Supervised Pre-trained neural networks, classifying DNA amplification data

Andreas Ährlund-Richter

Arpad Szell

Department of Computer and Systems Sciences

Degree project 15 credits
Computer and Systems Sciences
Degree project at the bachelor level
Spring term 2020
Supervisor: Jaakko Hollmén
Swedish title: Klassifierande av amplifierad DNA data, genom kontrollerade och för-tränade neurala nätverk.

Stockholm University

Abstract

Deep neural networks are a popular technique for cancer classification but they are known to be sensitive to the initial settings. Trying to stabilize the network early by using a range of techniques called pre-training is worth exploring. A previous **supervised pre-training** method useful for noisy and small datasets was chosen. **A cancer dataset** collected from a compilation of cancer research results was identified as interesting. The dataset had previously been explored with unsupervised machine learning algorithms, but not with neural networks. DNA amplification is a condition of multiple copies of a region of DNA. The found dataset contained DNA amplification information in cancer patients. This cancer dataset was used in the study to investigate neural network classification accuracy on cancer data, and possible classification improvement using the chosen pre-training method. The chosen technique is based on creating data by shuffling the values in the columns, thus creating fake data from the real data. The fake and real data is labelled as new classes to classify. The network distinguishes the fake data from the real data, the trained model settings are transferred to another model. This model is used to classify the classes in the unmodified dataset. This pre-training algorithm was only tested on binary problems. A comparison using the supervised pre-training method with the whole, contra specific parts of the dataset was performed. **Multi and binary class experiments** were conducted. Solving multi-class problems using the supervised pre-training technique was novel for this study. Binary class experiments were conducted to compare results with previous studies. **Results** of the experiments show that the deep learning neural network had low accuracy on the dataset. Sample size greatly impacted the result, with larger sample size increasing accuracy. The suggested pre-training had slight improvement on binary, but harmed multi-class classification accuracy. In **conclusion** no clear benefit was found between pre-training and non-pre-training. However the sparse DNA amplification dataset might correspond to the only dataset that the previous supervised pre-training fared worse on.

Keywords: Supervised pre-training, Neural Networks, Cancer classification

Synopsis

Background	This thesis is largely built on recent research done on supervised pre-training and how it can be applied to neural networks to increase their performance. The supervised pre-training learns to identify real data from shuffled data. Study on this supervised pre-training shows that the performance increases when the datasets have higher dimensionality and noisy attributes and therefore it is interesting to apply it on a neural network that tries to classify on a DNA amplification dataset. The thesis is in the field of Data science as it compares neural network performance and various techniques of supervised pre-training.
Problem	DNA amplification datasets have high dimensionality, can be sparse and noisy. Supervised pre-trained neural networks can potentially classify such datasets well. It is interesting to investigate to what extent supervised pre-training can improve how well neural networks classify such datasets.
Research Question	The overarching question of the thesis is "To what extent can pre-training improve deep neural network classification accuracy on a DNA amplification dataset?". By further testing the supervised pre-training on multi-class classification problems and binary classification problems on the DNA amplification dataset. The thesis will try to provide a better insight to the limitations and capabilities of the pre-training technique and hopefully this technique can be further improved upon.

Method	An experiment is conducted where neural networks are trained and tested on a DNA amplification dataset. Both pre-trained and not pre-trained neural networks were compared, as well as multi-class classification and binary classification. The neural networks had to classify depending on different representation thresholds of the cancers. There were also experiments where the pre-training was allowed to use the full dataset. Data collection was performed through observation and descriptive statistics was used to analyse the data.
Result	There were cases where pre-training improved accuracy, the highest measured improvement was 9% on average with multi-class classification. The highest measured improvement on binary classification was 5%. The results show that supervised pre-training can potentially improve deep neural network classification accuracy but on average it is about equal in performance with no pre-training.
Discussion	The thesis main limitation is that the neural network used for the dataset was not properly designed for that problem which harmed how significant the results were. The limitations of the supervised pre-training technique becomes obvious and suggestions on how to further develop the pre-training technique to improve it are highlighted. Using the supervised pre-training on multi-class classification and on the DNA amplification set is the novelty of the thesis. The findings can help improve and help those that are developing deep neural networks for cancer classification.

Acknowledgement

Great thanks to Jaakko Hollmén for his helpful and encouraging supervision, and for working tirelessly even when sick, and even during the stressful times of the corona outbreak.

We would like to highlight the tremendous performance of Andreas laptop during testing, a grueling 38h nonstop testing phase.

Lastly we want to thank ourselves for not giving up even though the results were hard to digest.

Contents

List of Figures

List of Tables

List of Abbreviations

PT - Pre-training
Peng-PT - Pre-training of neural networks, using normal and shuffled data as used by Peng et al. (2019)
he_normal - a type of initialization method for weights in a neural network before training of the model.

Chapter 1

Introduction

Designing a system that can extract features from raw data which a machine can learn from, to classify or detect patterns is at the core of machine learning (LeCun et al. 2015). Neural networks, a type of machine learning, have gained popularity in recent years as it allows a machine to automatically discover the representation needed for classification straight from the raw input. Neural networks consist of multiple nodes arranged in layers. When classifying, the nodes pass values to each other adjusted by weights (Le et al. 2015). As the neural network trains on data, it adjusts the weights to get closer to the correct classification of the dataset records. Each layer transforms the value, given enough transformation even complex non-linear functions can be represented, and difficult classification problems can be solved (LeCun et al. 2015). Neural networks that have multiple layers are called deep neural networks (Deng & Yu 2014). For consistency, this thesis will use that definition of deep neural networks.

Due to the aforementioned strength of deep neural networks, the method has been applied to cancer data with great success. Daoud & Mayo (2019) summarized recent research that has used a neural network based cancer prediction model and categorized the functionality of neural networks as following: filtering methods, predicting methods and clustering methods. The summarization shows just how versatile and useful these methods can be with a complex dataset. One of the strengths of deep neural networks is the ability to discover complex structures in high dimensional datasets.

Cancer is a common genetic disease (Siegel et al. 2020). A common cause of cancer is DNA copy number amplification. DNA copy number amplification can be described as several repeated copies of the same gene sequence in a genome. This has been shown to play a role in cancer pathology (Myllykangas et al. 2008). In this paper, DNA copy number amplification is referred to as DNA amplification. Research on DNA amplification in different neoplasms has been conducted. Neoplasms are abnormal growth, with the more invasive or malign type being referred to as cancer (Cooper 1992). In this paper, all the neoplasms are referred to as being cancers. The DNA amplification cancer

dataset was created with the help of *bibliomics* survey (Myllykangas et al. 2006). Bibliomics is a thorough summary of data on a particular topic, in this case DNA amplification studies.

Amplified genes are a great target for diagnostics and prognostics. The amplification dataset that Myllykangas et al. (2006) created and used was scarce, with high dimensionality and some noisy attributes which limits what methods that could be applied. Scarcity refers to few records per cancer type, and sparsity low information in many attributes. High dimensionality means that the data has many attributes. Noisy attributes refers to information that does not contribute to classification of the records. Deep neural networks are especially suitable for high dimensionality and complex datasets, like the DNA amplification dataset (Hugo & Yoshua 2010).

Much research has been done to try to optimize the initial weights to speed up deep network training compared to shallow (Glorot & Bengio 2010). When initialized properly however, deep networks are only linearly slower to learn than shallow (Saxe et al. 2013). Peng et al. (2019) Shows that many common *initialization methods* are not necessarily optimal. Initialization refers to the method of setting the starting weights in the neural network, here referred to as initial weights. Peng et al. (2019) showed that inappropriate initial weights lead to a lower classification accuracy. A suggested solution for the problem is *supervised pre-training* of neural networks which generally enhances the performance (Peng et al. 2019). Supervised pre-training refers to using weights from a previous model trained on a similar dataset, as the initial weights. The results of Peng et al. (2019) show that their type of pre-training, here referred to as Peng-PT, gave improvements, compared to other methods. This occurred when the data was scarce, had similar classes, high dimensionality and noisy attributes. A weakness for the method, is redundant attributes, which can be copies of the same attribute, or very similar attributes. Lastly Pengt-PT was only tested on binary classification problems.

