



DEGREE PROJECT IN COMPUTER ENGINEERING,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2019

Evaluating different training techniques for a convolutional neural network that classifies Alzheimer's disease

KARL LUNDSTIG

Evaluating different training techniques for a convolutional neural network that classifies Alzheimer's disease

KARL LUNDSTIG

Degree Project in Computer Science, DD142X

Date: June 7, 2019

Supervisor: Jeanette Hällgren Kotalleski

Examiner: Örjan Ekeberg

School of Electrical Engineering and Computer Science

Swedish title: Utvärdering av olika träningstekniker för ett konvolutionellt neuralt nätverk som klassificerar Alzheimers sjukdom

Abstract

Effective computer diagnosis of Alzheimer's disease could bring large benefits to the millions of people worldwide who does or will suffer from dementia. One popular method for trying to achieve this is the training of convolutional neural networks to classify MRI brain scans. An abundance of training methods exists that aims to improve the performance of these neural networks, but their effectiveness and eventual disadvantages are not always clear.

This study evaluates the performance of a CNN on the OASIS-3 neuroimaging dataset. The CNN's task is to classify each MRI scan into one of four classes which correspond to the stage of Alzheimer's disease. Two different training methods are compared to a training baseline. The first method evaluated is class-weighting, a technique that tries to compensate for rare classes in imbalanced datasets. The second method is data augmentation, a technique that extends the dataset in an attempt to reduce overfitting and increase performance.

Class-weighting was found to improve the classification performance significantly on the rarest class, while not having too large effect on the other classes. Data augmentation was not found to improve performance in general, but did improve the recall on some classes.

Sammanfattning

Effektiv datorbaserad diagnos av Alzheimers sjukdom skulle kunna medföra stora fördelar för de som lider av eller kommer drabbas av demens. En populär metod för att åstadkomma detta är tränandet av konvolutionella neurala nätverk (CNNs) som klassificerar magnetröntgenskanningar av hjärnan. Ett överflöd av träningsmetoder finns som försöker förbättra de här neurala nätverkens träffsäkerhet, men deras effektivitet och eventuella nackdelar är inte alltid tydliga.

Den här studen undersöker träffsäkerheten av ett CNN på hjärnavbildningsbanken OASIS-3. Nätverkets uppgift är att klassificera varje MRI-bild som en av fyra klasser, vilka motsvarar stadiet av Alzheimers sjukdom. Two olika träningsmetoder jämförs mot en grundmetod. Den första metoden som provas är klassviktning, en teknik som försöker kompensera för ovanliga klasser i en datamängd. Den andra metoden är datautökning, en teknik som utökar datamängden i ett försök att minska överträning och öka träffsäkerheten

Klassviktning visades tydligt öka klassificeringsträffsäkerheten på den ovanligaste klassen, utan att ha alltför stor effekt på säkerheten för de andra klasserna. Datautökning ökade inte träffsäkerheten generallt, men förbättrade täckningen på vissa klasser.

Contents

1	Introduction	1
1.1	Research Question	2
2	Background	3
2.1	Alzheimer's Disease	3
2.1.1	Benefits of early detection	3
2.1.2	Brain changes	4
2.1.3	Clinical dementia rating	4
2.2	Artificial neural networks	5
2.2.1	Structure	5
2.2.2	Activation functions	6
2.2.3	Error functions	7
2.2.4	Gradient descent	8
2.2.5	Learning rate	9
2.3	Convolutional neural networks	9
2.3.1	Motivation	9
2.3.2	Structure	9
2.3.3	Convolutional layers	10
2.3.4	Pooling layers	11
2.3.5	Fully-connected layers	11
2.3.6	Common architectures	11
2.4	Evaluation of classification systems	12
2.5	Previous work	13
2.5.1	Training CNNs on medical images	13
2.5.2	Handling imbalanced datasets	14
2.5.3	Classification of Alzheimer's disease	14
3	Methods	16
3.1	Training methods	16

3.1.1	Baseline	16
3.1.2	Class-weighting	17
3.1.3	Data augmentation	17
3.2	OASIS-3 dataset	17
3.3	Labeling	18
3.4	Image extraction	20
3.5	Hyperparameters	20
3.6	Network structure	20
4	Results	21
4.1	Baseline	21
4.2	Class-weighted training	23
4.3	Training with data augmentation	25
4.4	Summary of training results	27
5	Discussion	29
5.1	Method	29
5.2	Results	30
5.2.1	Baseline training	30
5.2.2	Class-weighted training	30
5.2.3	Training with data augmentation	31
5.3	Future work	32
6	Conclusions	33
7	Acknowledgments and source code	35
	Bibliography	36

Chapter 1

Introduction

In the last couple of years machine learning, and more specifically deep learning, have been applied successfully on many different problems. These problems cover various areas such as speech recognition, image recognition [1], and even board games [2]. One place where effective image recognition and analysis could be applied to great benefit is in the field of medical diagnostics. Computer-aided detection and diagnosis can help physicians interpret medical images, decreasing the time spent on interpretation and possibly even increasing their accuracy [3].

Dementia is an umbrella term for several diseases affecting memory and cognitive abilities, and in 2015 affected 47 million people worldwide (5 % of the elderly population). This number is predicted to increase to 75 million by 2050 [4]. The leading cause of dementia is the neurodegenerative *Alzheimer's disease*. No cure currently exists for this brain disease, but early detection is still important as it carries with it several benefits for the patient [5]. Alzheimer's causes brain atrophy, which can be detected using MRI imaging. Several studies [6, 7] shows promising results for identifying and classifying Alzheimer's disease using convolutional neural networks (CNNs) trained on these structural MRI images.

Machine learning is a complex field, and CNNs can be used in a wide variety of ways. They can have different activation functions, different number of layer and filters, and varying depths. The training of a CNN can also be performed in various ways, with different hyperparameters, error functions, and dataset layouts.

1.1 Research Question

Choosing the correct way of training a CNN can have a significant impact on the resulting performance. This study aims to evaluate two training methods that are often used and relevant for CNN models:

1. The use of class-weighting in the error function. This can be applied to alleviate bias when the classes in the dataset are of very different sizes.
2. Data augmentation. This can be used to reduce overfitting and improve performance when datasets are small.

These methods tries to solve different problems and can each be applied independently to an existing model. They can be useful for all kinds of CNNs, but this study will focus on their effects on Alzheimer's diagnosis using MRI imaging. To this end, a simple CNN will be created, and its performance will be assessed when training, first without any of these methods, then with *class-weighting*, and finally with *data augmentation* and class-weighting. The following questions will be answered:

- Does class-weighting improve the performance on rarer classes?
- Does data augmentation improve general performance?

Chapter 2

Background

2.1 Alzheimer's Disease

Dementia is a group of symptoms that may have many different causes. One common cause is Alzheimer's disease, a degenerative brain disease that is estimated to occur in 60 % to 80 % of dementia cases. In half of these cases of dementia Alzheimer's is the only cause, while many other also have evidence of other changes to the brain [5]. Individuals with Alzheimer's disease may experience multiple symptoms, which may vary over the course of the illness [5].

2.1.1 Benefits of early detection

Early (and accurate) diagnosis of Alzheimer's disease has several benefits. Those who do not have Alzheimer's but are diagnosed with mild dementia, might after testing realize that they're suffering from some other, treatable condition. There are also medical and social benefits to get an Alzheimer's diagnosis early. Neither a curing nor a slowing treatment exist, but individuals can still take some measures that helps them retain their cognitive function for as long as possible. Aerobic exercise, mental activity and social engagement may help delay cognitive decline, while medication and other intervention can help with managing symptoms. An early diagnosis can also help reduce anxiety by giving noticed symptoms a name, for both affected individuals and their family members. Finally, early diagnosis gives individuals time to make plans for the future while still having good cognitive function [5, p. 406-409].

