University of York

Department of Computer Science

Final Year Project (B.Eng)

Using 2D Convolutional Neural Networks for Electrocardiogram Abnormality Detection

Luke Jenkinson

supervised by
Dr. Simon O'Keefe

April 30, 2018
All appendices should be marked.

Abstract

Computerised methods of Electrocardiogram (ECG) interpretation play an increasingly important role in modern healthcare, with some physicians relying on their results entirely. It is therefore advantageous to investigate how the performance of such systems can be improved.

This project outlines a novel computerised method for detecting abnormal ECG recordings. Specifically, it detects whether an ECG recording is of a patient suffering a myocardial infarction, or is the ECG of a healthy control patient. ECG recordings from the PTB Diagnostic ECG Database (an open access database) were split into overlapping windows equating to 10 seconds in length. This data was then used to train and validate the performance of 84 different neural networks (60 Multi Layered Perceptrons and 24 Convolutional Neural Networks) providing a comprehensive picture of the classification ability of the different networks. When subsequently tested on unseen data, an 18 layer 2D Convolutional Neural Network is able to achieve an test accuracy of 86.36% compared to an accuracy of 59.36% for a Multi Layered Perceptron.

Acknowledgements

I would like to thank my supervisor, Dr Simon O'Keefe, for all of his help and time during the project. I would also like to thank Dr Chris Hayes for his help with the clinical aspects of the project. I would also like to thank Hilary, Steven and Jack for help with proof reading.

Statement of Ethics

The work undertaken as part of this project has been completed in an ethical way.

While the data used in the project is medical data, it was taken from a public data source, and as such, all participants will have provided informed consent for their data to be released. As the data is public, there are no requirements for storage or disclosure. The data has been anonymised, containing only clinical notes and ECG recordings.

There are possible ethical implications in the use of Artificial Intelligence and Machine Learning for medical diagnosis and detection. One such ethical concern is that the algorithm may "reflect the biases in the data used to train them". For example, if one sex or race is over-represented or under-represented in the training data, this imbalance could lead to misdiagnosis or poorer quality of care. Another ethical concern is the possibility that programs could be designed to recommend certain clinical interventions over others (in particular, interventions that the program designers have an economic stake in) [1].

As my work is not used for clinical intervention, these ethical implications do not affect my research. However, if this work is pursued further, these implications should be considered.

Contents

List of Figures

1

Introduction

The ultimate goal of computational medicine is to create an artificially intelligent computer program that is able to perform all of the duties of a human physician, with the benefit of never making mistakes. The program should also be able to use the latest knowledge about a condition, thereby improving patient outcomes. This is an extremely hard task due to a number of reasons: there is a large variation in the symptoms that a patient can show for the same condition; much of the information in the field is unstructured, therefore, it is not easily accessible to software; and ethical problems with a piece of software treating a patient, such as those outlined in the statement of ethics. This project describes one way in which software can be used to help diagnose conditions from patient data.

Electrocardiograms (ECGs) are a diagnostic tool used by physicians to help detect problems affecting the heart. A wide array of conditions can be detected using an ECG including arrhythmias (where the heartbeat is too fast, too slow, or irregular [2]), cardiomyopathy (where the heart walls become stretched, thickened or stiff [3]); and myocardial infarctions (commonly known as heart attacks [4]). After the data from the ECG is collected, it is generally reviewed by a cardiac specialist who will try to diagnose any problem. This analysis is a time consuming task, which would be made longer with a longer ECG recording. Therefore, ECG recordings are often very short in duration, typically 30 minutes, to reduce the time burden on the physician. The condition may be asymptomatic during the ECG and so an underlying problem may be missed by the specialist.

While machine learning has previously been applied to ECG interpretation, this work has mainly been limited to classical approaches to machine learning [5] [6]. Recent advances in deep learning have produced incredible results

in many domains [7] [8], however, there has been very little work applying this new found success to ECG interpretation. If deep learning can be shown to produce state of the art performance, these algorithms could be used in commercial medical systems, which would hopefully reduce the number of diagnoses that are missed.

This report will explain basic Machine Learning and Deep Learning concepts and how they are currently applied to computerised interpretation of ECGs. The report will then go on to explain the problem specification, with experiments that use a small sample of the data to show that it is possible to use Convolutional Neural Networks for this task. It will subsequently discuss how Convolutional Neural Networks perform in this task. Finally, it will summarise any limitations with this work, and outline possible future work.

2

Literature Review

This section will start with a description of basic concepts: what an ECG is; descriptions of machine learning; and deep learning concepts. These sections are included to give the reader the general information needed to understand the rest of the work. The section will then go on to outline current methods for computerised ECG interpretation along with other works which have applied deep learning to this particular problem. This should demonstrate to the reader how this work integrates into the wider body of knowledge.

2.1 The ECG

'An electrocardiogram is a simple test that can be used to check [a] heart's rhythm and electrical activity' [9]. It works by detecting the depolarising and repolarising of the heart during a heartbeat. During normal heart activity, this electrical signal originates in the SA node (the pacemaker of the heart). It then moves to the AV node (with a slight delay to allow the contracting atria have enough time to pump the blood into the ventricles), and then down along the Bundle of His, into the Bundle branches and Purkinje Fibres [10]. See Figure 2.1 for a diagram of the heart. The ECG machine is able to detect this electrical activity, and uses it to output a graph of calculated potential difference over time (this is known as a lead), an example of which can be seen in Figure 2.2.

2.2 12 Lead ECG

There are several different types of ECG recordings, however, the most common is the 12 lead ECG. This type of ECG uses 10 electrodes placed in different locations around the body (see Table 2.1): by doing this the ECG

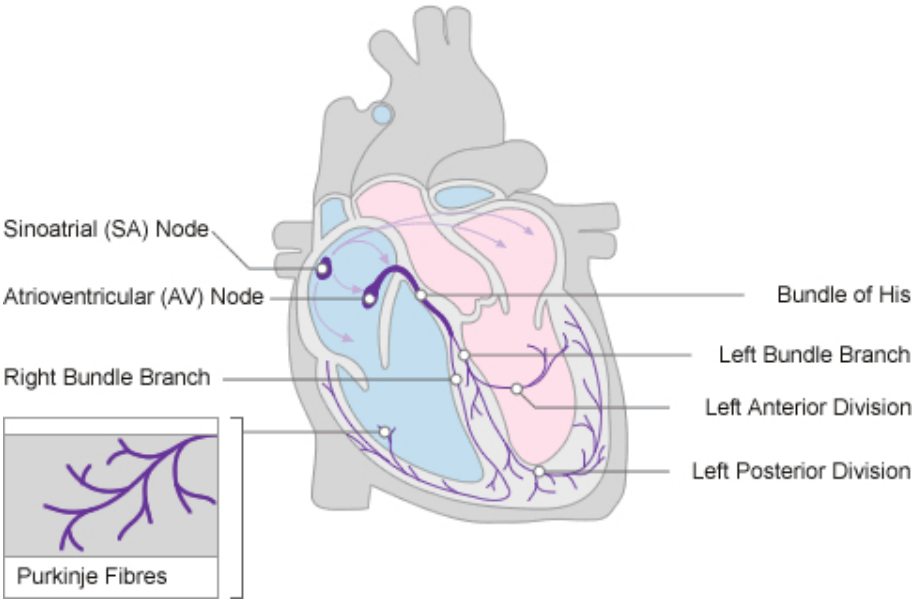


Figure 2.1: A diagram of the heart

machine is able to calculate 12 leads, each of which have a different name and give a different "view" of the heart. A lead is the name given to a trace calculated by the ECG machine. This is calculated by combining the values in different ways, which is done using the following equations (the variables are defined in Table 2.1):

