DEGREE PROJECT, IN COMPUTER SCIENCE , FIRST LEVEL
STOCKHOLM, SWEDEN 2015

The Impact of Imbalanced Training Data for Convolutional Neural Networks

PAULINA HENSMAN AND DAVID MASKO

KTH ROYAL INSTITUTE OF TECHNOLOGY

CSC SCHOOL

KTH Computer Science
and Communication

The Impact of Imbalanced Training Data for Convolutional Neural Networks

Paulina Hensman
David Masko

Degree Project in Computer Science, DD143X
Supervisor: Pawel Herman
Examinator: Örjan Ekeberg

May 8, 2015

Abstract

This thesis empirically studies the impact of imbalanced training data on Convolutional Neural Network (CNN) performance in image classification. Images from the CIFAR-10 dataset, a set containing 60 000 images of 10 different classes, are used to create training sets with different distributions between the classes. For example, some sets contain a disproportionately large amount of images of one class, and others contain very few images of one class. These training sets are used to train a CNN, and the networks' classification performance is measured for each training set. The results show that imbalanced training data can potentially have a severely negative impact on overall performance in CNN, and that balanced training data yields the best results. Following this, oversampling is used on the imbalanced training sets to increase the performances to that of the balanced set. It is concluded that oversampling is a viable way to counter the impact of imbalances in the training data.

Abstract in Swedish

Detta kandidatexamensarbete utför en empirisk studie av den påverkan ojämnt fördelad träningsdata har på bildklassificeringsresultat för *Convolutional Neural Networks*(CNN). Bilder från datamängden CIFAR-10, bestående av 60 000 bilder fördelade mellan 10 klasser, används för att skapa träningsdatamängder med olika fördelningar mellan klasserna. Exempelvis innehåller vissa mängder oproportioneligt många bilder av en klass, medan andra innehåller väldigt få bilder av en klass. Dessa datamängder används för att träna ett CNN, och nätverkets klassificeringsresultat noteras för varje datamängd. Resultaten visar att ojämt fördelad träningsdata kan ha en markant negativ påverkan på de genomsnittliga resultaten för CNN, och att balanserad träningsdata ger bäst resultat. *Oversampling* används på de ojämnt fördeladade träningsdatamängderna vilket resulterar i samma resultat som för den balanserade träningsdatamängden. Detta visar att *oversampling* är ett gångbart sätt att motverka effekterna av ojämnt fördelad träningsdata.

Contents

1 Introduction

Over the past few years Artificial Neural Networks (ANN) have received major attention due to breakthroughs in several fields, such as computer vision[1], voice recognition[2] and natural language processing[3]. With statistical methods[4] these networks are able to approximate underlying functions and patterns in large amounts of data without any prior knowledge or assumptions about it.

Two special types of ANN known as Deep Neural Network (DNN) and Convolutional Neural Network (CNN) are today the state-of-the-art approach to solving several complex problems. One of these problems is *image classification*[1], the task of identifying which class an image belongs to given a number of options. Image classification is useful in several contexts, among them optical character recognition (commonly known as OCR)[5], image search, and self-driving cars[6]. DNN and CNN currently outperform all of the previous machine learning approaches to this problem[7]. The downside of these networks is that in order for them to be trained to a satisfying level, a lot of data is required[8]. Image classification using CNN requires labeled images, and as the labeling has to be done by humans to be reliable, acquiring the data is hard.

Several labeled image datasets are publicly available to provide researchers and machine learning practitioners benchmarking resources[5][9][10][11]. This base of benchmarking sets enables comparison[11] between image classification methods and is used to prove progress in the field[1]. The favored datasets for single class classification benchmarking, for example MNIST[5] and CIFAR[9], contain the same amount of images for each class. This is known as *balanced* or *even* datasets[12]. Balanced datasets have been empirically shown to outperform *imbalanced* datasets[13][14][15]. However, in most real-life situations, the available datasets are imbalanced.

Dealing with imbalanced data is a well known challenge in machine learning, and several methods to lessen the impact of imbalanced datasets exist[13][14][15]. A simple method is *oversampling*, duplicating instances of under-represented classes until a balanced dataset is created[13][14]. Although such methods are known to perform well in machine learning algorithms[13][14] there is no research on the effects of imbalanced data on DNN and CNN with the benchmarking datasets.

1.1 Problem statement

The aim of this thesis is to approximate the loss (or possibly the increase) in CNN image classification performance due to imbalanced distribution in training data. This will give insight into what types of distributions cause underperforming, and how successful oversampling is in increasing the performance. The thesis aims to investigate the following:

- How important is a balanced distribution in training sets for CNN?
- How is the CNN performance affected by different distributions in training data?
- Can performance be improved by adjusting the distribution in training data, and what methods are available for adjusting it?

1.2 Scope

A single dataset will be used to create subsets with different distributions of data between the classes, and these will be used to train a CNN. The limitation of one dataset avoids performance differences due to varying suitability between CNN configurations and datasets. The aim of this thesis is not to reach high classification performance, but to compare performances between networks trained with differently distributed datasets. Thus, only a single simple CNN implementation will be used.

It is expected that training a CNN with a balanced dataset will lead to the best performance, compared to CNN trained with other distributions with the same total amount of training data. Oversampling the classes with less data in imbalanced datasets is expected to increase the overall performance of the network, but it will not reach the performance of a CNN trained with an originally balanced dataset.

1.3 Thesis overview

Section 2 introduces image classification and how imbalanced data can affect the performance of image classifiers. It talks about the principles of ANN and describes some notable implementations, and mentions a number of popular datasets and related work. In section 3, the choices of dataset, distributions and network parameters are motivated, and the testing procedure is explained. In section 4, the results are presented. In section 5, the results are analyzed, limitations are discussed, and a conclusion is presented.

2 Background

This section introduces image classification and the effect of imbalanced data on the performance of machine learning algorithms and ANN. Some solutions to improve the performance with imbalanced data will be presented followed by an explanation of the state-of-the-art ANN approaches used in image classification. Finally, multiple datasets used to benchmark image classification algorithms will be presented, as well as some related work.