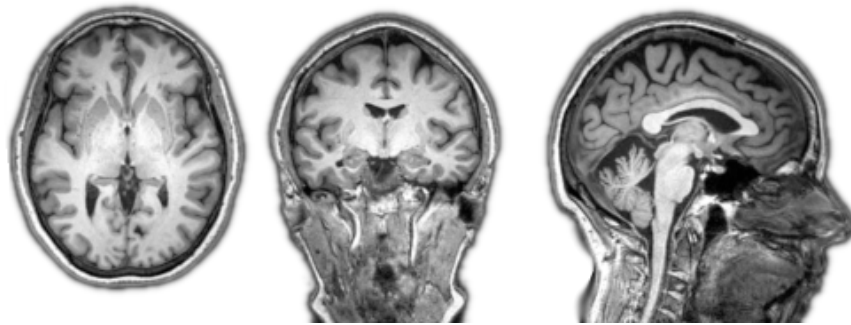


Figure 2.1: The three planes of a brain MRI scan.

2.1.2 Brain changes

The abnormal accumulation of two proteins (into what is called beta-amyloid plaques and tau tangles) inside the brain is one of several brain changes associated with Alzheimer's disease. These proteins interfere with neuron communication and nutrient transportation, while also triggering inflammation. Brain shrinkage, or *atrophy*, eventually occurs due to cell loss. These protein changes may begin as early as 20 years or more before symptoms appear [5].

Brain atrophy occurs in a relatively late disease stage, when neurodegeneration has started to occur. This can be detected and measured using structural MRI imaging. Atrophy is a medically established predictor of an individual progressing from mild cognitive impairment to Alzheimer's disease. In addition, there seems to be a direct relationship between the severity of the atrophy and time-to-dementia. Brain atrophy is nevertheless not a unique sign of Alzheimer's, but a measure of neural degeneration in general. Degeneration might be caused by other conditions [8].

2.1.3 Clinical dementia rating

Clinical dementia rating (CDR) is a score that is used to classify the stage of Alzheimer's disease. Scores 0.0 through 2.0 is described in Table 2.1. There also exists a final score of 3.0 [9]. This score is not relevant to this study, as the disease in this stage is too severe to partake in medical studies.

The CDR is derived from an interview with the patient as well as an informant, where impairment to six cognitive categories are scored: *Memory*, *Orientation*, *Judgment and Problem Solving*, *Community Affairs*, *Home and Hobbies*, and *Personal Care*. The final CDR score is a combination of all the categories, with *Memory* being given extra weight [9].

Score	Severity of AD	Description
0.0	None	No consistent forgetfulness. Cognitively normal.
0.5	Very mild	Slight, 'benign' forgetfulness. Slight impairment to problem solving, social activities, and hobbies. Able to fully care for oneself.
1.0	Mild	Moderate memory loss. More difficult chores and interests abandoned. Moderate difficulty with time relationships. Need prompting for personal care.
2.0	Moderate	Severe memory loss. New material learnt is rapidly lost. Usually disoriented to time, often to place. Requires assistance with dressing and hygiene.

Table 2.1: Clinical dementia rating definitions [9]

2.2 Artificial neural networks

Artificial neural networks (ANNs) are computational modeling tools inspired by the neural networks found in biology. ANNs exhibit several information processing characteristics which make them useful for modeling complex real-world problems, among them non-linearity, their ability to handle imprecise information, and their ability to generalize [10]. A common machine learning task is *classification*. Here the machine learning algorithm is presented with a dataset containing many data points, each having some number of features and an assigned class. The goal of the machine learning model is then to learn a mapping from features to class, such that the model can be applied to new data where the class is unknown. ANNs are often used successfully for this task, but they do have the drawback that the models are opaque, or in other words that it is hard to reason about why they give the output that they do [11, p. 3, 8–9].

2.2.1 Structure

Artificial neural networks can be constructed in different ways, but the most common is the so called *multi-layer feed-forward neural network*, illustrated in Figure 2.2. It consists of neurons ordered into multiple layers, the first layer

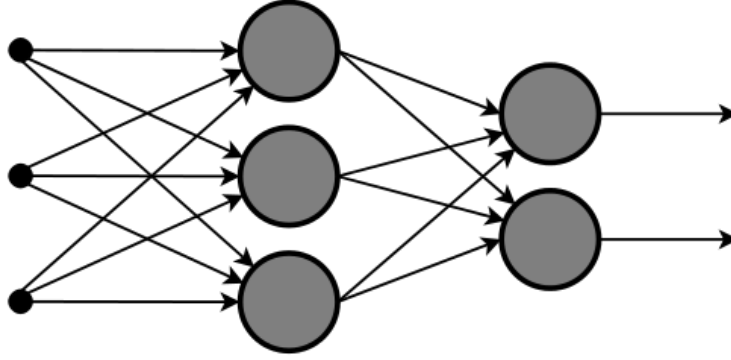


Figure 2.2: An illustration of a multi-layer artificial neural network.

Author: Offnfopt / CC-BY-SA 3.0.

being the *input layer* and the last one the *output layer*. The layers in between are called *hidden layers*. Each neuron is connected to all the neurons in the next layer, with every connection from neuron i to neuron j having a weight coefficient w_{ij} . In addition, every neuron has a threshold coefficient, or bias b_i . Using these coefficients and all the inputs from the previous layer, every neuron produces an output x_i , which can be formally written as follows:

$$x_i = \varphi \left(\sum_{j \in \Gamma^{-1}(i)} w_{ij} \cdot x_j \right)$$

Here $\Gamma^{-1}(i)$ is notation for the set of all neurons in the previous layer, and φ is some *activation function*. The activation function squeezes the weighted sum of all input neurons into the range $[0, 1]$, in preparation of passing the output along to the next neuron [12].

2.2.2 Activation functions

There are many possible choices for the activation function, but three of the most common ones can be found in Table 2.2. The softmax activation function is different from the others since it does consider only one neuron at a time, but instead all neurons in the layer. In doing this, the softmax function can ensure that the sum of all the output values are equal to 1, which is a desirable thing for the output layer when classifying into K different classes. This also allows the use of the *cross-entropy error function* (see later section), which seems to give higher accuracy than using the sigmoid function and the *least squares error function* [13].

Name	Function $\varphi(x)$
Sigmoid	$\frac{1}{1 + e^{-x}}$
Rectifier	$\max(0, x)$
Softmax	$\frac{e^{x_i}}{\sum_j^K e^{x_j}}$

Table 2.2: A few common activation functions.

2.2.3 Error functions

To be able to train our network, we need some way to determine how ‘bad’ a certain output is. This is done using an *error function*, or *loss function*. This function is averaged on the network’s output on all training samples. In a classification task, a low average loss means that the network is good at predicting the correct class from a given input data. If the network performs perfectly, the error function is generally 0. *Least squares error* is a traditional error function, and if you let x and \hat{x} be the network’s output and the wanted output respectively, the least squares error is defined as [12]

$$E = \sum_i \frac{1}{2} (x_i - \hat{x}_i)^2$$

There is however also another error function that was mentioned earlier, the *cross-entropy error* function. This has some theoretical advantages over least squares error, such as better handling when the probabilities estimated by the network is small, as well as being less dominated by large outliers. Experiments also suggest that these advantages are large enough to be useful in practice. The cross-entropy is defined as [14]

$$E = \sum_i \hat{x}_i \left(\frac{x_i}{\hat{x}_i} \right)$$

2.2.4 Gradient descent

To train a neural network, we want to change the weights and biases (the network's parameters) such that the error on the training set becomes as low as possible. This is done using a process called *gradient descent*. Given the error function and the dataset, the neural network can be viewed a giant function, taking all the parameters as input. By finding the slope or *gradient* of the parameters in the direction that decreases the error, the parameters can gradually be adjusted to find a minimum of the error function. This process is illustrated in Figure 2.3. Since this gradient is calculated using the entire dataset to perform a single step, it is known as *batch* gradient descent. Due to this batch gradient descent can become very slow on large datasets, and does not work at all if the dataset does not fit in memory [15]

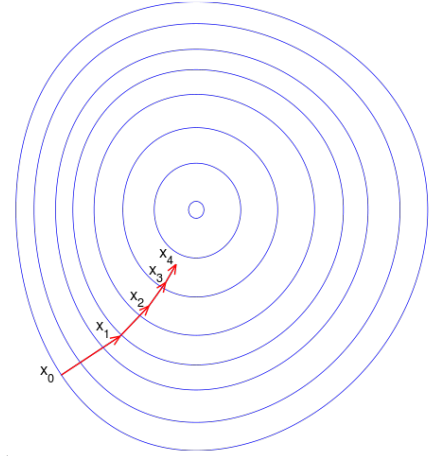


Figure 2.3: Illustration of gradient descent.