1.1 Problem

A couple of interesting problems have been lifted in the introduction. The high dimensionality, scarcity and sparsity in cancer data. Also, deep neural networks suitability for high dimensional datasets. Lastly, usefulness of Peng-PT for scarce, noisy and high dimensional datasets was highlighted. The sections below describe the problems more in depth.

1.1.1 Weight initialization

Several types of weight initialization exist, one is called pre-training. Pre-training belongs to two groups, supervised and unsupervised pre-training. Supervised uses a similar dataset to the dataset of interest to generate weight values. Unsupervised generates new values from a mathematical distribution or algorithm. An "optimally bad" neural network weight initialization method

was created in previous research by Peng et al. (2019). Shuffling the values in a dataset randomly, and then using this dataset for supervised pre-training. It was proven to be worse performing than both initializing with random weights, and common weight initialization algorithms (Peng et al. 2019). Peng et al. (2019) highlighted with this experiment the importance of good weight initialization.

1.1.2 Pre-training and limited record data

A supervised pre-training method was invented by Peng et al. (2019). It duplicated the data to classify, shuffled the values in one copy, creating fake data. A model was trained to classify the real from fake data. The weights from this model were used as the starting weights. Peng-PT proved to be an improvement on several classification tasks compared to baseline pre-training methods, as well as for artificial datasets with different amounts of noise distribution of attribute values and dataset size (Peng et al. 2019). The algorithm was never tested for *non-binary* classification problems, and performed badly on redundant datasets (Peng et al. 2019). Non-binary refers to classification problems with more than two classes to identify.

A benefit of Peng-PT was the pre-training utilizing a copy of the data with shuffled attribute values for each record. This duplication and pre-training using the regular and shuffled data copy, allows the data to be fully used. In regular neural network classification, only a part of the data is used, some for training and some for evaluating the accuracy of the model. The Peng-PT in effect, enables the model to train on all the data in the pre-training step. This is especially useful for datasets with few records.

1.1.3 Classifying medical datasets

The DNA amplification dataset provided by Myllykangas et al. (2006) has not previously been explored by classification algorithms. Instead different clustering algorithms have been applied to find correlations and causations in different cancer types. This thesis does not compare the usefulness of different classification algorithms, but does apply deep neural networks to investigate possible problem areas when applying classification on the dataset.

There is low representation of many cancer types in the DNA amplification dataset, with most being scarce, having 12 out of 73 being below 10 records in number, and only 13 being above 100 records. Few records is also a common problem in medical science at large, a literature review publication by Hudson (2000) found that most medical science studies are performed on classes having less than 10. In light of this, deep networks are especially interesting, as they perform better on scarce datasets than shallow networks do (Bengio et al. 2007). As stated earlier, pre-training used by (Peng et al. 2019) found good accuracy improvement on scarce datasets compared to conventionally trained deep neural networks. Another issue with the DNA amplification dataset is that it can be considered noisy (Myllykangas et al. 2006). The attributes are also sparse, meaning mostly no amplification for a gene. The Peng study finds

acceptable but not as impressive results classifying noisy sets with the described pre-training method (Peng et al. 2019). As previously mentioned Peng-PT did not manage redundant attributes (Peng et al. 2019). If the sparse attributes in the DNA amplification dataset can function like redundant ones are not previously investigated.

1.2 Research Question

This paper tries to answer an overarching question **"To what extent can pre-training improve deep neural network classification accuracy on a DNA amplification dataset?"** With the help of the following three questions:
1) How well does deep neural networks classify the DNA amplification dataset, with and without Peng-PT?
2) When using Peng-PT, is it preferable to use all the data when pre-training, or is it better to pre-train with filtered data, only containing evenly balanced classes of interest? Here referred to as unfiltered versus filtered pre-training.
3) How well does Peng-PT improve accuracy in binary versus multi-class classification tasks?

1.2.1 Classification of DNA amplification dataset

As discussed in section 1.1.3, the DNA amplification dataset has not been previously explored by classification algorithms. The benefit of increased accuracy on scarce data for Peng-PT neural networks is possibly applicable on this dataset. However as noted by Myllykangas et al. (2006), the dataset contains a lot of information classified as noise. This is because as previously mentioned the dataset is sparse with mostly non-amplified records for most attributes. Although noise is possibly managed well by Peng-PT according to the Peng et al. (2019) study, it is difficult to say if sparse attributes will behave as noisy, or redundant in the context of Peng-PT. Peng-PT was shown to be sensitive to datasets with attributes that are very similar, also known as redundant. Two attributes containing mostly zeroes, can possibly function redundantly (Peng et al. 2019).

1.2.2 Unfiltered pre-training

The DNA amplification dataset has noisy data, and is sparse, meaning a high proportion of genes not being amplified and not contributing to cancer classification. As the dataset is scarce, there are few records per cancer type. It is interesting to see if pre-training could use every available record to improve accuracy, even if DNA amplification between cancer types is highly varied, and class separation is high.

1.2.3 Pre-training on multi-class problems

The pre-training performed by Peng et al. (2019) was not investigated with multi-class prediction tasks. This thesis performs further research proposed by Peng et al. (2019), by comparing multi-class and binary classification on the Peng-PT pre-training accuracy improvement.

Chapter 2

Theoretical Background

2.1 Introduction to neural networks

Deep neural networks have been briefly touched upon in the introduction. It is important to define how neural networks work and explain what it means in the context of this thesis, to better understand the limitations and the benefits of deep neural networks. Both deep and shallow neural networks use a neural network as an architecture.

The network consists of layers where the initial layer is called an input layer, the last layer is the output layer and between these layers are the hidden layers which consist of the neurons, (see figure 2.1). Neurons are also called nodes. One can imagine the neurons to be nodes in a graph, and the edges that connect them have attached weights to them. The weight represents the strength of the connection between two nodes and thus a weight determines how much influence the input will have on the output of the receiving node (Le et al. 2015). Weight close to zero implies that changing the input will not affect the output, a negative value means that increasing this input will decrease the output (Alpaydin 2014).

The most important part of a neuron is the activation function. This function defines the output of a neuron given the weighted inputs. All neurons in the network have an activation function and as the neurons are connected to each other, meaning that the final output of a neural network is determined by the activation function. A classic example of an activation function is *sigmoid activation function* as it is common to use in classification problems. The output of a sigmoid activation is always 1 if the value given to the function is greater than 0.5, and it is 0 if the value is less than 0.5 (Le et al. 2015). The reason for the value 0.5, is because several perfectly linear activation functions become linear when combined, and can only solve linear problems. Although activation functions are usually homogeneous in the neural network, to get binary output to simplify interpretation of classification results, a *softmax* function is used in the final layer (Bishop 2007). Softmax function makes all the outputs sum up to 1, with all the inputs into the softmax being divided depending on their size.

For example, A softmax function with inputs 70 and 30, will have the output values 0.7 and 0.3. This attribute of a softmax node having several adjusted outputs makes it useful for multi-class classification problems as well.

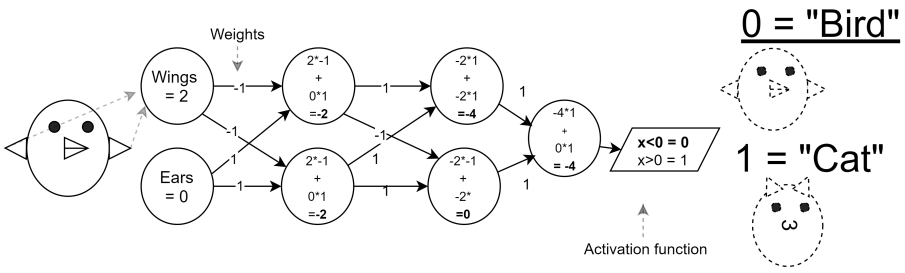

Figure 2.1: A simplified trained neural network.