2.1 Image classification

Image classification is the process of selecting which class a given image belongs to, i.e. what objects an image contains. When classifying images, there are two categories of annotations. *Image-level annotation* is a binary label saying whether or not an object class exists in an image, e.g "this image contains a cat". *Object-level annotation* is specific as to where in an image a certain object can be found, e.g. “there is a screwdriver centered at position (20,25) with width of 50 pixels and height of 30 pixels”[7]. This thesis will focus on image-level annotations.

2.2 Classification with imbalanced data

Imbalanced data means that the data used in machine learning training has an imbalanced distribution between the different classes. Imbalanced data poses a challenge in classification problems, since algorithms trained with balanced datasets surpass those trained with imbalanced datasets in performance[13][14][15]. In practice, the available data is often imbalanced[14]. However, most machine learning algorithms assume a balanced distribution or the same distribution of classes in new, unlabeled data as in the known training data[15]. Such algorithms underperform if the training data does not have the same distribution as the unknown data that needs to be classified[13]. Furthermore, most machine learning algorithms aim to minimize the overall error rate which results in worse performance for the classes that are under-represented in the training data[13]. This can have a very negative impact if the rare classes are of importance, for example in rare disease diagnostics[14]. However, imbalanced data has received a great deal of research interest[13] and there are many successful methods of countering it.

2.2.1 Improving performance of imbalanced data

Solutions to the imbalance problem aim to reduce the bias towards the plentiful classes and can be divided into three categories; *sampling techniques*, *cost sensitive techniques*, and *one-class learning*[13][14].

Sampling techniques create balanced distributions by altering the original dataset[13]. Simple sampling techniques include duplicating instances from the minority classes until a balanced distribution is reached (*oversampling*) or removing instances from over-represented classes (*undersampling*)[13][14]. However, it is suggested that a combination is the best solution for extremely imbalanced distributions[14]. Advanced sampling techniques consist of generating new data in minority classes based on the current data[13].

Cost sensitive techniques address the learning itself, keeping the original dataset intact. These approaches increase the bias towards the minority-classes[13]. For example, a higher penalty can be given to the network when it misclassifies the minority classes during training, thus encouraging it to learn those classes better[14].

One-class learning only provides training data from a single class, aiming to provide a tight boundary for the class, serving as anomaly detection[13].

2.3 Artificial neural networks

This section will introduce the principles of ANN and two specialized implementations, CNN and DNN. It will also talk about Convolutional Deep Neural Networks (CDBN), a combination of CNN and DNN.

2.3.1 Principles of artificial neural networks

ANN is an attempt to imitate the behaviour of the information processing in the biological nervous system, and use it to solve various problems. The basics of this structure is creating an interconnected network of neurons and process input by sending it through the network. This is similar to the impulses travelling between the millions of synapses in the biological nervous system. The neurons in an ANN are connected with weights, which set the amount of influence the output of a neuron has on another neuron. The structures and paradigms of ANN are many but share the common factor of being only partially pre-determined. The networks are trained with a selection of learning algorithms, which are automatically adaptive in nature rather than explicitly programmed. This makes ANN suitable for machine learning and pattern recognition. A key characteristic in an ANN is defined by whether the output signal from a neuron will affect its input, hence creating a recursive effect. Such networks are called recurrent in contrast to feed-forward networks that lack connection loops.[4]

A feed-forward network is divided into multiple layers (Figure 1). Each layer is computed by applying the weights to the output of the previous layer. The data passes through the network layerwise in a defined order. The first layer is always the input layer, which takes the input data. The final layer is thus the output layer. There may be several layers between the input and the output layers. However, most learning algorithms have difficulty training a network with a lot of layers[16][4].

The learning of an ANN is generally either supervised or unsupervised. The difference between these two paradigms is that supervised training provides the network with a pair of input data and output data, called labelled data. With the given data,

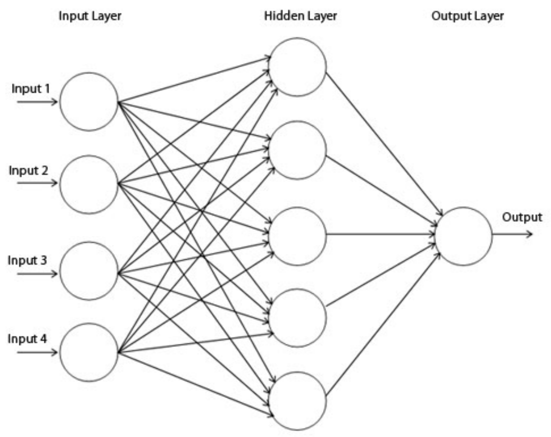

Figure 1: An example of a feed forward network with 2 layers and 1 output node. Source: www.codeproject.com[17]

the network has to change the weights so that the input in the training set will provide the given output after passing through the network. Unsupervised learning on the other hand only provides the input, and the output is unknown, unlabelled data. In the unsupervised case, the ANN has to try to find some hidden structure in the data without getting negative nor positive feedback on its progress[4].

2.3.2 Convolutional neural networks

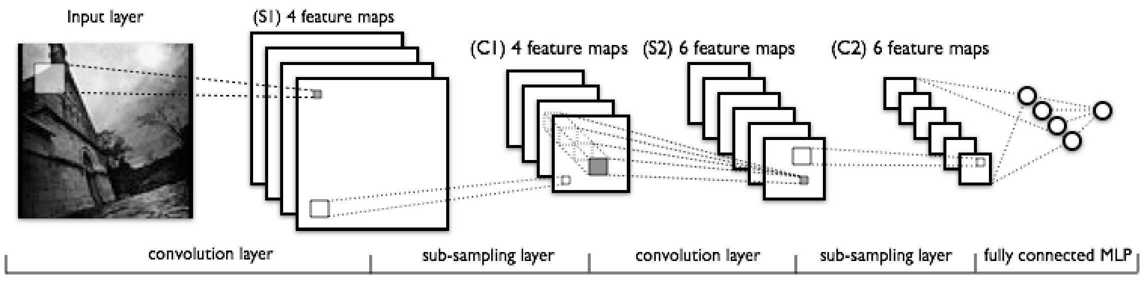

Figure 2: A visual representation of CNN
Source: deeplearning.net[18]

The CNN is a specialized feed-forward network that is a state-of-the-art model for image classification[19][20][11]. The structure is inspired by the stimulus processing in the biological brain's primary visual cortex[20]. The inspiration lies within the addition of convolutional layers doing subsampling of data that combines continuous areas of adjacent pixels into single values[21]. This subsampling makes up the output of each convolutional layer as seen in Figure 2.