The common lead:

$$Vw = \frac{1}{3}(RA + LA + LL)$$

The limb leads are calculated as followed:

$$I = LA - RA$$

$$II = LL - RA$$

$$III = LL - LA$$

Figure 2.2: A one lead ECG recording

The three augmented limb leads are calculated as follows:

$$aVR = RA - \frac{1}{2}(LA + LL) = \frac{3}{2}(RA - Vw)$$

$$aVL = LA - \frac{1}{2}(RA + LL) = \frac{3}{2}(LA - Vw)$$

$$aVF = LL - \frac{1}{2}(RA + LA) = \frac{3}{2}(LL - Vw)$$

The other leads (i.e. V1,V2,V3,V4,V5,V6) are calculated as the potential difference between the relevant electrode and the common lead.

In a normal clinical setting, this is displayed in a user friendly format as can be seen in Figure 2.3.

2.3 What is an Artificial Neural Network?

Artificial Neural Networks are a relatively old subsection of Computer Science, with their roots from within biology and neuroscience. They were originally used to try to mimic the way that neurons in the brain behaved,

Name	Placement
RA	Right Arm
LA	Left Arm
RL	Right Leg
LL	Left Leg
V1	Fourth Intercostal space, right of the sternum
V2	Fourth Intercostal space, left of the sternum
V3	Midway between V2 and V4
V4	Fifth intercostal space at the midclavicular line
V5	Anterior axillary line at the same level as V4
V6	Midaxillary line at the same level as V4 and V5

Table 2.1: The different placements of the electrodes in a 12 lead ECG

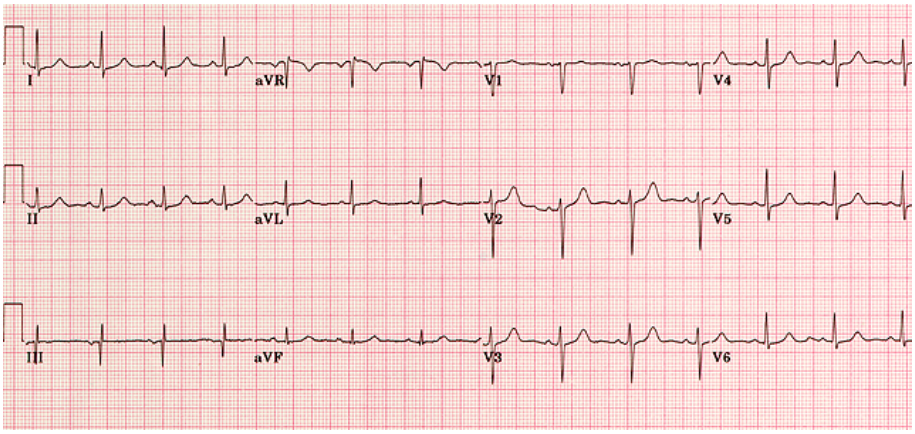


Figure 2.3: A twelve lead ECG recording

however, more recently they have become a purely theoretical model for the brain, rather than trying to mimic the behaviour exactly.

The first Artificial Neural Network was known as a Perceptron, which consisted of an input layer, weights, an activation function (step function) and a single output neuron (Figure 2.4). This network could classify linearly separable classification problems, however, it was largely ignored by the mainstream research as it failed to solve the XOR problem [11](i.e linearly inseparable classes). It wasn't until many years later, when another layer (a so-called hidden layer) was added to the Perceptron, that it was able to solve the XOR problem. This new type of network combined multiple Perceptrons into multiple layers, and is known as a Multi Layered Perceptron (MLP), an example of which can be seen in Figure 2.5.

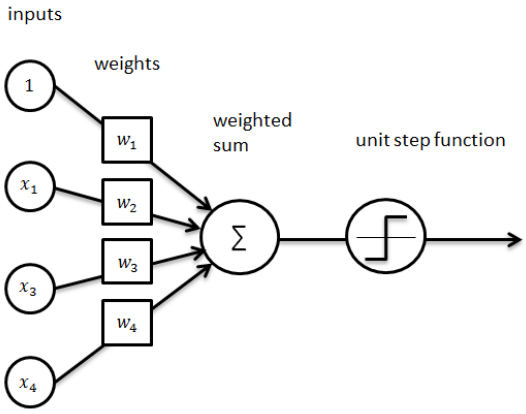


Figure 2.4: A representation of a perceptron

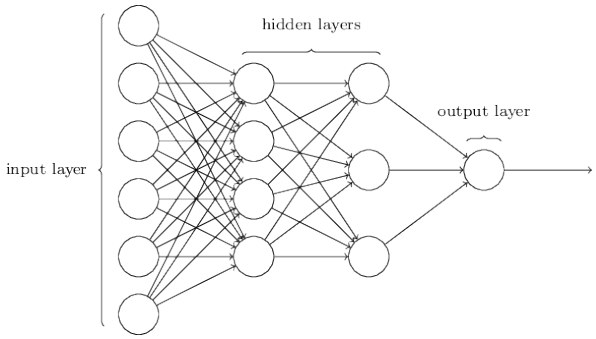


Figure 2.5: A representation of an MLP

Neural Networks 'learn' through a process known as backpropagation of errors, or more simply as backpropagation [11]. This process is where the weights of the connections between neurons have to be updated during training, to minimise the error between the input and the expected output. There are a number of techniques to decide which parameters should be changed including Stochastic Gradient Descent (SGD) [11], Adam optimizer [12] and Adagrad [12]. Such backpropagation algorithms try to minimise the error between the output of the network and the expected output of the network. Algorithms such as Adam are quite well suited to climbing out of local minimums in the loss function due to their inclusion of a 'momentum' term.

2.4 Deep Learning

Deep learning has received a large amount of attention in the past decade, particularly in the past 5 years, due to its impressive performance [13]. While there is no universally accepted definition of how deep a network has to be before it is considered 'deep', a network with 4 or more hidden layers would most likely fit the definition [13]. Deep learning has become more popular recently for two reasons: larger training datasets are available; and computer hardware, such as General Purpose Graphical Processing Units (GPGPUs) and Tensor Processing Units (TPUs) [14], has become able to compute fast enough to process the data. Huge public datasets have become available recently, allowing even amateur enthusiasts to access large volumes of training data, for example, the Common Crawl Web Corpus [15] contains Petabytes of data which are easily accessible on Amazon Web Services. Deeper networks generally give better performance: the size of neural networks has doubled roughly every 2.4 years and it is expected that this trend will continue for the foreseeable future as hardware and techniques become more advanced [13].

2.4.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks are a class of network which have kernel convolutions in at least some of the layers [16]. A convolutional section is typically composed of three stages: a convolution stage; a detector stage (i.e. the activation function); and a pooling stage.

