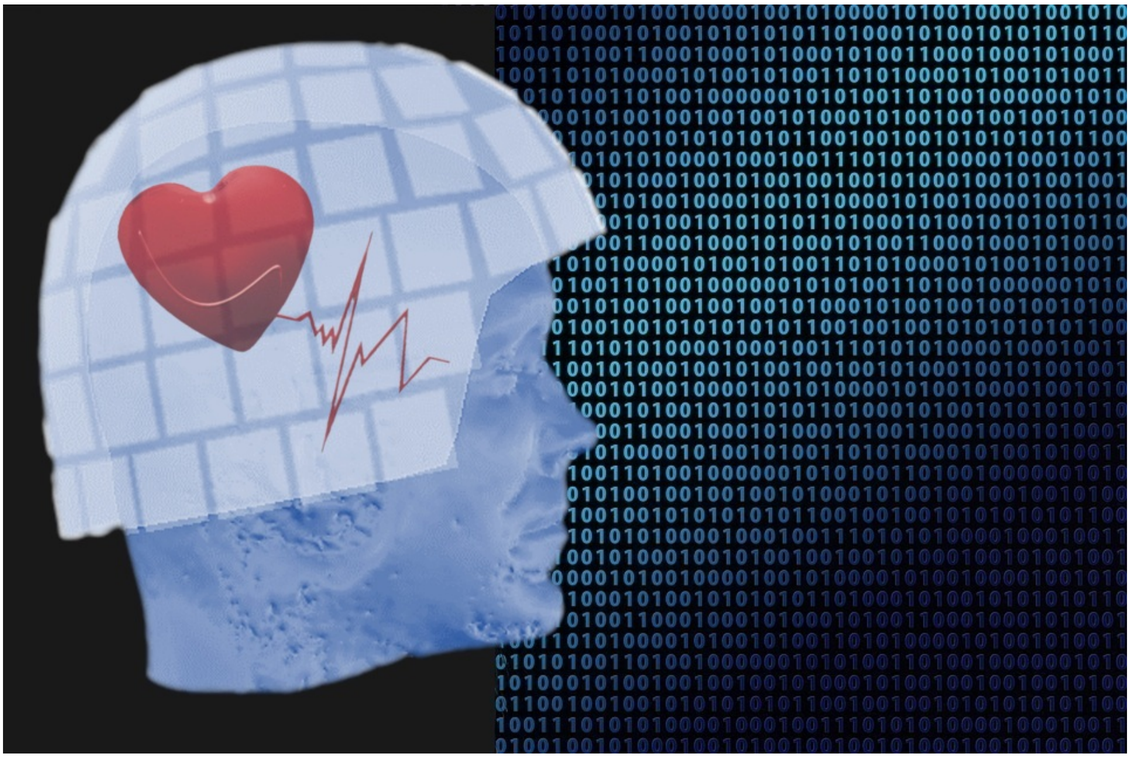
CHALMERS
UNIVERSITY OF TECHNOLOGY



Deep Learning for Brain Activity Analysis

Applying Artificial Neural Networks on Magnetoencephalography Data to Predict the Risk of Developing Essential Hypertension

Anna Bakidou
Jonas Nord Odhner

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

Deep Learning for Brain Activity Analysis

Applying Artificial Neural Networks on Magnetoencephalography Data to Predict the Risk of Developing Essential Hypertension

Anna Bakidou
Jonas Nord Odhner

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2020

Deep Learning for Brain Activity Analysis
Applying Artificial Neural Networks on Magnetoencephalography Data to Predict the Risk of Developing Essential Hypertension
Anna Bakidou
Jonas Nord Odhner

Supervisor: Nima Hajiabdolrahim, Electrical Engineering
Supervisor: Jonas Andersson, Syntronic Research and Development AB
Examiner: Professor Paolo Monti, Electrical Engineering

Master's Thesis 2020
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: A collage of three figures illustrating the main topics of the thesis.
Figure 1: Binary, Code, Privacy Policy, Woman, Face, View. Pixabay license. Adapted with permission.
Figure 2: Illustration of a MEG helmet. Image courtesy of J. Schneiderman. Adapted with permission.
Figure 3: Door sign, Billboard, Heart, Curve, Smile, Health. Pixabay license. Adapted with permission.

Typeset in LaTeX
Gothenburg, Sweden 2020

Abstract

Nearly one third of all deaths in the world are caused by cardiovascular diseases, making it the leading cause of death for both males and females. One of the major risk factors is essential hypertension. Due to natural variations in blood pressure, predicting the development of essential hypertension is difficult. Several studies have revealed an individual correlation between hypertension and the sympathetic nervous response elicited by an arousing stimulus. Half of the test group exhibited a relatively elevated sympathetic nerve activity, which may indicate an increased risk of developing essential hypertension. Thus, the relationship could potentially serve as a biomarker. However, the method used to measure the sympathetic nervous response is invasive and complicated. As such, there exists a need to identify and evaluate non-invasive measurement techniques.

The purpose of this work is to investigate if artificial neural networks can be used to predict the development of essential hypertension, by analyzing brain functional responses detected by magnetoencephalography (MEG). Two different types of artificial neural networks were developed for classification: one multi-layer perceptron network and one long short-term memory network. Additionally, an autoencoder neural network was developed to examine whether applying high-frequency filters to the data could increase the classification accuracy.

The classification networks identified measurement specific patterns in the sensor data, but these patterns did not generalize well to other individuals. It was found that both networks could predict, with high accuracy, from which subject a MEG measurement originated. This could indicate that brain functional activations, like finger prints, are individual and unique. When training the artificial neural networks on data filtered by the autoencoder, the classification accuracy decreased. The result may imply that the high frequency content of the MEG signal is an important part for characterizing an individual's brain activation response.

Keywords: Hypertension, Magnetoencephalography, Cardiovascular Diseases, Artificial Neural Networks, Multi-Layer Perceptron, Long Short-Term Memory, Time Series Analysis, Muscle Sympathetic Nerve Activity, Neuroimaging, Beta Oscillation

Acknowledgments

Here we are, finally on top of the mountain after five years of studying at Chalmers University of Technology, with the flag proudly in our hands. It feels surrealistic to see the view and realize that we have reached our finish line. We remember our first day in August 2015 like it was yesterday, dressed in yellow shirts and excited for the upcoming years. The journey would not have been possible without the people we have met during our time at Chalmers, where we have gained friendships we value and cherish. You know who you are and we are grateful for all the laughter and all fond memories we have shared during these years. Thank you for always pushing and supporting us when the climb was steep. University would not have been the same without you and we hope our friendships will last for many years to come.

We want to thank everyone at Syntronic Research and Development's office in Gothenburg. Thank you for instantly making us feel welcome and part of the group. Especially, we would like to give an extra warm thank you to Alice Deimante Neimantaite, algorithm developer at Syntronic, for initiating the collaboration between Syntronic and Medtech West and thus making this thesis possible. We are deeply thankful for your input, feedback, knowledge and continuous involvement in our thesis.

Thank you Bushra Rias, PhD at Medtech West, for taking the time to meet with us to transfer the data and explain how it is structured and processed. Thanks to Nima Hajiabdolrahim, our supervisor at Chalmers, for your guidance through all the formalities and your valuable feedback on our thesis. We would also like to direct our gratitude to Henry von Hörsten for an excellent opposition, which demonstrated deep insight with interesting comments and questions.

We want to send warmly thanks to Justin Schneiderman, senior lecturer at the University of Gothenburg and MEG researcher at MedTech West. Your passion for conducting meaningful research inspired us to set the bar high. The time and commitment you have put into our thesis have been invaluable and it has been a true privilege to get to know you and be mentored by you.

We owe our greatest and heartfelt thanks to the fantastic role model Jonas Andersson, our supervisor at Syntronic. You have really gone out of your way and been so much more than a supervisor. Your knowledge, support, insight and leadership are never-ending sources of inspiration. There are not enough words to express how thankful we are for everything you have done.

Finally, our deepest gratitude goes to our respective families. Your love, support and guidance have been, and continue to be, what drives us forward. Thank you for always standing by our side.

We wish the best for all of you.

Anna Bakidou & Jonas Nord Odhner, Gothenburg, June 2020

Contents

Acronyms

AI Artificial Intelligence 2
ANN Artificial Neural Network 9–13, 18–22, 25, 27, 30, 38–43, 55
ANS Autonomic Nervous System 4, 5

FFNNs Feedforward Neural Networks 13, 15, 16

ICA Independent Component Analysis 9, 49

LSTM Long Short-term Memory 16, 19, 22–25, 27, 28, 31–33, 35, 37, 40, 55

MEG Magnetoencephalography 1–4, 8, 9, 19–21, 26, 27, 30, 31, 34, 35, 37–43, 46, 49, 50, 52, 53
MLP Multi-layer Perceptron 13, 14, 16, 19, 22, 23, 25, 27, 28, 30–33, 35, 37, 40, 55
MSE Mean Squared Error 11, 25, 26
MSNA Muscle Sympathetic Nerve Activity 1, 4, 7, 20, 21, 29, 31, 32, 34, 35, 38–43, 49, 50, 52

RNNs Recurrent Neural Networks 16

SQUID Superconducting Quantum Interference Device 8
SSS Signal Space Separation 9, 49

List of Figures

List of Tables

1
Introduction

A cascade of central and peripheral nerve signals is initiated when humans are subjected to stressful or attention-demanding distractions [1]. The standard peripheral nerve response induces dilation of the muscle vasculature so they can fill with blood, which prepares humans for physical exertion [1]. Recent research has shown that the strength of this regulatory signal varies across the population; many people's muscle vasculature constricts, rather than dilates, leading to long-lasting blood pressure elevation with each stressful distraction [1]. Such variable muscle sympathetic nerve activity (MSNA) responses lead to differences in the risk for development of essential hypertension, which is the leading risk factor for cardiovascular diseases and early death [1].

The individual difference in MSNA response can be identified by studying the brain activity initiated by stressful stimuli using magnetoencephalography (MEG) [1]. MEG is a non-invasive neuroimaging technique with high temporal and spatial resolution [2]. Currents generated on the cellular level induce a weak magnetic field which propagates relatively unaffected through the brain tissues [2]. The induced magnetic field is recorded with sensors placed in a grid pattern around the head [2]. The location of the current source is then determined from analysis of the measured magnetic field [1].

Prediction of hypertension development is important for the development of preventive treatments. However, an effective predictor does not exist [1]. The relationship between the MSNA response and elevated blood pressure suggests that MSNA could be a potential biomarker for predicting the risk of developing hypertension [1]. Deep learning models have been used to predict different medical conditions in various research studies and could be advantageous in predicting the risk of developing hypertension as well. However, the prediction procedure by a deep learning model is often less interpretable compared to traditional means of analysis, but the outcome can result in better performance [3]. Therefore, applying deep learning algorithms on MEG measurements on subjects whose MSNA responses are characterized could result in a reliable predictor.

In the interest of using MSNA activity as a predictor for essential hypertension development, a study indicated a correlation between the MSNA activity and neural oscillations in the beta band [1]. The definition of a neural oscillation is a wave generated by the superimposed electric activity of neural communication [4]. The beta band consists of frequencies between 13 to 30 Hz and are modulated by many factors, for instance mental activity and touch sensations [5]. Using MEG sensors located around brain regions with strong beta activity could therefore be used to search for brain responses that correlate with the MSNA responses [1].

1.1 Background

Syntronic Research and Development AB and MedTech West have recently initiated a joint effort to apply new techniques in artificial intelligence (AI) and machine learning to experimental data on neural activity to learn more about the human brain. The thesis was formed as a result of this newly formed collaboration and aims to develop and improve current algorithms for analysis of experimental neuroimaging data recorded with MEG. The project is of interest for Syntronic and MedTech West, as well as the scientific community, since the results may improve the analysis of various neuroimaging techniques and could lead to new technological and scientific insights.

1.2 Aim

The purpose of the thesis is to investigate the potential of applying deep neural networks on MEG data to predict the risk of developing essential hypertension. The available data set is of limited size and contaminated with noise, therefore, a first step will be to investigate autoencoder networks and explore their efficiency in reducing experimental noise levels. The study will lay a foundation for further research which aims at testing the relative predictive power of multi-layer perceptrons and recurrent neural networks. A long term goal is to develop a software which utilizes the benefits of AI to assist clinicians in prediction of hypertension development risk.

1.3 Delimitations

The scope of the thesis does not allow for the inclusion of evaluating all available neural networks for denoising and classification tasks. As such, autoencoder neural networks will be used to improve the signal-to-noise ratio of the MEG data. For the classification, multi-layer perceptrons and a special kind of recurrent neural network, long short-term memory, will be evaluated. Thus, only networks relying on supervised learning will be considered. By definition, when describing how an artificial neural network is trained, supervised learning is implied. Unsupervised learning techniques will not be addressed. The data used to train and evaluate the neural networks is limited to the data provided by Medtech West, which was obtained using low critical-temperature MEG measurements on a known demography.

The usage of deep learning algorithms in the field of biomedical engineering and usage of MEG raises several ethical questions. For instance, the limited data size requires analysis of the risk of data bias. If the software reaches clinical practice, the bias will affect the patients differently depending on e.g. gender and ethnicity. The negative impact of this potential scenario strengthens the need for considering the ethical aspects of the thesis. Hence, ethical aspects will be addressed. However, societal and ecological aspects will not.

The thesis is an important step for the long term vision of clinical implementation. However, such implementation aspects are beyond the scope of the thesis. As such, user interface, regulations and CE marking for the developed software will not be considered.

1.4 Research Questions

To reach a conclusion to the aim of this study, the following research questions will be examined:

a) Can hypertension be predicted by an artificial neural network applied on MEG data?

b) What are the arguments for using an artificial neural network as a predictor?

c) Is there a part of the MEG signal that is particularly correlated with the brain activation response?

d) Is there a difference in MEG measurements between e.g. age and gender?

2

Theory

This chapter presents theory which is important to fully understand the need for the thesis and the selected methods. First, an overview of the autonomic nervous system is given with a brief explanation of its subsystems, the nerve cell and neural communication. The information forms a basis for deep learning theory, since the concept of artificial neural networks is inspired by biological neural networks.

Thereafter, an introduction to blood pressure and its connection to the autonomous nervous system is provided. The section ends with an outline of hypertension, its connection to cardiovascular diseases and findings from previous work regarding the MSNA response.

Next, an overview of MEG theory is presented. The section begins with a presentation of the instrumentation and functionality and continues with a description of measurement artifacts and how they are removed. Details on the MEG data used in this thesis can be read in Appendix A and its metadata in Appendix B.

The chapter ends with a section about deep learning. First, a general description of artificial neural networks is given. This is followed by an overview of how neural networks are trained. Thereafter, three different network architectures are introduced: the multi-layer perceptron, the autoencoder neural network, and the long short-term memory network. Lastly, the section ends with the challenges of implementing deep neural networks in the health care sector, which is important to consider when evaluating the potential of the developed classifiers.

2.1 The Autonomic Nervous System

The autonomic nervous system (ANS) is one of two functional divisions of the central nervous system and manages involuntary, vital signs [6]. For example, it coordinates blood pressure and maintains homeostasis [6]. The balance is kept with the ANS's three subdivisions, which handle involuntary control: the sympathetic, parasympathetic and enteric nervous systems [7]. The sympathetic nervous system actives when the human body believes to be in a fight-or-flight situation [7]. It typically increases the rates or levels of vital signs [6]. The parasympathetic nervous system is active during normal conditions and decreases the rates or levels of vital signs [7]. The enteric nervous system works with gut functions [7]. The ANS consists of both high and low level structures including the cerebral cortex, hypothalamus, brainstem, spinal cord, peripheral nerves and receptors and target effector organs [8].

2.1.1 Neuronal Communication

Communication within the ANS is mainly operated through neurons [9]. A neuron, also known as a nerve cell, is the central nervous system's main internal communication unit [9]. The neuron consists of a cell body, which holds the nucleus, extended cell body branches called dendrites, nerve fibers named axons and an axon hillock which connects the cell body with the axon [9]. Together, the dendrites and cell body form an input surface that receives information from connected axons of other cells [9]. The information is then processed into the cell body for evaluation: if the incoming information exceeds a threshold, the axon hillock generates an output signal which travels through the axon to the nerve endings [10]. At the endings, there are synaptic boutons which connect to nearby neurons' dendrites or other cells [9]. Figure 2.1 illustrates the main components of a neuron.