The layer structure of a CNN is one, or several convolutional layers in succession, followed by fully connected regular ANN layers. This structure is visualized in Figure 2. The convolutional layers act as feature extraction layers and the last fully connected ANN layers act as the classification module. The convolutional layers consist of three different layers; the *filter bank layer*, the *non-linearity layer*, and the *feature pooling layer*[22].

The input and output of each convolutional layer is a 3D matrix where the first two dimensions are the image height and width, and the third is the number of feature maps. The feature maps for an RGB image would for example be three; one for each color channel. Thus, such an image would be represented with three separate images in the initial input, each with a single unique color channel[23]. The output feature maps are undefined in nature and depend on the training. The deeper the network, the higher the level of features in the output[22].

The *filter bank layer* is constructed of several trainable kernels that each recognize a particular feature. This feature extraction is done at every location of the input, and thus the convolutional layer is not sensitive as to where a feature is located in the input image.

The *non-linearity layer* will apply a non-linear sigmoid transfer function on the entire output from the filter bank layer. This transfer function comes in a number of variants that have shown prowess in different ways, for example by creating local competition between adjacent features. [22]

The *feature pooling layer* is the layer that performs the sub-sampling of the convolutional layer. This sub-sampling is done for each feature map and results in a lower resolution representation of the map, but with increased noise resistance. The sub-sampling could for example be done by setting each value in the feature map as the average of its neighbours within a certain range. [22]

After the final convolutional layer the output will be passed to the classification module. The classification module consists of several fully connected layers. These layers result in an output specifying the class, or classes, contained within the image[22].

2.3.3 Deep learning networks

DNN is a feed-forward type of ANN that solves the efficiency difficulties that arise when applying learning on a feed-forward network with a large amount of layers[21][16]. Its significance is due to the fact that adding a layer with N nodes results in better ability to create complex approximations than just adding N nodes to an existing layer[24]. The learning is performed greedily on each set of weights between each layer using the Restricted Boltzmann Machine algorithm (RBM). This implementation of the RBM finds the maximum likelihood for each weight in the connections between two bipartite layers given the input [21][16]. RBM can be used on DNN since DNN are feed-forward networks and will always fulfill the requirement of a bipartite graph[16].

Deep learning networks benefit from being able to be trained in an unsupervised manner during a pre-training stage followed by supervised training. Combining these learning paradigms has been shown to improve the results of DNN[8][25]. A problem with supervised learning is that it is subject to overfitting - a phenomenon that occurs when a network becomes too niche in relation to its learning data causing it to under-perform on unlabeled data after the training[4]. By introducing unsupervised training prior to supervised training, the issue of overfitting is reduced. This is a result of supervised learning being amenable to slightly overfit supervised data introduced early in the training[8]. This suggests that this network type inherently improves performance of imbalanced data.

2.3.4 Convolutional deep neural networks

The CDBN is a combination of the CNN and the DNN. The CDBN combines the stacked convolutional layers with subsampling and the greedy RBM training approach from DNN[26]. This is achieved through an altered RBM, a convolutional RBM (CRBM). The alteration of the CRBM is that instead of evaluating each weight separately the weights are shared within different groups, resembling the subsampling of CNN [26].

Since the CDBN is trained in a similar manner as DNN it is possible to train a CDBN with unsupervised data prior to the labeled training. The unlabeled data has also been shown not to be restricted to the classes of the actual labeled training data[26]. CDBN has been successful in almost replicating the performance of the state-of-the-art CNN implementations with a lot of available labeled data. Additionally, CDBN has been shown to surpass the performance of such implementations when the amount of labeled data is restricted[26].

2.4 Datasets and benchmarks

There are several datasets available to benchmark image classification implementations. These datasets consist of a combination of labeled and unlabeled images of various quantities. This section will cover the most popular and recognized datasets.

2.4.1 ImageNet

Figure 3: Example images from ImageNet 2011 Fall Release
Source: image-net.org [27][28][29]

The current standard image classification and object detection benchmark is the ImageNet Large Scale Visual Recognition Challenge, which has been running annually since 2010[10]. The participants of the challenge receive a large number (over a million) of manually annotated images as a training set, as well as a test set without annotations. Some image examples are shown in Figure 3. After training their algorithms with the training set, they let the algorithms annotate the unlabeled images and send them in for evaluation[10].

The year 2012 marked a turning point in the challenge results, as large-scale CNN was introduced. This approach won both tasks of the 2012 challenge with a big margin, with an error rate of 15.3% compared to 26.2% achieved by the second best approach, which used Fisher vectors [30]. Due to the success of the CNN in the 2012 challenge, in 2013 the vast majority of the teams used CNN[10]. There has been a lot of improvement over the years of the challenge. For Image Classification, the error rate of the winner has dropped from 28.2% in 2010 to 6.7% in 2014[10].

Data from previous years is available, and researchers may ask for access to the material and also test their networks and algorithms on the automated judge system[10].

2.4.2 MNIST

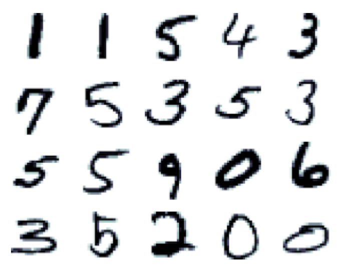

Figure 4: Example digits from MNIST
Source: Classification Datasets Results [11]

The Mixed National Institute of Standards and Technology (MNIST) database is a large database of 70 000 handwritten digits (Figure 4). MNIST is a mixture of subsets from the different National Institute of Standards and Technology (NIST) datasets that were collected from high school students and Census Bureau employees that have also been normalized to a 20x20 size[5]. The MNIST dataset has been widely used to benchmark image classification machine learning algorithms[5][11] with a lowest error rate of 0.21% achieved in 2013 by Li Wan et al[31].