Because of this a variant of gradient descent named *stochastic* gradient descent (SGD) is sometimes used, where the gradient is calculated using only a single training example at a time. This makes updates to the parameters much faster, but also causes the error function to jump around. This fluctuation adds noise, but also helps the model to not get stuck in local minima. SGD also has the benefit of allowing online learning, where the network can train on new data as it becomes available. One possible disadvantage of SDG is that the overhead of calculating the gradient many times might have a negative influence on the final performance [15].

There is also a third option, *mini-batch* gradient descent, which aims to combine the best of both batch and stochastic gradient descent. Here the gradient calculation is performed on *mini-batches* of n training examples. This can help making the updates more stable, and can also allow the use of optimized matrix libraries that enables quickly calculating the gradient for a mini-batch. This is the method typically used when training neural networks in practice, and as such the term SGD is usually used to refer to this specific method [15].

Adam optimizer

Adam is another, more recent method for gradient-based optimization which works in a similar manner as mini-batch SDG. Its hyperparameters (parame-

ters that adjust how the optimization method works) are intuitive and require little tuning, at the same time as Adam has high efficiency. It compares favorably to other optimization methods, and is well-suited for use in machine learning problems [16].

2.2.5 Learning rate

The learning rate is a hyperparameter that determines how large steps should be taken when performing the gradient descent. Too large a learning rate, and the network will not converge. Too small, and the convergence will take too long. A common strategy is to keep decreasing the learning rate as training progresses [17].

2.3 Convolutional neural networks

2.3.1 Motivation

One of the largest limitations of traditional artificial neural networks is that they struggle with the complexity of images. The input of a grayscale image of just 112×112 pixels require 12 544 connections for every neuron in the first hidden layer. This means that the network has to have a very large number of parameters, which results in training of the network becoming very computationally demanding, as well as increasing the chances of overfitting. A model with fewer parameters to train makes overfitting less likely, which improves the network's performance [18].

2.3.2 Structure

Convolutional neural networks (CNNs) are specifically designed to work with images as input, and therefore has a different structure than traditional neural networks. The neurons in a layer are not connected to all the neurons in the preceding layer, instead only connecting to a small region of it. CNNs usually consists of three different types of layers: *convolutional* layers, *pooling* layers, and *fully-connected* layers. The layers are said to have three dimensions: height, width, and depth. The initial height and width correspond to the input image dimensions, while the depth represents different *feature maps*. If the images the network trains on are colored with three color channels, and has size 112×112 , the input 'volume' will have the dimension $(112, 112, 3)$ [18].

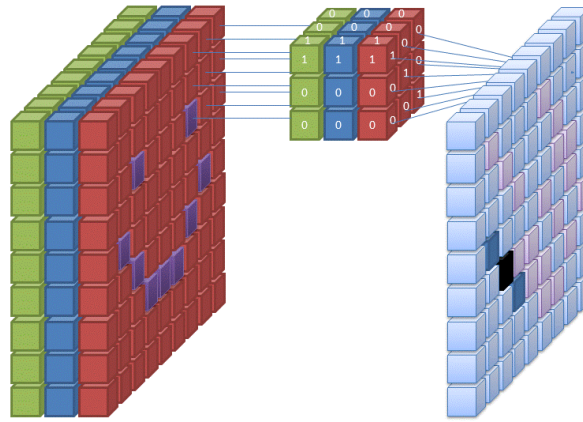


Figure 2.4: Illustration of a filter on a color image.

Author: Cecbur / CC-BY-SA 4.0.

2.3.3 Convolutional layers

Convolutional layers consists of a number of different *filters*. These filters usually have small dimensions, say $(3, 3)$, but works on the entire depth of the input and are used to sweep, or *convolve*, over the input's width and height. A 3×3 filter would have $3 \times 3 \times \text{depth}$ adjustable parameters, which forms the filter's *kernel*. The output value, which will be located at the pixel in the center of the filter, is the dot product of the kernel and the current input region [18]. For an illustration of this see Figure 2.4.

The depth of a layer's output volume in the CNN can be increased by increasing the number of filters used. The filters can pick up on different features, of which the intensity in different regions of the image will be outputted in the different depth dimensions, and picked up by later layers.

Another parameter that can be adjusted for the layer is the *stride*. A stride of 1 means that the filter is moved one step at a time, while a higher stride of n can be used to make the filter only evaluate every n^{th} possible position along each axis. This reduces overlap, but has the benefit of shrinking the output volume.

For filters of size larger than 1×1 , the output will have slightly smaller dimensions since the filter can't go outside the edge of the input. To avoid missing features on the edges, *zero-padding* is often used. This pads the input volume with enough zeroes to make the output volume the correct size,

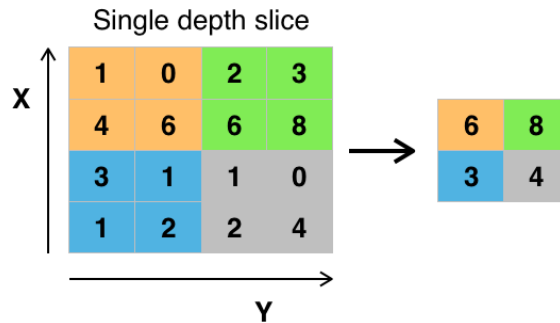


Figure 2.5: Example of max pooling.
Author: Aphex34 / CC-BY-SA 4.0.

enabling the filter to be positioned right on the edge [18].

2.3.4 Pooling layers

A different way of shrinking the layers' spatial dimension are *pooling layers*. These are used to gradually reduce the representation's dimensionality, which reduces the total amount of parameters that need to be trained. The most common method is by using 2×2 max-pooling layer. This is applied with stride 2 on the input volume, shrinking the output width and height by 2 and reducing the total output size to 25 %. As the name implies, the output of the filter is the maximum of the input values in the region as shown in Figure 2.5. Pooling retains the number of depth dimensions [18].

2.3.5 Fully-connected layers

The fully connected layers work just like they do in traditional neural networks, and are often placed at the end of a CNN network. This lets them transform all the data available in the final volume into the final output of the network.

2.3.6 Common architectures

Commonly a CNN is constructed by stacking pairs of convolutional and pooling layers, before finally ending with fully-connected layers. The convolutional layers' outputs are passed through the rectifier activation function. Another common option is to have two convolution layers before the pooling layers. This is recommended since it allows the network to learn more complicated features. The general trend of the layer volume dimensions is that they

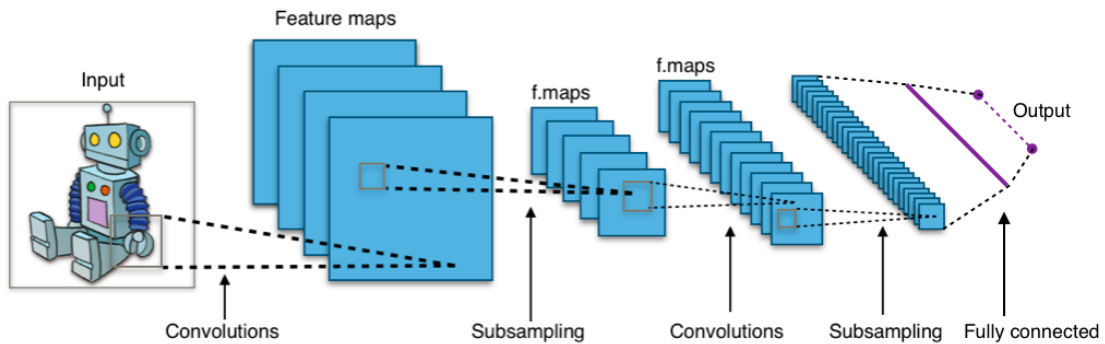


Figure 2.6: Illustration of a typical CNN architecture.

Author: Aphex34 / CC-BY-SA 4.0.

become smaller but deeper towards the end of the network [18]. For an illustration of a typical complete CNN architecture, see Figure 2.6.