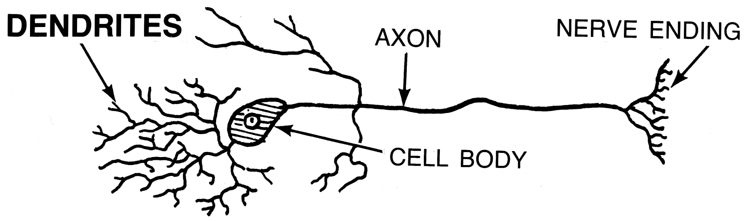


Figure 2.1: Schematic depiction of a neuron, with dendrites highlighted. From [11].

Communication between two neurons occurs in a synapse [12]. The most common synapse is based on chemical reactions where a presynaptic neuron transmits information to a postsynaptic neuron via signaling molecules [12]. The transfer initiates when the impulse traveling through the axon of the presynaptic neuron reaches the neuron's message depots and causes release of neurotransmitters [12]. The transmitters diffuse over the synaptic cleft, a separation space between the neurons, and binds to specific receptors at the postsynaptic neuron [12]. The binding enables ions to flow into the postsynaptic neuron [12]. Figure 2.2 presents a chemical synapse.

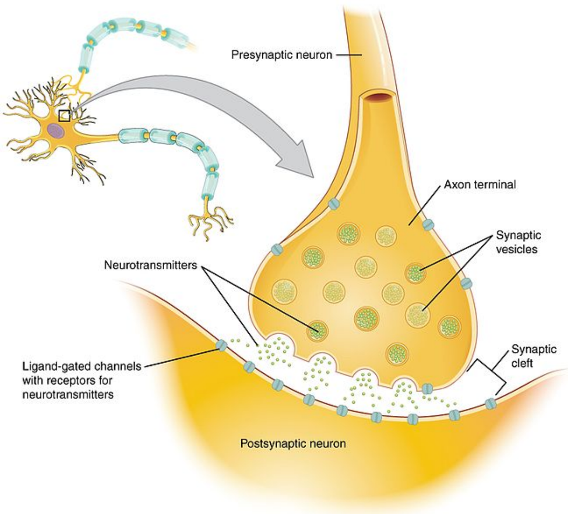


Figure 2.2: An illustration of a chemical synapse between two neurons. From [13].

2.2 Definition of Blood Pressure

Blood pressure is defined as the varying force on the arterial walls caused by the heart's blood pumping [14]. The pumping rhythm consists of two phases, the diastolic and systolic phase, which together form a cardiac cycle [14]. The flow is regulated in the heart's four chambers by pressure differences, created by opening and closing valves [14]. The diastolic phase initiates when the right atrium receives deoxygenated blood and the left atrium receives oxygenated blood [14]. As the atria are filled with blood, the pressure increases, which creates pressure gradients between the chambers [14]. The change in pressure triggers the opening and closing of valves and enables blood to enter the ventricles [14]. After the ventricles are filled, new pressure gradients are created and the systolic phase begins [14]. The pressure differences activate the valve mechanism and the chambers push deoxygenated blood to the lungs and oxygenated blood to the body [14]. The two phases are illustrated in Figure 2.3.

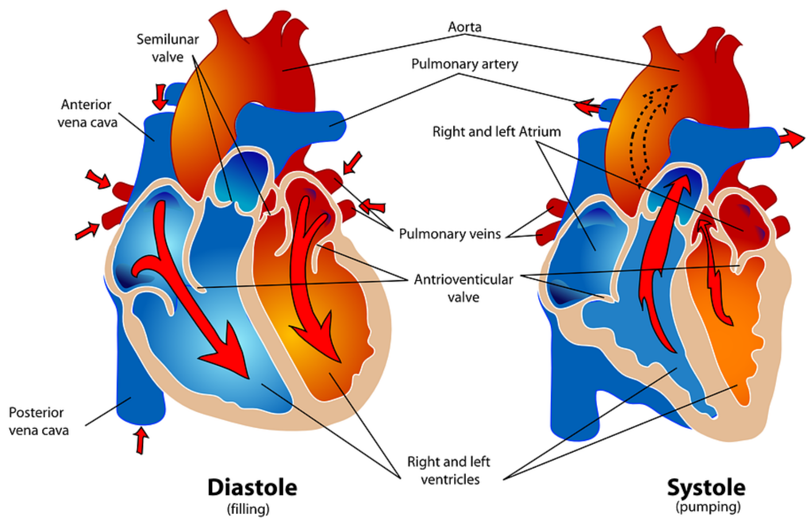


Figure 2.3: Illustration of a cardiac cycle including the diastole (left) and systole (right) phase, where the arrows indicates the direction of blood flow. From [15].

Blood pressure can be measured for both the diastolic and the systolic phases [14]. A measurement taken during the diastolic phase measures the lowest pressure level within the arteries [14]. The highest pressure level within the arteries is sampled if the measurement is performed during the systolic phase [14]. Blood pressure is typically measured in millimeters of mercury, mmHg [14]. A measurement on a healthy individual is defined to be below 90 mmHg and below 140 mmHg for diastolic and systolic phases, respectively [14]. In clinical settings, the average of the blood pressure during a cycle is often used [14]. A measurement on a healthy individual is then redefined to be in the range of 60 to 90 mmHg [14].

The sympathetic and parasympathetic nervous systems work together to balance the blood pressure level [14]. Receptors located within the arteries recognize pressure changes by sensing whether the artery diameter changes [14]. Depending on the changes in diameter, the sympathetic and parasympathetic systems respond differently [14]. The sympathetic system is stimulated during a stressful situation and triggers responses that increases the blood pressure [12]. For instance, stimulation of the sympathetic system causes the release of adrenal hormones [12]. These hormones increase heart rate, the strength of each heart contraction and narrow the arterioles and veins in certain organs [12].

2.2.1 Hypertension

Hypertension is defined as blood pressure which exceeds the defined thresholds for a healthy measurement [14]. The condition has a strong correlation with development of cardiovascular diseases [16]. People between 40 and 69 years of age with hypertension are in higher risk of developing cardiovascular diseases, and have a higher mortality rate [16]. Nonetheless, the symptoms of hypertension are often not recognized by the individual, and the condition is usually not discovered until a myocardial infarction or a stroke occurs [16]. Lifestyles affect the risk of developing hypertension where obesity, alcohol and dietary patterns are known risk factors [16]. Nevertheless, the cause of hypertension is unknown in the majority of cases [16].

An individual's blood pressure increases during mental stress due to the stimulation of the sympathetic nervous system [17]. However, the strength of the stimulus and where it is acting result in the individual's sympathetic response [17]. There is a hypothesis that the sympathetic response due to mental stress is individual [17]. The hypothesis comes from a study where the index finger was stimulated and the MSNA response was recorded [17]. The results showed that 50 % of the participants were able to inhibit the increased MSNA while the remaining 50 % could not [17]. The participants who could not inhibit the response had higher blood pressure during a longer time compared to the ones who were able to inhibit it [17]. The result, that was reproduced half a year, suggested that the MSNA inhibition is an individual trait [17].

Today, there is no reliable clinical tool available to predict the risk of hypertension [1]. One suggestion for a new biomarker is the MSNA response to stress, according to the strong relationship with increased blood pressure during mental stress [1]. By inducing stress and at the same time studying the MSNA response, a potential conclusion could be drawn about the person's risk of developing hypertension and, in the long-term perspective, the risk of developing cardiovascular diseases [1]. However, studying MSNA is not trivial due to its short-term nature and the invasive nature of the measurement techniques [1]. A recent suggestion to capture the MSNA activity is to record the neuronal activity in the brain when a stress stimulus is applied [1]. The neurons within the brain would react quickly to the stimulus, which would require a recording technology with high temporal and spatial accuracy [1].

2.3 Magnetoenchephalography

Brain activity can be studied by recording currents in the cerebral cortex, which is located in the uppermost layer of the brain and contains more than 10^{10} neurons

[2]. Dipole currents are generated when the neurons communicate with each other, which induces a magnetic field between 50 to 500 fT [2]. To fully understand the challenge of recording such a weak field, it can be compared to the earth's magnetic field in Europe, which is close to 50 μT [18]. Therefore, the recording equipment must be able to sense a magnetic field which is as much as several million times weaker in magnitude than the earth's field [2].

MEG is a neuroimaging technique for measuring the magnetic fields induced by the brain's neural activity [2]. The technique is based on a few hundred superconducting quantum interference device (SQUID) sensors organized in a helmet [1]. A flux transformer induces the brain magnetic flux to the SQUID sensors, which in turn converts flux changes to voltage [1]. A SQUID sensor is only able to operate in temperatures below 10 K and liquid helium is therefore used to manage the cooling inside the helmet [1]. Because of the low temperatures, insulation of a few centimeters is needed [1]. The general layout of a MEG helmet is presented in Figure 2.4.

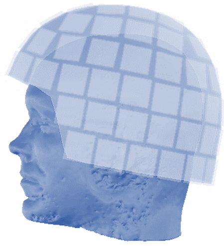


Figure 2.4: Illustration of a MEG helmet. The SQUID sensors, visualized as squares in the helmet shaped object, are organized in a structured pattern around the head to record the magnetic field generated by neural activity. From [19]. Modified with permission.

The conversion of magnetic flux changes to voltage requires the MEG to solely sense the brain's magnetic flux [2]. However, equipment in the surrounding environment close to the MEG recording can induce external magnetic flux to the flux transformer [2]. This interferes with the flux conversion and causes inaccurate and noisy measurement readings [2]. To minimize the risk of picking up external magnetic flux, the room in which the recording is performed is usually magnetically shielded [1],[2]. The shielding is performed by having special metal walls, which target the spectrum of the external noise [2]. Furthermore, two different flux transformers with different noise sensitivities are used: magnetometers and gradiometers [1]. A magnetometer has a single coil while a gradiometer has two coils [1]. The gradiometer is less sensitive to background noise compared to the magnetometer, since the functionality of the two coils cancels out the homogenous fields generated by external noise sources far from the sensors [1].

2.3.1 Artifact Removal

The high sensitivity of MEG causes the data to be corrupted by both physiological and environmental artifacts despite the precaution of recording in a magnetically shielded room [20]. Examples of such artifacts are cardiac, ocular and muscular movements, malfunctioning sensors and environmental frequency disturbances [20],[21]. To increase the understanding of the human brain, recordings of the brain activity should only consist of the signals of interest. Methods to remove these artifacts from the MEG recordings are therefore important, and current MEG research targets the development of new artifact removal methods [21].

Signal space separation (SSS) is a method to remove external sensor disturbances and can compensate for head movements during recording [22]. The MEG recording zone around the head is divided into subspaces: an internal space that represents the sources within the head, an external space that represents the sources of the external disturbances and a space between the internal and external which represents the MEG helmet [22]. The internal and external subspaces are separately composed of linearly independent components [22]. The measured magnetic field in the sensor subspace is then modulated by expressing the magnetic field from the two connecting subspaces [22]. A modern version of the SSS method is called temporally-extended SSS and is used to remove overlapping disturbances between the subspaces in time [1].

An additional method to compensate head movements during recording is to use head position indicator coils [1]. The coils are attached to the subject's head during the MEG recording and each coil can be identified with its unique powering frequency [1]. This allows to track the head position relative to the MEG helmet during the measurement while compensating for the movement [1]. The method is typically combined with the SSS method to model the fields in different subspaces [1].

A commonly practiced method for artifact removal is called independent component analysis (ICA) [20],[21]. The concept is to separate the measured signal into statistically independent components and remove those that are correlated with known artifact behaviors [20]. The remaining components are considered to be parts of the signal of interest and are used to reconstruct the recording [20]. ICA assumes that the independent components are non-gaussian distributed with variance equal to one [20]. In parallel with the MEG recordings, known artifact contributors, as for instance ocular and cardiac activity, are measured [20]. The statistically identified independent components can then be correlated with the measured artifact activity and be marked as disturbances [20]. After all components have been analyzed, those classified as disturbances are removed and the cleaned MEG recording remains [20].

2.4 Artificial Neural Networks

An artificial neural network (ANN) is a computational network which tries to simulate, in a rough manner, the decision process of the mammalian central nervous system [23]. The building blocks consist of simplified mathematical models of nerve cells, called perceptrons, illustrated in Figure 2.5.

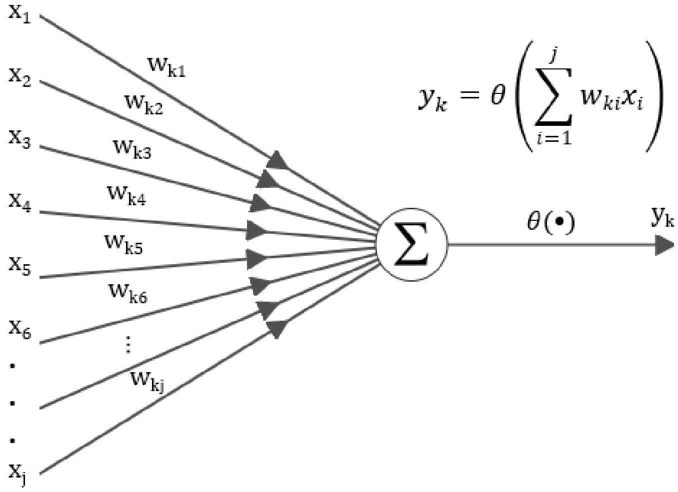
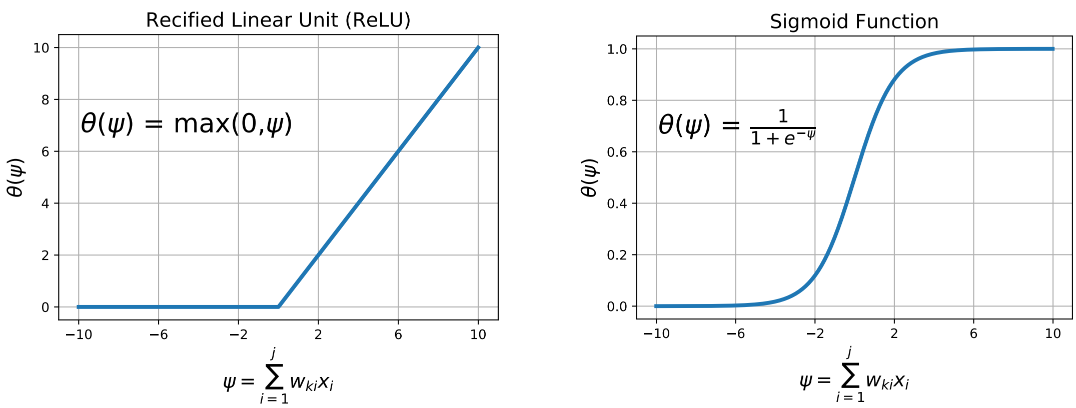


Figure 2.5: Schematic diagram of a perceptron. The k:th neuron, y_k, receives information from j incoming connections from other neurons, x_1, ..., x_j. Each connection is weighted by its corresponding weight, w_{kj}. The weighted sum is subjected to an activation function, denoted by θ, resulting in the output y_k. Note the similarity to the biological neuron in Figure 2.1, where the input connections are analogous to the dendrites and the output connection analogous to the axon.

Just like a biological neuron, the artificial neuron receives information from several incoming connections from other neurons. The strength of the connections are weighted and summed before being subjected to an activation function, which works as an abstraction of the action potential of a biological neuron [23]. Several activation functions exist, both linear and non-linear. Two commonly used, rectified linear unit (ReLU) and the sigmoid function, are illustrated in Figure 2.6.



(a) **(b)**

Figure 2.6: Illustration of two commonly used activation functions, (a) rectified linear unit and (b) the sigmoid function.