2.4.3 Tiny images dataset

The Tiny Image dataset is a collection of almost 80 million 32x32 images of various kinds. These images consist of varying labeling quality from correctly labeled, wrongly labeled, to not labeled at all[32]. For example images, see Figure 5.

2.4.4 CIFAR-10 and CIFAR-100

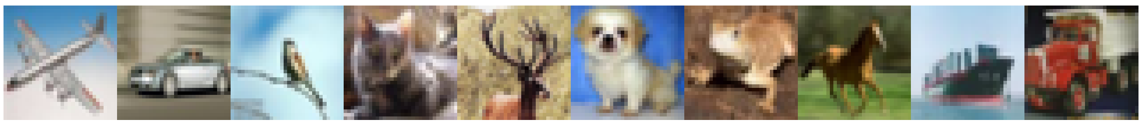

Figure 5: Example images from CIFAR-10, one from each class
Source: CIFAR-10 [33]

The CIFAR datasets are subsets of 60 000 images from the Tiny Image dataset with 10 different classes in the CIFAR-10 set (Figure 5), and 100 classes in the CIFAR-100 set[9]. The subject may not be centered in the image, it may only be partial within the image, and the backgrounds vary[20]. The selected classes in CIFAR-10 are mutually exclusive, so there is no overlap between the classes[9]. The challenge of classifying the CIFAR-10 dataset has been attempted by a human who reached 94% accuracy[34]. The current best result by machine learning was achieved by Chen-Yu Lee et al in 2014 with an accuracy of 91.78% [1]. Lee et al also hold the best score of CIFAR-100, with the same implementation as for CIFAR-10, with 65.43% accuracy[1].

2.5 Related work

In 2003, Weiss and Provost[35] empirically analyzed the best distributions for training data when some data had to be removed due to a restriction on the data amount. This study compared subsets of 26 natural imbalanced datasets. The study showed that is was best to remove data from the over-represented classes, giving the resulting set a balanced distribution. Retaining the original distribution when removing data also gave a good performance. The study also introduced their own ”budget-sensitive” progressive sampling algorithm that showed the best results[35].

In 2013, Song, Morency, and Davis showed the negative effect of imbalanced training data using datasets named MSRC-12 and NATOP[12]. The balanced distribution yielded the best performance. The study successfully increased the performance of the imbalanced datasets with a combination of a cost-sensitive technique and undersampling, but did not reach a performance above the original balanced distribution[12].

3 Methods

A single dataset was selected and partitioned into subsets with different artificially created distributions, in order to measure the impact of differently distributed training data on CNN performance. Caffe[36] was used as the CNN library to conduct performance tests on each subset several times to gauge the average performance. Oversampling was performed on the subsets with imbalanced distributions to evaluate and compare the effectiveness of oversampling on different distributions.

3.1 Dataset

The CIFAR-10 dataset was used for the experiments, as its advantages were several. It is limited to only 10 classes with 6000 images each. This made it possible to test multiple subsets with imbalanced distributions of images from each class in isolation. Imbalanced distributions would have been difficult to obtain from CIFAR-100, as it only contains 600 images of each class. Simple implementations for CNN classifying CIFAR-10 already existed, providing suitable network parameters[37]. The network was also able to be trained at a fast rate due to the small image sizes. This was an important consideration due to limited available computational power and amount of different subsets.

The ImageNet database was discarded mainly due to the larger image sizes. ImageNet also provides thousands of classes, while CIFAR-10 consists of a small mutually exclusive selection of classes, making CIFAR-10 more accessible. The Tiny Images database was discarded as an option due to lack of organized structure and inconsistent label quality.

The MNIST dataset was a possible choice fulfilling the criteria above. However, CIFAR-10 consists of images from a variety of angles, MNIST does not. It was concluded that the CIFAR-10 dataset provided a more complex classification task.

Using multiple datasets with different original distributions was left out due to the risk of introducing differences in performance due to the difference between the datasets themselves. By using a single dataset, all images were of the same size and the network parameters did not favor any particular dataset. By restricting the tests to only one dataset any differences in performance were only due to the distributions themselves.

3.2 Subsets

The training set of CIFAR-10, consisting of 5 000 images per category, was partitioned into different subsets. The test set with 1 000 images of each category remained unchanged throughout all tests. Thus, all references to subsets are referring to the subsets of the training data.

The selection of distributions for the subsets was arbitrary. Distributions with extreme imbalances were excluded, as they would limit the total subset size too much. The subsets contained 65% of the images in the original CIFAR-10 set. Higher percentages were not possible as certain amounts of data needs to be removed to achieve the imbalanced distributions, and smaller percentages were excluded since such subsets would not train the network sufficiently. The distributions were selected to be as mutually exclusive from each other as possible.

3.2.1 Distributions

The distributions were as presented in Table 1.

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
Dist. 1	10	10	10	10	10	10	10	10	10	10
Dist. 2	8	8	8	8	8	12	12	12	12	12
Dist. 3	6	6	6	6	6	14	14	14	14	14
Dist. 4	12.25	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75	9.75
Dist. 5	14.5	9.5	9.5	9.5	9.5	9.5	9.5	9.5	9.5	9.5
Dist. 6	7.3	10.3	10.3	10.3	10.3	10.3	10.3	10.3	10.3	10.3
Dist. 7	6.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4
Dist. 8	8.24	8.63	9.02	9.41	9.80	10.20	10.5	10.99	11.37	11.76
Dist. 9	7.58	7.68	7.91	8.29	8.83	9.57	10.52	11.70	13.11	14.79
Dist. 10	12.5	12.5	8.33	8.33	8.33	8.33	8.33	8.33	12.5	12.5
Dist. 11	15.22	15.22	6.52	6.52	6.52	6.52	6.52	6.52	15.22	15.22

Table 1: A table over the different distributions. For each of the 11 distributions it shows how many percent of the set is made up by each CIFAR-10 class. For example, in Dist. 4, airplanes make up 12.25% of the total set.