Training neural networks

The main goal of training the neural network is to update and adjust the weights. After predicting a record, a loss function is used to calculate the difference between the real value, and the predicted value, this is the error. The error is propagated back using an algorithm called gradient descent. The gradient descent adjusts the weights between each layer to lower error at that layer (Russell & Norvig 2009). Gradient descent also takes something called the learning rate into account, and does not completely remove all error (Le et al. 2015). The learning rate determines how seriously the model takes the error. A high learning rate adjusts weights heavily in response to error, while a low learning rate makes smaller adjustments (Le et al. 2015). High learning rate can make the model learn quickly, but risks missing the optimal solution. One piece of the dataset of interest is used as a training dataset, and another piece of the dataset is set aside as a test dataset. A possible problem that can occur when the neural networks becomes too good at classifying the training dataset is referred to as overfitting (Alpaydin 2014). When a neural network is overfitted it is not general enough to be able to classify the test data. This is one of the reasons large datasets are useful, they reduce the risk of overfitting (Alpaydin 2014). Each step of training on a record and adjusting weights is called an epoch. Converging is when the modelling stops adjusting its weights after several epochs, as the same level of accuracy on the training is not surpassed after many new epochs.

2.1.1 Deep neural networks

The main advantage of a deep neural network is that it can solve more complex problems than shallower architectures (LeCun & Bengio 2007). Deep neural networks work on the principle that if you have a complex problem it is better to break it down into smaller problems that are solvable and merge the results

(Le et al. 2015). Another benefit is a reduction in needed nodes and data (LeCun & Bengio 2007).

A problem with gradient descent is more pronounced in deep neural networks. This causes both more steps to calculate, and potentially reaching very large or small values as the gradient progresses through the layers. The weights set at initialization can also affect the gradient more strongly, making initialization more important for deep networks. For deep neural networks issues with initialized weights or gradients will escalate for each layer, and the output will be too extreme to be able to classify a problem space (He et al. 2015). This is because when the error is propagated back through the neural network the weights are adjusted more in final layers of the neural network. The level of adjustment decreases for each layer away from the output layer (Nielsen 2015). As the activation functions in a node are often non-linear, small changes will barely register on the output from the node. This causes the output from nodes in layers far from the output to be virtually the same each epoch, and the adjustment keeps being small (Nielsen 2015). In the end, nodes close to the input layer might stop to learn completely and get stuck in the same position (Nielsen 2015). This is also referred to as vanishing gradient.

2.2 DNA amplification dataset

The DNA amplification dataset used for this study was first created and researched on by Myllykangas et al. (2006). The dataset was created to investigate DNA amplifications in different neoplasms. A neoplasm can be described as any new abnormal growth of cells in a specific part of the body, and is characteristic of cancers (Cooper 1992). The dataset was created from a bibliomics survey of 838 published chromosomal comparative genomic hybridization studies published between 1992 and 2002. Amplification data at chromosome band resolution was collected from more than 4500 cases. The amplification data in the dataset is presented in a binary fashion, a vector of zeros and ones, not degrees of amplification or number of repetitions of genes (Myllykangas et al. 2006). The study determined amplification patterns for 73 distinct neoplasms. In a later study in 2008 the data set was further explored with unsupervised learning algorithms. With the help of probabilistic clustering on each chromosome Myllykangas et al. (2008) identified 111 amplification models and divided the cancer cases into clusters, the new approach disregarded cancer type information and modelled amplifications based on case specific data vectors. The inherent structure in the clustering suggests that the amplifications are non-randomly selected according to the biological background of cancers (Myllykangas et al. 2008). The DNA amplification dataset can be described as a sparse dataset with high dimensions, and few records per cancer type. Sparse meaning mostly empty values, zeroes, or "not amplified" in the attributes. There is also a large class imbalance meaning some cancers have a larger representation and some cancers only have few cases. This can lead to some issues as they can be seen as noise, or create attribute redundancy, this needs to be accounted for.

2.2.1 Cancer prediction

Both binary class problems and multi-class problems for cancer classification have utilized neural network classifiers. The architecture that is applied can range from deep multi-layered models as shown in Mandal & Banerjee (2015) study, to single layered networks highlighted by Yen-Chen et al. (2014). The number of hidden layers or neurons to use has been observed to vary in the studies and the decision is often based on trial-and-error. For binary classification problems the neural network learns how to diagnose based on samples of cancerous or non-cancerous, alternative discriminate one type of cancer from another. The latter approach is more applicable to this thesis because the dataset only consists of cancers. Daoud & Mayo (2019) points out the output layer should consist of one or two neurons if the classification problem is binary. For multi-class problems the network learns how to discriminate between multiple types of cancer, Daoud & Mayo (2019) suggests an approach of having multiple binary classifiers to solve the problem.

2.3 Unsupervised and supervised pre-training

In neural network research, deeper networks using multiple layers were shown to perform better than shallow on high-dimension classification problems (Hugo & Yoshua 2010). Another benefit of deep neural networks was managing scarce datasets better than shallow networks (LeCun & Bengio 2007). Scarce datasets are datasets containing classes with few records. However, research also showed that deep networks relied more on good weight initialization. Bad initialization would cause failure to converge and learn, meaning failing to create a classification model for the problem (Glorot & Bengio 2010).

As stated in the introduction pre-training is a subtype of weight initialization of neural networks. Pre-training uses some form of training of a network, and then transfers the weights into the model that classifies the actual dataset of interest. Supervised pre-training uses some form of similar data to the data of interest in the pre-training step. Unsupervised pre-training uses different forms of regularization or restriction to capture values that are within the range of the input data to initialize the weights (Erhan et al. 2010). A common unsupervised weight initialization is He_normal. He_normal extracts random values from a normal distribution as the initial weights. The normal distribution used, is centered around 0, but has a standard-deviation of 2 divided by the number of input nodes for the node, squared (He et al. 2015). Weight initialization is also useful in context of activation functions. When researching activation functions, reliance on sigmoid activation units was shown to be less useful for deep-layers because of vanishing gradients. Binary activators were discussed, but ReLu activation functions were invented, and gained traction to be one of the more dominant activation functions (LeCun et al. 2015). ReLu converts any negative value to 0, but otherwise outputs the complete input value unmodified. ReLu however had a tendency to develop "dead" neurons or "dying ReLu", where

the neurons reach an output value of 0 and cannot be changed during training. This however was shown to be solvable with the correct weight initialization, at least for some training problems (Lu et al. 2019). At this stage, initialization was still unsupervised (Glorot & Bengio 2010).

Peng et al. (2019) did research on supervised pre-training. The purpose of the research by Peng et al. (2019) was not only to improve initialization, but also to improve generalization, making the network better at classifying. Also, improvement was found on the amount of data needed for classification. Some initialization-based accuracy improvements had been done using unsupervised methods (Dmytro & Jiri 2015). Research on transferability of weights between neural networks, found some positive effect of pre-training using datasets containing different classes, but similar types of data (Yosinski et al. 2014). However, the amount of data needed for accuracy was not investigated. The Peng-PT initialization was compared with the commonly unsupervised initialization he_normal as a baseline (Peng et al. 2019).

2.3.1 Peng supervised pre-training

The Peng 2019 research was performed due to classification accuracy improvement not being a previous focus in other pre-training papers. Previous focus had mainly been on pre-training preventing convergence issues or dead neurons in the neural networks. The supervised pre-training algorithm used by Peng (see figure 2.2) utilizes the complete dataset of interest for pre-training. A copy of the data has all attributes shuffled, to create an incorrect model of the problem environment, and is labelled as "fake". The rest of the data is labelled as "real". The real and fake data is combined into a pre-training dataset. The pre-training then involves performing a regular classification training on the pre-training dataset, classifying fake data from real data. The weights in the resulting models weights are transferred to the model used for the actual classification.