The motivation to use ANNs instead of traditional programming, i.e. hard coding the behavior of the program, is their information processing characteristics [24]. ANNs offers many advantages such as high parallel processing capability, the ability to handle imprecise and noisy data and their generalizability as function approximators [24]. These characteristics ensure fast processing, the ability to respond to a changing environment by updating the networks internal structure and accurate prediction even in the presence of noisy and inaccurate data [24]. Additionally, ANNs may discover information in a set of data that is not inherent to a human.

2.4.1 Training an Artificial Neural Network

The goal of training an ANN is to maximize the classification accuracy, which is achieved by determining the optimal set of weights [25]. This is a complicated task since weights in an ANN are interdependent [25]. As a consequence, if the value of one weight is changed, it does not only affect the connecting neuron but all other neurons having a connection to that neuron as well, including the output neurons [25]. In other words, one can not expect to maximize the classification accuracy by optimizing one weight at a time. Instead, a loss function, sophisticated algorithms, appropriate network structure and a large amount of data are used to iteratively update the weights and efficiently train an ANN.

2.4.1.1 Loss Functions

The purpose of the loss function is to measure the performance of the ANN [25]. In practice, the loss function measures the distance between the predicted output value and the desired target value [25]. Several loss functions exist, but most penalize the ANN based on the distance between predicted value and target value [25]. This procedure allows the network to make greater modifications to the weights if the loss is high, while ensuring that only small modifications are made as the loss decreases. Commonly used loss functions include mean squared error (MSE), binary cross entropy and categorical cross entropy.

2.4.1.2 Stochastic Gradient Descent and Optimizers

Finding parameters that extremize some objective function is a common problem when working with mathematical optimization. Many of these optimization problems, such as training ANNs, rely on gradient descent [25]. At the start of the training, the weights of the network are initialized randomly, often following a normal or uniform distribution with values close to zero [23]. After the first epoch, i.e. cycle of training, the cumulative loss is calculated [23]. To estimate in which direction the loss function decreases the most, the gradient with respect to all the parameters is calculated [23]. A small step, regulated by the network's learning rate, is taken in that direction which successively decreases the total loss [25]. After a number of epochs, the cumulative loss will reach a minimum and additional epochs will not further increase the accuracy of the network [23],[25]. Since ANNs often are nonlinear and the objective function may contain many local minima, some stochasticity is introduced to the gradient, resulting in the stochastic gradient descent algorithm [23],[25]. When the gradient of the loss function is computed, the actual gradient is replaced with an estimated value, which is calculated from a random subset of the data [23],[25]. The stochasticity can allow the optimization algorithm to escape local minima and search for a better solution [23],[25].

The learning rate regulates how fast the ANN may converge to a solution [23],[25]. If the learning rate is too low, convergence can take a very long time [23],[25]. On the other hand, if it is set too high, it may overshot the minimum and never converge [23],[25]. To address this issue, an adaptive optimization method to update the learning rate during training can be used [25]. One of the most common ones, Adam, adapts the learning rate for each trainable parameter based on past gradients [25].

2.4.1.3 Backpropagation

Backpropagation, short for "backward propagation of errors", is the heart of training an ANN [26]. The algorithm calculates the partial derivatives of the loss function with respect to the weights, starting with the gradient of the final layer of weights and ending with the gradient of the first layer of weights [26]. Partial calculations of the gradient from one layer are reused when calculating the gradient in a previous layer, which allows for efficient computations [26]. The error δ_j^{l-1} in the $(l-1)$:th layer depends on the error δ_i^l in the l:th layer. In practice, the loss based on the distance between the target output, y, and the predicted output, $\hat{y}$, constitutes the error in the last layer of weights [23],[26]. The error in a previous layer is calculated as a product sum of the error in the next layer, scaled with a factor $g'(b_j^{l-1})$, where g' is the derivative of the activation function and b_j^{l-1} is the product sum for neuron j in layer $l-1$ [23],[26]. The backpropagation algorithm and updating of weights is summarized in Algorithm 1 below.

Algorithm 1: Backpropagation and updating of weights

compute errors for the output layer: $\delta_i^L \leftarrow g'(b_i^L)(y_i - \hat{y}_i)$
for l = *L, L-1, ..., 2* **do**
  propagate error backwards: $\delta_j^{l-1} \leftarrow \sum_i \delta_i^l w_{ij}^l g'(b_j^{l-1})$;
end
for l = *1, 2, ..., L* **do**
  update weights: $w_{mn}^l \leftarrow w_{mn}^l + \eta \delta_m^l V_n^{l-1}$;
end

In Algorithm 1, η is the learning rate, V_n^{l-1} is the value of the n:th neuron in the $l-1$:th layer and w_{ij}^l is the value of the weight in the i:th row, j:th column of the l:th layer.

2.4.1.4 Hyperparameters and Overfitting

An always existing difficulty when training an ANN is to avoid overfitting. Overfitting occurs when the network over-optimizes the target values in the training data, with a loss of generality as expense [25]. To address the effects of overfitting, the data set can be divided into a training set, a validation set and a test set [25]. As the name implies, the training set is used to train the network. To further decrease the possibility of overfitting, the training set is often divided into smaller batches and the network is trained on one batch at the time [23]. There is always a risk that some of these batches are not representative for the overall data set, which would result in a poor estimation of the gradient. To prevent this, one often shuffles the data samples and create new batches after each epoch [23].

The validation set is used to monitor the performance of the ANN and provide an unbiased evaluation of the model [23]. At the start of the training, the accuracy obtained from the training data and the validation data is expected to be similar [25]. As the training progresses, the accuracy obtained from the training data may continue to increase while the accuracy obtained from the validation data starts to saturate [23],[25]. Such results may indicate that the network starts to overfit the training data and training should be interrupted [23],[25].

Another motivation for using a validation set is to evaluate the ANNs performance while using different combinations of hyperparameters [25]. Hyperparameters are an umbrella term for parameters that are typically tuned manually by the developer, e.g. number of neurons, number of layers and the learning rate [25]. When the network architecture yielding the highest accuracy on the validation set has been selected, its overall performance is evaluated on the test set [25]. The test set contains data that has never been seen by the ANN and provides a reliable approximation of the network's ability to generalize and correctly classify new data [23],[25].

Regularization can be applied to further prevent the risk of overfitting [23],[25]. The techniques involve imposing constraints on the network to restrain the magnitude of the weights from growing too large [25]. A common technique is dropout, introduced in 2014 [25],[27]. If dropout is applied to a layer, some percentage of the neurons and their connecting weights are deactivated during training [27]. The deactivated neurons are chosen randomly in each epoch [27]. The effect of dropout is that the ANN can not depend solely on a specific set of neurons since the connections between them are not active all the time [27]. As a result, the network is forced to learn a more balanced representation, which in turn reduces the network's risk of overfitting [27].

2.4.2 Feedforward Neural Networks

A common type of ANN architecture is feedforward neural networks (FFNNs). Neurons in a layer of FFNNs are fully connected, meaning that each neuron in a previous layer connects to all neurons in the next layer [28]. The connections are unidirectional, i.e. there are no feedback connections where the output from a neuron is fed back to itself or a previous neuron [28]. Common types of FFNNs are multi-layer perceptrons, convolutional neural networks and autoencoder neural networks.

2.4.2.1 Multi-Layer Perceptrons

The multi-layer perceptron (MLP) is a general function approximator which can be used to model or approximate mappings between given input and output spaces [23]. A popular application of MLP is classification of datasets [23]. The data to be classified can be of both binary and multiclass nature, which is reflected by the number of units in the output layer [29]. The architecture of a MLP consists of an input layer, hidden layer(s) and an output layer [23],[29], schematically illustrated in Figure 2.7. The optimal layout of the hidden layer(s) for a given task is difficult to predict analytically [30]. Instead, a manual testing procedure is often used to determine the optimal architecture [30].

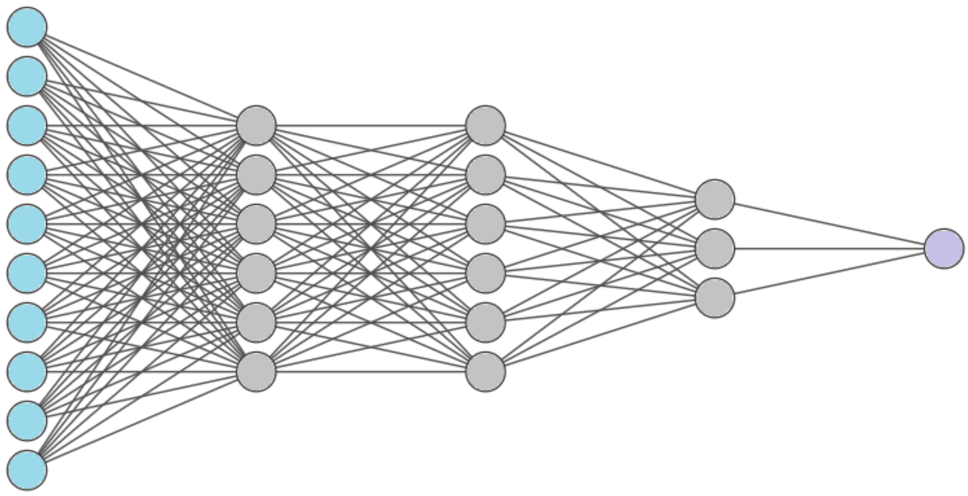
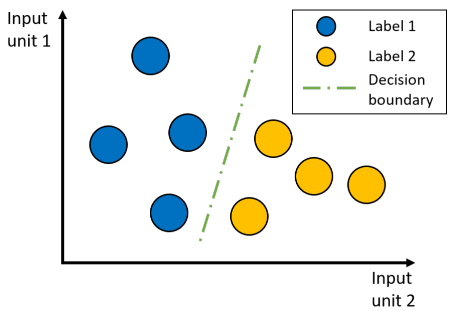
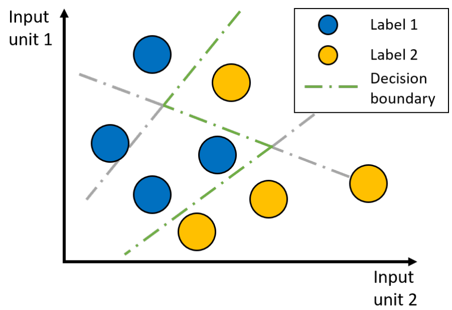


Figure 2.7: Schematic over a MLP architecture for binary classification. The MLP has ten input neurons (cyan), three hidden layers (gray) and one output neuron (lavender).

In a MLP network the classification is based on the input data and the corresponding target labels [23]. The input data is fed to the network and an output is produced [23]. The error between the target label and the produced output is then used to update the weights using backpropagation and stochastic gradient descent [23]. The updating of weights forms improved decision boundaries which can classify the data [23]. Figure 2.8 illustrates how a decision boundary can be formed based on the input data.



(a) Linearly separable data.



(b) Piecewise linearly separable data.

Figure 2.8: Classification by MLP. (a) The input data is linearly separable, since a single decision boundary can be placed between the two groups of different labels. (b) The input data is not linearly separable, but piecewise linearly separable when adding hidden layers and units. The joined decision boundary can then correctly separate the two groups of different labels.

In Figure 2.8a, the data consists of eight objects with the binary labels blue and orange. The objects with the same labels are clustered together and can be linearly separated by a line [23]. This line is called the decision boundary and acts as a divider: on one side the label is blue and on the other side the label is orange [23]. If a ninth, unseen object is feed into the network, the decision boundary is used to predict the corresponding label as the network's output [23]. In Figure 2.8b, the classes follow a more complicated structure and cannot be separated by a single, linear decision boundary [23]. The network can then be expanded by additional hidden layers and hidden units to add several linear decision boundaries [23]. Together, they form a piecewise, linear decision boundary which can separate the two labeled groups [23].

2.4.2.2 Autoencoders

Autoencoder neural networks constitutes a special class of FFNNs which may be trained with the same techniques [31]. Typically, minibatch stochastic gradient descent is used, where the weights of the network are updated using backpropagation with the input as target value [31],[32]. Simply copying the input to the output would not be especially useful. Instead, the hidden layer(s), commonly referred to as the bottleneck or the latent space, is often restricted [31],[32]. The most common method of restriction is to constrain the latent space to having a smaller dimension than the input, illustrated in Figure 2.9 [31],[32].

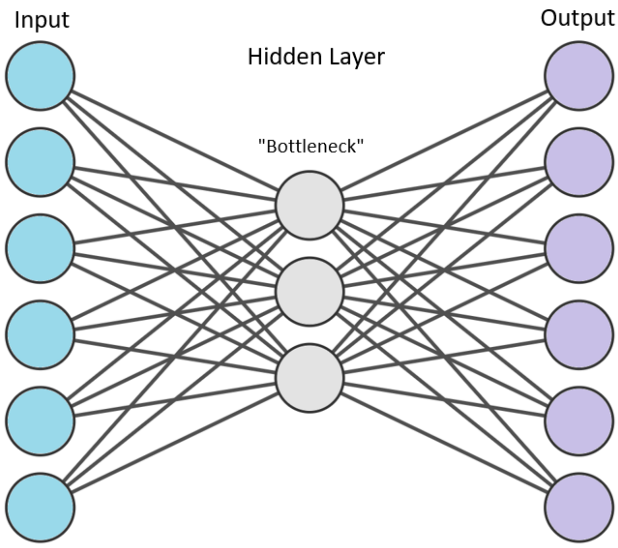


Figure 2.9: Schematic illustration of an autoencoder neural network. By restricting the hidden layer to have a smaller dimension than the input, the network is forced to ignore signal noise and only encode the most relevant features of the data. From the encoding, an approximate, denoised representation of the input can be reconstructed.

The network architecture forces the autoencoder to learn the most prominent features of the input [32]. If the input to the network was completely random, e.g. independent and identically distributed Gaussian noise, it would be difficult for the network to learn anything useful. However, if there is any structure in the data, the latent space variables would be forced to remember only relevant features of the input while unimportant features such as noise are forgotten [32]. For the network to be successful, the autoencoder must be restricted in a way that allows it to build a reconstruction accurately, but not reconstruct the input perfectly [31],[32].

2.4.3 Recurrent Neural Networks

Traditional neural networks, such as MLPs, excel at recognizing patterns. However, they lack persistence and the ability to understand context [33]. As an example, while reading a sentence, the human brain can understand the information presented in the ending of the sentence based on the words in the beginning of the sentence, due to our short-term memory. Traditional neural networks fail to do so. In an attempt to model this capability, recurrent neural networks (RNNs) were developed [33].

RNNs are derived from FFNNs and are similar in architecture, however, in RNNs the connections between neurons are not unidirectional [34]. Aside from feeding information from one layer of the network to the next, the output from previous predictions are also fed back to the input [34], illustrated in Figure 2.10. This allows the network to have, in some sense, a short-term memory, where the current prediction is dependent on previous predictions [34]. In other words, RNNs exhibit temporal dynamic behavior, making them ideal for sequential data.

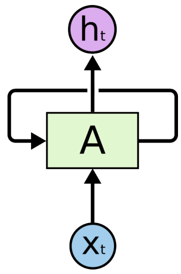


Figure 2.10: Illustration of a RNN. The network, A, receives an input vector, x_i, and produces an output prediction, h_t, The prediction is fed back to the input, providing additional information to be used for future predictions. Reproduced with permission from [33].

RNNs excel at recognizing patterns in time series data such as language modeling and translation, but can be difficult to train effectively [33],[35]. The difficulty stems from the exploding/vanishing gradient problem. The cause of these problems is not within the scope of this thesis. For the interested reader, a thorough discussion is provided in [35].

2.4.3.1 Long Short-Term Memory

As mentioned in the previous section, RNNs suffer from the exploding/vanishing gradient problem and struggle to learn to connect the information as the gap between information grows (e.g. the length of a sentence) [33]. These difficulties can be mitigated by a network structure called long short-term memory (LSTM) [33]. LSTM networks, illustrated in Figure 2.11, can be thought of as several copies of the same network that are connected through time. It is a modified version of RNNs, capable of learning short-term dependencies, even if the gap between information grows large [33].