2.4 Evaluation of classification systems

To evaluate a classification system an objective scoring method is needed. The most common method uses the two values *precision* and *recall*, together with the combined measure *F1-score*. These can be calculated from the the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Descriptions of these can be found in Table 2.3. Using precision and recall instead of accuracy (the ratio of correct classifications) provides more information about what kinds of mistakes a model makes.

Name	Meaning
TP	Hits, number of correct identifications.
TN	Number of correct rejections.
FP	False alarms, number of incorrect identifications.
FN	Misses, number of incorrect rejections.

Table 2.3: Explanation of classification results.

Precision

In the context of classification, the fraction of guesses on a certain class that is correct is said to be the *precision* of the model on this class [19, p. 5].

$$P = \frac{TP}{TP + FP}$$

Recall

In the context of classification, the fraction of instances of class which are correctly identified is said to be the *accuracy* of the model on this class [19, p. 5]. This is also known as *sensitivity*.

$$R = \frac{TP}{TP + FN}$$

F1-Score

The F1-score can be used to combine the precision and recall scores into a single measure. It is defined to be the harmonic mean of precision and recall [19, p. 156-167], which can be written as

$$F_1 = \frac{2PR}{P + R}$$

2.5 Previous work

Studies focusing specifically on the effect of different training methods on neural networks, convolutional or otherwise, seems to be rare. This is especially true for the smaller subject of machine learning on medical images. What do exist, however, is *guidelines* for how you should train networks in various scenarios, as well as studies where researches create and evaluate a specific CNN model. Three previous works will be highlighted here: One for the case of medical images, one for the handling of imbalanced dataset, and finally one study which achieves high performance on classification of Alzheimer's disease. A few additional studies for classification of Alzheimer's disease will be mentioned, without going into detail.

2.5.1 Training CNNs on medical images

In the book *Deep Learning for Medical Image Analysis*, suggestions for how to train machine learning models in the context of medical images are de-

scribed. They suggest three general methods for improving learning: The use of rectified linear units (ReLU), dropout, and batch normalization [20, p. 20-22] These will not be evaluated in this study, however ReLU units will be used in the CNN (see Section 3). In addition, *Deep Learning for Medical Image Analysis* highlights the usefulness of data augmentation, where geometric transforms are applied to the training data in order to extend it. This is said to typically increase performance by about 3 % [20, p. 31].

2.5.2 Handling imbalanced datasets

Training of machine learning models for classification problems often suffers from imbalance in the dataset between the different classes. This is studied in “Survey of resampling techniques for improving classification performance in unbalanced datasets”, by More [21].

Three general solutions, along with more or less complicated variants, were evaluated on a synthetic imbalanced dataset. The goal is to improve recall on the rare class, while not worsening the precision of the common class. First of the methods is *class-weighting*, where the error function is modified to weigh each class equally, disregarding their relative sizes. Second is *undersampling*, where samples of the common class are skipped when training. Finally there is *oversampling*, where the rare class is in some way replicated in order to make it larger. In their experiment, a variant of oversampling called *SMOTE* in combination with class-weighting was found to be the most effective. Granted that, the article also notes that the most effective method may vary depending on the distribution of your dataset [21].

2.5.3 Classification of Alzheimer’s disease

Many studies exist describing systems classifying MRI images as either having Alzheimer’s disease or not. Many different methods have been used, including linear support vector machines by Cuingnet et al. [22], deep CNNs by Sarraf and Tofghi [23], probability distribution functions by Beheshti, Demirel, Initiative, et al. [24], as well as strategies combining MRI with multiple biomarkers and machine learning models such as the one by Zhang et al. [25].

Multi-class classification

Islam and Zhang [7] studied a variation of Alzheimer’s disease classification, where classification is done into four different classes (based on CDR scores), instead of just two. Their method was a variant of a CNN that they trained

CDR Class	Precision	Recall	f1-Score	Support
Non-demented (0.0)	0.99	0.99	0.99	73
Very mild (0.5)	0.75	0.50	0.60	6
Mild (1.0)	0.62	0.71	0.67	7
Moderate (2.0)	0.33	0.50	0.40	2

Table 2.4: Results by Islam and Zhang [7].

on the OASIS dataset [26]. Their CNN uses a densely connected architecture that seems identical to DenseNet [27]. While their model showed very high performance on identifying non-demented patients, its performance on classifying demented patients was significantly worse, which they speculate could be because of the limited number of samples [7] in OASIS. Their performance can be viewed in Table 2.4.

Chapter 3

Methods

This report will evaluate two different strategies for training a convolutional neural network, comparing them to a baseline. The structure of the CNN used is described in Section 3.6. This CNN was trained and evaluated using the OASIS-3 dataset, presented in Section 3.2. The training methods examined were:

1. Baseline
2. Training with equal weighting of the different classes (*class-weighting*)
3. Training utilizing *data augmentation*

The network was trained for 1000 epochs (one epoch is one run through all training data) for the *Default* and *class-weighted* training regimes. The last training method using data augmentation was allowed to run for 5000 epochs, since convergence was much slower. The accuracy on the different classes were measured for each method. Most similar studies report their performance using precision, recall, and f1 score. The same performance metrics will be used in this study, both for their inherent usefulness and to make comparisons possible.

3.1 Training methods

3.1.1 Baseline

Baseline training is the least sophisticated method, and is what would be used by someone who does not put any thought into how the training should be

done. This is what popular machine learning frameworks will provide by default, and will provide a performance baseline. The other two methods are meant to counter the problems the baseline training strategy might have in our use case.

3.1.2 Class-weighting

OASIS-3 contains many times more images of patients without or with mild Alzheimer’s disease as with more severe cases (see Section 3.2). One way to try and correct this is by weighing each class equally in the error function, no matter how large part of the dataset each class actually represents. This will be the second training method tested, to see if class weighting solves the issue of an unbalanced dataset and improves the performance on the rarer classes.

3.1.3 Data augmentation

When the size of the dataset is not too large, there is a risk of overfitting. Overfitting is when the network ‘memorizes’ the data used in the training set. This causes it to have very high performance and low error on the training data, while not being able to perform well in general which will be visible when testing on the validation data. One way to battle this is data augmentation, where the dataset is in some way extended. For this test the images will be mirrored so that the right side of the brain will become the left, as well as zoomed in or out a random amount at most 20 %. This will, in effect, produce an infinite training dataset (albeit a quite homogeneous one). This larger dataset will in theory make overfitting less likely. However, this will not change the proportion of the training classes. When training, an image is selected from the original dataset. This image is then used to create an augmented image, which keeps the class ratio the same.

3.2 OASIS-3 dataset

OASIS-3 is a longitudinal neuroimaging dataset, with data collected over the course of 30 years from 1098 adult participants from multiple studies. Of these participants 809 were cognitively normal, while 489 were at various stages of cognitive decline. Accompanying the neural imaging is (among other things) clinical assessments of cognitive function [28]. I will use the T1-weighted brain scans and the mental assessments done using the CDR system. The

dataset contains in total 3393 T1-weighted MRI images and 4089 different CDR evaluations.

The subset of the dataset that was used contains 2782 images (see skipped images in Section 3.4). Of these images 20 % were selected to be used as validation data, while the other 80 % was used to train the network. Since the network has never seen the validation data during training, it can be used to evaluate its expected performance on unknown MRI images. Just as in the paper by Islam and Zhang [7], the CDR scores were used to divide the images into four classes: *non-demented*, *very mild dementia*, *mild dementia*, and *moderate dementia*. These classes are not equally represented in the dataset, with the more serious dementia cases being much less frequent. The total count of each class is shown in Table 3.1, together with the image count in the training and verification sets.

CDR Class	Total	Training set	Validation set
Non-demented (0.0)	2162	1722	440
Very mild (0.5)	442	362	80
Mild (1.0)	152	121	31
Moderate (2.0)	22	17	5
Sum	2778	2222	556

Table 3.1: Count of images belonging to each class in the dataset.

3.3 Labeling

The assessment of the subjects' CDR score was used to label every MRI image with the stage of dementia. However, since OASIS-3 is a longitudinal study, the MRI scans and the clinical assessments were not performed at the same time. In fact, the time between MRI scans and the closest CDR assessment can be relatively large ($\mu = 116$, $\sigma = 107$). The distribution is shown in Figure 3.1. For most of the data the time from an MRI scan to the closest CDR assessment is less than 200 days, while for some images the delay can be several hundred days.