Convolution stage:
In this stage, a filter (of any dimensionality) is applied to the input. This can be thought of as a filter or mask applied to the input of the layer which outputs the required information. The filter works by multiplying its values by the values in the input, then summing up all the values. The filter is then moved along the image by a set number of pixels, known as a stride, until it has been applied to the whole image. A visual representation of this can be seen in Figure 2.6.

Detector stage:
This stage consists of a non-linear activation function, most commonly, ReLU (Rectified Linear Unit). A ReLU layer applies the function max(0,x) to the input of the layer, providing the required nonlinearity in neural networks. ReLU has been found to be more successful than other functions for use in CNNs. A graphical representation can be seen in Figure 2.7.

Pooling:
Pooling is used to reduce the size of an output of a layer. Pooling splits the

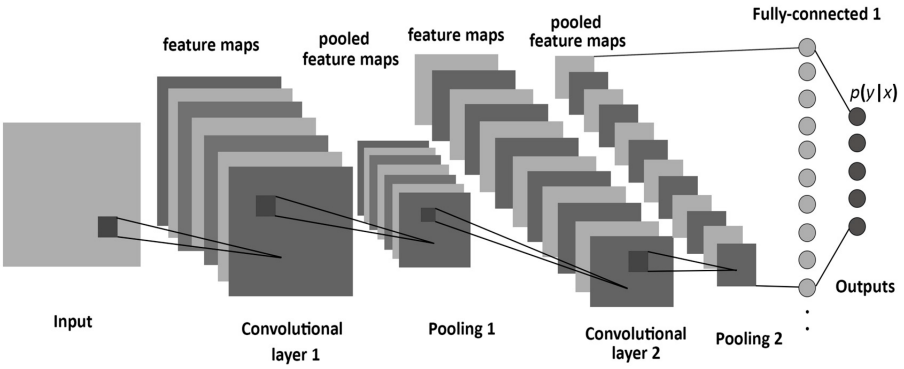


Figure 2.6: A representation of a CNN

input into non-overlapping regions. These regions are then mapped into a single number (see Figure 2.8). Two types of pooling are the most commonly used; max pooling(where the maximum value in the region is the output) and average pooling (where the average of the values is used). As the convolution stage can be thought of as a feature detector, which outputs a higher number as it detects more of that feature, pooling, therefore, reduces the dimensionality without losing too much information.

2.4.2 Regularisation

Regularisation is the term used to describe techniques that apply a penalty to some aspect of the network. This increases the generalisation of the algorithm. Sometimes these constraints or penalties are designed to encode specific domain or prior knowledge. Dropout [17] is a technique that reduces overfitting (i.e. high training accuracy, but low validation accuracy) by randomly removing, with some probability, a node from the layer with dropout. Dropout forces the network to improve the accuracy of that pathway. This increases robustness as complex co-adaptations cannot form. Dropping 20% of input neurons and 50% of hidden neurons was found to be optimal [17].

2.5 Computerised ECG Interpretation

The classical approach to computerised electrocardiogram abnormality detection is to extract features, such as R-R interval; QT interval and PR interval, from the data. These features are then input into a statistical classifier which will give the predicted heart condition.

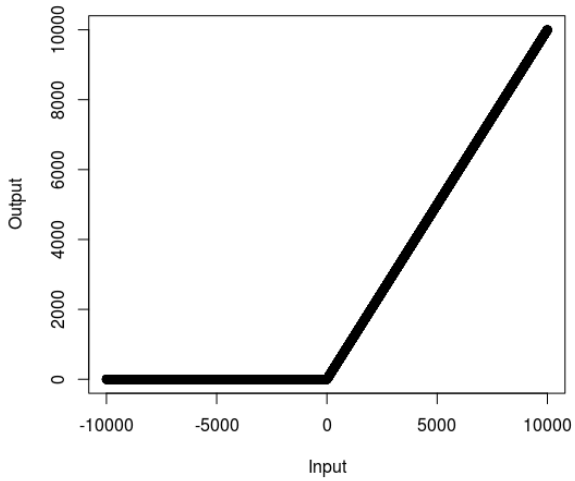


Figure 2.7: A graphical representation of the ReLU function

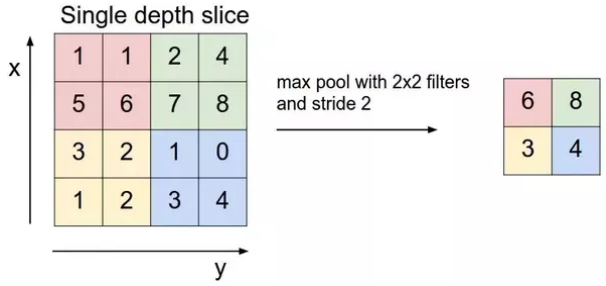
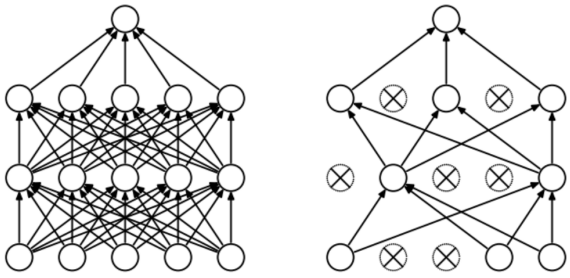


Figure 2.8: A representation of max pooling

In "Study of Features Based on Nonlinear Dynamical Modeling in ECG Arrhythmia Detection and Classification" [5], Owis et al. compare a new method of feature extraction to the classical methods of detection, such as frequency domain features analysis, and wavelet transform. They then used a statistical classifier to predict the class of the sample.

In "Automatic arrhythmia detection based on time and time frequency analysis of heart rate variability" [6], Tsiposras and Fotiadis extract features from the time domain and time frequency analysis of the R-R interval. This data was then used to train a neural network as a statistical classifier. They achieved a sensitivity of 89.95% and a specificity of 92.91%.



(a) Standard Neural Net (b) After applying dropout.

Figure 2.9: A representation of a dropout

In "Computerized Interpretation of ECGs: Supplement Not a Substitute" [18], Estes is highly critical of automated ECG tests: stating that relying on an automated test can result in inappropriate treatment or a missed diagnosis. This stance agrees with the opinion of the cardiologist interviewed as part of this project, who said that doctors who are less familiar with ECGs tend to rely on the automated diagnosis resulting in poorer patient outcomes. Estes focuses on QT interval abnormality detection, as this can be affected by numerous medical conditions, thus, it is a useful diagnostic technique. He suggests that automated methods should take into account parameters that have an impact on QT interval such as heart rate, sex, age, and QRS prolongation.

2.6 Expert Opinion of Computerised ECG Interpretation

A senior cardiologist, Dr Chris Hayes, at York Hospital was interviewed to provide expert insight into how computerised ECGs are currently used within a clinical setting and to also provide opinion on how they could be improved. It was found that, generally, cardiologists don't rely on the diagnosis provided by the machine as they have a large amount of expertise in interpreting ECGs manually, but will look at the provided diagnosis after, as a check. However, people with less experience working with them, such as GPs, may rely more heavily on the machine's diagnosis, and therefore, it is believed, a more accurate machine would result in less misdiagnosis and mismanagement of treatment.
When asked about the accuracy of the machines, Dr Hayes said the diagnosis hardly ever gave false negatives, although it wasn't as accurate in giving true positives. This is the more preferable outcome as, in a clinical setting, it is

better to refer a few people to a specialist unnecessarily than to not treat a heart condition.
In Dr Hayes' opinion, current methods are good for finding abnormalities in each complex (the part of the ECG that represents the depolarising and repolarising of the heart), however, they are not so good at finding abnormalities over a period of time.
In Dr Hayes' opinion, improving the quality of automated ECG interpretation methods would result in greater patient outcomes, especially when used by non-cardiologists.
He also stated that ECGs are not stored digitally at York Hospital so ex post facto analysis would not be possible. Therefore, the algorithm would need to be able to run in real-time on the machine .