The following list explains each distribution from Table 1.

1. This is the balanced distribution and was included as a benchmark since it should yield the best preformance[13][14].
2. A minor 50-50 split. This distribution is interesting since it divides the classes into two. This was included to see how much the minority classes would underperform.
3. A major 50-50 split. The difference between a lower and higher imbalance had potential to show the fragility of the 50-50 distribution.
4. A minor singular over-representation. This was a good comparison to the 50-50 split, examining what the effect would be if only a single class was in majority. It was also an interesting distribution to apply oversampling on.
5. A major singular over-representation.
6. A minor singular under-representation. This was a distribution that was expected to have a smaller impact on the performance since only one class was affected negatively.

7. A major singular under-representation. Due to the clear minority of a single class, this distribution was expected to show the limits of oversampling improvement.
8. A linear imbalance included due to the smooth transition. The previous distributions were discretely separated, this distribution was chosen to represent natural smooth imbalance.
9. An exponential imbalance included to contrast the linear transition and represent a major imbalance.
10. A minor split between the animal classes and the vehicle classes. A similar case to distribution 2, but it was more intuitive that similar classes share the same quantity. Earlier research has shown that when a CNN makes mistakes in CIFAR-10 classification, it usually mistakes an animal for another animal or a vehicle for another vehicle[38].
11. A major split between the animal classes and the vehicle classes.

3.2.2 Partitioning CIFAR-10

The CIFAR-10 dataset was partitioned into subsets by for each distribution parsing through the entire dataset and selecting the first N occurrences of every class, where N is the wanted number of images from that class to achieve that distribution. Since there was no simple way to determine what images were better or worse than others in the dataset, a more advanced selection method was deemed unnecessary.

3.2.3 Oversampling CIFAR-10 subsets

To balance the data, oversampling was performed on the under-represented classes in the subsets. Random images of each under-represented class were duplicated until all classes had the same amount of images as the largest class.

3.3 CNN

3.3.1 Framework

The neural network framework Caffe[36] was used to create and train a CNN. Caffe is an open source framework developed by the Berkeley Vision and Learning Center and and by community contributors. It is both very fast and simple to use, which made it a good fit for the tests.

3.3.2 Parameters

The network used in the experiments was one of Alex Krizhevsky's setups for CIFAR-10 classification[37]. They are commonly used network setups where a simple and straightforward setup is required. The fastest one was picked because it would allow time to run more tests, and as the goal was not to reach high performance but to compare performances between networks trained with different subsets. The network had 3 convolutional layers and 10 output nodes. It was trained with learning rate 0.001 for 8 epochs, and then with learning rate 0.0001 for 2 epochs The momentum was set to 0.9 and weight decay to 0.004 [39][40].

3.3.3 Testing method

For each subset, the network went through training with the parameters explained in 3.3.2 three times, with performance being tested after each time. It was tested on the test data from each of the classes separately. The mean results of the three runs were recorded, and the total mean and variance for the entire test set was calculated. Hypothesis testing was used when comparing the total means of the imbalanced subsets to the mean of the balanced subset.

3.4 Definition of performance

The performance was defined by the percentage of correct answers the CNN produced during testing. A correct answer means the network classified an image as the class it belonged to. The performance was measured for each individual category and a mean result for the entire subset was calculated.

4 Results

4.1 Distribution performance

Table 2 shows the results of each distribution for each class. The performances are presented in decimal form, e.g. Table 2 shows that the CNN trained with subsets created with distribution 4 made a correct classification on 24% of the automobiles in the CIFAR-10 test set on average. The mean performance on the complete CIFAR-10 test set is presented in the *Total* column.

	Total	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Dist. 1	0.73	0.78	0.84	0.62	0.57	0.70	0.62	0.80	0.76	0.84	0.80
Dist. 2	0.69	0.74	0.75	0.58	0.33	0.58	0.65	0.84	0.78	0.87	0.79
Dist. 3	0.66	0.71	0.75	0.59	0.30	0.52	0.61	0.79	0.77	0.85	0.73
Dist. 4	0.27	0.78	0.24	0.12	0.08	0.19	0.24	0.33	0.27	0.21	0.27
Dist. 5	0.10	0.999	0	0.002	0.0003	0	0	0.001	0	0	0
Dist. 6	0.73	0.74	0.86	0.65	0.53	0.71	0.63	0.81	0.76	0.83	0.79
Dist. 7	0.73	0.75	0.86	0.66	0.52	0.71	0.63	0.80	0.78	0.84	0.79
Dist. 8	0.66	0.63	0.75	0.55	0.35	0.51	0.58	0.82	0.74	0.84	0.80
Dist. 9	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
Dist. 10	0.69	0.75	0.77	0.56	0.42	0.66	0.63	0.76	0.70	0.81	0.79
Dist. 11	0.69	0.74	0.82	0.58	0.44	0.59	0.64	0.80	0.69	0.83	0.81

Table 2: The total and the individual class performance ratios.

The balanced distribution had the best overall performance, as shown in distribution 1 in Table 2. Distributions 4, 5, and 9 underperformed severely. Distribution 2, 3, and 8 perform under the median. All distributions gave worse total performances compared to the balanced distribution at a significance level of 0.1%, except distributions 6 and 7 that did not give a significantly worse performance. The *cat* class has the worst results in all distributions. *Automobile* followed by *frog* are the best performing classes overall. The very best individual class performances are found in the distributions that also contain the worst individual performances, e.g. in distribution 5. The balanced distribution has the highest minimum performance for all individual classes.