To try to get reasonable CDR labels even when the time differences are large, a linear interpolation of the two closest CDR assessments considering the time of the MRI scan was used when this was possible. If the MRI scan

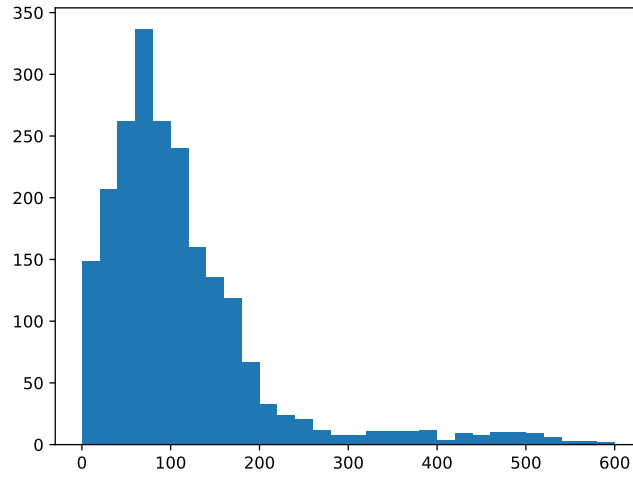


Figure 3.1: Histogram of time between MRI scans and the closest CDR assessment (in days).

was done either before or after all of this subject's CDR assessments, the MRI scan was tagged with the closest CDR value.

This CDR calculation can be written as follows:

$$\text{cdr} = \begin{cases} \text{cdr}_{\text{next}} & \text{if before all assessments} \\ \text{cdr}_{\text{prev}} & \text{if after all assessments} \\ \text{cdr}_{\text{prev}} + x * (\text{cdr}_{\text{next}} - \text{cdr}_{\text{prev}}) & \text{otherwise} \end{cases}$$

$$\text{where } x = \frac{\text{day}_{\text{MRI}} - \text{day}_{\text{prev}}}{\text{day}_{\text{next}} - \text{day}_{\text{prev}}}$$

Here $0 \leq x \leq 1$ is the interpolation factor between the two assessments closest to day_{MRI} , the day the MRI scan was done.

3.4 Image extraction

During the preprocessing step a single image plane was extracted from each MRI scan, of the brain as viewed from above. For simplicity the plane chosen was the middle one along the axis. The raw intensity values from each scan were flattened into the normal black/white range of PNG images.

Some MRI scans are rotated the wrong way, even after trying to orient them correctly using the metadata from the scan. These images were not corrected but instead not included in the dataset.

3.5 Hyperparameters

The Adam optimizer was used to train the network. It was instantiated with a learning rate of 10^{-5} , a β_1 of 0.9, a β_2 of 0.999, an epsilon of 10^{-7} , and no weight decay. When training a batch size of 64 was used.

When choosing hyperparameters, the learning rate of the Adam optimizer was found to be important. The default learning rate of 0.001 appears to be too large, and results in the network not improving. A lower rate of 10^{-5} resulted in better and quicker results during training.

3.6 Network structure

The exact structure of the CNN this study is going to use is not considered important, since only the relative performance of the different training methods will be compared. The chosen structure is mostly based on the basic CNN design suggested by O'Shea and Nash [18], with double convolutional layers joined by pooling layers. The first two layers are taken from Islam and Zhang [7], and consists of a convolutional layer with filters of size 7×7 and stride 2, followed by a pooling layer. The sequence of layers can be seen in Figure 3.2. All the other convolutional layers use filters of size 3×3 and a stride of 1. Every convolutional layer uses the linear rectifier as activation function. At the end of the network there are 128 fully connected neurons which finally connect into the four output neurons. Each output neuron represents the probability of each class. The densely connected layer also uses rectified linear activation, while the output layer uses softmax activation and the cross-entropy loss function.

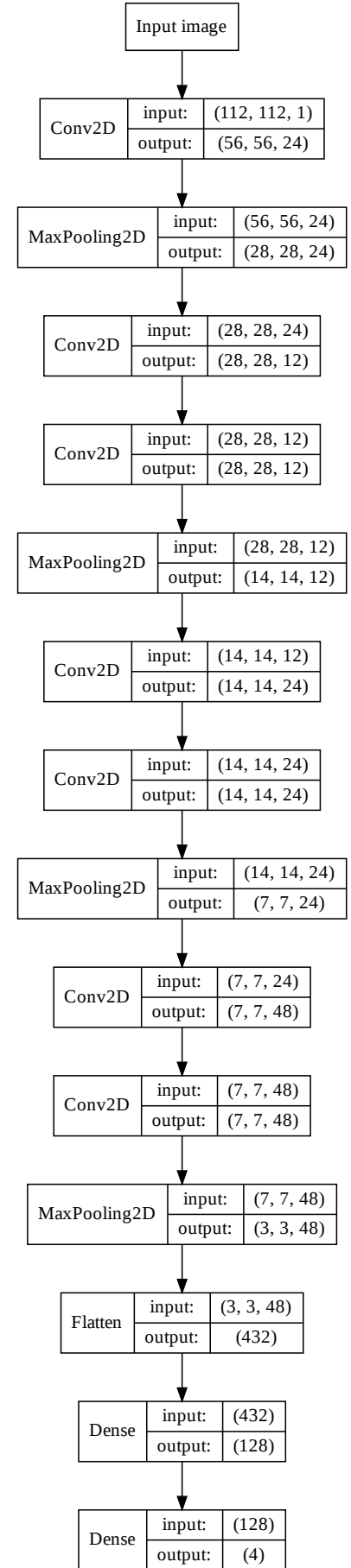


Figure 3.2: Diagram of the CNN used in this study.

Chapter 4

Results

For each of the two methods, in addition to the default baseline, the results will be presented as one table, and two diagrams. The table presents the precision, recall, and f1-score for each method on each class. The support column of the table shows the number of data points in each class. Then follows the two diagrams. The first shows the evolution of the loss function for both test and validation data during the training, on a logarithmic scale, while the second shows the evolution of the accuracy.

4.1 Baseline

CDR Class	Precision	Recall	f1-Score	Support
Non-demented (0.0)	0.88	0.91	0.89	440
Very mild (0.5)	0.51	0.46	0.49	80
Mild (1.0)	0.40	0.32	0.36	31
Moderate (2.0)	0.33	0.20	0.25	5

Table 4.1: Baseline results.

As can be seen in Table 4.1, the network achieves high performance on the *Non-demented* class (f1-score of 0.89), but as the classes get rarer performance deteriorates. Especially bad is the recall, with a value of just 0.20 on the *Moderate* class.

The validation loss, presented in Figure 4.1, can be seen to quickly decrease together with the training loss, and then slowly start increasing while the training loss decreases. For the accuracy presented in Figure 4.2, note the ‘bump’ in accuracy at the beginning, and the following gap as test accuracy rises to 1 while validation accuracy stays around 0.80.

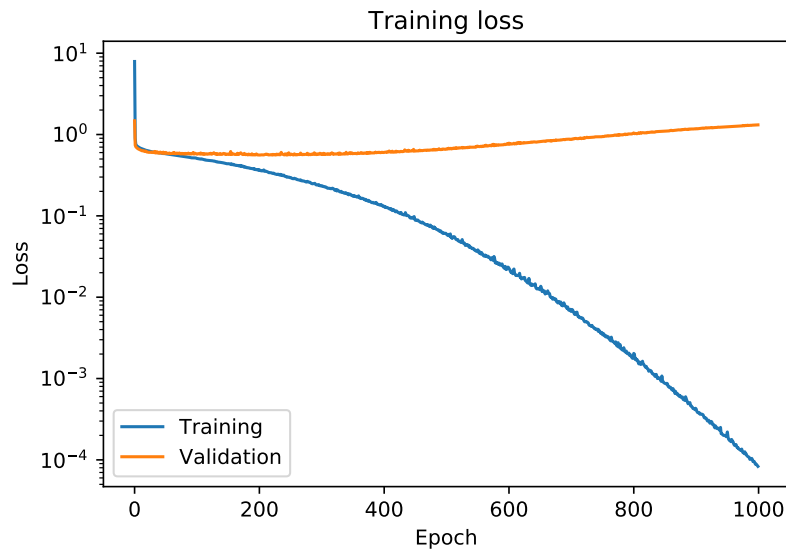


Figure 4.1: Loss during training with the baseline training scheme.