2.7 Deep Learning for ECG Interpretation

In "Real-Time Patient Specific ECG Classification by 1D Convolutional Neural Networks" [19], Serkan et al. describe a neural network that can achieve an accuracy of 99% after being trained on 1% of the dataset and then tested on 99% of the total available data. This result shows that 1D Convolutional Neural Networks can be trained on a small supervised dataset and then achieve good performance on a large test set. This would be extremely useful in a clinical setting where a physician can give examples of the abnormal beats they would like to find in the large dataset (for example, the output from a Holter Monitor) and then the network can find areas of interest for the physician. This would significantly improve the time taken to analyse the large amount of data that Holter Monitors can produce, as they are often recording for 2 weeks at a time. While this paper achieves impressive accuracy, it looks at each complex individually: using beat detection algorithms to segment the data. One of the problems with computerised ECG interpretation highlighted by the cardiologist interviewed as part of this project, was that the current methods did very well at detecting problems within complexes, but failed at detecting abnormalities over time.

Rajpurkar et al. outline a deep Convolutional Neural Network for arrhythmia detection in "Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks" [20]. They use a 34 layer CNN with residual connections (described in [21]), which makes a prediction of whether the ECG is exhibiting arrhythmia every 1 second of each 30 second recording. Rajpurkar et al. created the largest dataset of its type, consisting of 64,121

1-lead ECG records from 29,163 patients. The training, testing and validation datasets were labelled by a committee of 3 cardiologists to reduce the possibility of a mislabelled classification. The testing performance of the network was then compared to 6 cardiologists working individually. The CNN was able to exceed the average performance of the cardiologist, both in recall and precision.

2.8 Conclusion

This section has explained what an ECG is, basic machine learning and deep learning concepts, and current work into how these methods can be applied to automated ECG interpretation.
While Deep Learning and Convolutional Neural Networks have both been applied to the problem of automated ECG interpretation (both producing state of the art results); only 1 dimensional Convolutional Neural Networks have been investigated. 12 lead ECGs are the most common form of ECGs, however, no work has been conducted applying CNNs to this problem, instead, focusing on 1 lead ECGs. Multi-dimensional Convolutional Neural Networks are able to detect features that are present across different dimensions, therefore, I believe that they could achieve greater performance than their single dimensional counterparts.

3

Problem Analysis

3.1 Problem Objectives

The objective of this project is to produce a 2D Convolutional Neural Network capable of distinguishing an abnormal ECG from a healthy control group and to compare the result of this network to the result from a Multi-Layer Perceptron. This problem has been constrained to distinguishing the difference between a Myocardial Infarction and a healthy control group.

3.2 Requirements

- The data should be minimally preprocessed, this allows the network to find the best representation of the data.
- The network should be able to be trained in a reasonable amount of time. Due to time and budget constraints of the project, this means that a subset of the data will have to be used instead of the whole dataset. This also means that smaller network sizes will need to be used. This requirement is derived from the fact that this is a time limited project, and thus I cannot train overly complicated networks and finish on time.
- The network should be able to run in a reasonable amount of time, as any commercial application of this research would need to run on a physician's computer. A network that is computationally cheap to run also opens the possibility of running the network on the collection device itself, thereby allowing real time interpretation. This requirement is derived from comments by the cardiologist consulted as part of

this project. Dr Hayes stated that the finished product would be much more useful if it were to be integrated into the collection device.

3.3 Methodology

This project will use the PTB Diagnostic ECG Database first described in work by Bousseljot:
"The ECGs in this collection were obtained using a non-commercial, PTB prototype recorder with the following specifications:

- 16 input channels, (14 for ECGs, 1 for respiration, 1 for line voltage)
- Input voltage: 16 mV, compensated offset voltage up to 300 mV
- Input resistance: 100 Ohms (DC)
- Resolution: 16 bit with 0.5 V/LSB (2000 A/D units per mV)
- Bandwidth: 0 - 1 kHz (synchronous sampling of all channels)
- Noise voltage: max. 10 V (pp), respectively 3 V (RMS) with input short circuit " [22]

The full dataset contains 549 records collected from 290 subjects. The subjects have a mean age of 57.2 years. The data consists of the conventional 12 leads (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6) plus three frank leads, which can be derived from the 12 other leads.

The full dataset consists of:

Diagnostic class	Number of subjects
Myocardial infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	52
N/A	22

The patient files used as part of this project are taken from myocardial infarction and healthy control groups; and are described in Appendix 1. The data will be split into training, validation and testing sets.

Before researching whether Convolutional Neural Networks can be used for this task, I will implement a Multi Level Perceptron on a small subset of the data to prove that neural networks can work on the dataset. If this works, I will implement a Convolutional Neural Network on the small subset. By implementing the network on a smaller subset of the total data, I am able to investigate whether CNNs are able to work in this particular domain, in a far smaller amount of time than training a network on the full dataset. I will implement the smaller test network on a set of 5 Myocardial Infarction patients' ECGs and 5 healthy control patients' ECGs. This data is split: 8 recordings for training (4 Healthy, 4 Myocardial Infarction) and 2 for testing (1 Healthy, 1 Myocardial Infarction).

If the results of the initial experiments are positive, I will implement a Multi Layered Perceptron on the whole dataset to get a baseline performance. I will implement a grid search of 60 different networks: 2, 3 and 4 layer networks each tested on 50-500 (step 50) neurons per layer, tested on each of the 30 epochs (i.e. a forward and backward pass on all the training data), and each of the architectures will be tested with and without dropout.
I will then implement Convolutional Neural Networks on the full dataset, and compare the accuracy of these networks to the accuracy achieved by the Multi Layered Perceptron.

These networks will be trained on the training set and evaluated on the validation set. The results from these will be analysed to find the optimal hyperparameters. A test network will be implemented using these hyperpa-

rameters, and tested on the testing set. This should give an approximation to the accuracy of the network on an unseen ECG.

3.4 Initial Experiments

In the initial experiments, it was found that there was an error (most likely an overflow or underflow error happening within the Keras or tensorflow libraries) when training the MLP. This caused the loss of the network to be given as NaN (Not a Number), meaning that network could not learn. It was found, with some experimentation, that scaling the inputs down by a factor of 1000 (therefore making the inputs between 0 and 1) would fix this error, thus allowing the network to learn.

Once the error had been fixed, a Multi Layered Perceptron was able to achieve an accuracy of 93.54% on the small subset of the data, therefore showing that neural networks are able to be used on this particular dataset.

A Convolutional Neural Network was then implemented on the small subset of the data. A network diagram is shown in figure 3.1.
It was also found, in these initial experiments, that it is not necessary for the data to be scaled down by a factor of 1000 for the CNN to learn.

Using a random search of the possible hyperparameters, it was found that having a width of 100 nodes for the densely connected layers had the highest accuracy.