The first part of the Peng study also proves that initialization does matter for classification accuracy. By comparing a model where only "fake" data was used for the pre-training, while still classifying half of the data as "real". This "optimally bad" model was outperformed even by a completely randomized set of weights. The correctly used Peng pre-training improves classification for many different types of datasets, when compared to he_normal initialized models. The Peng-PT models performed better for almost all datasets explored, except for a "redundant" dataset. In the redundant dataset attributes were duplicated. The redundant dataset not only lowered the improvement effect of Peng-PT, but would stop Peng-PT models from converging. The other factors evaluated for Peng-PT models had generally good results. The factors were noise, distribution type, size of the dataset, and class separation. Class separation can be explained as how similar or dissimilar classes are between each other. A clear positive side-effect of using supervised pre-training with the dataset was an ability to perform better classification accuracy with less data (Peng et al. 2019). This is doubly positive, as deep neural networks are proven to be more

sensitive to initialization, but also better at predicting with less available data than shallow networks (LeCun & Bengio 2007). There was no clear increase in computational time for pre-training and convergence compared to he_normal initialized classification running time for reaching convergence. Another issue lifted by Peng was the fact that the initialization has just been tested for binary problems and not multi-class.

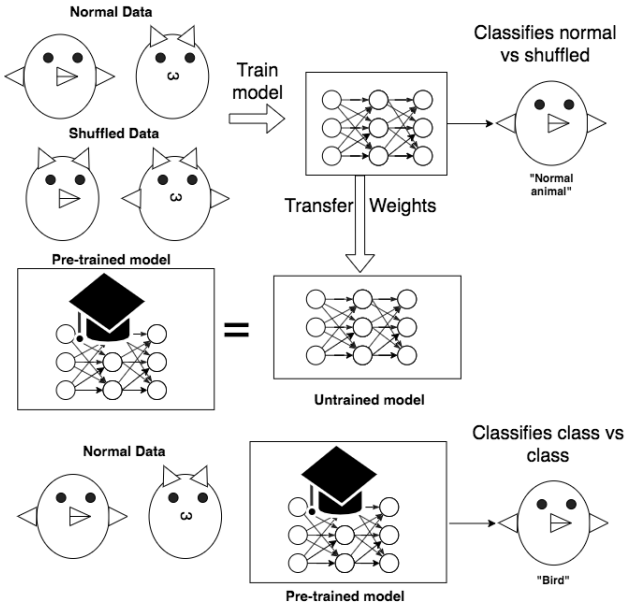

Figure 2.2: Peng-PT Pre-training. The animals displayed represent real records of known classes. These are shuffled into new fake animals or records using existing attributes like wing or ears. The classifier trains to identify the real versus shuffled data. The weights are then transferred to a new model, this model is now a Peng-PT model. The model can then more accurately classify the specific classes in normal non-shuffled data.

Chapter 3

Methods

3.1 Research strategy

The overarching research question for this thesis is "To what extent can pre-training improve deep neural network classification accuracy on a DNA amplification dataset?". As such it is fitting to conduct an experiment, because it investigates cause and effect relationships (Johannesson & Perjons 2014). Additionally Peng et al. (2019) used an experiment in their study which further encourages the usage of experiment because the focus of this thesis is applying Peng-PT. Having a dataset available and computational experiments not costly in further material, it is reasonable to argue that an experiment is more fitting from a test-material perspective as well.

In an experiment you have an independent variable and a dependent variable, as Johannesson & Perjons (2014) defines it, the independent affects one or more dependent variable. In this study the independent variable is pre-training and the dependent variable is the test-accuracy received from the neural network. It is important to mention that the results of an experiment can only increase or decrease support for a research question (Johannesson & Perjons 2014). As the main goal of an experiment is to study a cause and effect relationship it is important that other factors do not interfere and invalidate the result and therefore the experiments need to be conducted carefully (Johannesson & Perjons 2014). A drawback to experiments is that the result that it produces runs the risk of not being generalised, which harms the thesis (Johannesson & Perjons 2014). A benefit of experiments is that it is repeatable (Denscombe 2014).

An alternative research strategy could be a document survey, the aim of a survey as a research strategy is to map out some world (Johannesson & Perjons 2014). As defined by Johannesson & Perjons (2014) a survey is a good research strategy for collecting data on narrow and well defined topics. However, surveys are not suited for studying complex phenomena in greater detail. A document survey would use data gathered from other articles, and as it has been observed there is a lot of research done on the topic of neural networks and cancer pre-

diction however, there has been less research conducted on pre-training. As the overarching research question "To what extent can pre-training improve deep neural network classification accuracy on a DNA amplification dataset?" is well defined, a document survey can help map out how pre-training has been used, what were the limitations and how it fared on improving the results. A strength of survey as research strategy is that it enables both quantitative and qualitative data to be collected in an inexpensive way (Johannesson & Perjons 2014). A challenge with this strategy would be to get a good enough sample size that accurately represents research conducted with pre-training on deep neural networks on a dataset that is similar to the DNA amplification dataset. This strategy is suitable when an actual dataset to test on is not available.

3.1.1 Data collection method

Johannesson & Perjons (2014) mentions several data collection methods such as interviews, questionnaires, documents and observation. Neither interviews or questionnaires are feasible for this experiment as the topic at hand is novel and narrow. This makes it hard to find suitable persons to conduct interviews that will give enough information to answer the research question. Documents can be a reasonable choice as discussed in the research strategy section, however not applicable to experiments as research strategy. Lastly, the most flexible would be an observation. The method is versatile and can be applied to many research strategies (Johannesson & Perjons 2014). A systematic observation approach is suitable for collecting the necessary data as it is structured and rigorous, which in turn helps produce reliable and objective results (Johannesson & Perjons 2014). A drawback with this data collection method is that the data may be superficial and as Johannesson & Perjons (2014) explain that this method can bias researchers and in turn only focus on easily observed events in isolation and miss the context of the events.

3.1.2 Data analysis method

The data collected from the observations will be of quantitative nature. Descriptive statistics will be used as the data analysis method as it is a well suited method for describing a sample of data (Johannesson & Perjons 2014). Descriptive statistics describe the collected data sample with the help of tables, charts and various aggregate measures such as mean, ration and median to name a few (Johannesson & Perjons 2014).

An alternative analysis method could be inferential statistics, this method aims to reach a conclusion, as Johannesson & Perjons (2014) summarises, it is used for making inference from collected data to more general conditions. Because of the novelty of the Peng-PT and that the study aims to describe and summarise how Peng-PT neural networks fare compared to non-pre-trained neural networks it is more fitting to use descriptive statistics instead.

3.1.3 Ethical considerations

When choosing a research strategy, data collection method and data analysis method one has to consider ethical aspects (Denscombe 2014). This experiment uses patient data collected from previous studies, however the data cannot be traced back to any individual. The data collected and analysed will only regard the neural networks performance. Therefore no ethical consequences regarding human life can be found.

3.2 Application of method

Test Dataset

The dataset used consists of 4590 patients. Amplification status of 393 specific areas covering several genes was recorded for each patient (Myllykangas et al. 2006). There are 73 different cancer classes represented in the dataset. The dataset has a skewed class distribution, with the average representation being 1.3% of the records/patients. The median cancer class composed 0.5% of the dataset. Twelve percent (12%) of the datasets entries belonged to the maximally represented class. For a visualization, (see fig 3.2).

Data pre-processing

For each experiment a dataset with equal class distributions was used, with 50, 100 and 250 records per class classified. Pre-training with and without using the complete data set was also conducted, categorized as "filtered" and "unfiltered". Unfiltered pre-training uses all the classes in the dataset to pre-training the model. Filtered pre-training only uses classes of interest in the dataset, with balanced class-distribution. (see figure 3.1).

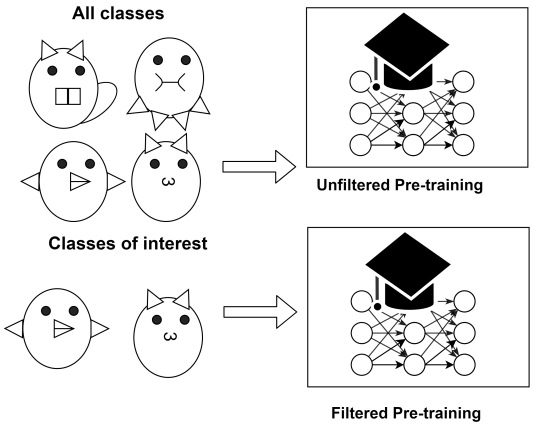

Figure 3.1: Unfiltered pre-training uses all the classes in the pre-training method. While filtered only uses the classes of interest.