4.2 Oversampling performance

	Total	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Dist. 2	0.72	0.77	0.80	0.57	0.51	0.68	0.64	0.81	0.78	0.84	0.82
Dist. 3	0.73	0.73	0.80	0.59	0.53	0.63	0.65	0.82	0.81	0.87	0.83
Dist. 4	0.73	0.76	0.82	0.60	0.54	0.68	0.63	0.81	0.78	0.85	0.83
Dist. 5	0.73	0.80	0.84	0.61	0.55	0.68	0.63	0.82	0.79	0.82	0.81
Dist. 6	0.73	0.75	0.86	0.65	0.51	0.65	0.66	0.81	0.78	0.85	0.80
Dist. 7	0.73	0.73	0.85	0.62	0.51	0.71	0.66	0.81	0.79	0.86	0.80
Dist. 8	0.73	0.78	0.84	0.62	0.56	0.66	0.64	0.81	0.77	0.83	0.80
Dist. 9	0.72	0.73	0.84	0.58	0.55	0.68	0.59	0.79	0.78	0.83	0.82
Dist. 10	0.73	0.78	0.85	0.63	0.50	0.67	0.63	0.82	0.75	0.87	0.80
Dist. 11	0.72	0.82	0.87	0.58	0.64	0.64	0.49	0.78	0.69	0.84	0.80

Table 3: The total and the individual class performance ratios with oversampling. Dist. 1 is not included since it was already balanced and thus was not oversampled.

All imbalanced subsets showed improved performance after oversampling. The oversampled subsets showed very similar mean performances and individual class performances, as seen in Table 3. The performances are presented in decimal form, e.g. Table 3 shows that the CNN trained with the oversampled subset created from distribution 4 made a correct classification on 82% of the automobiles in the CIFAR-10 test set. The mean performance on the complete CIFAR-10 test set is presented in the *Total* column. Figure 6 illustrates a comparison between the performance of the original subsets and their oversampled counterparts. The performances reached by the oversampled subsets show no relation with the original sets' performance.

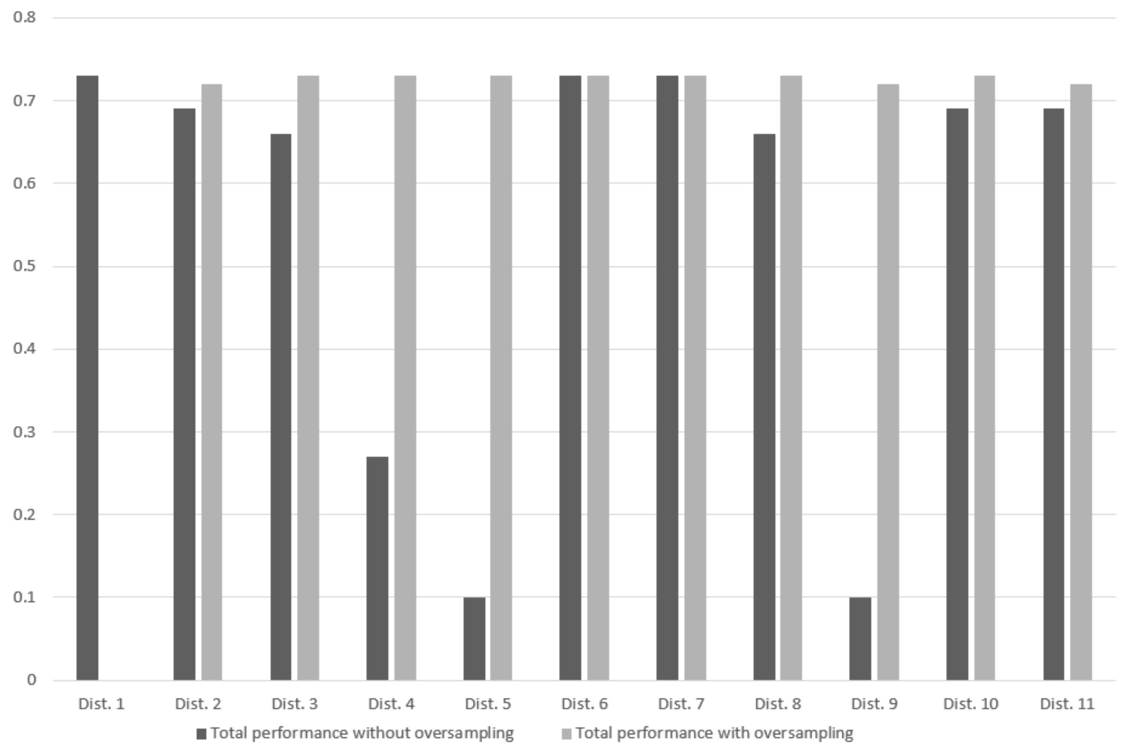

Figure 6: The total performance ratio of each distribution with and without oversampling from Table 2 and Table 3. Dist. 1 shows no result after oversampling since it was already balanced.

5 Discussion

5.1 Results analysis

The results show that the distribution of training data has significant impact on the performance of CNN. The balanced distribution yielded the best performance as suggested by earlier research. All distributions except 6 and 7 gave a worse performance than the balanced distribution on a statistically significant level. As seen in Table 2 the heavier the imbalance, the worse the total performance. The most imbalanced distributions 5 and 9, the major single-class overload and the exponential set, had such a negative impact that the CNN only guessed the majority class repeatedly. As the over-represented classes only made up 14.5% and 14.79% of the entire set in distribution 5 and 9 respectively, the fragility when using imbalanced data in the CNN training algorithm is clearly shown.

Comparing distributions 4 and 5 and distributions 8 and 9 shows that reduced imbalance yields a better performance. Distribution 4, where the over-represented class made up 12.25% of the set gave almost three times better performance than distribution 5, and the performance yielded by distribution 8 was close to the mean. The 50-50 split distributions gave a performance only slightly worse than the balanced distribution, suggesting that single class imbalances are the worst case. However, the under-represented class distributions 6 and 7 gave a performance on par with the performance from the balanced distribution, which suggests that imbalances in a single class only affects overall performance if that class is over-represented.