A Convolutional Neural Network with a 3 x 3 convolution filter, a ReLU activation function, a 2 x 1 max pooling layer, followed by 3 densely connected layers and a softmax output layer was able to achieve a 91% accuracy.

3.4.1 Time and Memory Requirements

I also investigated the training time and memory requirements for the neural network. It was found that 64 or 128 convolutional filters in each layer would result in the training time of the network being too great. It was therefore decided that 8 or 16 filters should be used.

The initial size of the dataset was 88 training samples and 27 validation samples. It was found that the training time for the full dataset was too long, therefore I reduced my training dataset size to 32 recordings and my

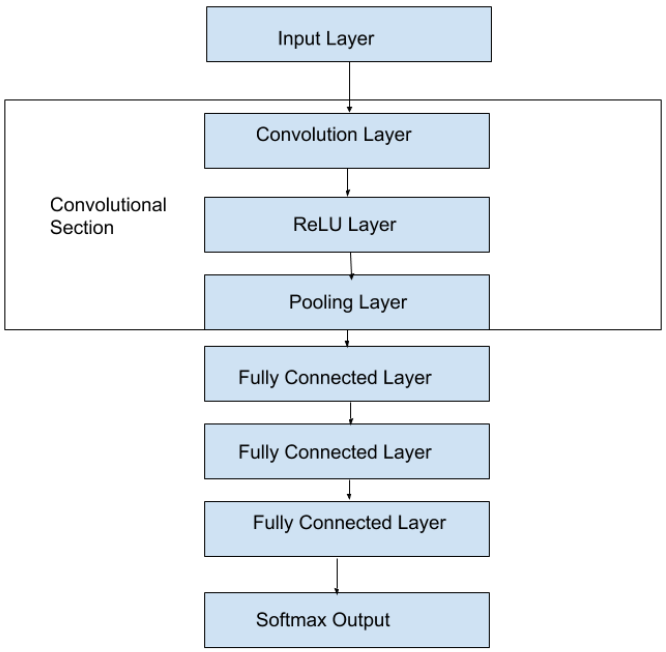


Figure 3.1: Network digram of the CNN used in testing

validation dataset size to 8.

I also reduced the memory requirements of my network, so rather than holding the whole dataset in memory, it loads part of the data in when needed. This means that I am able to run the code in parallel on smaller machines. Thus, while the time taken to run a single network is increased, the speed at which I can run a grid search of different possible architectures has also increased.

4

Implementation

I have carried out two implementations of Neural Networks as part of this project: a Convolutional Neural Network as the network architecture I am investigating; and a Multi Layered Perceptron that will be used as a reference guide to show how well other architectures are able to perform.

4.1 Data

The original data was in a Waveform Database format [23] (in the MIT signal file format). The data was converted to CSV format, using the WFDB software package [24], so it could be imported into the program. Once imported into the program, the data was down sampled to a rate of 100Hz (down from 1000Hz). I chose a rate of 100Hz, comparable to the commercial systems used in York Hospital, which use a sample rate of 140Hz. By doing this, it ensured that it was possible to make diagnoses from data sampled at this rate. Also, by down sampling the data, I reduced the number of parameters that the network had to learn, therefore both increasing the speed of training, and also increasing the generalisation ability of the network. After down sampling, the data was segmented using overlapping windows of 1000 samples in length, which corresponds to a length of 10 seconds. Ten seconds was chosen as, according to a cardiologist (Dr Hayes), it will contain the necessary information needed to make a diagnosis. The output, i.e. whether the data was from a healthy control or from a patient having a myocardial infarction, was then encoded as a one hot vector.

As it was found in the initial experiments that the Multi Layered Perceptron was not able to learn if the data was not scaled down, I have kept this preprocessing step. However, this step has not been included in the Convolutional Neural Networks, as it was found not to be necessary, thereby

achieving the requirement that the data be minimally preprocessed.

4.2 Hyperparameter Optimisation

The hyperparameters of a network are the parameters that are set before training and which define the network, for example: the network width and network depth are hyperparameters, as well as the kernel size and the type of pooling in a Convolutional Neural Network. The hyperparameters chosen can have huge impacts on the effectiveness of the network. There are no definitive rules for choosing hyperparameters, as it entirely depends on the problem. I have therefore implemented a grid search (also known as a brute force search) of possible network hyperparameters. To reduce the number of hyperparameters to optimise, I decided that the network width should be consistent throughout the network, i.e. all hidden layers should be the same size. This is a common approach for neural networks.

4.3 Multi Layered Perceptron

I have implemented 60 Multi Layered Perceptrons, running a validation test after each epoch between 1 and 30, therefore testing 1800 possible different values for the hyperparameters. I have tested two, three and four hidden layer networks. I have also tested network widths between 50 and 500 with a stride of 50. Half of the networks were trained with dropout and half without, but with the same network structure. This allowed me to explore the impact of dropout of the generalisation ability of the network.

4.4 Convolutional Neural Network

I have implemented 24 convolutional neural networks, running a validation test after each epoch between 1 and 10. I have varied the number of convolutional filters in each network to be able to explore the impact this has on the network's ability. As with the Multi Layered Perceptron, the impact of dropout on the network ability has been explored.

4.5 Training

All networks tested as part of this project were optimised using Adam optimizer [12] , and used categorical cross entropy (Figure 4.1) as the loss

function.

$$H(p,q) = -\sum_{n=1}^{n=numberofvals} p(x)log(q(x))$$

Figure 4.1: The formula for categorical cross entropy

4.6 Testing

There are different ways in which the network's performance can be measured. Accuracy (Figure 4.2) was chosen for measuring the performance of the network in this study, however, alternatives of accuracy are discussed in the "Discussion and Future Work" section.

$$\frac{TP + TN}{TP + FP + TN + FN}$$

Figure 4.2: The equation for accuracy (where: TP = True Positive; TN = True Negative; FP = False Positive; FN = False Negative)

4.7 Tools

The network was implemented with Python 3.6 [25] using Keras 2.0.8 framework [26] with a TensorFlow [27] backend (using Conda package manager [28]). The network has been trained on a variety of hardware, depending on availability:

PC: Intel Core i7 - 4770 @ 3.4GHz, 15 GiB RAM, Linux 4.4.0

Compute Server 0: 4 x AMD Opteron 6386 SE, 503 GiB RAM, Linux 4.4.0

Compute Server 1: 2 x Intel Xeon E5 - 2680 @2.4 GHz, 503 GiB RAM, Linux 4.4.0

Compute Server 2: 2 x Intel Xeon E5 - 2680 @2.4 GHz, 503 GiB RAM, Linux 4.4.0

The hardware chosen does not have an impact on the ability of the network, purely the speed at which the network can be trained.

5

Results

This section will do two things. Firstly, it will analyse the validation results; by doing this, I am able to make my choice as to the best hyperparameters for the network: For the MLP, I will examine the effect of varying network width and depth on the validation performance. For the CNN, I will examine the effect the number of filters has on the validation performance. All the validation performance figures are derived from the validation set, this consists of 8 ECG recordings not used in training.

I then, secondly, apply these network hyperparameters to 2 networks, resulting in the 'test MLP' and the 'test CNN'. The performance of these network will be examined through the use of an unseen dataset; thus providing a realistic test of the real world performance of the network.