Training and test dataset

Thirty percent (30%) of the data was used as the test set. The remaining data was split into labelled training data (21%) and unlabelled data in concordance with Peng et al. (2019).

Setup of experiment

The neural network architecture is made as similar as possible to the one used in the study by Peng et al. (2019). The neural network has, input, output, and five hidden layers, fully connected. Each hidden layer consists of 1024 nodes. The activation function is ReLU, except for the output layer that uses a softmax function. The input layer contains as many nodes as attributes, meaning it will always be 393 nodes. Standard gradient descent is used with constant learning rate 0.01. Number of epochs was 1000 and set to convergence for both pre-trained and non-pre-trained methods. Epochs were however only 10% of Peng et al. (2019) study epochs, Peng used 10000 and 50000 epochs. The neural network has converged if the training accuracy reaches 100% or the training loss stops decreasing, with 20 additional runs of non-improving epochs allowed before convergence being counted as reached. This value is referred to as "patience". All layers except the output layer were transferred from pre-training to the classifier. The output layer was initialized with he_normal and consisted of two neurons for binary experiments and for multi-class experiments the output layer had as many neurons as classes to classify.

Table 3.1: Experiment dataset types

Pre-training data	Representation Threshold	Binary/Multiple
unfiltered	50	Binary
filtered	100	Multiple
...	250	...

3.2.1 The experiment

Johannesson & Perjons (2014) points out that it is important to show that a single factor has a certain effect on another factor, however there are risks that other factors in the experiment disrupt the results. The experiments in this study were conducted on the same machine, the same algorithms for pre-processing the dataset was used and the same general neural network architecture was used. By being consistent with how the experiment is conducted the risk of other factors disrupting the result was minimized. Each experiment was run ten times to enable a good sample size for the data analysis. The multi-class experiments consisted of three classes for simplicity and time constraints reasons.

3.2.2 Data collection from experiments

The test accuracy for each experiment was collected through observing what the neural network evaluation outputs. Accuracy is a common metric and it evaluates the overall efficiency of an algorithm (Akosa 2017). As highlighted by Akosa (2017), accuracy can be a misleading evaluation measure when the classes are imbalanced. Due to the equalization performed for all class distributions that the neural network tries to classify class imbalance is not an issue. Additionally, accuracy as a metric works for both binary and multi-class classification problems while other metrics are more specialized towards one of the classification problems. Using accuracy will enable a more fair comparison for the two classification problems at the cost of not having a more nuanced evaluation of the performance.

3.2.3 Data Analysis

The data collected was parsed and summarized using a self written python script. The script calculates the following aggregate measures, mean, min and max accuracy. Min and max were picked out for all the experiment types. The results from the script were then transferred to an Excel spreadsheet. With the help of Excel a final column describing the ratio was created. Ratio is calculated by dividing the mean accuracy of two independent variables, in this case pre-training and no pre-training or unfiltered pre-training and filtered pre-training. The ratio provides a metric to see how much accuracy was gained or lost. All the descriptive statistics is summarized in a table to create a clear visualization of the results as suggested by Johannesson & Perjons (2014).

3.2.4 Tools

Python with the package ScikitLearn was used as programming environment to keep implementation similar to the settings in the study by Peng et al. (2019). The neural network was implemented with Keras and Tensorflow. All code, datasets, and results, can be found at *https://github.com/AndreasAAR/ExamThesis*. The aim being reproducibility and transparency, in line with the benefits of the experiment stated by (Denscombe 2014).

3.2.5 Research ethics

To reiterate, all data gathered for the dataset is patient data, the data has been collected from published findings in journals. Connection with patients is therefore untraceable. Out of the ethical considerations mentioned by Denscombe (2014). There are no apparent ethical issues regarding how the experiment was conducted and how the dataset was used, the thesis has no financial funding or support.

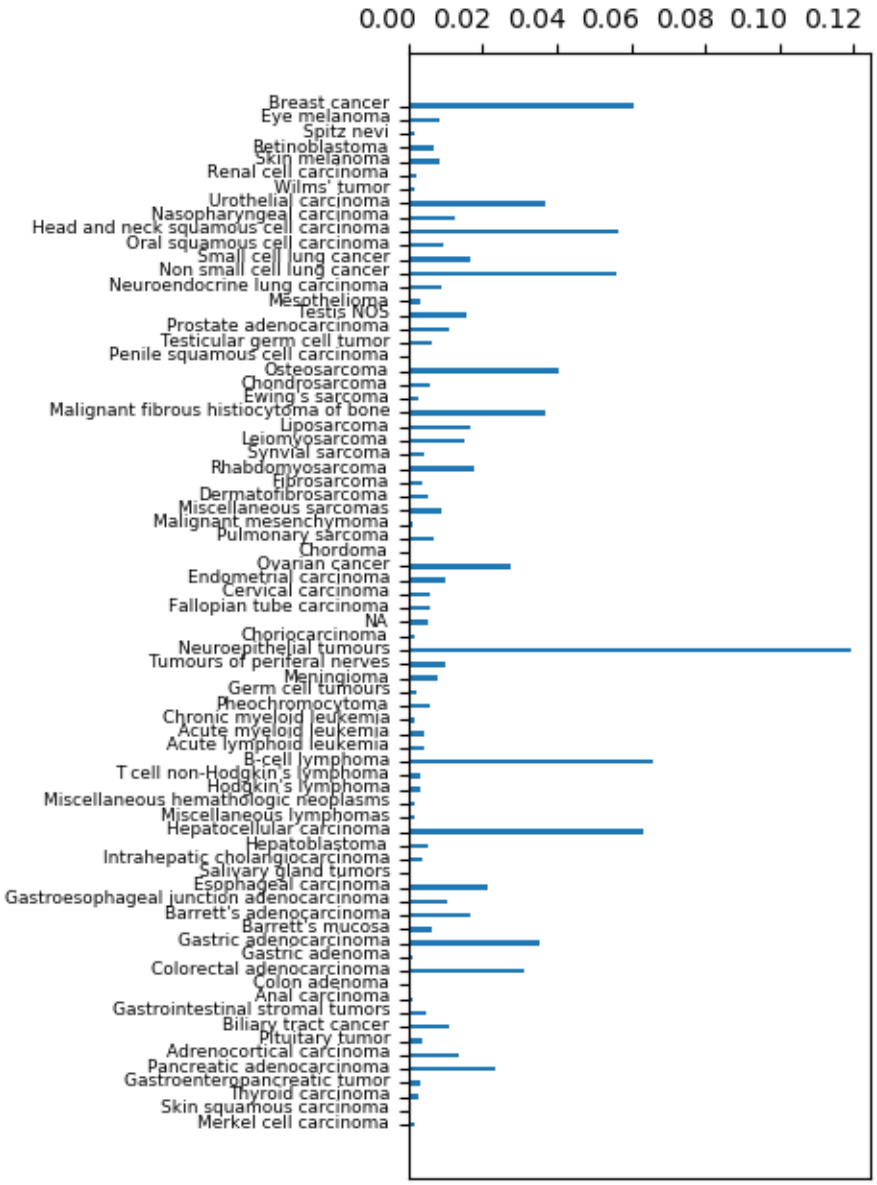

Figure 3.2: Class divison in dataset

Chapter 4

Results

This section will emphasize on mean, min, max and ratio of the classification accuracy. These metrics are easy to interpret and they give a good overview of the general performance. Ratio enables a fair comparison of results and gives a good indication of how much of a performance was gained or lost when the independent variable was changed. The tables in this section use bold to highlight best results and italic together with underline for worst results.

4.1 Overall results

The worst measured average results were 8.8% accuracy for unfiltered pre-trained multi-class classification with sample size of 50. The best average accuracy was 76.87% experiment using filtered pre-training with sample 250 on a binary classification problem. However, the average max result for each experiment type was similar for pre-training and no pre-trained model. Instead when looking at the ratios for pre-training compared to no pre-trained average accuracy for all experiment types, the highest improvement by pre-training was 9% for filtered pre-trained multi-class experiment with a sample size of 250.