A relationship between the class representation and the performance for each class can be seen in distribution 4 and 8 by comparing Table 1 and Table 2. Such a relationship is not visible in distributions with higher total performance. This suggests that the imbalanced distributions gave a worse performance due to the difference in distribution between training and test data, as has been shown in earlier research. Since CNN is based on statistical models, this result was expected.

The oversampling technique proved effective. All oversampled subsets yielded a performance similar to the balanced subset as seen in seen in Figure 6. The results do not show any major differences in performance between the subsets in the total or for any individual class (Table 3) and this suggests that the oversampling is not overly dependent on the original distribution. The unanimous performance results suggest that the distribution of training data is more important than the number of unique instances in training data for CNN; although several images were duplicates the oversampled subsets yielded a performance similar to that of the balanced subset. This supports the conclusion that the original data distribution is of less importance if oversampling is used.

5.2 Limitations

The results of this study add to previous empirical results[12][13][14][15][35] showing that balanced distributions yield the best performance. This study was restricted to balanced test data and performance was defined as the overall performance, with the effect that the results of this study will not be applicable in situations where some classes are more important than others, such as positive cases in rare disease diagnosis. By using balanced test data across all tests, it is assumed that unknown data is balanced. This means the tests conducted in this study do not indicate a general direction for distributions in all situations. However, this study provides incentive to have balanced training data if the unknown data is expected to be balanced, or if maximum overall performance is sought.

Choosing the CIFAR-10 dataset introduced restrictions of available tests. Having a limited amount of available images for each class limited the tests to only moderately imbalanced subsets. CIFAR-10 also removed the possibility of testing the distributions over a large number of classes. The study is thus unable to claim that some distributions always yield a better performance, only to suggest it. Since only the CIFAR-10 dataset was analyzed it is not certain that the findings are universal, although it is unlikely that CIFAR-10 would be a unique occurrence and we expect that similar results would be found in other datasets. CIFAR-10 does suffer from not being truly mutually exclusive; e.g. dogs and cats have several similarities. This means that some distributions may perform differently depending on different orderings of the classes. However, this study assumes that these differences are of less impact than the distributions themselves.

The same network parameters for the CNN were used for all tests, and different results may have been obtained with other parameters. A more advanced CNN may have countered the performance drops caused by the imbalanced data. However, the selected network parameters are commonly used and have a high performance considering the simplicity and as the scope of the study was on relative performance, conducting the tests on a more advanced network was not considered valuable.

5.3 Future research

Since no subsets underperformed on the test data after oversampling, future research could look into whether even larger imbalances reduce the effect of oversampling and approximate at what level of imbalance oversampling is no longer enough to improve the results. Furthermore, as this study only conducted tests with a single quantity of images, finding similar results on larger quantities would strengthen the results from this study. Repeating the successful results from this thesis on other datasets would suggest that CNN is a preferred ANN implementation for imbalanced subsets combined with oversampling techniques.

Both CNN and DNN are state-of-the-art in image recognition, but this study only performed tests on CNN. The same tests with CIFAR-10 and oversampling could be conducted with DNN, more advanced ANN types such as the CDBN, or more advanced CNN implementations.

The good overall results of the distributions 10 and 11 with different quantities for animals and vehicles suggest that the performance from imbalanced sets also depends on the similarities of the classes. This relationship would be interesting to examine in future research.

5.4 Conclusion

The results show that the distribution of the training data has a big impact on CNN performance. As expected, a balanced training set was optimal. A relationship was found between larger imbalances and worse performances, but only when some classes were over-represented. Under-representations of single classes showed no significant impact on the overall performance. It is noted that balanced distributions in training data are most relevant when the unknown data is expected to be balanced or when a high overall performance is sought. Using the sampling technique oversampling on the imbalanced data increased the CNN performances to that of the CNN trained with balanced data. This makes oversampling a promising method of countering imbalanced data. However, these experiments need to be repeated with other datasets than CIFAR-10 for confirmation.

References

[1] Chen-Yu Lee et al. “Deeply-Supervised Nets.” In: *CoRR*. 2014. URL: http://arxiv.org/abs/1409.5185.

[2] Frank Seide, Gang Li, and Dong Yu. “Conversational Speech Transcription Using Context-Dependent Deep Neural Networks”. In: *Interspeech 2011*. International Speech Communication Association, 2011. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=153169.

[3] Ronan Collobert and Jason Weston. “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Helsinki, Finland: ACM, 2008, pp. 160–167. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390177. URL: http://doi.acm.org/10.1145/1390156.1390177.

[4] R. Rojas. *Neural Networks: a Systematic Introduction*. Springer-Verlag, 1996.

[5] *THE MNIST DATABASE of handwritten digits*. http://yann.lecun.com/exdb/mnist/. Accessed: 2015-03-23.

[6] Ionut Alexandru Budisteanu. In: ().

[7] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. 2014. eprint: arXiv:1409.0575.

[8] Aaron Courville Pierre-Antoine Manzagol Pascal Vincent Samy Bengio Dumitru Erhan Yoshua Bengio. “Why does Unsupervised Pre-training Help Deep Learning?” In: *Journal of Machine Learning Research* (2010), pp. 625–660. URL: http://jmlr.csail.mit.edu/papers/v11/erhan10a.html.

[9] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[10] *ImageNet*. http://www.image-net.org/. Accessed: 2015-03-23.

[11] *What is the class of this image? Discover the current state of the art in objects classification*. http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130. Accessed: 2015-02-29.

[12] Morency Song and Davis. “Distribution-sensitive learning for imbalanced datasets.” In: *FG*. IEEE, 2013, pp. 1–6. ISBN: 978-1-4673-5544-5. URL: http://dblp.uni-trier.de/db/conf/fgr/fg2013.html#SongMD13.

[13] Haibo He and Edwardo A. Garcia. “Learning from Imbalanced Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. ISSN: 1041-4347. DOI: http://doi.ieeecomputersociety.org/10.1109/TKDE.2008.239.

[14] Chao Chen, Andy Liaw, and Leo Breiman. *Using Random Forest to Learn Imbalanced Data*. Tech. rep. Department of Statistics, University of Berkeley, 2004. URL: http://www.stat.berkeley.edu/users/chenchao/666.pdf.