5.1 Multi Layered Perceptron Results

5.1.1 Effect of Network Depth and Network Width on Network Ability

A 2 way ANOVA test was performed to analyse whether network depth and/or network width had a significant effect on the validation accuracy. There was a significant effect of width on the validation accuracy (ANOVA: $F = 117.709$; d.f. $= 1$; $p = 2e\text{-}16$) but there was no significant effect of depth on the validation (ANOVA: $F = 0.724$; d.f. $= 2$; $p = 0.485$). A graph of network width against validation accuracy has been plotted and can be seen in Figure 5.1, this shows a positive correlation between network width and validation accuracy.

On the basis of these results, the test MLP will have a width of 500 neurons, and a depth of 2 hidden layers. As there is no statistical effect of depth on performance, I have chosen a computationally less complex value.

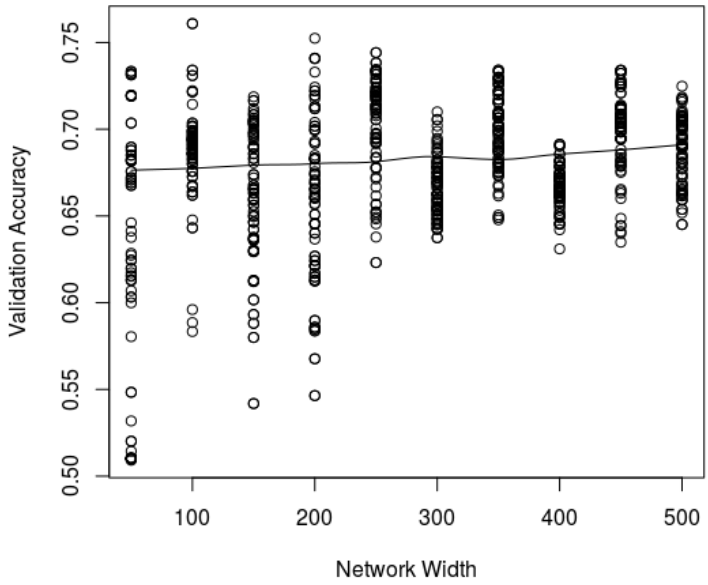


Figure 5.1: This figure shows the effect of width (in neurons) of an MLP on that network's validation accuracy

5.2 Convolutional Neural Network Results

5.2.1 Effect of Number of Filters on Network Ability

The effect of the number of filters on a network's ability was tested by running multiple different networks, some containing 8 filters per layer, some containing 16 filters. The results of these tests are examined here.

A Wilcoxon Signed-Rank Test indicated that the validation accuracy of the network when using 16 filters was not statistically significantly different than the validation accuracy of the network when using 8 filters ($V = 221$, $P = 0.8236$).

However, as can be seen in Figure 5.2, there are some outlier values, if they are removed we can see that the validation performance of the networks with 16 filters have a higher mean accuracy (96.1% compared with 93.0%). A Wilcoxon Signed-Rank Test indicated that this had a P value of 0.06364 and therefore, was not statistically significant. I will choose to use 16 filters in my test network as it had the higher mean validation accuracy, even though it is non-significant.

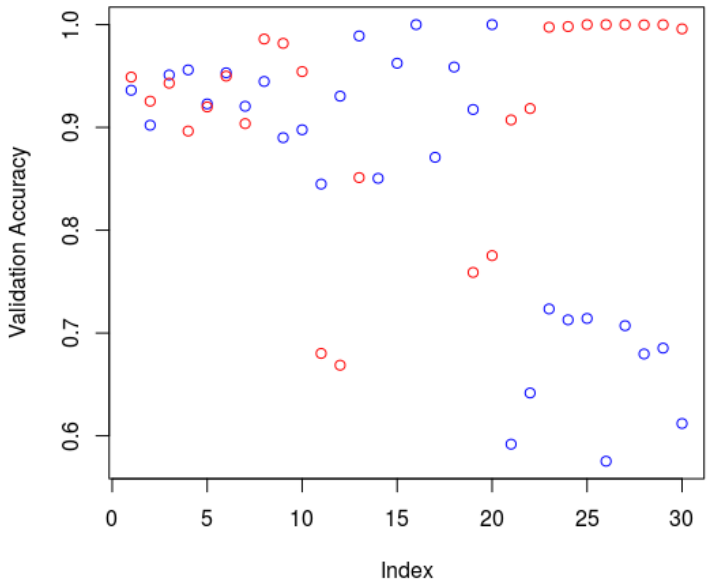


Figure 5.2: This figure shows the effect of varying the number of convolutional filters on the validation accuracy of the Convolutional Neural Network with dropout. 8 convolutional filters 16 convolutional filters

5.2.2 Effect of Number of Epochs on Network Ability

As can be seen in Figure 5.3, there is a positive relationship between the number of epochs and the validation accuracy of the Convolutional Neural Networks with dropout. It is probable that increasing the number of epochs, to more than 10, would increase the performance of the network. While this may be the case, increasing the number of epochs would increase the training time of the network to an unreasonable amount of time, thus not achieving

a requirement of the project. Consequently, the test network will be trained for 10 epochs, as to achieve this requirement.

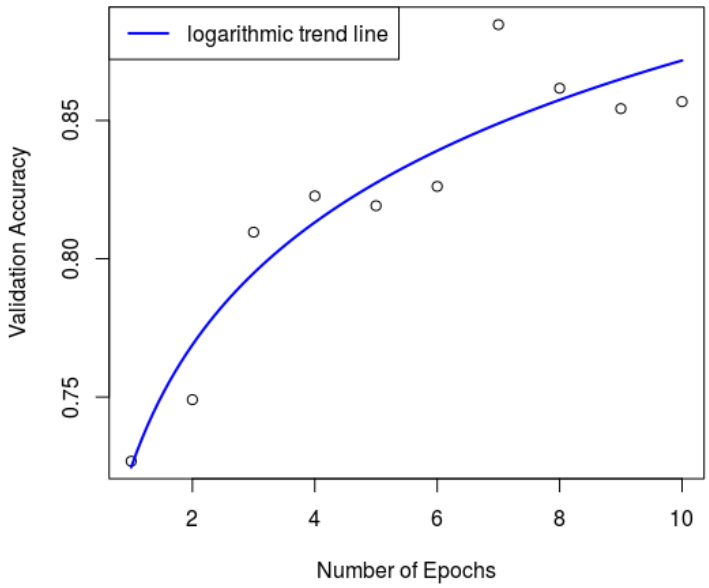


Figure 5.3: This figure shows the effect of varying the number of epochs has on the validation accuracy of the Convolutional Neural Network with dropout.

5.3 Other Results

5.3.1 Effect of Reducing the Training Data on the Generalisation of the Network

A single MLP was trained on the full dataset, this was then compared to the same network, trained on the partial dataset. The network had higher performance when trained on the full dataset: A Wilcoxon Signed-Rank Test indicated that the validation accuracy of the full dataset was statistically significantly higher that the validation accuracy of the partial dataset ($V = 52$, $P = 0.009766$). The results of these tests can be seen in Figure 5.4.