4.2 Binary test cases

Looking at the binary test cases, the differences between pre-training and no pre-training were small. The largest performance increase using pre-training versus no pre-training was 5% for the experiment with filtered pre-training and sample size of 250. The average results of having pre-training was 54.50% and the average results for no pre-training was 54.44%. Binary test cases with sample size of 50 had the worst accuracy overall. Unfiltered pre-training with sample size 50 had the lowest accuracy of 16.67%. The lowest accuracy for no pre-training was on sample size 50 which measured 23.33%. The highest measured accuracy was 84%, for no pre-training with a sample size of 250. For

pre-training the highest measured accuracy was 82.67% which was measured on both unfiltered and filtered pre-training with a sample size of 250.

Table 4.1: Binary test results

Case	noPTmean	noPTmin	noPTmax	PTmean	PTmin	PTmax	PT/noPT Ratio
Unfiltered 50	*32.33*	23.33	40.00	33.33	*16.67*	40.00	1.03
Filtered 50	34.67	33.33	40.00	31.67	20.00	40.00	*0.91*
Unfiltered 100	56.17	48.33	60.00	56.67	60.00	56.67	1.01
Filtered 100	56.67	48.33	58.33	53.67	46.67	58.33	0.96
Unfiltered 250	74.27	60.67	80.00	74.80	62.00	82.67	1.01
Filtered 250	73.53	62.00	**84.00**	**76.87**	71.33	82.67	**1.05**
Mean	54.44	46.00	60.39	54.50	44.72	60.06	0.99

4.3 Multi-class test cases

Multi-class test cases also had marginal differences in average accuracy. The highest measured accuracy was 65.67% with no pre-training and sample size 100. The highest measured accuracy with pre-training was 64.89%, a filtered pre-training with sample size of 250. An interesting observation is that on sample size of 50 the pre-training harmed the results significantly, unfiltered measuring an accuracy of 21.11% and filtered an accuracy of 20.44%. The average accuracy for no pre-training was 44.04% and with pre-training it was 43.22%.

Table 4.2: Multi-class test results

Case	noPTmean	noPTmin	noPTmax	PTmean	PTmin	PTmax	PT/noPT Ratio
Unfiltered 50	26.00	17.78	42.22	21.11	*8.89*	53.33	0.81
Filtered 50	28.44	15.56	53.3	*20.44*	11.11	24.44	*0.72*
Unfiltered 100	47.44	42.22	56.67	47.89	38.89	56.67	1.01
Filtered 100	47.34	33.33	**65.67**	49.44	43.33	55.56	1.04
Unfiltered 250	58.76	52.89	63.56	57.91	48.89	64.44	0.99
Filtered 250	56.27	43.11	63.11	**61.29**	54.22	64.89	**1.09**
Mean	44.04	34.15	57.43	43.01	43.22	53.22	0.94

4.4 Unfiltered pre-training compared to filtered pre-training

On average the difference between unfiltered and filtered pre-training was small, 48.90% for filtered and 48.62% for unfiltered. As the sample size increased the ratio decreased. For sample sizes of 50 and 100, unfiltered pre-training increased accuracy with a ratio of 3-6%. Multi-class with 100 samples showed that filtered pre-training was better with 3%. Both binary and multi-class classification had ratio 3% and 6% respectively, for filtered pre-training on sample size of 250. The highest total accuracy was 82.67% on unfiltered pre-trained binary

classification with a sample size of 250. For filtered pre-training the highest measured accuracy was 79.33% on binary classification with sample size 250. Unfiltered pre-training also had the lowest accuracy of 8.88%, a multi-class classification with sample size 50.

Table 4.3: Unfiltered pre-training compared with filtered

Case	FilteredMean	FilteredMin	FilteredMax	UnfilteredMean	UnfilteredMin	UnfilteredMax	Ratio D/C
Binary 50	31.67	20.00	40.00	33.33	16.67	40.00	1.05
Multi 50	*20.44*	11.11	24.44	21.11	*8.89*	53.33	1.03
Binary 100	53.67	46.67	58.33	56.67	51.67	60.00	**1.06**
Multi 100	49.44	43.33	55.56	47.89	38.89	56.67	0.97
Binary 250	**76.87**	71.33	79.33	74.80	62.00	**82.67**	0.97
Multi 250	61.29	54.22	64.89	57.91	48.89	64.44	*0.94*
Mean	48.90	41.11	53.76	48.62	37.84	59.52	1.00

Chapter 5

Discussion

In general the performance of the neural networks was poor and many tests had an accuracy around 50%. However, it was not in the studies focus to create an optimized neural network for the DNA amplification set for all different sample sizes. Rather, the aim was to compare and investigate Peng-PT, and investigate neural network classification on the DNA amplification dataset. A trend in the results is that with a larger sample size the accuracy improved across all the different variations of pre-training and even reaching some good accuracy of 70-80%. This is in line with how deep neural networks function, that the model improves with more data at hand. The neural network used had a similar architecture of what Peng et al. (2019) used, this enables a more fair comparison. Instead using a more thought out and better designed neural network that fits the DNA amplification dataset would have probably increased the accuracy, and might have given more significant results when applying the pre-training. To optimize the pre-training a better stopping condition could have been used, also picking certain layers that performed well could improve the results.

Dimensionality

The DNA amplification dataset is very sparse compared to even the intentionally noisy set in Peng et al. (2019), with 20 real attributes and 20 noisy attributes, compared to the 393 attributes in the DNA amplification dataset. The model trained on the noisy dataset in Peng et al. (2019), had 10 examples to classify, and had an accuracy of 82-86%. This is clearly superior to the 40-60% smaller 50 record experiments on the DNA amplification set. However the ratio of pre-training to no pre-training is similar to what was found for the noisy dataset. Also, the dimensionality between the sets is different, with almost 20 times more attributes in the DNA amplification dataset. This could indicate that the DNA amplification dataset is more difficult to classify, but Peng-PT could arguably give about the same improvement. As previously mentioned, the worst result in the Peng-PT study was the performance on redundant datasets (Peng et al.

2019). This is in line with the concern in one sub research question, that the DNA amplification dataset can function as an extremely redundant set. As most attributes will be 0 for most records, this sparsity could act similar to the duplicated columns in the Peng et al. (2019) redundant dataset. This could explain why no pre-training sometimes outperformed pre-training.

Binary classification

Generally, the average difference between pre-training and no pre-training was small, however this was expected. Looking at the results from Peng et al. (2019) their highest measured accuracy was 33.3% with pre-training on a small dataset with noisy attributes. The best ratio measured for binary pre-training in this study was 5% which is in accordance with the findings of Peng et al. (2019). The results on the binary test cases varied and a pre-training did not always improve the classification on the DNA amplification dataset. When looking at the difference between a unfiltered pre-training and no pre-training it showed that unfiltered pre-training on average improved the accuracy by 1-3% for all the sample sizes. A possible explanation for this difference in performance is that the unfiltered pre-training uses the whole dataset for pre-training which feeds the neural network more data to train on and it can better generalize.

Multi-class classification

The highest measured increase in performance between pre-training and no pre-training in this study was 9%, multi-class classification with filtered pre-training with sample size of 250. However no pre-training generally outperformed pre-training on multi-class classification. The pre-training technique used for multi-class was identical to the one in binary testing, and the way the multi-class neural network was designed it probably could not utilize the weight that well and that is in accordance with the findings is Peng et al. (2019) where it is emphasised the importance of good initialization. The architecture of the neural network was planned to be as Daoud & Mayo (2019) suggested, where you have multiple independent binary neural networks that together can do multi-class classification, however due to technical limitations and time constraints such a network could not be successfully implemented. It is an interesting idea to explore how well the Peng-PT would improve such a solution, but seeing the small differences in improvement one can think that those small improvements stacked on top of each other might be more significant and help when classifying multiple classes in an ensemble fashion.