[15] Slobodan Vucetic and Zoran Obradovic. “Classification on Data with Biased Class Distribution”. English. In: *Machine Learning: ECML 2001*. Ed. by Luc De Raedt and Peter Flach. Vol. 2167. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, pp. 527–538. ISBN: 978-3-540-42536-6. DOI:

10.1007/3-540-44795-4_45. URL: http://dx.doi.org/10.1007/3-540-44795-4_45.

[16] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Comput.* 18.7 (July 2006), pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. URL: http://dx.doi.org/10.1162/neco.2006.18.7.1527.

[17] *Financial Predictor via Neural Network.* http://www.codeproject.com/Articles/175777/Financial-predictor-via-neural-network. Accessed: 2015-03-23.

[18] *Convolutional Neural Networks (LeNet).* http://deeplearning.net/tutorial/lenet.html. Accessed: 2015-03-23.

[19] Rob Fergus Matthew D Zeiler. "Visualizing and Understanding Convolutional Networks". Version 3. In: *arXiv preprint arXiv:1311.2901v3* (2014).

[20] Jonathan Masci-Luca M. Gambardella Jurgen Schmidhuber Dan C. Cireşan Ueli Meier. "Flexible, high performance convolutional neural networks for image classification". In: *IJCAI'11 Proceedings of the Twenty-Second international joint conference on Artificial Intelligence* 2 (2011), pp. 1237–1242.

[21] Nicolas Le Roux and Yoshua Bengio. "Representational Power of Restricted Boltzmann Machines and Deep Belief Networks". In: *Neural Comput.* 20.6 (June 2008), pp. 1631–1649. ISSN: 0899-7667. DOI: 10.1162/neco.2008.04-07-510. URL: http://dx.doi.org/10.1162/neco.2008.04-07-510.

[22] Cĺement Farabet Yann LeCun Koray Kavukcuoglu. "Convolutional networks and applications in vision". In: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on* (2010), pp. 253–256.

[23] Chuan Yu Foo-Yifan Mai Caroline Suen Adam Coates Andrew Maas Awni Hannun Brody Huval Tao Wang Sameep Tandon Andrew Ng Jiquan Ngiam. *Convolutional Neural Network.* URL: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/ (visited on 03/01/2015).

[24] Merrick L. Furst, James B. Saxe, and Michael Sipser. "Parity, Circuits, and the Polynomial-Time Hierarchy". In: *FOCS*. IEEE Computer Society, 1981, pp. 260–270. URL: http://dblp.uni-trier.de/db/conf/focs/focs81.html#FurstSS81.

[25] Alexey Dosovitskiy et al. "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks". In: (2014). Ed. by Z. Ghahramani et al., pp. 766–774. URL: http://papers.nips.cc/paper/5548-discriminative-unsupervised-feature-learning-with-convolutional-neural-networks.pdf.

[26] Honglak Lee et al. "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations". In: *Proceedings of the 26th Annual International Conference on Machine Learning.* ICML '09. Montreal, Quebec, Canada: ACM, 2009, pp. 609–616. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553453. URL: http://doi.acm.org/10.1145/1553374.1553453.

[27] *ImageNet Tabby Cat.* http://farm1.static.flickr.com/111/316824947_b2da7a6e2a.jpg. Accessed: 2015-05-07.

[28] *ImageNet Sunflower*. http://farm2.static.flickr.com/1202/677592934_70f550f706.jpg. Accessed: 2015-05-07.

[29] *ImageNet Car*. http://farm4.static.flickr.com/3134/2735396242_616bef1619.jpg. Accessed: 2015-05-07.

[30] Jorge Sánchez, Florent Perronnin, and Zeynep Akata. “Fisher Vectors for Fine-Grained Visual Categorization”. In: *FGVC Workshop in IEEE Computer Vision and Pattern Recognition (CVPR)*. IEEE. Colorado Springs, United States, June 2011. URL: https://hal.inria.fr/hal-00817681.

[31] Li Wan et al. “Regularization of Neural Networks using DropConnect”. In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. Ed. by Sanjoy Dasgupta and David Mcallester. Vol. 28. 3. JMLR Workshop and Conference Proceedings, May 2013, pp. 1058–1066. URL: http://jmlr.org/proceedings/papers/v28/wan13.pdf.

[32] *Visual Dictionary*. http://groups.csail.mit.edu/vision/TinyImages/. Accessed: 2015-03-23.

[33] *CIFAR-10*. http://www.cs.toronto.edu/~kriz/cifar.html. Accessed: 2015-05-07.

[34] *Lessons learned from manually classifying CIFAR-10*. http://karpathy.github.io/2011/04/27/manually-classifying-cifar10/. Accessed: 2015-03-07.

[35] Gary M. Weiss and Foster Provost. “Learning when Training Data Are Costly: The Effect of Class Distribution on Tree Induction”. In: *J. Artif. Int. Res.* 19.1 (Oct. 2003), pp. 315–354. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=1622434.1622445.

[36] Yangqing Jia et al. “Caffe: Convolutional Architecture for Fast Feature Embedding”. In: *arXiv preprint arXiv:1408.5093* (2014).

[37] *cuda-convnet, High-performance C++/CUDA implementation of convolutional neural networks*. https://code.google.com/p/cuda-convnet/. Accessed: 2015-04-09.

[38] Hinton Ranzato Krizhevsky. “Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images”. In: (), p. 6. URL: http://www.cs.toronto.edu/~hinton/absps/ranzato_aistats2010.pdf.

[39] *26% error on CIFAR-10 in 80 seconds - layer definition file*. https://code.google.com/p/cuda-convnet/source/browse/trunk/example-layers/layers-80sec.cfg. Accessed: 2015-04-09.

[40] *26% error on CIFAR-10 in 80 seconds - layer parameter file*. https://code.google.com/p/cuda-convnet/source/browse/trunk/example-layers/layer-params-80sec.cfg. Accessed: 2015-04-09.