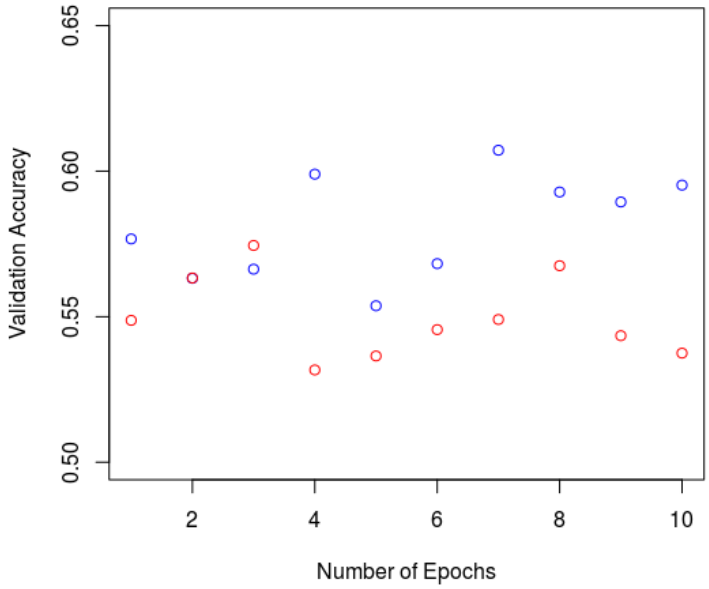


Figure 5.4: This figure shows the average validation accuracy of the full dataset and the average validation accuracy of the partial dataset when trained on a 2 hidden layer Multi Layered Perceptron.

5.3.2 Effect of Dropout on Network Ability

It was found that the Convolutional Neural Networks were not able to learn without dropout, achieving accuracies of approximately 50%. 2 layer Multi Layered Perceptrons were able to learn without dropout, however, 3 and 4 layer networks could not. With dropout, the deeper networks could achieve a higher accuracy than the 2 hidden layer network. Accordingly, dropout will be included in both the test MLP and test CNN.

5.4 Testing Results

A test Multi Layered Perceptron and test Convolutional Neural Network were implemented using the optimal hyperparameters, found using the statistical tests (see Figure 5.7 for the full architecture). Each of the networks were trained on same dataset as used before, however, an unseen test set (described in appendix 1) was used to examine the accuracy of the networks. These tests were run 3 times and the results averaged, as to limit the effect of initialisation

on the result. The CNN achieved a mean accuracy of 86.36% compared to a mean accuracy of 59.36% for the MLP. The mean confusion matrices can be seen in Figure 5.5 and 5.6.

	Predicted Class	
	MI	**Control**
True MI	35238	6841
True Control	4898	39110

Figure 5.5: The confusion matrix for the CNN testing results, where MI is myocardial infarction

	Predicted Class	
	MI	**Control**
True MI	18749	23331
True Control	11651	32357

Figure 5.6: The confusion matrix for the MLP testing results, where MI is myocardial infarction

5.5 Conclusion

Through the work in this chapter, we can see that 2D Convolutional Neural Networks outperform Multi Layered Perceptrons by 29 percentage points. We can also see that dropout and training set size both have a significant impact on network accuracy. Furthermore, increasing the number of epochs, increases the network performance.

Layer Number	Test CNN	Test MLP
1	input	input
2	(9x3), 16 Convolution	Fully connected (500 Nodes)
3	Activation (ReLU)	Activation (ReLU)
4	(1x10) Max Pooling	Dropout
5	(9x3), 16 Convolution	Fully connected (500 Nodes)
6	Activation (ReLU)	Activation (ReLU)
7	(1x10) Max Pooling	Dropout
8	Dropout	Output (Softmax)
9	Fully connected (100 Nodes)	
10	Activation (ReLU)	
11	Dropout	
12	Fully connected (100 Nodes)	
13	Activation (ReLU)	
14	Dropout	
15	Fully connected (100 Nodes)	
16	Activation (ReLU)	
17	Dropout	
18	Output (Softmax)	

Figure 5.7: The architectures of the test CNN and test MLP

6

Discussion and Future Work

6.1 Discussion

My work has implications not just for computerised ECG interpretation, but also for general signal processing. Recurrent Neural Networks have been used extensively for a number of years for signal processing, because of their ability to encode temporal data. My work for this project shows that Convolutional Neural Networks can also be used for the same task, achieving high accuracy. I hope that my work can be built on for other signal processing tasks.

6.2 Future Work

Due to the scope of the project, it has not been possible to do everything that I would have liked to have done. It is therefore recommended that more work could be done in this area. Some possible areas of research are:

6.2.1 Test More Architectures

The training time for the Convolutional Neural Networks was very large. CNNs also have more hyperparameters to tune than a Multi Layered Perceptron. These include: the kernel size, number of filters, and pooling layer shape. Due to both of these factors, it was not possible to explore as wide a number of network architectures and sizes than with the Multi Layered Perceptron. Further research, with a larger budget, could use GPGPUs to increase the speed at which networks can be trained, thereby allowing a wider variety of networks to be explored. The number of epochs explored was also capped at 10 for Convolutional Neural Networks due to the long training time. The number of epochs used in other pieces of work can be much larger

than this. For example, in Deep Residual Learning for Image Recognition [21], He et al. train the network for up to 600,000 epochs.

6.2.2 Use the Full Dataset

As discussed in Problem Analysis, I had to decrease the size of my dataset, as the training time of the networks became unusable. For training, 32 recordings were used, and for validation, 8 were used. By increasing the training dataset size, the network should be able to learn more robust features from the data. By increasing the validation and testing dataset size, it would ensure, to a greater degree, that the network has learned the difference between classes.

6.2.3 Randomise the Data

"As for any stochastic gradient descent method (including the mini-batch case), it is important for efficiency of the estimator that each example or minibatch be sampled approximately independently" [29]. In other words, for the network to learn efficiently, each mini-batch should have a random sample of the full dataset. It was not possible to fully randomise the data, due to hardware limitations. This limitation most likely had a detrimental impact on the training and overall result of the network. If further research is done within this area, I would recommend that the data be fully shuffled and thus randomised before training.

6.2.4 Increase the Number of Classes

My research has focused on detecting the difference between Myocardial Infarction and healthy control, however, the full data set contains ECG recordings for 9 different classes. This would introduce multiple problems to overcome, however, if they could be overcome: it would result in a much more commercially useful piece of software. One example of a problem needed to be solved, would be that the PTB Diagnostic ECG Database contains a class imbalance. There are 148 subjects exhibiting myocardial infarction, however, just 4 for Myocarditis. There are a number of methods for dealing with class imbalance including: reducing the size of the larger class; and more advanced techniques such as RUSBoost [30]. These techniques should be examined.

6.2.5 Compare the Results of a 2D Convolutional Neural Network to a 1D Convolutional Neural Network

Previous work has focused on 1 dimensional Convolutional Neural Networks trained on 1 dimensional data, and my work has explored 2 dimensional CNNs trained on 2 dimensional data. However, it is also possible to use 1D neural networks on the 2D data, by thinking of the second dimension as multiple channels, rather than as another dimension. Further work should be conducted, exploring the difference in performance between 1 dimensional CNNs acting on the data in multiple channels, and the performance of 2D CNNs acting on the data in 2D.

6.2.6 Alternatives to Accuracy

There are alternative metrics that could be used instead of accuracy for measuring network performance. Precision, recall and F1-score are all used as metrics for this purpose. If this technology were to be developed for commercial purposes, the stakeholders would have to decide what the minimum values for these metrics should be.