Unfiltered pre-training

The idea with the unfiltered pre-training was that it uses the whole dataset and with it comes more data, especially of the scarce cancer types. The unfiltered pre-training outperformed filtered pre-training on cases where the sample size was 100 or below on binary cases. This could be because unfiltered gives a

larger dataset to train with when the filtered version is limited for the small sample experiments. However, when the sample size was 250 it was preferred to have filtered pre-training and especially on multi-class classification. The DNA amplification set has a high variance in class distribution and this introduces the risk of the pre-trained model overfits because it learns the noise and therefore cannot be generalized well enough to successfully be applied on the test data. Another explanation is that the limitations of Peng-PT also applies to this variation of the pre-training, as Peng et al. (2019) points out that this method works best when class separation is small, which is not the case in the DNA amplification set.

Something worth discussing is that Myllykangas et al. (2008) could perform a clustering on some amplification profiles which implies that some cancer types are somewhat similar. The unfiltered pre-training could have found some of these patterns as well, and it might be an explanation why unfiltered pre-training sometimes had a better accuracy. The findings of these patterns would help the neural network generalize, which also explains why unfiltered pre-training generally performed worse with a bigger sample size. As sample size increases, the general model that the neural network uses is too general, and it cannot distinguish well enough. It was also observed that the difference between lowest measured accuracy and mean accuracy for unfiltered pre-training with low sample size was high. A reason for this can be that there were cases when the neural network distinguished these patterns and relied heavily on them and that in turn led to underfitting.

Impact of equalize

By always equalizing the class distribution to each other in the experiment datasets, it becomes easier to compare no pre-training with pre-training. However equalizing removes much of what makes the actual classification of cancer types hard. An equal class distribution is not common in cancer datasets and this is also true for the DNA amplification dataset. Equalization helped improve the reliability of the experiments. It could be argued that unequal class distribution accuracy would have been interesting to investigate, but due to time-constraints this could not be performed. Also, the unfiltered pre-training could compare well to a class-imbalanced pre-training scenario.

5.1 Limitations

Validity

Denscombe (2014) discusses validity in terms of internal and external factors and that validity refers to relevance and precision of the data. Internal validity refers to whether the right questions have been asked (Denscombe 2014). This study has used accuracy as a measurement to answer the research question. This is in line with metrics used by Peng et al. (2019). However, only using accuracy was a naive approach which probably harmed the internal validity.

Using metrics that are not as general would have helped drawing conclusions and give more nuance to the results on how the neural networks fared on the DNA amplification dataset. Denscombe (2014) highlights an aspect of external validity which is how the data compares to other studies in the field. This study has achieved similar results as the Peng study.

Reproducibility

By source code, data and results being accessible for public use on github, the study becomes reproducible and transparent.

Reliability

Reliability is a concept that refers to if the measuring instrument is neutral and would make the same measurements for the same situation across multiple occasions (Denscombe 2014). The measuring tool used is a predefined and well tested algorithm and therefore it should be consistent over multiple occasions. The results in this study are also in accordance with the findings of Peng et al. (2019) which strengthens that the results are reliable.

Generalizability

Denscombe (2014) highlights properties of generalizability which regards that the results from the study can be used to draw conclusions on a wider population. This study shows that the Peng-PT has the potential to increase performance in cancer datasets. However, the results in this study alone are not sufficient enough to be able to draw conclusions on a wider population. Because the dataset is unique and further research has to be performed with Peng-PT.

Credibility

Credibility of results regards to how free of bias and errors they are (Denscombe 2014). As much human error as possible has been avoided by using scripts to summarize and analyze the data. The researchers have no political or financial interest.

5.2 Results research ethics

According to the paragraph 16 in the Helsinki Declaration, harm and benefits to human life should be compared (World Medical Association 2013). The authors would argue that it is clear that the nonexistence of harm to patients, makes even a small potential contribution to deep learning and cancer classification a good motivation for this study being ethical. Cancer classification could potentially save lives, and deep learning has many potentially positive applications. The Association for Computing Machinery mainly argues for professional conduct rather than science, however, their ethical conduct could be argued to be

applicable to computer science as well. In the ACM Code of Ethics and Professional Conduct, 1.6 Respect privacy, it is argued that only the minimum amount of personal information necessary should be collected in a system (ACM Council 1992). Although the results and data have been published on Github, only the amplification dataset is available. The DNA amplification dataset is from a bibliomic study, making it impossible to trace.

5.3 Conclusions

One of the sub research questions was **classification accuracy improvements using Peng-PT** when applied to classifying the DNA amplification dataset. A deep learning neural network as conducted in this study with and without pre-training fared poorly on the DNA amplification dataset. The sample size to classify greatly impacted the results and a larger sample size often had higher measured accuracy. A novel technique tested in this study was **unfiltered pre-training**. Unfiltered pre-training had a slight accuracy improvement on binary classification with a low sample count. Multi-class classification accuracy was harmed by unfiltered pre-training. A sub research question was to investigate **multi-class classification with Peng-PT**. Generally it did not improve multi-class classification results, however it is worth noting that pre-training did improve accuracy by 9% on average when sample size was the largest. Binary classification did have more cases where pre-training improved the accuracy, however these improvements were only 1-5%. In conclusion the results of this study are not conclusive enough to answer the research question **"To what extent can pre-training improve deep neural network classification accuracy on a DNA amplification dataset"**.

5.4 Future Research

The potential of supervised pre-training as suggested by Peng et al. (2019) is a promising idea. The results from this study were too inconclusive which is dependent on many factors and one of those factors is probably the size of the dataset. To have more significant results the DNA amplification dataset can use simulated data to increase the representation of cases which would favour deep neural networks. As mentioned in the discussion, optimizing the pre-training phase more by cherry picking layers that are performing well and having better stopping conditions is an interesting area to explore as it would probably lead to more significant results. Further developing the pre-training technique to also apply the pre-trained output layer might be of interest to do. Reducing redundancy or using a different dataset would be interesting from a computer sciences perspective to more fairly investigate if multi-class classification is improved by Peng PT, and to what extent. In a cancer science perspective, it would be interesting to reduce the redundancy of the DNA amplification set, to see if cancer classification can be improved when using pre-training.

Bibliography

ACM Council (1992), 'ACM code of ethics and professional conduct'.

Akosa, J. (2017), Predictive accuracy: a misleading performance measure for highly imbalanced data, *in* 'Proceedings of the SAS Global Forum', pp. 2–5.

Alpaydin, E. (2014), *Introduction to Machine Learning*, The MIT Press.

Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. (2007), Greedy layer-wise training of deep networks, *in* 'Advances in neural information processing systems', pp. 153–160.

Bishop, C. M. (2007), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1 edn, Springer.

Cooper, G. M. (1992), *Elements of human cancer*, Jones & Bartlett Learning.

Daoud, M. & Mayo, M. (2019), 'A survey of neural network-based cancer prediction models from microarray data', *Artificial intelligence in medicine* .

Deng, L. & Yu, D. (2014), Deep learning: Methods and applications, Technical Report MSR-TR-2014-21, Microsoft.
URL: *https://www.microsoft.com/en-us/research/publication/deep-learning-methods-and-applications/*

Denscombe, M. (2014), *The Good Research Guide: For Small-Scale Social Research Projects*, Open UP study skills, McGraw-Hill/Open University Press.

Dmytro, M. & Jiri, M. (2015), All you need is a good init, Vol. 1511.06422, ICLR.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P. & Bengio, V. (2010), 'Why does unsupervised pre-training help deep learning?', *Journal of Machine Learning Research* pp. 625–660.

Glorot, X. & Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks, *in* 'Proceedings of the thirteenth international conference on artificial intelligence and statistics', pp. 249–256.

He, K., Zhang, X., Ren, S. & Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 1026–1034.

Hudson, C. (2000), *The Sciences of the Artificial intelligence for biomedical engineering*.

Hugo, L. & Yoshua, B. (2010), An empirical evaluation of deep architectures on problems with many factors of variation.

Johannesson, P. & Perjons, E. (2014), *An introduction to Design Science*, first edn, Springer.

Le, Q. V. et al. (2015), 'A tutorial on deep learning part 1: Nonlinear classifiers and the backpropagation algorithm', *Google Inc., Mountain View, CA* p. 18.

LeCun, Y. & Bengio, Y. (2007), 'Scaling learning algorithms towards AI', *MIT Press* .