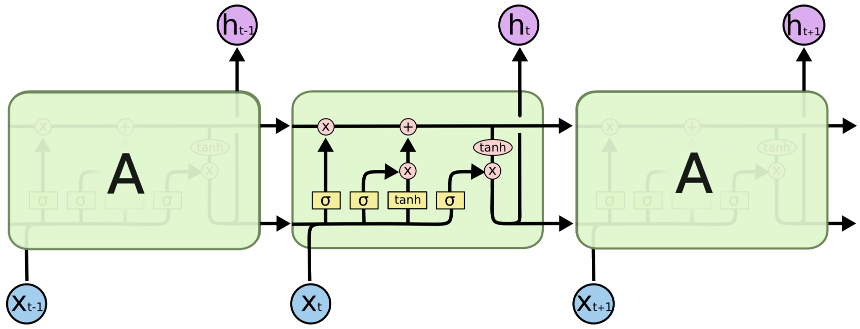
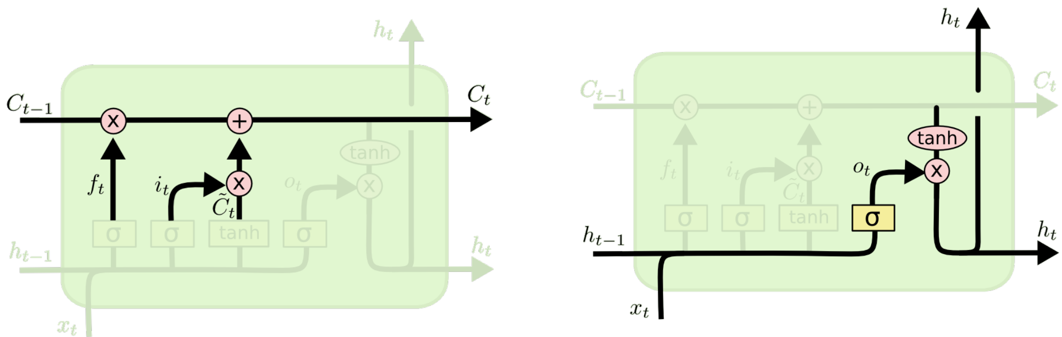


Figure 2.11: Illustration of a LSTM network structure. The network can be thought of as a chain of repeating blocks, A. The middle block illustrates the operations that are used to update the cell state and calculate the output, h_t. Reproduced with permission from [33].

The horizontal line running at the top of the middle block of Figure 2.11 is called the cell state [33]. The cell state transfers relative information between the blocks and can be though of as the network's "memory" [33]. Several carefully regulated gates are connected to the cell state with the possibility to alter it [33]. The first gate, a sigmoid layer, decides which information to disregard from the cell state [33]. The next gate has two parts, another sigmoid layer and a tanh layer. This second sigmoid layer decides which values should be updated, while the tanh layer creates a vector of values that are candidates to be added to the cell state [33]. These two vectors are combined and form an updated state [33].



(a) Illustration of the update of the cell state.

(b) Illustration of how the output, h_t, from each block in Figure 2.11 is decided.

Figure 2.12: (a) Clarifications of how the cell state C_t is updated and (b) how the output h_t is decided. Both figures show different parts of the network structure from the middle block of Figure 2.11. Reproduced with permission from [33].

Updating of the state is illustrated in Figure 2.12a. The old state, C_{t-1}, is multiplied by f_t, a vector containing values ranging from zero to one, deciding to which degree a given value in the state should be forgotten [33]. Next, i_t, the vector containing how much the values should be updated, is multiplied by the candidate values, $\widetilde{C}_t$ [33]. Then, the update of the cell state can be expressed as

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{2.1}$$

Lastly, the output, denoted by h_t in Figure 2.11 and 2.12b, is decided as a filtered version of the cell state [33]. A sigmoid layer generates values between zero and one, deciding which parts of the state to be transferred to the next block [33]. The same state passes through a tanh layer, forcing the values to range between minus one and one. The output from the tanh layer is multiplied by the output from the sigmoid layer to ensure that only the decided values are passed on to the next block [33].

2.4.4 Challenges with Clinical Application

Deep learning techniques demonstrate promising performances in solving medical problems, however, only a few systems have reached clinical implementation [36]. The gap between research and clinical application can be understood by looking at the core of deep learning from a clinical point of view [36]. The healthcare community is strictly bounded by regulations and ethical codes to offer a safe and equal care [36]. Therefore, it is essential for the deep learning systems to deliver equal and interpretable outcomes which the clinician can understand and communicate [36]. These requirements raise two major issues concerning the concept of ANNs: the available data and the decision making process [36].

The distribution of the accessible data has an impact on the ANN's performance because of its use during training, validation and testing of the system [37]. Given a large amount of quality data to train on, the system will be able to make correct predictions on unseen data [36],[37]. Nonetheless, medical data is normally corrupted with noise and is of limited size [37]. For instance, poor data acquisition and subjective labeling can occur which decrease the quality [36]. Furthermore, the available data the network is trained on may contain samples which do not represent the application domain properly [37]. The problems of working with a limited data set arouse as a result of the introduced bias and can cause the system to deliver incorrect predictions [36]. Common methods to reduce the impact of bias are to divide the sets in different ways and to change the sampling procedure [37]. If this is considered during the development of the deep learning algorithm, the bias can be reduced, but not eliminated [37].

The decision process of an ANN bases its predictions on a complex updating procedure and makes it noninterpretable [38]. This matter is commonly referred to as the black box which represents the hidden logic within the network [38]. One of the pillars in a clinician's oath is the full disclosure in relation to the patient. Thus, the use of an ANN becomes troublesome due to the clinician not being able to explain how the system reached its prediction [38]. In view of this issue, it was up until recently believed that improved performance of deep learning systems came at the cost of noninterpretability. [39]. However, new research presents the so called "modern machine learning methods" where the explainability of ANNs is comparable with traditional systems such as logistic regression [39]. Hence, incorporating the modern technology can increase the transparency and reduce the gap between research to clinical implementation [38],[39].

3
Methods

This chapter presents the methods used to approach the defined research questions. First, the basic steps to enable the application of deep learning algorithms are described. The software environment is presented where the data is processed and the deep learning models developed. Next, the procedure of converting the raw data to processed data packages which can be used to train and evaluate ANNs is presented. The data packages were constructed to enable analysis on different test cases.

Next, the chapter continues with the hyperparameter optimization strategy. The hyperparameters were tuned in an experimental fashion for the optimization of both the MLP and LSTM network. Continuing, a description of how the optimized networks were evaluated on the different test cases is provided. The processed data packages, which differs in data and size, were used to evaluate the ANNs. This allowed for evaluation of how the ANNs classification ability changed when the data size and quality varied.

Finally, an autoencoder network was used to filter the data within selected data packages. The development procedure for the autoencoder is provided followed by the evaluation process.

It is important to note that the term "epoch" is used differently in deep learning and MEG analysis. In MEG analysis, an epoch refers to the time interval between two stimuli, while an epoch in deep learning refers to a training cycle. It should be clear from the context which definition the term is referring to.

3.1 Software Environment

The programming language Python, version 3.7, was used for all software development. MNE, an open-source Python software, was used to read and process the MEG data files. The framework Keras, built on top of TensorFlow 2.0, was used when developing the ANNs. More specifically, the Keras' class Sequential was used to build, train and evaluate the networks. For data preprocessing, such as feature scaling and splitting the data into training and validation sets, the machine learning library scikit-learn was used.

3.2 Raw Data Processing

The data files provided by Medtech West were preproccesed according to the steps described in Section 2.3.1. 20 files in the format .fif were used, containing MEG data from 20 test persons. However, it was later discovered that one file was corrupted during a data transfer procedure and it was therefore excluded in this study. The corresponding person identified as an inhibitor. Thus, the available data contained MEG recordings from ten non-inhibitors and nine inhibitors.

The .fif-files were loaded into the workspace with the MNE-function *read_epochs*. The data was then converted to data frames using the function *to_data_frame* from Pandas library. From the data frames, either *all MEG channels*, only the *decent channels*[1], or only the *peak channels*[2] were selected together with data containing information about the stimulation train[3] and epoch length. Since Keras' Sequential model expects the data to be structured in a specific way, each data frame was converted to Numpy arrays. By using the sensor identifications and epoch length, the MEG data could be sorted into three different groups, where each group contained data from a specific pulse in the stimulation train. These groups will from now on be referred to as *pulse 1*, *pulse 2* and *pulse 3*.

As can be seen from the bottom diagram of Figure A.2 in Appendix A, each epoch is 4000 ms long. In order to reduce the dimensionality, and thus the complexity of training the neural networks, each epoch was truncated to only contain measurements close to the stimuli, which was issued at $t = 0$. As such, the epochs were cropped at $t = -100$ and $t = 1500$, resulting in each epoch containing 1600 measurement values.

Lastly, a label array was created. The label array enabled the input data to be mapped to a target value during classification. The labeling was based on a list provided by MedTech West where the MSNA inhibition value for each person was documented. Epochs from inhibitors were given label 1 and an epoch from a non-inhibitor was given label 0 for the binary classification. For multiclass classification, an epoch was instead classified with a number ranging from 0 to 18 corresponding to each subject.

3.3 Composition of Data Packages

After the processing, the data was composed to data packages intended for training, validating and testing the ANNs. Inspection of Table B.1 in Appendix B showed that the test person with ID 0293 had an MSNA inhibition value of 30.01 %. This value was considered to be a borderline case since the threshold between inhibitors and non-inhibitors was defined as 30 % according to the study conducted by MedTech West [1]. As such, a proper classification for this individual could not be done given the experimental uncertainty of the MSNA value. Including this individual with an arbitrary classification could potentially affect the training and have a negative impact on the validation and test accuracy.

To examine the influence of including a borderline case, two categories of data were assembled: category A where the data from the borderline case was excluded and category B where the borderline case was included and defined as an inhibitor. For both categories, three different data packages were constructed where each pack-

[1]Channels moderately activated by the applied stimuli. See Appendix A for more information.
[2]Channels strongly activated by the applied stimuli. See Appendix A for more information.
[3]See Appendix A for more information about the stimulus train.

age contained a training and validation set and a separate test set for the respective three pulses, with corresponding labels. The division is illustrated in Figure 3.1.

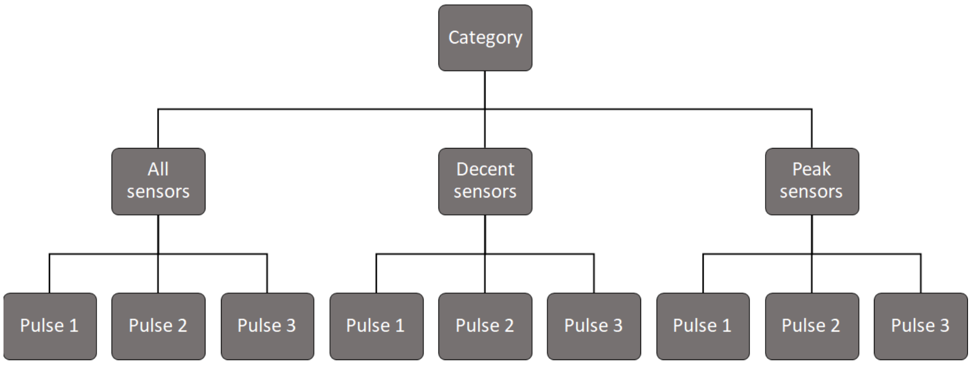


Figure 3.1: Illustration of how the different categories were divided into packages and subpackages.

The first package consisted of data from all active sensors during the MEG recordings. The second package consisted of data from the sensors in the decent sensor list and the third package consisted of data from the sensors in the peak sensor list. A random generator was used to pick one subject from each side of the MSNA threshold to be used as test set. The choice fell on subject 0179, having MSNA inhibition value 7.9 %, and subject 0283, having MSNA inhibition value 58.64 %. Thereby, the testing could be performed on data from both an unseen inhibitor and non-inhibitor.

To further test the ANNs classification capability, two similar categories were introduced, $\widetilde{A}$ and $\widetilde{B}$. Three packages for each category were constructed using the same procedure as for A and B, with the difference that the subjects for the test set were chosen manually. The two "extreme cases", ID 0301 and ID 0158, were chosen as they had the lowest, -132.47 %, and the highest, 83.74 %, MSNA inhibition values respectively.

Early findings in the thesis gave indications that the ANNs identified measurement specific patterns in the sensor data. However, it was observed that these patterns did not generalize to other individuals, which raised the issue of what the ANNs were learning from the data. A hypothesis was formed to address this issue, stating that brain functional activations may be individual. To test the hypothesis, an additional category was introduced, labeled X. Category X contained data from all subjects, including the borderline case. Instead of dividing it into training, validation and test sets based on subjects, it was divided based on percentage of epochs. In this way, each data set contained epochs from each subject. X differs from the other categories since it was used to train the networks to map a specific epoch to the corresponding subject. If this proved successful, the hypothesis may be confirmed.

To summarize, for each of the five categories, three data packages were constructed, corresponding to data from all sensors, sensors in the decent sensor list och sensors from the peak sensor list. Additionally, every data package was divided into subpackages containing data from either pulse 1, pulse 2 or pulse 3. The categories and their composition and purpose are summarized in Table 3.1.

Table 3.1: Summary of the content and purpose for the categories used for training, validation and testing of the ANNs.

Category	Inclusion of Borderline Case	Test Set	Purpose
A	No	IDs 0179, 0283	Binary classification of normal cases
B	Yes	IDs 0179, 0283	Binary classification of normal cases
$\widetilde{A}$	No	IDs 0301, 0158	Binary classification of extreme cases
$\widetilde{B}$	Yes	IDs 0301, 0158	Binary classification of extreme cases
X	Yes	10 % from all available epochs	Multiclass classification of ID dependence

3.4 Processing of Packaged Data

The data in each package was preprocessed before it was used to train and evaluate the networks. The training and validation data and corresponding labels were divided into a training set and validation set by applying the method *train_test_split* from the scikit-learn library. The splitting of training and validation data was randomized with the same randomization seed for the random number generators for each data package. This ensured that the same fraction of data allocated to the respective data sets each time, which guaranteed reproducibility and a fair comparison of the results between the different data packages and the different ANNs.

Next, the utility class *StandardScaler* from scikit-learn was used to standardize the data. An object of the class was created which allowed for the usage of class methods. The method *fit_transform* was applied on the training data, allowing the object to fit and transform the data. The same object and the method *transform* was later used to transform the validation set and the test set, thus ensuring that all sets have been scaled equally.

3.5 Optimization Procedure

Several different combinations of hyperparameters were tested while developing the MLP and LSTM networks. The objective was to find a network architecture that maximized the classification accuracy, evaluated on pulse 1 from data category A. Since the decent and peak channels represent subsets of all channels, the optimization was performed using data from all sensors.

Depending on the type of network being optimized, different hyperparameters were selected as fixed or variable. The hyperparameter fixation reduced the degrees of freedom and configurations of only those parameters that showed a clear impact on the validation accuracy were optimized in a combinatorial way. The hyperparameters were divided into two classes: local and global. Local hyperparameters represented specific characteristics of each layer and the global were general model parameters. The hyperparameters considered during optimization are summarized in Figure 3.2.

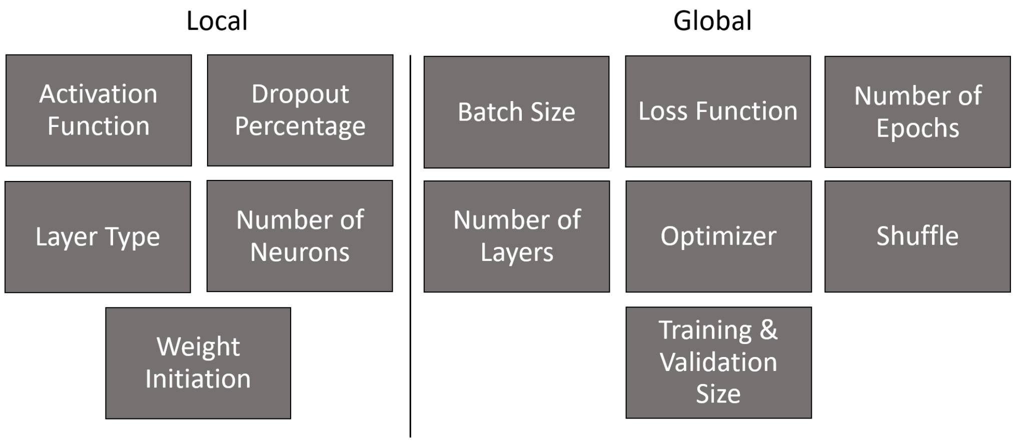


Figure 3.2: Local and global hyperparamters considered during network optimization.