7

Conclusion

The goal of this project was to explore whether 2D Convolutional Neural Networks can be used to detect abnormal ECG readings, and to compare the performance of these CNNs to Multi Layered Perceptrons.

Three requirements were outlined in the Problem Analysis section:

1. The data should be minimally preprocessed. This requirement has been met: the data for the CNNs is not preprocessed at all; and the data for the MLPs has only been scaled by a constant factor.

2. The networks should be able to be trained in a reasonable amount of time. While training did, initially, take quite a long time: this requirement was met. This was achieved by reducing the size of the networks, as well as parallelising the workload, so they were able to train in fewer than 2 days.

3. The networks should be able to run in a reasonable amount of time. This requirement was partially fulfilled. All networks are able to run in real time when running on a standard PC, however, they are too complex to run on a microcontroller. Therefore, in a commercial system, these networks could only be used in a system that uses a laptop or desktop for storage and processing.

The results of the project clearly show that Convolutional Neural Networks are better suited for the task of detecting abnormal ECGs than Multi Layered Perceptrons. The test CNN was able to achieve 86.36% test performance, however, it could achieve up to 100% accuracy on the validation set. This indicates that the accuracy could be improved through some of the improvements outlined in the future work section.

If these improvements were implemented, and further testing done to ensure the reliability of the system, I believe this work could be used within a commercial system to improve patient outcomes, and possibly save lives.

Bibliography

[1] D. S. Char, N. H. Shah, and D. Magnus, "Implementing machine learning in health care addressing ethical challenges," *New England Journal of Medicine*, vol. 378, no. 11, pp. 981–983, 2018.

[2] "Arrhythmia," 2015. https://www.nhs.uk/conditions/arrhythmia/.

[3] "Cardiomyopathy," 2016. https://www.nhs.uk/conditions/cardiomyopathy/.

[4] "Heart attack," 2016. https://www.nhs.uk/conditions/heart-attack/.

[5] M. I. Owis, A. H. Abou-Zied, A.-B. Youssef, and Y. M. Kadah, "Study of features based on nonlinear dynamical modeling in ecg arrhythmia detection and classification," *IEEE transactions on Biomedical Engineering*, vol. 49, no. 7, pp. 733–736, 2002.

[6] M. G. Tsipouras and D. I. Fotiadis, "Automatic arrhythmia detection based on time and time–frequency analysis of heart rate variability," *Computer methods and programs in biomedicine*, vol. 74, no. 2, pp. 95–108, 2004.

[7] J. Zhang and C. Zong, "Deep neural networks in machine translation: An overview," *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 16–25, 2015.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, *et al.*, "Going deeper with convolutions," Cvpr, 2015.

[9] "Ecg, nhs choices," 2015. https://www.nhs.uk/conditions/electrocardiogram/.

[10] "Cardiac conduction system - normal function of the heart - cardiology teaching package - practice learning - division of nursing - the university of nottingham, nottingham.ac.uk," 2017. http://www.nottingham.ac.uk/nursing/ practice/resources/cardiology/function/conduction.php.

[11] R. Beale and T. Jackson, *Neural Computing-an introduction*. CRC Press, 1990.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[14] "K. sato, an in-depth look at googles first tensor processing unit (tpu) — google cloud big data and machine learning blog — google cloud platform, google cloud platform," 2017. https://cloud.google.com/blog/big-data/2017/05/ an-in-depth-look-at-googles-first-tensor-processing-unit-tpu.

[15] "Common crawl, commoncrawl.org.," ND. https://commoncrawl.org/.

[16] S. O'Keefe, "Convolutional neural networks," 2018.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] N. M. Estes, "Computerized interpretation of ecgs: supplement not a substitute," 2013.

[19] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ecg classification by 1-d convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2016.

[20] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[22] "The ptb diagnostic ecg database, physionet.org," 2000. https://www.physionet.org/physiobank/database/ptbdb/.

[23] "The wfdb software package, physionet.org," ND. https://www.physionet.org/physiotools/wfdb.shtml.

[24] "Wfdb software package." https://www.physionet.org/physiotools/wfdb.shtml.

[25] "python programming language," 1991. https://www.python.org/.

[26] "Keras: The python deep learning library," ND. https://keras.io/.

[27] "Tensorflow: An open source machine learning framework for everyone," ND. https://www.tensorflow.org/.

[28] "Conda package manager," ND. https://anaconda.org/anaconda/conda.

[29] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*, pp. 437–478, Springer, 2012.

[30] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[31] "Sa node." https://www.medicinenet.com/script/main/art.asp?articlekey=5495.

[32] "Av node." https://www.medicinenet.com/script/main/art.asp?articlekey=2387.

[33] "Rr interval." https://medical-dictionary. thefreedictionary.com/R-R+interval.

[34] "Qt interval." https://www.merriam-webster.com/medical/QT%20interval.

[35] "Pr interval." https://medical-dictionary.thefreedictionary.com /P-R+interval.

[36] "Arrhythmia." https://www.heart.org/HEARTORG/Conditions/Arrhythmia/Arrhythmia-UCM-002013-SubHomePage.jsp.

[37] "Holter monitor." https://www.heart.org/HEARTORG/Conditions/HeartAttack/DiagnosingaHeartAttack/Holter-Monitor-UCM-446437-Article.jsp.

Appendix A

Data Used

This page details the patient files that were used for training, validating and testing the network. All the data files are taken from the PTB Diagnostic ECG Database [22].

Training Data Files:
- s0010re - s0028lre - s0060lre - s0067lre - s0035_re - s0044lre - s0021bre - s0017lre - s0021are - s0020are - s0037lre - s0042lre - s0071lre - s0015lre - s0064lre - s0056lre - s0476_re - s0453_re - s0475_re - s0311lre - s0300lre - s0302lre - s0561_re - s0461_re - s0458_re - s0415lre - s0305lre - s0287lre - s0482_re - s0436_re - s0459_re - s0304lre

Validation Data Files:
- s0063lre - s0053lre - s0075lre - s0070lre - s0470_re - s0480_re - s0468_re - s0462_re

Testing Data Files:
- s0081lre - s0065lre - s0085lre - s0149lre - s0487_re - s0503_re - s0478_re - s0496_re

Appendix B

Medical Glossary

SA node
'Sinoatrial node: The heart's natural pacemaker, one of the major elements in the cardiac conduction system, the system that controls the heart rate.' [31]

AV node
'Atrioventricular node: The electrical relay station between the upper and lower chambers of the heart.' [32]

RR interval
'The interval from the peak of one QRS complex to the peak of the next as shown on an electrocardiogram.' [33]

QT interval
'The interval from the beginning of the QRS complex to the end of the T wave on an electrocardiogram representing ventricular depolarization and repolarization and indicating the time during which ventricular contraction and subsequent relaxation occurs.' [34]

PR interval
'The interval from the beginning of the P wave to the beginning of the QRS complex on an electrocardiogram. It represents the atrioventricular conduction time, which normally is between 0.12 and 0.20 second.' [35]

Arrhythmia
'An arrhythmia is an abnormal heart rhythm.' [36]

Holter monitor
'A Holter monitor is a battery-operated portable device that measures and records [the] heart's activity (ECG) continuously for 24 to 48 hours or longer depending on the type of monitoring used.' [37]