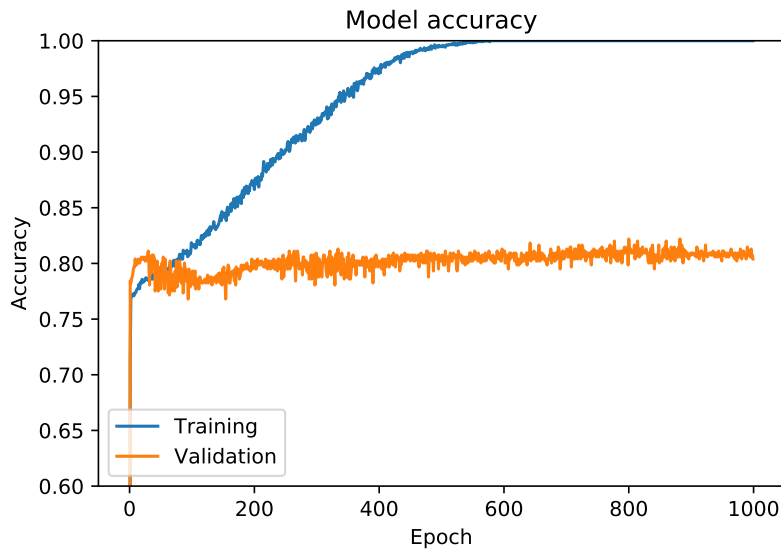


Figure 4.2: Accuracy during training with the baseline training scheme.

4.2 Class-weighted training

CDR Class	Precision	Recall	f1-Score	Support
Non-demented (0.0)	0.87	0.93	0.90	440
Very mild (0.5)	0.52	0.41	0.46	80
Mild (1.0)	0.43	0.29	0.35	31
Moderate (2.0)	1.00	0.60	0.75	5

Table 4.2: Results for class-weighted training.

As can be seen in Table 4.2, the networks achieves high performance on the *Non-demented* class (f1-score of 0.9). The model does not perform as well on the *Very mild* class (f1-score of 0.46), and even worse on the *Mild* class (f1-score of 0.32). Nevertheless, note the high performance of the *Moderate* class (precision of 1.0 and recall of 0.60).

The training and validation loss has a large gap, as can be seen in Figure 4.3. The accuracy shown in Figure 4.4 starts out low, below the bottom of the diagram at 0.60, but then increases. Training accuracy increases faster towards 1.0, while validation accuracy more slowly plateaus at 0.80.

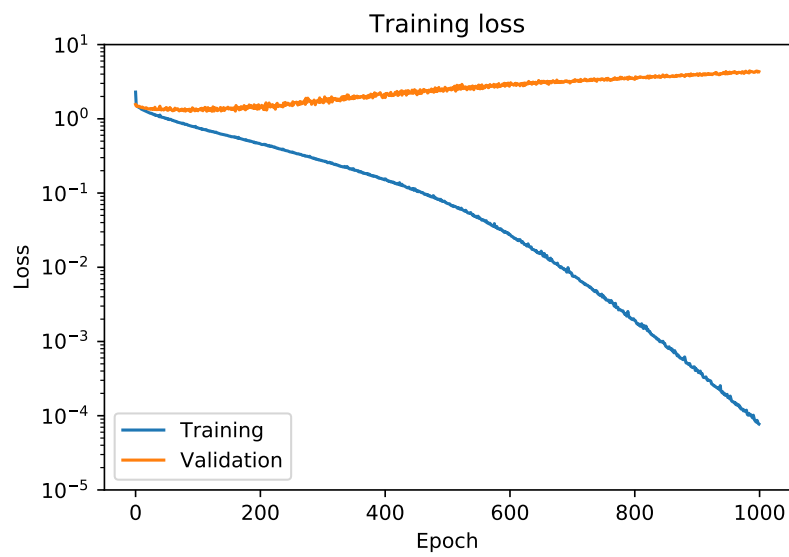


Figure 4.3: Loss during training with class-weighting.

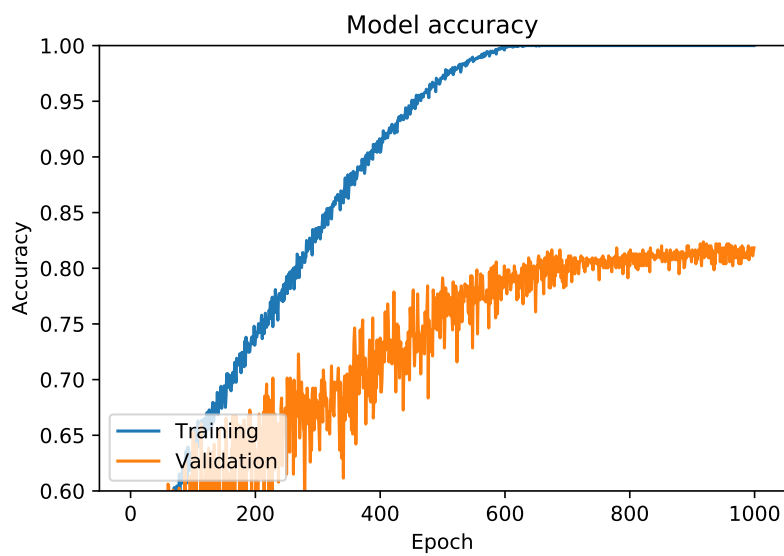


Figure 4.4: Accuracy during training with class-weighting.

4.3 Training with data augmentation

CDR Class	Precision	Recall	f1-Score	Support
Non-demented (0.0)	0.91	0.77	0.83	440
Very mild (0.5)	0.33	0.57	0.42	80
Mild (1.0)	0.40	0.55	0.46	31
Moderate (2.0)	0.67	0.40	0.50	5

Table 4.3: Results with data augmentation.

Looking at Table 4.3, *Non-demented* precision is high (0.91), but recall is lower (0.77). Precision on the *Very mild*, *Mild*, and *Moderate* classes was low (0.33, 0.40, and 0.67 respectively) with recall being slightly higher (0.57, 0.55, and 0.40 respectively).

The training and validation loss presented in Figure 4.5 follow each other relatively closely. Note also that the training accuracy in Figure 4.6 does not reach 1.0, instead following the validation accuracy without departing significantly.

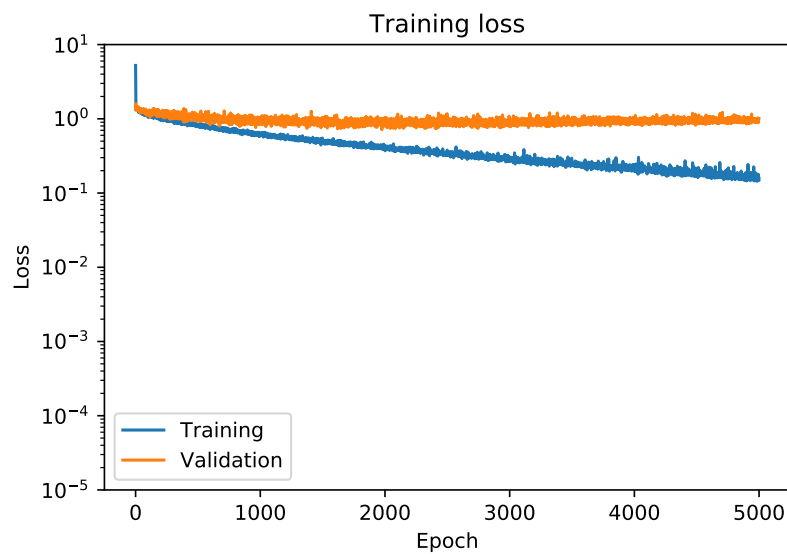


Figure 4.5: Loss during training with data augmentation.