Three mutual local hyperparameter selections were used for the MLP and LSTM optimization: sigmoid as activation function for the output layer, ReLU as activation function for the hidden layers and a dense layer as the output layer. Additionally, three mutual global hyperparameters were used: binary cross entropy as loss function, Adam as optimizer and the enabling of Sequential's shuffle feature, which shuffles the data before each epoch.

Specifically for the MLP optimization, the performance while using either one, two or three dense layer(s) as hidden layer(s) were evaluated. All layers had the same dimension, ranging from 20 to 800 neurons. Three different distributions were tested as weight initializers to evaluate how the weights initial state affected the rate of convergence: random normal, random uniform and Glorot uniform. 0-30 % were used as dropout to assess how the dropout percentage prevented overfitting. The network was trained for 150 epochs using a batch size ranging between 64 and 512 samples, using 80-90 % of the data for training and 10-20 % for validation. An overview of the hyperparameters used during the MLP optimization are presented in Table 3.2.

Table 3.2: Summary of tested hyperparameters during the optimization of the MLP network. The square brackets define an interval which includes the tested values.

Hyperparameter	**Input Layer**	**Hidden Layers**	**Output Layer**
Activation function:	-	ReLU	Sigmoid
Dropout percentage:	0 %	[0, 30] %	0 %
Kernel initiation:	-	Random normal, Random uniform, Glorot uniform	Random normal, Random uniform, Glorot uniform
Layer type:	-	Dense	Dense
Dimension:	1600	[20, 800]	1
		Global Hyperparameters	
Batch size:	[64, 512]	Epochs:	150
Hidden layers:	[1, 3]	Loss function:	Binary cross entropy
Optimizer:	Adam	Shuffle:	True
Training size:	[80, 90]	Validation size:	[10, 20] %

For the optimization of the LSTM network, two LSTM layers were used. Additionally, one dense layer was applied before the output layer to test how it affected the performance. Both LSTM layers had the same dimension, ranging between 100 and 300 neurons, while the dimension of the dense layer ranged between 0 and 120 neurons, where zero neurons should be interpreted as absence of the dense layer. The network was trained for 20 epochs using a batch size of 128 samples, with Glorot uniform initialization of the weights and 20 % dropout. 60 % of the data was used for training and 40 % for validation. The hyperparameters used to optimize the LSTM network are summarized in Table 3.3.

Table 3.3: Summary of tested hyperparameters during the optimization of the LSTM network. The square brackets define an interval which includes the tested values.

Hyperparameter	**Input Layer**	**Hidden Layers**	**Output Layer**
Activation function:	-	ReLU	Sigmoid
Dropout percentage:	0 %	20 %	0 %
Kernel initiation:	-	Glorot uniform	Glorot uniform
Layer type:	-	LSTM/Dense	Dense
Dimension:	1600	[100,300]/[0,110]	1
		Global Hyperparameters	
Batch size:	128	Epochs:	20
Hidden layers:	[2, 3]	Loss function:	Binary cross entropy
Optimizer:	Adam	Shuffle:	True
Training size:	60 %	Validation size:	40 %

TensorFlow's visualization toolkit was used to monitor the training and validation loss and accuracy. In particular, the validation loss was studied to determine whether or not the network started to overfit the training data. By this procedure, the appropriate number of epochs to minimize the risk of overfitting could be determined. When the different network architectures had been tested, the MLP and LSTM network with the highest validation accuracy were selected as the optimized network.

3.6 Evaluation of Optimized Networks

The optimized MLP and LSTM networks were evaluated on the constructed data packages. Recall the description of the division of packages in Figure 3.1. Separate evaluations on each branch was performed to examine the ANNs classification abilities. The training and validation sets were processed as described in Section 3.4 before training the ANNs. The training and validation accuracies were saved for each epoch during training and plotted in a graph which allowed for visual evaluation of the performance. Additionally, the final validation and training accuracies, i.e. the accuracies for the last epoch of training, were tabulated.

After the completed training, the test data was standardized according to Section 3.4. Keras' evaluation function was used to assess the trained network on the test data and the test accuracy was documented. The optimized network was then restored, i.e the neurons returned to their untrained states, and the procedure was repeated with the next branch. Once all branches for one data package had been used to evaluate the network, the same procedure was applied on the remaining data packages.

The evaluation of category X required adjustments to the optimized networks due to the change from binary to multiclass classification. Considering this, the loss function was changed to categorical cross entropy, the dimension of the output layer changed to 19 neurons and softmax was used as activation function. Apart from the mentioned modifications, the training, validation and test sets were processed as explained in Section 3.4. The optimized MLP and LSTM networks were then trained and evaluated on all data packages according to the procedure described for the binary classification.

3.7 Filtering With the Autoencoder

The autoencoder neural network used for denoising had the architecture illustrated in Figure 2.9, i.e. one hidden layer with smaller dimension than the input and output layers. Several different dimensions ranging between 10 and 100 were tested for the hidden layer in order to find suitable denoising and reconstruction properties. ReLU was used as activation function for the hidden layer while the output layer had a linear activation function. The network was trained for 20 epochs using a batch size of 512 samples, with Adam as optimizer and MSE as loss function. Pulse 1 and all sensors were used to train the network, where 80 % of the data was used for training and 20 % for validation. The data was shuffled before each epoch. The hyperparameters used to train the autoencoder are summarized in Table 3.4.

Table 3.4: Summary of tested hyperparameters used when developing the autoencoder. The square brackets define an interval which includes the tested values.

Hyperparameter	**Input Layer**	**Hidden Layer**	**Output Layer**
Activation function:	-	ReLU	Linear
Dropout percentage:	0 %	0 %	0 %
Weight initiation:	-	Glorot uniform	Glorot uniform
Layer type:	-	Dense	Dense
Dimension:	1600	[10,100]	1600
Global Hyperparameters			
Batch size:	512	Epochs:	20
Hidden layers:	1	Loss function:	MSE
Optimizer:	Adam	Shuffle:	True
Training size:	80 %	Validation size:	20 %

To evaluate the autoencoder's properties, four randomly chosen MEG epochs from the data set were plotted in the same window before and after being processed by the autoencoder. The results were then subjectively evaluated, where a smoothed version of the original signal without any high frequency components were desired.

The trained autoencoder was applied on data from selected packages to assess whether or not it could reduce experimental noise levels and thus improve the classification accuracy. The selected packages were then processed according the procedure in Section 3.4 and the evaluation was performed according to the method described in Section 3.6.

4

Results

The main purpose of this thesis was to investigate the potential of applying ANNs on MEG data to predict the risk of developing essential hypertension. This was approached via classification of subjects as inhibitors or non-inhibitors based on MEG data, recorded while the subjects were exposed to stressful stimuli. In this chapter, the results obtained to answer this question are presented. From the results, answers to the other research questions can be inferred.

First, the optimized MLP and LSTM network architectures are presented. The section provides a short description of how the hyperparameters affected the accuracy for each network. Next, the classification accuracies of the training, validation and test sets achieved by the MLP are presented, followed by the results obtained with the LSTM network. The chapter continues with the architecture of the developed autoencoder and its filtering properties. Finally, the accuracies obtained after the data was filtered by the autoencoder are provided.

Note that only classification accuracies obtained after fully training the networks are presented in this chapter. For the interested reader, the classification accuracies for each training epoch for both network types and all data packages are presented as complementary results in Appendix E.

4.1 Optimized Network Architectures

The hyperparameters tuned during the MLP network optimization can be examined in Table 3.2. It was found that one hidden layer was not sufficient to achieve adequate classification accuracies. Two hidden layers proved sufficient and the validation accuracy did not increase when adding a third. The validation accuracy continued to increase as more neurons were added to the hidden layers. To avoid overfitting the training data, the number of neurons were limited to be less than 800 in each hidden layer. 20 % dropout in each hidden layer proved to be a favorable trade off between mitigating overfitting, maximizing performance and shortening the training time it took to reach convergence. The network was trained for 150 epochs during optimization, however, it was observed that the validation loss did not decrease significantly after approximately 100 epochs. Concerning the batch size, an increased number of samples in the batch yielded higher validation accuracy. This is to be expected since more samples yield a better estimate of the "true" gradient. However, since the computational cost increases with the batch size, batches of more than 512 samples would not be practically feasible. The hyperparameters determined from the MLP optimization are presented in Table 4.1.

Table 4.1: Summary of hyperparameters for the optimized MLP network.

Hyperparameter	Input Layer	Hidden Layers	Output Layer
Activation function:	-	ReLU	Sigmoid
Dropout percentage:	0 %	20 %	0 %
Kernel initiation:	-	Glorot uniform	Glorot uniform
Layer type:	-	Dense	Dense
Dimension:	1600	800	1
Global Hyperparameters			
Batch size:	512	Epochs:	100
Hidden layers:	2	Loss function:	Binary cross entropy
Optimizer:	Adam	Shuffle:	True
Training size:	85 %	Validation size:	15 %

For the LSTM network, the classification accuracy increased as the number of neurons in the LSTM layers increased, up to 260 neurons. Adding a dense layer before the output layer had no impact on the classification accuracy. 20 epochs were used while optimizing the network, however, the network started to overfit the training data after approximately 10 epochs. The result from the optimization, i.e. the optimized hyperparameters, are presented in Table 4.2.

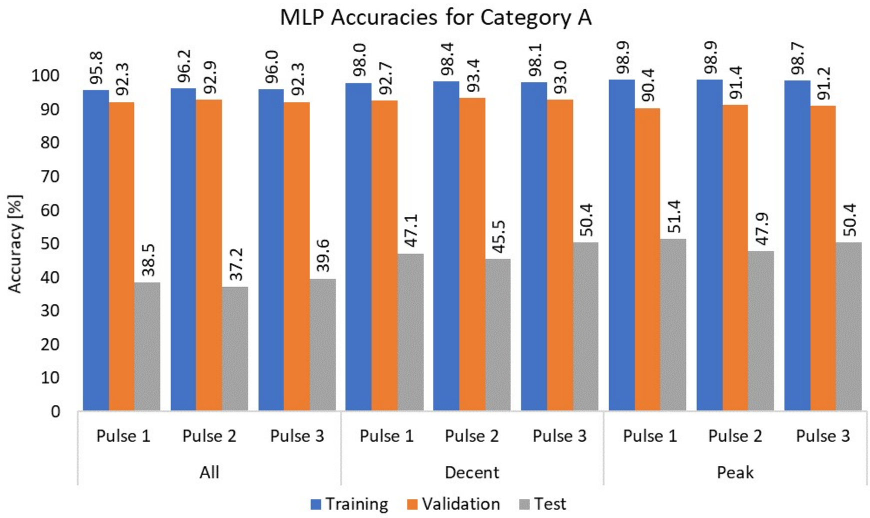
Table 4.2: Summary of hyperparameters for the optimized LSTM network.

Hyperparameter	Input Layer	Hidden Layers	Output Layer
Activation function:	-	ReLU	Sigmoid
Dropout percentage:	0 %	20 %	0 %
Weight initiation:	-	Glorot uniform	Glorot uniform
Layer type:	-	LSTM	Dense
Dimension:	1600	260	1
Global Hyperparameters			
Batch size:	128	Epochs:	10
Hidden layers:	2	Loss function:	Binary cross entropy
Optimizer:	Adam	Shuffle:	True
Training size:	60 %	Validation size:	40 %

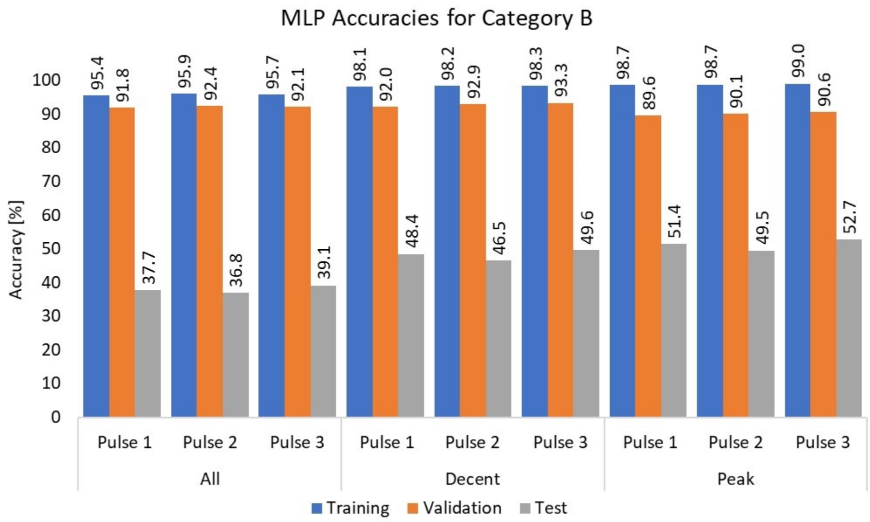
4.2 Multi-Layer Perceptron Accuracies

The accuracies achieved by the MLP network for data categories A and B are presented in Figure 4.1. The training and validation accuracies are, with one exception, over 90 % for both categories and all data packages. However, the test accuracies are consistently equal to, or worse, than chance (50 %).

Remember that the only difference between category A and B is the inclusion of the borderline case. For most of the data packages, category A, which did not include the borderline case, has somewhat higher accuracies (< 1 percentage point difference for all cases but one). This is most likely explained by statistical variations rather than influence from the borderline case.



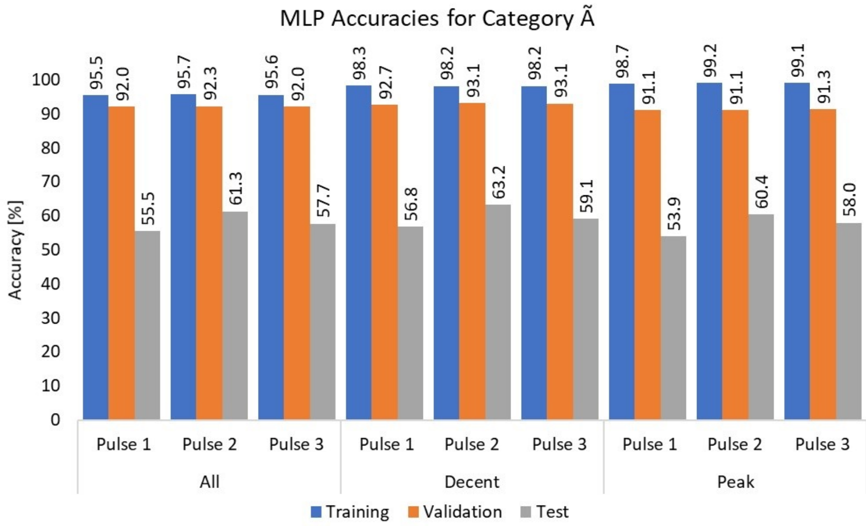
(a)



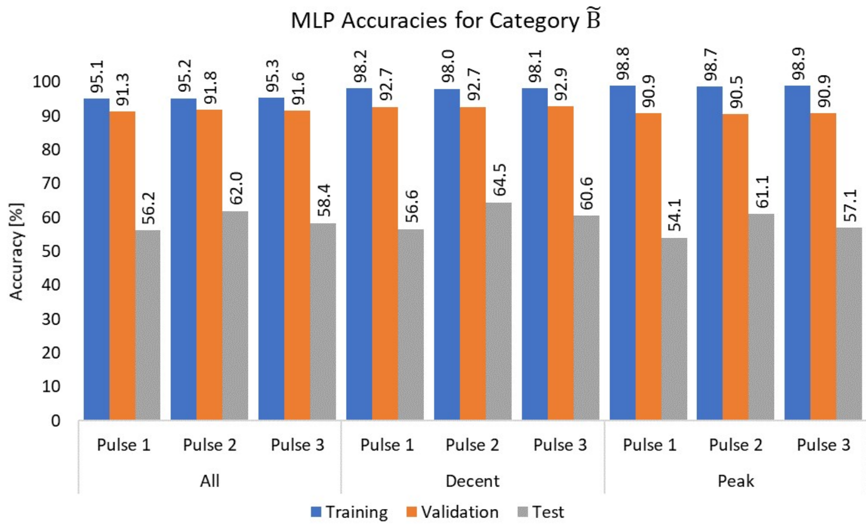
(b)

Figure 4.1: Training, validation and test accuracies for the MLP network after 100 epochs of training on (a) category A and (b) category B.

