just the given examples. At the same time, the results achieved are still achievable without the validation sets. Given this fact, the best way to interpret the results of this study is as the optimum of the exact model used.

Another issue is the uncertainty in some of the result data. We can see that the results of the CIFAR-10 experiments with 100 generated images and the CIFAR-10 experiments with 300 generated images do not align with regards to the accuracy of the training classifiers when tested with a trained and untrained CNN. Due to time constraints, additional experiments were not run, but might have cleared up the uncertainty.

We see a similar issue with the results on MNIST with 300 generated images. There appears to be a dip in performance for seven original images, but there is no simple explanation as to why. More experiments might have clarified if this was a continuous trend or uncertain data that should be discarded. It is also possible that running experiments for original images of all values between 1 and 15 would have been insightful.

The way the data was split up for each experiment also has some underlying problems. In order to avoid making any of the binary classifiers biased towards positive or negative images, each training or test set was kept to half positive images and half negative images. This does, however, lead to each set of images including mostly images of a single category, while nine other categories need to share the remaining space. This can skew the learning in a different way, as the features of the positive category will be trained more during the training of that classifier. The images in the other half of the training set were also picked at random, with no regular cycle. It is therefore possible that some images were seen more often, and others were seen less often.

5.2 Ethics and sustainability

When studying how to learn with limited data one worry is the possibilities enabled by machines that can learn from only a few examples. Just as technology can be used for positive means, it can also have problematic consequences. Today machine learning in general is commonly used to track people unwillingly, coordinate drone strikes, influence public opinion, etc [21] - none of which we want more of. On the other hand, it is also used for voice assistants, medical diagnostics, and improving infrastructure, among other things. Enabling machines to learn with less data strengthens both of those sides. While we acknowledge the potential for problematic uses of such technology, we are hopeful that it will provide a mostly positive impact on the world. Another worry is one related to model bias. With low amounts of data, the risk of bias is high. This puts few-shot models more at risk for discriminating against various groups of people, something that is already a problem for regular machine learning models [21]. This could in other words pose a threat to social sustainability. To combat bias it is important to consider the quality of the data used.

Chapter 6

Conclusions

Based on our experiments and the previous discussion we have reached the following conclusions.

Training the shared CNN with repeatedly switching classifiers does not show any clear benefit compared to only training the classifiers. We can see that the model does not perform as well as methods using statistical comparison when training on only a few examples. However, despite the failure of the model to outperform previous methods, it seems that there is potential for traditional neural networks to learn from limited data when assisted by image augmentation. It is evident from the results that repeated category switching of binary classifiers does not lead to better performance than that of previous methods, but that image augmentation still has a positive effect.

6.1 Future work

From our findings, it seems that repeated switching of classifiers results in no apparent synergy. Instead, it seems that repeated switching to some degree merely undoes any previous progress achieved during training of the CNN. If the underlying problem is the lack of synergy between the training tasks then maybe better results could be achieved if the CNN was trained on one single task instead. Such a task could involve multi-way classification of categories or self-supervised learning of categories. The experiments also showed that it could be of interest to examine the effects of an increasing number of generated images or even the effect of different augmentation magnitudes.

Bibliography

[1] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database", *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[2] A. Krizhevsky, "Learning multiple layers of features from tiny images", *University of Toronto*, Apr. 2009.

[3] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[4] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts", *Cognitive Science*, vol. 33, 2011.

[5] E. Bart and S. Ullman, "Cross-generalization: Learning novel classes from a single example by feature replacement", in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, 672–679 vol. 1.

[6] Y. Bengio, A. Courville, and P. Vincent, *Representation learning: A review and new perspectives*, 2014. arXiv: `1206.5538 [cs.LG]`.

[7] G. R. Koch, "Siamese neural networks for one-shot image recognition", 2015.

[8] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, *Matching networks for one shot learning*, 2016. arXiv: `1606.04080 [cs.LG]`.

[9] B. Athiwaratkun and K. Kang, *Feature representation in convolutional neural networks*, 2015. arXiv: `1507.02313 [cs.CV]`.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[11] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning”, *Journal of Big Data*, vol. 6, pp. 1–48, 2019.

[12] Q. Yu, Y. Yang, Y.-Z. Song, T. Xiang, and T. Hospedales, *Sketch-a-net that beats humans*, 2015. arXiv: `1501.07873 [cs.CV]`.

[13] P. Cui, S. Liu, and W. Zhu, “General knowledge embedded image representation learning”, *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 198–207, 2018.

[14] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, and A. Y. Ng, *Zero-shot learning through cross-modal transfer*, 2013. arXiv: `1301.3666 [cs.CV]`.

[15] U. M. B. Neto and E. R. Dougherty, “Error estimation for pattern recognition”, 2015.

[16] L. Perez and J. Wang, *The effectiveness of data augmentation in image classification using deep learning*, 2017. arXiv: `1712.04621 [cs.CV]`.

[17] A. Torralba, K. P. Murphy, and W. T. Freeman, “Sharing features: Efficient boosting procedures for multiclass object detection”, in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, 2004, pp. II–II.

[18] Li Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories”, in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 178–178.

[19] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database”, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[20] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, and S. Zhang, *Rethinking the usage of batch normalization and dropout in the training of deep neural networks*, 2019. arXiv: `1905.05928 [cs.LG]`.

[21] R. Vinuesa, H. Azizpour, I. Leite, M. Balaam, V. Dignum, S. Domisch, A. Felländer, S. D. Langhans, M. Tegmark, and F. Fuso Nerini, “The role of artificial intelligence in achieving the sustainable development goals”, *Nature Communications*, vol. 11, no. 1, Jan. 2020, ISSN: 2041-1723. DOI: `10.1038/s41467-019-14108-y`.