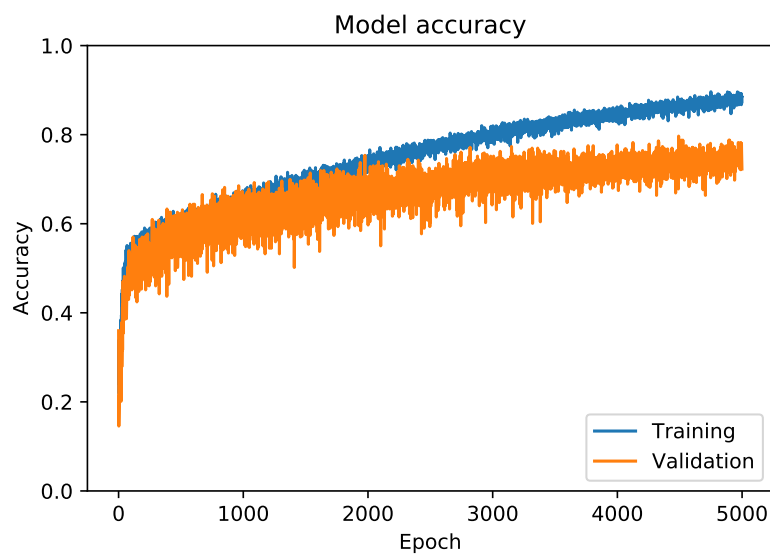


Figure 4.6: Accuracy during training with data augmentation.

4.4 Summary of training results

For easy visual comparison, the precision and recall for the three training methods on each class can be found in Figure 4.7 and Figure 4.8, respectively.

Class-weighting is seen to not decrease precision much on any class, while improving the precision significantly on the *Moderate class*. Recall was found to deteriorate slightly on *Very mild* and *Mild* classes, while improving slightly on *Non-demented* and significantly on *Moderate*.

Data augmentation increases precision slightly on the *Non-demented* class, while decreasing precision heavily on the *Very mild* and *Moderate* classes. Recall is decreased on *Non-demented* and *Moderate*, while increasing for the other two classes.

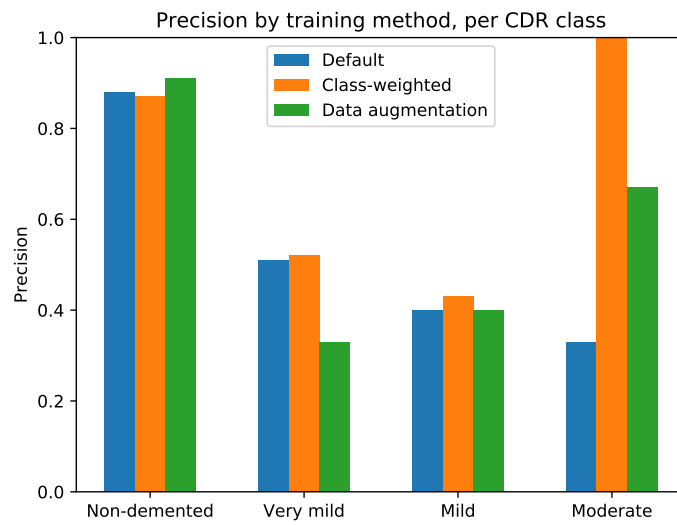


Figure 4.7: Comparison of the precision resulting from the different training methods.

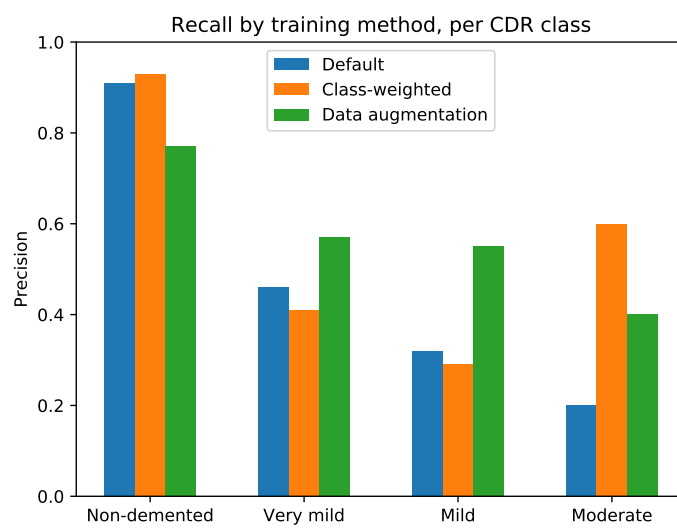


Figure 4.8: Comparison of the recall resulting from the different training methods.

Chapter 5

Discussion

5.1 Method

It is debatable exactly what good performance is in the area of computer-aided medical diagnostics. Is it more important that the model does not miss any cases of disease (minimize false negatives), or that the model does cause any false alarms (minimize false positives)? Additionally, in multi-class classification like the one performed here, are some classes more important than others? The answers to these questions are not obvious. Most likely the model should prioritize finding all cases of disease. If some false positives also occur that is fine, as long as the rate is low enough for the model to be useful. For Alzheimer's diagnosis, the most interesting class is probably *Very mild*. Diagnosis of this class would bring the most benefit to patients, as the disease is still not very apparent and in an early stage.

Just as in the study by Islam and Zhang [7], the low frequency of samples in the dataset showing progressed dementia is an issue. Thanks to OASIS-3 containing many times the number of MRI images as the original OASIS does, this problem is not as severe during training, with all classes having a decent amount of images (maybe besides *Moderate*, which only has 22). The low frequency is however very noticeable during validation, when it results in single-digit counts of the rarest class *Moderate*. This gives very rough knowledge of how accurate the model actually is on this type of data, as a single extra correct classification increases recall by 0.20. To lessen this it might have been useful to increase the ratio of validation to training data, for more stable results. On the other hand, detecting these progressed cases of dementia might not be very useful in practice, since by then the symptoms will be so pronounced that diagnosis using MRI imaging is not required. In the long

term it would be beneficial if new and larger datasets were created, that would both give more training data and enable researchers to evaluate their models' performance with more confidence.

The exact CNN model and hyperparameters used were assumed to not have any effect on the relative performance of the different training methods. However, this is just a guess, and might not be true. One must consider the possibility that other network structures and hyperparameters might benefit more or less from the two training methods.

5.2 Results

5.2.1 Baseline training

When training normally, the network starts out with an accuracy of just below 80 %. The training data mostly contains images of cognitively normal brains, making up $\frac{1724}{2162} \approx 79.7$ % of the dataset. This suggests that the network quickly learns to classify everything as this class, for a large and quick gain in accuracy which will just as quickly decrease the loss. Inspection of the output reveals that yes, this is actually the case the first few epochs. However, after training some more the network begins to try and also classify the other classes, managing a precision ranging from 30 % to 50 % on the three demented classifications. The accuracy steadily decreases as the classes get rarer, most likely because the loss function does not encourage exploration in order to get better on them. Since all input samples are weighted equally in the eyes of the loss function, performance on the ca. 20 images showing moderate dementia is worth 100 times less than the performance on the ca. 2000 images showing no dementia. If getting better on the rare classes means getting slightly worse on the much more common one, improvement is not likely. Overfitting seems to be a large problem, with the network quickly reaching 100 % accuracy on the training data, with nowhere near as good results on the validation set.

5.2.2 Class-weighted training

Weighting each class equally was very important for the most rare class, *Moderate*. The precision on this group increased from 0.33 to 1.0, and the recall from 0.20 to 0.60. This happened at the same time as the performance on the other classes remained on a similar, slightly higher level, besides on the *Non-demented* class. These decrease was however very small, and not significant.

In conclusion, class-weighting was important for the precision of the very rare class, but not as much for the moderately rare.

The recall was also found to improve significantly on the *Moderate* class, as well as increasing slightly on the *Non-demented* class. Interestingly, recall decreased on the *Very mild* and *Mild* classes, despite increased recall on rare classes being one of the main supposed benefits of class-weighting. Perhaps these classes are not rare enough, with only the *Moderate* class being rare enough to improve, or maybe the method simply doesn't work that well on this dataset. Overfitting is still a problem, with training validation reaching 100 %.