The results are similar for categories $\widetilde{A}$ and $\widetilde{B}$, presented in Figure 4.2. However, the test accuracies are considerably higher, reaching over 60 % for some of the data packages. Remember that the only difference between categories A and B and categories $\widetilde{A}$ and $\widetilde{B}$ is which individuals that were used as test set. Since MSNA response is believed to be approximately normally distributed across the population, the majority of individuals are within one standard deviation from the mean MSNA inhibition value. The responses from these individuals are thus expected to be similar and can be difficult to distinguish from each other. This would explain why the network fails to classify on categories A and B and why it is more successful on categories $\widetilde{A}$ and $\widetilde{B}$. From a clinical point of view, these results are good news. It is the individuals exhibiting an "extreme" deviation from the mean on the non-inhibition side of the spectrum that are likely to be in acute need of preventive interventions. The results are insufficient to be of practical research or clinical importance, but indicate that these individuals may be identifiable with the approach used in this thesis. This implies that ANNs applied to brain response MEG data may be used as a biomarker for identifying the development of essential hypertension.



(a)



(b)

Figure 4.2: Training, validation and test accuracies for the MLP network after 100 epochs of training on (a) category $\widetilde{A}$ and (b) category $\widetilde{B}$.

Category X was introduced to test the hypothesis whether brain functional activations are individual or not. As such, the MLP network was trained to map a MEG epoch to the individual from which the measurement originated. The results are presented in Figure 4.3. Recall that there is data from 19 subjects, i.e. approximately 5 % classification accuracy would indicate random guesses. Evidently, the classifier successfully identifies the origin for a majority of the MEG epochs. This may seem to contradict what was previously stated; that measurements from individuals with MSNA inhibition values near the average may be hard to distinguish from each other. However, it is important to differentiate the purpose between the different data categories. A, B, $\widetilde{A}$ and $\widetilde{B}$ were used to learn the classifier to map a MEG epoch to "inhibitor or non-inhibitor". Hence, the predictor is forced to learn to connect MSNA inhibition values to the MEG signals. This may be difficult since the MEG signal contains additional information that may not be specific to the MSNA response. On the other hand, for category X the predictor learns to map a MEG epoch to the individual from which the signal originated. As such, the network is not restricted and can base its prediction on all information contained in the signal.

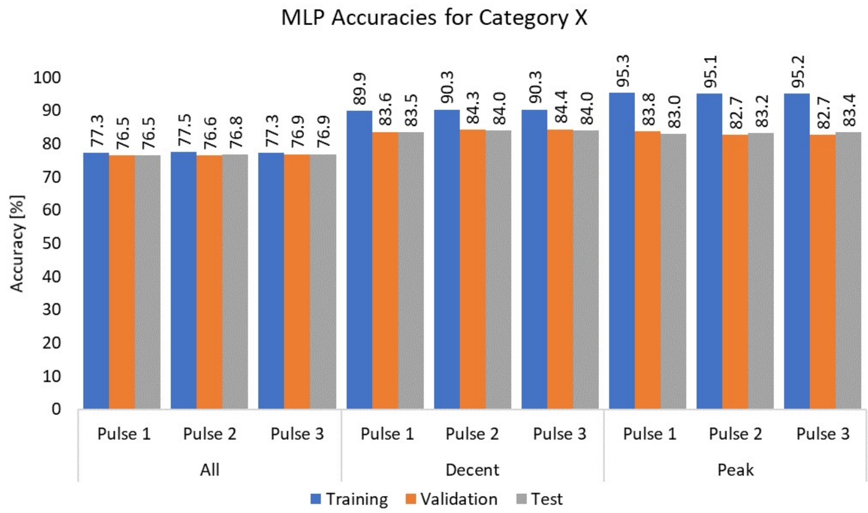
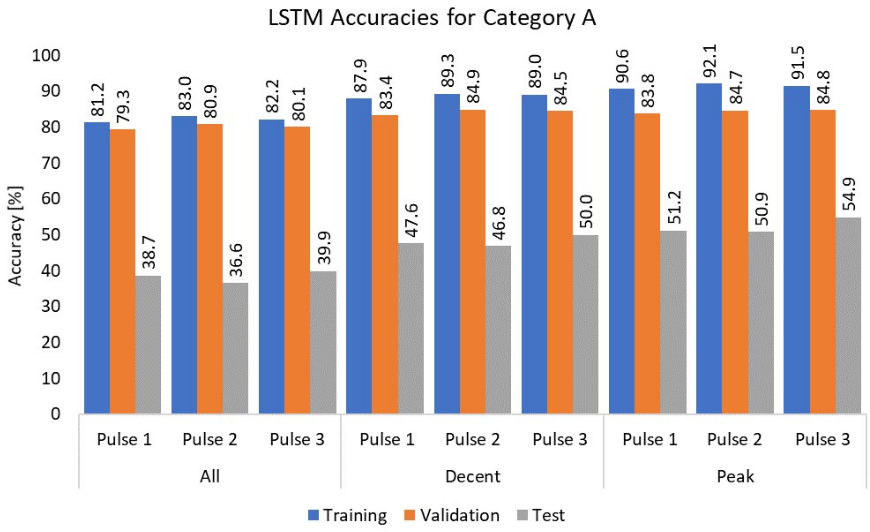


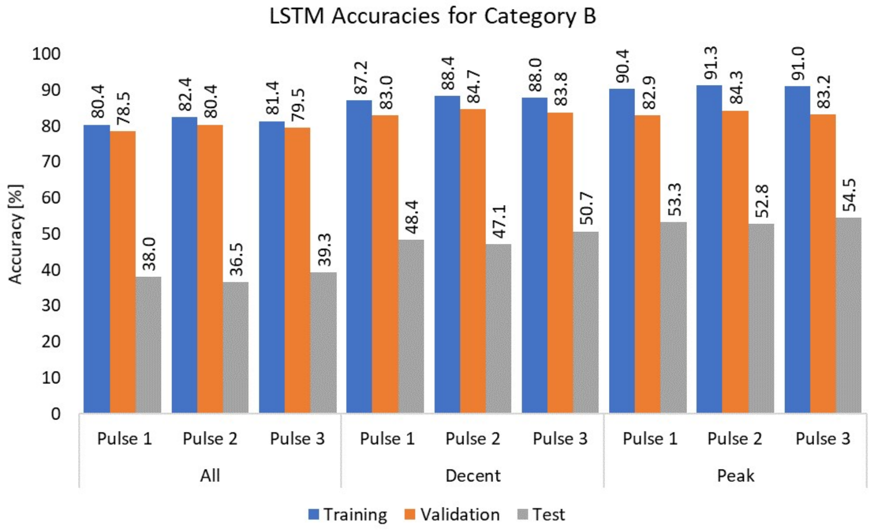
Figure 4.3: Training, validation and test accuracies for the MLP network after 100 epochs of training on category X.

4.3 Long Short-Term Memory Accuracies

The classification accuracies obtained with the LSTM network for data categories A and B are presented in Figure 4.4. As with the MLP, the LSTM network failed to classify correctly and was, as best, guessing whether the test subjects were an inhibitor or non-inhibitor. No statistical differences can be observed between category A and B, implying that the borderline case has no, or at most negligible, impact on the network's ability to classify correctly.



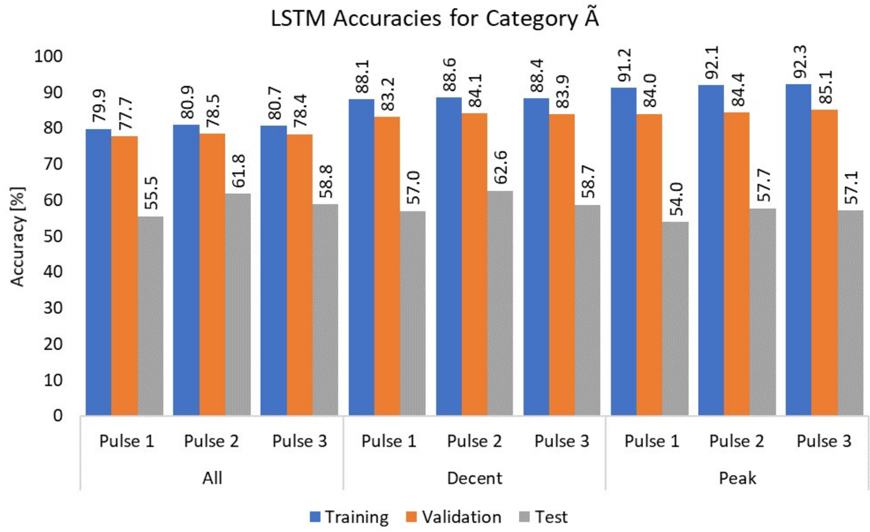
(a)



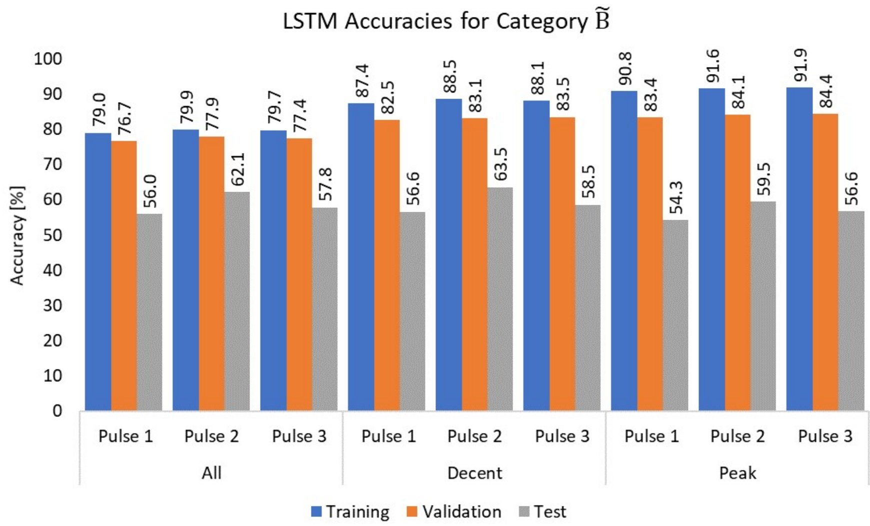
(b)

Figure 4.4: Training, validation and test accuracies for the LSTM network after 10 epochs of training on (a) category A and (b) category B.

Like the MLP, the LSTM network was able to significantly increase the test accuracy when classifying on subjects with an "extreme" MSNA inhibition value, see Figure 4.5. Especially interesting to note is that the test accuracy for pulse 2 is consistently higher than the accuracy for pulse 1 and pulse 3. The same can be observed for the MLP. This is consistent with recent findings from MedTech West. The findings indicate, for reasons that are currently unknown, that the second pulse in the stimulus train triggers the MSNA response most effectively.



(a)



(b)

Figure 4.5: Training, validation and test accuracies for the LSTM network after 10 epochs of training on (a) category $\widetilde{A}$ and (b) category $\widetilde{B}$.

The training, validation and test accuracies obtained for category X are presented in Figure 4.6. The results are good enough to suggest that it is possible to classify from which subject a measurement originated. However, the accuracies are considerably lower than the ones achieved by the MLP. It should be noted that these accuracies can be significantly improved by modifying the LSTM network's architecture. The interested reader can find a minor discussion about it in Appendix D.

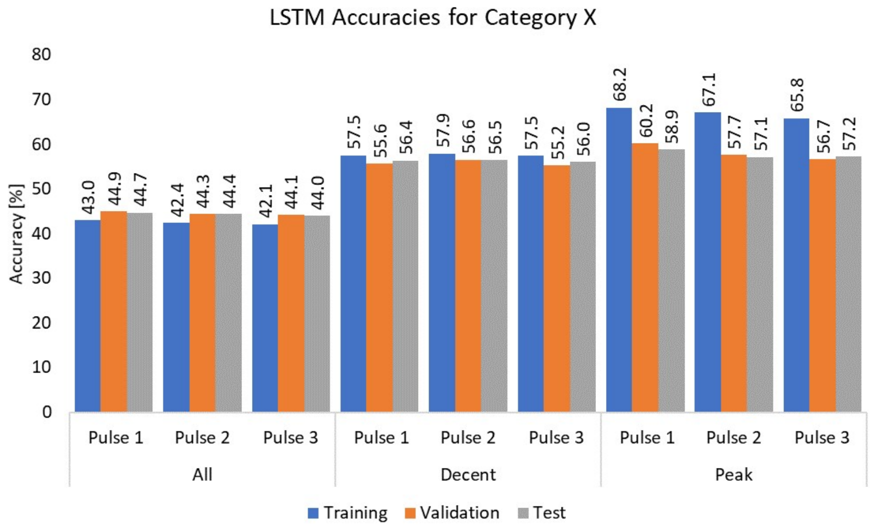


Figure 4.6: Training, validation and test accuracies for the LSTM network after 20 epochs of training on category X.

4.4 Denoising with the Autoencoder

The final autoencoder neural network had the architecture presented in Table 3.4, with 30 neurons in the hidden layer. The autoencoder functioned as a broad band-pass filter when applied on the MEG signal, illustrated in Figure 4.7. Note that the reconstructed signal follows the original signal considerably well, indicating that the autoencoder learned global measurement patterns and ignored individual differences. The discovery indicates that there is a structure in the MEG data which can be identified with deep learning models. However, this does not necessarily imply that the identified structure is correlated with MSNA response.

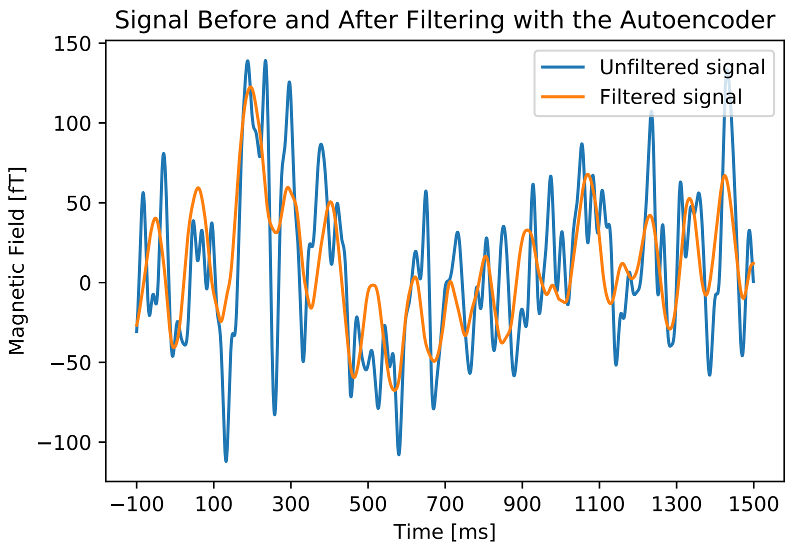


Figure 4.7: An unfiltered signal from a randomly chosen MEG epoch (blue) and the same signal after filtration with the autoencoder (orange).

4.4.1 Accuracies After Applying the Autoencoder

The autoencoder neural network was applied on pulse 1 from category $\tilde{A}$, $\tilde{B}$ and X. These categories were selected based on their higher test accuracies with the MLP and LSTM classifiers, as compared to A and B. Category X was included to examine how applying frequency filters to the data affected the classifier's ability to successfully map a MEG epoch to the corresponding individual. It was considered sufficient to examine one of the three pulses to evaluate how the autoencoder affected the classification accuracy.