LeCun, Y., Bengio, Y. & Hinton, G. (2015), 'Deep learning', *nature* **521**(7553), 436–444.

Lu, L., Yeonjong, S., Yanhui, S. & Karniadakis, G. E. (2019), 'Dying relu and initialization: Theory and numerical examples', *ArXiv* **abs/1903.06733**.

Mandal, S. & Banerjee, I. (2015), 'Cancer classification using neural network', *International Journal* **172**.

Myllykangas, S., Himberg, J., Böhling, T., Nagy, B., Hollmén, J. & Knuutila, S. (2006), 'DNA copy number amplification profiling of human neoplasms', *Oncogene* **25**(55), 7324–7332.

Myllykangas, S., Tikka, J., Böhling, T., Knuutila, S. & Hollmén, J. (2008), 'Classification of human cancers based on DNA copy number amplification modeling', *BMC medical genomics* **1**(1), 15.

Nielsen, M. A. (2015), *Neural networks and deep learning*, Vol. 2018, Determination press San Francisco, CA, USA:.

Peng, A., Koh Yun, S., Riddle, P. & Pfahringer, B. (2019), *Using Supervised Pretraining to Improve Generalization of Neural Networks on Binary Classification Problems: Recognizing Outstanding Ph.D. Research*, pp. 410–425.

Russell, S. & Norvig, P. (2009), *Artificial Intelligence: A Modern Approach*, 3rd edn, Prentice Hall Press, USA.

Saxe, A. M., McClelland, J. L. & Ganguli, S. (2013), 'Exact solutions to the nonlinear dynamics of learning in deep linear neural networks', *Computing Research Repository* **abs/1312.6120**.

Siegel, R. L., Miller, K. D. & Jemal, A. (2020), 'Cancer statistics', *CA: A Cancer Journal for Clinicians* **70**(1), 7–30.

World Medical Association (2013), 'World medical association declaration of Helsinki: ethical principles for medical research involving human subjects', *JAMA* **310**, 2191–2194.

Yen-Chen, C., Wan-Chi, K. & Hung-Wen, C. (2014), 'Risk classification of cancer survival using ann with gene expression data from multiple laboratories', *Computers in biology and medicine* **48**, 1–7.

Yosinski, J., Clune, J., Bengio, Y. & Lipson, H. (2014), How transferable are features in deep neural networks?, *in* 'Advances in neural information processing systems', pp. 3320–3328.

Appendix A

Reflection Document Arpad

Our study achieved many of the goals that regard finding scientific literature, criticizing it and implementing relevant methods. The thesis leans heavily on previous study and how the findings of those studies might be applicable to the dataset we have. Finding relevant papers was something that actively tried to do. It also became apparent that it requires a lot of knowledge for successfully implementing neural networks. Seeing the knowledge gap in our own work is something we tried to emphasize and highlight in the discussion section. Ethical issues and reflection of those were limited in this study because of the nature of the dataset and the experiments, I do believe those goals were achieved however not to the same extent as many others.

In general the planning was good. Me and Andreas have collaborated on multiple larger projects which helped with understanding each other and divide and plan the workload. Having many fixed small deadlines set by our supervisor helped a great deal to keep up the pace of the writing. The deadlines help with planning against other courses as well. If we saw a gap in a course we would take the opportunity to get ahead in the thesis writing which made the process much smoother. In general I am very content with how the planning went as most of the work was done within the goals that our supervisor layed out or even before. However, worth mentioning that we had a goal of getting a draft ready for first review by 16/02 which we achieved however we did not plan well enough with our supervisor, this resulted in the work being sent for review a month later as planned earlier by the supervisor. In practice this did not affect the work. Lastly the planning of the programming part could have been done better, all the implementations we wanted to do did not really fit and we had to start cutting some ideas and try to minimize test cases so that we actually can get the thesis completed in time. Learning to program in python, learning all the various packages and getting it to work together was challenging and demanded more hours than planned for.

The thesis in general falls under the Data Science and the course *Data Mining in Computer and Systems Sciences* helped a great deal with understanding neural networks and general Data Science related terms. No other course was

really relevant for the study, except for the obvious *Scientific Writing* and *Scientific Methodology and Communication in Computer and Systems Sciences* . Having a better background in the field of Data Science would have certainly helped but our supervisor helped out with many questions that we had and the joint meetings with other students that were in the field of data science helped.

As the thesis falls under Data Science and Health informatics it will not be that relevant in the early days of my career, however Data Science is taking a bigger role and I find the topic very intriguing, in the long run it might be of great use. Health informatics will not be relevant for me but it added a nice twist to the project. The implementation and discussion that me and Andreas had that revolved around the results and implementation decision is something that was very rewarding and it something that I regard as valuable for the future.

I am very pleased with the thesis and the quality of the work. The results were hard to digest as it there were no significant differences. However, as our supervisor pointed out it is important to not fall in love with your hypothesis. There are certainly some things that in hindsight could have been done to possibly improve the results. However the thesis highlighted many limitations of the supervised pre-training and how to apply neural networks on DNA amplification dataset and there is room to base future work on our findings and in my opinion that is a great result. We have already discussed many ideas for a new thesis by implementing new ideas and further explore how to improve the supervised training. The results regarding multi-class classification were encouraging, especially seeing that when it was better on average it was between 4-9% better.

Appendix B

Reflection Document Andreas

Our results were quite good in correspondence to the course goal, but more in certain areas. I think we had a good introduction with good background. The problem was broad to begin with, as we were interested both in solving a difficult dataset and we used a novel method, Peng Pre-Training for doing so. The research strategy, method and data-collection was well in line with the guidelines. We used the thesis template extensively when designing our methods and writing the thesis.

One of the big problems was not accounting for the sparsity and redundancy in the dataset we received, and we did wanted to define the research question early, so we could do quick tests on our practical steps. The supervisor had a very meritable and strong Bioinformatics background with high-grade publications, but a perfect computer science thesis and subject requires a strong focus on comparing and improving and investigating algorithms, rather than investigating the underlying biological research problem directly. At the same time this novel algorithm was from some perspective a good fit for attacking the problem of analyzing the dataset, and I think we did manage to put the focus on the computer science in our experiments.

The planning was quite good. We identified the potential risks of doing a computer science thesis not focused enough on the algorithms, or the research questions being a bit in conflict. From experience we also knew that implementing novel methods in a novel language risks exponentially growing in time requirements. However, we did not identify the sparsity of the dataset as having a potentially similar effect as the redundancy issue that crippled the algorithm used in our background-thesis we compared our results to until later in the research process. It did bring up interesting questions and we did have good results to discuss, and I think we drew the proper conclusions as well.

The most helpful courses before my thesis work, have been Data Mining, as well as Data Mining 2, also known as Research-Topics in Data Science. The

concepts of neural networks presented in the courses were very useful for understanding the most important terminology when designing experiments, reading the thesis, planning and coding. However even the first Object-Oriented course had some merit, as building the program into modules was very important to avoid issues when we wanted to change our experimental settings or do many different test runs. Most non-automated work was testing if a function worked, or adjusting small things in the experiment pipeline.

I currently have employment at Handelsbanken backend part-time, and am working as a trial-employee post my degree at Pensionsmyndigheten, both are Java backends. However, I do hope that machine-learning will either enter those companies in the future and I can transfer position, or that the usage will "trickle down" into the backend-layers, as classification is quite exciting to work with. Ive also done a bit of Bioinformatics Research in the past before studying Computer Science, and continued my research during summers and weekends. I do aim to go to hackathons and hopefully some of these will be Bioinformatical- it would be **amazing** to be able to solve a similar issue like this with likeminded people in the future.

Looking back at the process, both me and Arpad are very satisfied, although we created a very large problem to try to answer, and did so to a limited extent, I still think we focused on the most important questions and have discussed the limitations of us results well. There is never time to do everything, but this thesis would be a great jumping of point for future research. If I or Arpad continue part-time with a master at Stockholm University, I will very likely suggest that we use the previous research and coded framework as a jumping-off point. After 2-3 days of waiting for the computations to end, I was surprised how thrilling it was to read and analyze the results.