In general, if the dataset contains any rare classes like the *Moderate* class, making sure to not disfavor them could be very important. These rare classes might be obvious, as they are here, but might also be hidden. For example, if a dataset mostly contains data collected from male subjects, a model could be biased during training and perform less well on female data, and vice versa. In cases like this class-weighted training could be essential for fairness.

5.2.3 Training with data augmentation

The results when using data augmentation need to be compared to the class-weighted results, since the data augmentation was used on top of the class-weighting. At a first glance, the results were to my surprise not very impressive. The network still showed signs of overfitting, albeit not as much, and performance did not consistently improve. This suggests that data augmentation is not a universal solution, and other adjustments should be considered to combat overfitting

Precision improved slightly on *Non-demented*, but at the same time recall decreased significantly which is not desirable. The performance *Moderate* decreased heavily, with both precision and recall dropping. The two classes that did improve with data augmentation, interestingly, were *Very mild* and *Mild*. The precision on *Very mild* dropped from 0.52 to 0.33, but on the other hand recall increased from 0.29 to 0.55. Recall improved similarly on *Mild*, from 0.29 to 0.55 but without the precision decreasing.

Despite the regression of *Very mild* precision, I think these changes would be beneficial in a practical application. In a practical application these low severity classes would be the ones which the scan would try to detect. In this situation, the lower precision is outweighed by more instances being detected, thanks to the higher recall.

5.3 Future work

This study does not provide a definite guide for how to train a CNN on medical imaging. There exists an abundance of different ways to perform the training process, and only two were examined here. A more thorough evaluation of how to effectively train CNNs would be very useful in the development of future applications.

In addition, this study only used a single CNN structure. Different structures might perform differently, and react differently to various training schemes. An exploration of what, if any, effect the CNN structure has on the efficiency of the two training methods this report focused on would be needed to know if this studies' results hold in general.

Chapter 6

Conclusions

A convolutional neural network was found to be a powerful model for diagnosis of Alzheimer's disease using MRI images. Even though this study's CNN was not very advanced or developed for maximum performance, it still achieved decent performance on a problem that is difficult for non-trained humans. The way the model was trained was found to be an important factor of the network's performance.

Does class-weighting improve the performance on rarer classes?

This study finds that class-weighting brings clear performance improvements on the rarest class, tripling both precision and recall on the OASIS-3 dataset. This happens without worsening performance on the most common class. Performance on the two semi-rare classes did not see such performance improvements, with precision increasing only slightly and recall in fact decreasing slightly.

Does data augmentation improve general performance?

Data augmentation was not found to increase performance in general. Despite this, the recall on the two semi-rare classes was found to improve significantly, without at the same time hurting precision overly so. Data augmentation can however be done in many ways, so this result may not hold in general.

To conclude, both class-weighting and data augmentation were found to be useful techniques, but with some caveats. Their application does not guarantee universal performance improvements, instead increasing and reducing

performance to various degrees on different parts of the dataset. As such their usefulness will have to be evaluated on a case-by-case basis for each application.

Chapter 7

Acknowledgments and source code

Data were provided by OASIS-3: Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P50AG00561, P30NS09857781, P01AG026276, P01AG003991, R01AG043434, UL1TR000448, R01EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly owned subsidiary of Eli Lilly.

The source code used to run the experiments for this study can be found at <https://github.com/lundstig/kexjobb>.

Bibliography

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.
- [2] David Silver et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: *Science* 362.6419 (2018), pp. 1140–1144.
- [3] Bradley J Erickson et al. “Machine learning for medical imaging”. In: *Radiographics* 37.2 (2017), pp. 505–515.
- [4] World Health Organization et al. “Global action plan on the public health response to dementia 2017–2025”. In: (2017).
- [5] “2018 Alzheimer’s disease facts and figures”. eng. In: *Alzheimer’s & Dementia* 14.3 (2018), pp. 367–429. ISSN: 15525260.
- [6] Jyoti Islam and Yanqing Zhang. “A novel deep learning based multi-class classification method for Alzheimer’s disease detection using brain MRI data”. In: *International Conference on Brain Informatics*. Springer. 2017, pp. 213–222.
- [7] Jyoti Islam and Yanqing Zhang. “Early Diagnosis of Alzheimer’s Disease: A Neuroimaging Study with Deep Learning Architectures”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1881–1883.
- [8] Clifford R Jack Jr et al. “Brain beta-amyloid measures and magnetic resonance imaging atrophy both predict time-to-progression from mild cognitive impairment to Alzheimer’s disease”. In: *Brain* 133.11 (2010), pp. 3336–3348.
- [9] john c morris john c. “the clinical dementia rating (cdr): current version and scoring rules.” In: *neurology* (1993).

- [10] Imad A Basheer and Maha Hajmeer. “Artificial neural networks: fundamentals, computing, design, and application”. In: *Journal of microbiological methods* 43.1 (2000), pp. 3–31.
- [11] Sholom M Weiss, Ioannis Kapouleas, and JW Shavlik. “An empirical comparison of pattern recognition, neural nets and machine learning classification methods”. In: *Readings in machine learning* (1990), pp. 177–183.
- [12] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. “Introduction to multi-layer feed-forward neural networks”. In: *Chemometrics and intelligent laboratory systems* 39.1 (1997), pp. 43–62.
- [13] Rob A Dunne and Norm A Campbell. “On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function”. In: *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*. Vol. 181. Citeseer. 1997, p. 185.
- [14] Douglas M Kline and Victor L Berardi. “Revisiting squared-error and cross-entropy functions for training neural network classifiers”. In: *Neural Computing & Applications* 14.4 (2005), pp. 310–318.
- [15] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *CoRR* abs/1609.04747 (2016). arXiv: 1609.04747. URL: <http://arxiv.org/abs/1609.04747>.
- [16] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [17] Léon Bottou. “Stochastic Gradient Descent Tricks”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–436. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_25. URL: https://doi.org/10.1007/978-3-642-35289-8_25.
- [18] Keiron O’Shea and Ryan Nash. “An Introduction to Convolutional Neural Networks”. In: *CoRR* abs/1511.08458 (2015). arXiv: 1511.08458. URL: <http://arxiv.org/abs/1511.08458>.
- [19] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2009. ISBN: 978-0-521-86571-5. URL: <http://nlp.stanford.edu/IR-book/information-retrieval-book.html>.

- [20] S.K. Zhou, H. Greenspan, and D. Shen. *Deep Learning for Medical Image Analysis*. Elsevier Science, 2017. ISBN: 9780128104095. URL: <https://books.google.pt/books?id=WVqfDAAAQBAJ>.
- [21] Ajinkya More. “Survey of resampling techniques for improving classification performance in unbalanced datasets”. In: *arXiv preprint arXiv:1608.06048* (2016).
- [22] Rémi Cuingnet et al. “Automatic classification of patients with Alzheimer’s disease from structural MRI: a comparison of ten methods using the ADNI database”. In: *neuroimage* 56.2 (2011), pp. 766–781.
- [23] Saman Sarraf and Ghassem Tofghi. “DeepAD: Alzheimer’s Disease Classification via Deep Convolutional Neural Networks using MRI and fMRI”. In: *bioRxiv* (2016). DOI: 10.1101/070441. eprint: <https://www.biorxiv.org/content/early/2016/08/21/070441.full.pdf>. URL: <https://www.biorxiv.org/content/early/2016/08/21/070441>.
- [24] Iman Beheshti, Hasan Demirel, Alzheimer’s Disease Neuroimaging Initiative, et al. “Probability distribution function-based classification of structural MRI for the detection of Alzheimer’s disease”. In: *Computers in biology and medicine* 64 (2015), pp. 208–216.
- [25] Daoqiang Zhang et al. “Multimodal classification of Alzheimer’s disease and mild cognitive impairment”. In: *Neuroimage* 55.3 (2011), pp. 856–867.
- [26] Daniel S Marcus et al. “Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults”. In: *Journal of cognitive neuroscience* 22.12 (2010), pp. 2677–2684.
- [27] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
- [28] Pamela J LaMontagne et al. “OASIS-3: LONGITUDINAL NEUROIMAGING, CLINICAL, AND COGNITIVE DATASET FOR NORMAL AGING AND ALZHEIMER’S DISEASE”. In: *Alzheimer’s & Dementia: The Journal of the Alzheimer’s Association* 14.7 (2018), P1097.