The obtained accuracies are presented in Figure 4.8 for the MLP and Figure 4.9 for the LSTM network. Observe how the test accuracies for $\tilde{A}$ and $\tilde{B}$ are unaffected by the filtering, yielding similar results as when classifying on the unfiltered data. The same is not true for category X. Training, validation and test accuracies are considerably lower than when classifying on the unfiltered data. There may be multiple reasons behind these results. However, it is plausible that the global measurement patterns identified by the autoencoder have a correlation with MSNA response, which would explain why $\tilde{A}$ and $\tilde{B}$ are unaffected by the filtering. Additionally, evaluation on data category X suggests that high frequency content in the MEG signal is an important part for characterizing an individual's brain activation response.

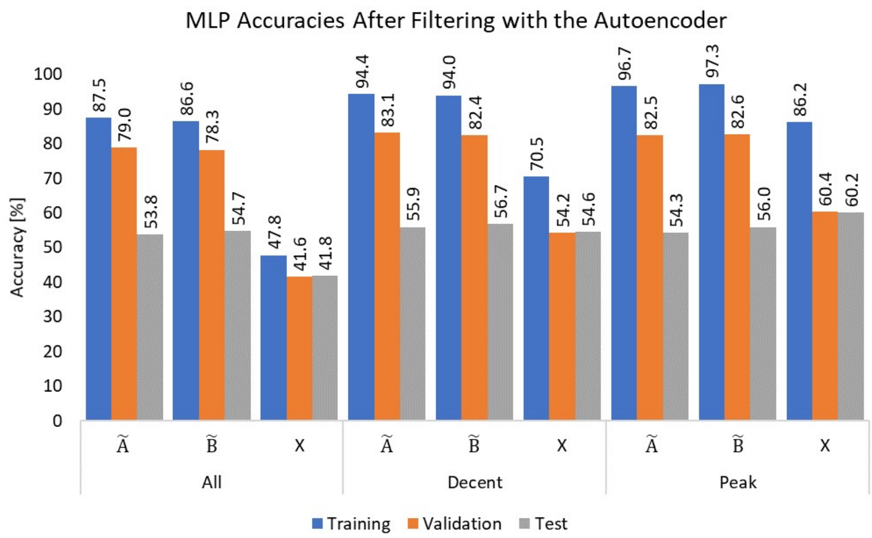


Figure 4.8: Training, validation and test accuracies for the MLP network after applying the autoencoder. Evaluated on pulse 1 from $\tilde{A}$, $\tilde{B}$ and X. The network was trained for 100 epochs.

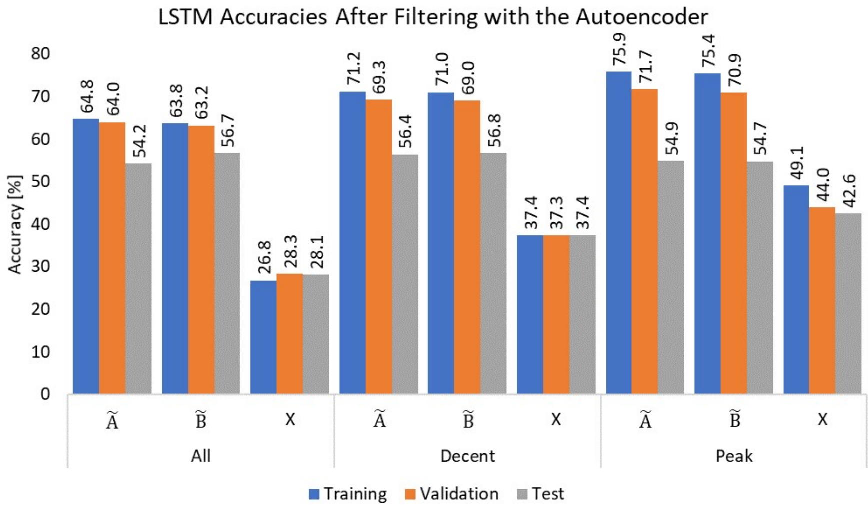


Figure 4.9: Training, validation and test accuracies for the LSTM network after applying the autoencoder. Evaluated on pulse 1 from $\tilde{A}$, $\tilde{B}$ and X. The network was trained for 10 epochs on $\tilde{A}$ and $\tilde{B}$ and 20 epochs on X.

5

Discussion

In this chapter, the interpretations, implications and significance of the results are discussed. First, the probable reasons behind the disparity between the training/-validation accuracy and the test accuracy are addressed. Next, points made in the preceding discussion are used to draw inferences about the ethical difficulties with clinical implementation, with focus on the potential impact of data bias. The chapter continues with a critical review of the methods used to obtain the results, with suggestions on future improvements in regard to the knowledge acquired from this thesis. The final part of the chapter covers aspects concerning the autoencoder neural network. The autoencoder's architecture and the training procedure are discussed together with suggestions on improvements, including recommendations on how to accurately examine the signal content of the MEG measurements. For ease of reference, the chapter is concluded with a summary of the main points discussed.

As previously mentioned, the results reveal that there is a clear discrepancy between the training/validation accuracy and the test accuracy for all binary classification cases. This is not uncommon in deep learning applications and there are many possible reasons why it can occur. One possible explanation is an insufficient amount of training data, but this is likely not the main reason for the discrepancy. Every data package contained at least 52 000 samples (peak sensors) and, at most, 410 000 samples (all sensors). Hence, the amount of samples should be sufficient to learn a relationship in the data, even if it is very subtle. Left to consider are mainly three possible explanations: there is no relationship in the data, the chosen deep learning algorithms are not suitable for this application, or the data sets are not sampled from the same distribution. The three reasons will be discussed in the following three paragraphs, arguing that the most reasonable explanation for the discrepancy is that the data sets are not sampled from the same distribution.

The MLP network is a universal function approximator that can provide approximate solutions by regression analysis, even if the solution is strongly nonlinear and complex. Given a suitable network structure, sufficient number of samples to train on and a relationship in the data, the MLP will likely succeed in finding an approximate solution to the problem. The results indicate that the MLP identifies a relationship in the data and can classify $> 99\,\%$ correctly for the training set and $>$ $93\,\%$ for the validation set. This suggests that the MLP is well suited for identifying real patterns in the the data analyzed in this thesis. Like the MLP, the LSTM network can classify most of the samples in the training set ($> 92\,\%$) and validation set ($> 85\,\%$) correctly. Hence, there are strong indications that the developed networks are suitable for this application and that there exist identifiable structures in the data. With that said, it is unlikely that the best network architectures were used in this thesis, even after a thorough optimization procedure. However, finding the best architectures was beyond the scope of this thesis; the ANNs were optimized to be sufficient to evaluate if they were capable to predict the risk of developing essential hypertension from MEG data. The results indicate that it might be feasible, at least for individuals with MSNA response values far away from the mean value. It would thus be interesting to test additional network architectures including deeper networks, and evaluate if the test accuracy can be increased. It would also be interesting to apply and evaluate other types of deep learning algorithms. For example, convolutional neural networks could be applied to MEG epoch image maps[1].

When constructing the data sets, 17 subjects were allocated to be used as training and validation sets, while 2 subjects were allocated for the test set. The data from the 17 subjects was shuffled before being divided into the training and validation sets, ensuring that each set contained MEG epochs from each of the 17 individuals. As such, the data in the training and validation sets were in some sense interchangeable. The same is not true for the test set, which contained MEG epochs from individuals that the networks had never seen before. The results show that the patterns that the network identified were not generalizable to new individuals. It might be that the ANNs learned to identify from which subject a measurement originated, rather than to binary classify the corresponding MSNA inhibition values. This was the reason to why category X was introduced. Evaluation on category X proved that it is possible to map a MEG epoch to the individual from which the signal originated. The MEG data was recorded under strict research conditions with careful precautions taken both during recording and processing of the raw data, to minimize the influence of disturbances and ensure reproducibility. However, it can still not be ruled out that the classifiers learned to identify measurement specific noise, rather than individual differences in brain activation responses.

The strength of the ionic current in the synapses and the correspondingly induced magnetic field is dependent on how many of the nerve cells that synchronously receive inputs in response to the stimulation. Furthermore, the anatomy of the brain, and thus the synaptic connections, is determined by genetic factors, individual life experiences and life circumstances (e.g. psychical and mental shape, alcoholism, neurological disorders etc). As such, it is entirely plausible that brain functional activations are individual and unique. This, together with MSNA inhibition value being approximately normally distributed, makes it problematic to divide the population into two groups (inhibitors or non-inhibitors). It might be that the individual variation in MSNA response within the two groups are too substantial to be binary separable. Since this could make the MSNA response relationship both complicated and subtle, the ANN instead learns to identify individual differences in the data.

Note that individual differences do not necessarily mean that there is a difference in MEG recordings between e.g. age and gender. However, since the brain deteriorate and nerve cells die as we get older (in varying rate dependent on individual factors); and the male brain on average is somewhat larger than the female brain, it is likely that there is a difference that is not only individual. This is problematic when developing a classifier to be used in clinical settings, since lots of diverse data would be needed in order to avoid a bias when training the classification network.

[1]See Figure A.2 in Appendix A.

Collecting this data would be expensive and involve lots of manual labor. Additionally, recall from Section 2.4.4 that the healthcare community are governed by strict regulations and ethical codes. These codes and regulations are difficult to maintain when using deep learning algorithms for clinical purposes, due to the "black box problem". With this in regard and the problem with data bias, it is unlikely that a clinical decision support system based on MEG data will reach clinical practice in the near future. However, the results obtained in this thesis imply that it might be feasible to use ANNs to predict the risk of developing essential hypertension. With improved classification accuracy, ANNs applied to MEG may still be used in research applications.

The discovery that MSNA correlates with MEG signals is very recent. As such, upon initiation of this thesis, little was known about the correlation. As a first step, it was decided that this thesis should focus on analyzing time series data. It was believed that the MSNA response should be strongest in time-windows close to the stimuli. Therefore, the MEG epochs were cropped 100 milliseconds before and 1500 milliseconds after the stimuli. The decision was based on an educated guess; there is no way to know if the data analyzed contained too much or too little information. Analyzing too much data increases the dimensionality and the complexity of the ANNs which makes it harder for the networks to provide an accurate prediction. On the other hand, if the data contains too few time steps, the available information might be insufficient for the neural networks to learn anything of interest. Fundamentally, further studies needs to be conducted in order to gain a better understanding of the correlation. With more knowledge, the MEG signal that are correlated with the MSNA response could potentially be isolated and characterized both in the frequency and time domain. This would significantly raise the probability of successfully training an accurate classifier.

In addition to not knowing which parts of the MEG signal that are correlated with the MSNA response, the exact location of the activity in response to the stimulus is also unknown. This is why the data was divided into all, decent and peak sensors, where the peak sensors are concentrated around the Rolandic area; the part of the brain believed to be most strongly activated by the applied stimuli. The findings from this thesis indicate that it might be true. With few exceptions, the classification accuracies obtained when training on data recorded from the Rolandic region are higher relative to when training on data from the decent sensors or all sensors (which covers a larger area of the brain). This is especially interesting since the neural networks were optimized using data from all sensors, which contained eight times as many data samples compared to the peak sensors. As such, the optimized networks were oversized, i.e. had too many trainable parameters, when training on data from the peak sensors. Thus, if the networks were instead optimized using data from the peak sensors, higher training and validation accuracies could have potentially been achieved. However, due to the difficulty of generalizing to other individuals, it is unsure whether the test accuracies would have been affected. With this in mind, in future research where MSNA is studied by applying ANNs to MEG data, it is recommended to compose the data of measurements from sensors covering regions with strong activation due to the applied stimulus, such as the Rolandic area.

The magnetic field induced in the brain is several million times smaller in magnitude than the Earth's magnetic field. This, together with bodily artifacts and disturbances from the surrounding environment, makes the application of MEG difficult. Even with strict precautions and extensive data processing, the signal will be contaminated with noise. Here, noise refers to both noise from the surrounding environment and parts of the signal that are not correlated with MSNA. With this in mind, it was believed that the data needed to be processed to attenuate the noise before training the ANNs. Upon visual inspection of MEG signals from different subjects, a low-frequent similarity in the signal was observed. This led to the assumption that high frequencies constitutes individual differences in the data (i.e. noise). However, without any prior knowledge about what frequencies in the MEG signal that are correlated with the MSNA response, it was not practical to simply hard code the cutoff frequency. Instead, with the expectation of finding the global features and attenuating individual features, it was decided to filter the data using an autoencoder neural network.

When developing the autoencoder's architecture, the amount of variable hyperparameters was limited to the dimension of the hidden layer. Since the autoencoder's objective is to encode the signal and then rebuild it, zero loss would indicate that the signal was reconstructed perfectly. As such, when "optimizing" the autoencoder, it would be irrational to use the loss function as the main performance measure. Instead, the performance was subjectively evaluated, where a smoothed version of the original signal without any high frequency components were desired. As a result, low frequency components were kept due to the believed correlation with the MSNA response, while high frequency components were filtered out due to the assumption that they were noise. This procedure implies that important features in the data can be lost when using the autoencoder. It would thus be interesting to test additional autoencoder architectures and evaluate the performance of the classification networks on data with different frequency contents. This would also yield a better understanding of which parts of the MEG signal that are of importance in relation to MSNA. For instance, increasing the dimension of the hidden layer allows for more information to be kept, while decreasing it would focus more on the global features of the data.

Furthermore, the data used to train the autoencoder is an important aspect to consider. In this thesis, the autoencoder was trained on data from pulse 1 and all sensors. However, the results show that pulse 2 yielded the highest classification accuracies for both the MLP and LSTM network. The finding is reassuring, since researchers at Medtech West have found the same correlation by using conventional, statistical methods involving time-frequency analysis of the three stimulation pulses. Additionally, as previously discussed, data from the peak sensors seem to contain most information. Thus, it could be more suitable to train the autoencoder on pulse 2 and the peak sensors, since the data would presumably be more strongly correlated with the MSNA response. Secondly, a thorough examination of the signal content in the MEG measurements should be performed. By doing so, a better understanding of what parts of the signal that should be kept after filtration would be obtained, which reduces the arbitrariness when determining the dimension of the hidden layer.

There are several approaches to examine the signal content of the MEG measurements. For instance, the frequency spectrum could be investigated thoroughly by converting the signal to the Fourier domain. It would be interesting to compare the frequency content in a MEG signal where no stimuli was applied and the signal after applied stimuli. The differences in the signals could give an indication of what frequencies that are correlated with the MSNA response. Another approach could be to perform a time-frequency analysis of the MEG signal. Such analysis could be used to identify frequencies that are strongly activated when the stimuli is applied, as a function of time. From the analysis, inferences could be made in regard to how the frequency content of the MEG signal changes in response to the stimuli. The information could serve as a decision basis when deciding how to filter the data, since frequencies evoked close to the stimuli most likely are correlated with the MSNA response. Additionally, recall that beta oscillations, with a range of 13 to 30 Hz, have been suggested to be strongly correlated with the MSNA profile. However, the data used in this thesis was band-pass filtered between 0.5 and 40 Hz. Therefore, one approach to improve the signal-to-noise ratio would be to band-pass filter the MEG data between the range of the beta frequencies before applying the autoencoder neural network. By doing so, the measurements would be more compressed, which could potentially make it easier for the autoencoder to identify patterns correlated with the MSNA response.

Upon applying the autoencoder to data from $\widetilde{A}$ and $\widetilde{B}$, it was observed that the training and validation accuracies decreased while the test accuracies remained unchanged. For category X, the classification accuracies for all three data sets decreased significantly. As such, it could be argued that the autoencoder neural network is a superfluous part of the method since none of the classification accuracies were improved. However, it should be noted that even though the training and validation accuracies decreased, they were still high and considerably better than chance. This, together with the unaffected test accuracies for $\widetilde{A}$ and $\widetilde{B}$, indicates that the autoencoder may have learned global features that possibly are correlated with the MSNA response. Additionally, it is reasonable to believe that the decrease in training and validation accuracies for $\widetilde{A}$ and $\widetilde{B}$ depends on the autoencoder's smoothing of individual features. Evaluation on filtered data from category X seem to confirm this, where the smoothed signal made it more difficult for the classification networks to predict from which subject the signal originated. This is an additional indication that brain functional activations are individual and unique and that high-frequency content is an important part for characterization. This also implies that the classification networks may base its predictions mostly on individual features and less on MSNA response.

To summarize, more research is needed to gain a better understanding of the correlation between MSNA and brain functional activation recorded with MEG. If the content of the MEG signal was more accurately known, ANNs could be applied more effectively. Findings in this thesis imply that the highest classification accuracies can be achieved when training on data measured with the peak sensors. As such, it is suggested for future studies to focus on analyzing data recorded with MEG around the Rolandic area. Additionally, the highest binary classification accuracies were obtained from data measured during the second pulse of the stimulation train. The reasons are unknown and would be interesting to investigate further. In the interest of developing a general ANN predictor for evaluating the risk of developing essential hypertension from MEG data, it might be that the MSNA inhibition values from individuals near the mean value are too similar. This would make it challenging for an ANN to accurately divide the population into two groups. Thus, it is suggested to divide the population into percentiles (e.g. no risk, low risk, medium risk etc). This could increase the possibility of accurately classifying individuals exhibiting an "extreme" deviation from the mean on the non-inhibition side of the spectrum, i.e. the individuals with a high risk of developing essential hypertension.

6
Conclusions

Indications were found that individuals exhibiting an "extreme" MSNA response may be identifiable with ANNs. However, it is inconclusive whether the neural networks based their predictions on MSNA response or measurement specific noise. To further study if development of essential hypertension can be predicted with ANNs, it is recommended to divide the population into percentiles, rather than as inhibitors or non-inhibitors.

Several arguments exist for using ANNs as predictors. They have the ability to find complex patterns in data that could have been difficult and time consuming for humans to identify. Additionally, they have the potential to analyze large amount of data in a brief time. If a reliable predictor was to be developed, it would save researchers lots of manual labor.

Parts of the MEG signal that could be correlated with the brain activation response have been identified. Evaluation of the classification networks yielded best accuracies when applied to the second stimulation pulse, measured from sensors covering brain regions with strong beta oscillations. It was also found that high frequency components are important for the characterization of an individuals brain activation response. However, further studies are needed to ensure that the correlation is due to the brain activation response and not measurement noise.

The classification networks were able to accurately classify from which subject a MEG measurement originated from. Thus, it is likely that the brain activation response is dependent on individual traits, e.g. age and gender. However, since all test subjects were males and their exact age unknown, only assumptions can be made. If this is true, clinical implementation would be difficult since lots of diverse data would be needed in order to avoid a biased classifier, which could discriminate towards certain groups.

References

[1] B. R. Syeda, "Bringing MEG towards clinical applications", PhD thesis, Department of Clinical Neuroscience and Rehabilitation, University of Gothenburg, Sahlgrenska Academy, Gothenburg, Sweden, 2018. [Online]. Available: https://gupea.ub.gu.se/handle/2077/56334 (Accessed on: Jan. 24, 2020).

[2] M. Hämäläinen *et al.*, "Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain", *Reviews of Modern Physics*, vol. 65, no. 2, pp. 413–497, Apr. 1993. DOI: 10.1103/RevModPhys.65.413. (Accessed on: Jan. 24, 2020).

[3] R. Elshawi, M. H. Al-Mallah, and S. Sakr, "On the interpretability of machine learning-based model for predicting hypertension", *BMC Medical Informatics and Decision Making*, vol. 19, no. 1, p. 146, Dec. 2019. DOI: 10.1186/s12911-019-0874-0. (Accessed on: Jan. 24, 2020).

[4] G. Buzsáki, "Neural oscillation", in *Encyclopaedia Britannica*, Encyclopaedia Britannica, inc., 2018. [Online]. Available: https://www.britannica.com/science/brain-wave-physiology (Accessed on: May 18, 2020).

[5] A. K. Engel and P. Fries, "Beta-band oscillations—signalling the status quo?", *Current Opinion in Neurobiology*, vol. 20, no. 2, pp. 156–165, Apr. 2010. DOI: 10.1016/j.conb.2010.02.015. (Accessed on: May 15, 2020).

[6] C. R. Noback *et al.*, "Autonomic nervous system", in *The Human Nervous System: Structure and Function*, 6th ed. Totowa, NJ, USA: Humana Press, 2005, ch. 20, pp. 349–369, ISBN: 978-1-59259-730-7. (Accessed on: Feb. 12, 2020).

[7] P. A. Low *et al.*, *Primer on the Autonomic Nervous System*, 3rd ed. ProQuest Ebook Central: Elsevier Science & Technology, 2011, ISBN: 978-0-12-386526-7. (Accessed on: Feb. 10, 2020).

[8] W. Wiltschko *et al.*, "M", in *Encyclopedia of Neuroscience*, Berlin, Heidelberg, Germany: Springer, 2009, pp. 2197–2526. [Online]. Available: https://doi.org/10.1007/978-3-540-29678-2_13 (Accessed on: Feb. 10, 2020).

[9] A. G. Brown, "Introduction to nerve cells and nervous systems", in *Nerve Cells and Nervous Systems: An Introduction to Neuroscience*, 2nd ed. London, United Kingdom: Springer, 2001, ch. 1, pp. 1–14, ISBN: 978-1-4471-0237-3. (Accessed on: Feb. 11, 2020).

[10] H. Lodish *et al.*, "Overview of neuron structure and function", in *Molecular Cell Biology*, 4th ed. New York, NY, USA: W. H. Freeman, 2000, ch. 21.1, pp. 187–221. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK21535/ (Accessed on: Apr. 24, 2020).

[11] P. S. Foresman, Raw image from Pearson Scott Foresman donation, 2008. [Online]. Available: https://commons.wikimedia.org/wiki/File:PSF-D240001.png#/media/File:Dendrites_(PSF).png (Accessed on: Feb. 11, 2020).

[12] G. J. Tortora and B. Derrickson, *Introduction to the human body : the essentials of anatomy and physiology*, 10th ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2015, pp. 230–232, 261–270, ISBN: 978-1-118-58318-0. (Accessed on: Feb. 13, 2020).

[13] K. Young *et al.*, Version 8.25 from the Textbook, 2016. [Online]. Available: https://commons.wikimedia.org/wiki/File:1225_Chemical_Synapse.jpg (Accessed on: Feb. 13, 2020).

[14] P. A. Iaizzo, *Handbook of Cardiac Anatomy, Physiology, and Devices*, 3rd ed. Cham, Switzerland: Springer, 2015, pp. 235–250, 307–320, ISBN: 978-3-319-19464-6. (Accessed on: Feb. 11, 2020).

[15] Free Image on Pixabay - Human, Heart, Pumping, Care, 2020. [Online]. Permission: Pixabay License. [Online]. Available: https://pixabay.com/vectors/human-heart-pumping-care-medicine-41546/ (Accessed on: Feb. 11, 2020).

[16] S. Nadar and G. Y. H. Lip, "Part 1 : Epidemiology, pathogenesis, and diagnosis", in *Hypertension*, 2nd ed. ProQuest Ebook Central: Oxford University Press, Incorporated, 2015, ch. 1, pp. 3–44, ISBN: 978-0-19-100546-6. (Accessed on: Feb. 11, 2020).

[17] V. Donadio *et al.*, "Muscle sympathetic response to arousal predicts neurovascular reactivity during mental stress", *The Journal of Physiology*, vol. 590, no. 12, pp. 2885–2896, May 2012. DOI: 10.1113/jphysiol.2012.228981. (Accessed on: Feb. 12, 2020).

[18] A. Mohorič and J. Stepišnik, "NMR in the earth's magnetic field", *Progress in Nuclear Magnetic Resonance Spectroscopy*, vol. 54, no. 3, pp. 166–182, Apr. 2009. DOI: 10.1016/j.pnmrs.2008.07.002. (Accessed on: Apr. 16, 2020).

[19] J. Schneiderman, Illustration of a MEG helmet, 2020. [Online]. Retrieved via email in February 2, 2020. Modification: Color changes according to Chalmers' thesis guidelines.

[20] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications", *Neural Networks*, vol. 13, no. 4, pp. 411–430, Jun. 2000. DOI: 10.1016/S0893-6080(00)00026-5. (Accessed on: Feb. 14, 2020).

[21] D. Mantini *et al.*, "Improving MEG source localizations: An automated method for complete artifact removal based on independent component analysis", *NeuroImage*, vol. 40, no. 1, pp. 160–173, Mar. 2008. DOI: 10.1016/j.neuroimage.2007.11.022. (Accessed on: Apr. 10, 2020).

[22] S. Taulu, M. Kajola, and J. Simola, "The signal space separation method", *arXiv*, Jan. 2004. [Online]. Available: http://arxiv.org/abs/physics/0401166 (Accessed on: Apr. 16, 2020).

[23] B. Mehlig, *LECTURE NOTES: Artifical Neural Networks*. Gothenborg, Sweden: Department of Physics, University of Gothenburg, 2019. [Online]. Available: http://physics.gu.se/~frtbm/joomla/media/mydocs/ann.pdf (Accessed on: Feb. 12, 2020).

[24] I. A. Basheer and M. Hajmeer, "Artificial neural networks: Fundamentals, computing, design, and application", *Journal of Microbiological Methods*, vol. 43, no. 1, pp. 3–31, Dec. 2000. DOI: 10.1016/S0167-7012(00)00201-3. (Accessed on: Apr. 7, 2020).

[25] Github, How neural networks are trained. [Online]. Available: https://ml4a.github.io/ml4a/how_neural_networks_are_trained/ (Accessed on: Apr. 15, 2020).

[26] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015. [Online]. Available: http://neuralnetworksanddeeplearning.com (Accessed on: Apr. 16, 2020).

[27] N. Srivastava *et al.*, "Dropout: A simple way to prevent neural networks from ovefitting", *The journal of machine learning research*, vol. 15, no. 1, p. 30, Jun. 2014. [Online]. Available: http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf (Accessed on: Apr. 16, 2020).

[28] A. Jain, M. Jianchang, and K. Mohiuddin, "Artificial neural networks: A tutorial", *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996. DOI: 10.1109/2.485891. (Accessed on: Feb. 22, 2020).

[29] Stanford University, Unsupervised Feature Learning and Deep Learning Tutorial: Multi-Layer Neural Network, UFLDL Tutorial. [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/ (Accessed on: Apr. 15, 2020).

[30] D. Stathakis, "How many hidden layers and nodes?", *International Journal of Remote Sensing*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009. DOI: 10.1080/01431160802549278. (Accessed on: Apr. 15, 2020).

[31] Stanford University, Unsupervised Feature Learning and Deep Learning Tutorial: Autoencoders. [Online]. Available: http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/ (Accessed on: Feb. 12, 2020).

[32] I. Goodfellow, Y. Bengio, and A. Courville, "Autoencoders", in *Deep Learning*. MIT Press, 2016, ch. 14, pp. 499–523. [Online]. Available: http://www.deeplearningbook.org (Accessed on: Feb. 13, 2020).

[33] C. Olah, Understanding LSTM Networks – colah's blog. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (Accessed on: Apr. 9, 2020).

[34] J. P. Mejia, The Ultimate Guide to Recurrent Neural Networks (RNN) - Blogs SuperDataScience - Big Data | Analytics Careers | Mentors | Success, 2018. [Online]. Available: https://www.superdatascience.com/blogs/the-ultimate-guide-to-recurrent-neural-networks-rnn (Accessed on: Apr. 9, 2020).

[35] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks", in *Proc. of the 28th international conference on machine learning (ICML-11)*, 2011, p. 8. [Online]. Available: http://www.cs.toronto.edu/~ilya/pubs/2011/LANG-RNN.pdf (Accessed on: Apr. 9, 2020).

[36] R. Challen *et al.*, "Artificial intelligence, bias and clinical safety", *BMJ Quality & Safety*, vol. 28, no. 3, pp. 231–237, Mar. 2019. DOI: 10.1136/bmjqs-2018-008370. (Accessed on: May 12, 2020).

[37] G. D. Tourassi and C. E. Floyd, "The effect of data sampling on the performance evaluation of artificial neural networks in medical diagnosis", *Medical Decision Making*, vol. 17, no. 2, pp. 186–192, Apr. 1997. DOI: 10.1177/0272989X9701700209. (Accessed on: May 12, 2020).

[38] E. Vayena, A. Blasimme, and I. G. Cohen, "Machine learning in medicine: Addressing ethical challenges", *PLoS Medicine*, vol. 15, no. 11, Nov. 2018. DOI: 10.1371/journal.pmed.1002689. (Accessed on: May 12, 2020).

[39] E. Zihni *et al.*, "Opening the black box of artificial intelligence for clinical decision support: A study predicting stroke outcome", *PLOS ONE*, vol. 15, no. 4, e0231166, Apr. 2020. DOI: 10.1371/journal.pone.0231166. (Accessed on: May 12, 2020).

Appendix

A Magnetoencephalography Data Structure

The MEG data used in this thesis is provided by MedTech West and has been recorded at The Swedish National Facility for Magnetoencephalography at Karolinska Institutet in Stockholm, Sweden. [1]. The data was recorded for 20 healthy male participants with an Elekta Neuromag TRIUX system with 306 sensors inside a magnetically shielded room [1]. The sensors are both magnetometers and gradiometers arranged together in 102 locations on the helmet [1]. The stimulus to trigger the MSNA response is an electrical pulse applied on the participant's index finger 200 milliseconds after the cardiac recording has reached its maximum [1]. The stimulus train consists of three stimuli with a delay of two seconds between each [1]. After the offset of the third stimulus, an interstimulus interval between 30 to 60 seconds is applied [1]. In total there are 72 stimuli for each participant [1].

After the acquisition, the data is processed [1]. First, the recordings are band-pass filtered between 0.5-40 Hz [1]. Next, a manual inspection of the data is performed to remove artifacts with ICA and bad segments [1]. Elekta's own software is used to compensate for head movements with head position indicator coils and the SSS method [1]. The data is then divided into epochs of four seconds each, centered around the stimulus [1]. Figure A.1 visualizes a portion of the MEG data from Medtech West after acquisition and processing.

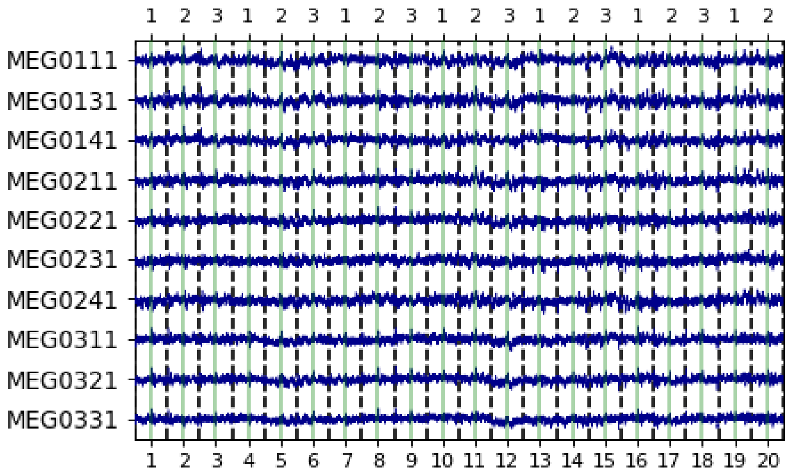


Figure A.1: Visualization of the recorded magnetic fields for ten sensors during the first 20 epochs. The top axis indicates where the the stimulus of each pulse starts. The bottom axis marks the epochs in chronological order. Note that the stimulus is applied in the middle of each epoch.

The recordings are displayed with the publicly available MNE software package. The information in Figure A.1 are the magnetic fields for one subject during the first 20 epochs of recording. The black-dashed lines indicates the end and the beginning of the epochs and each epoch is centered around the stimulus. 10 out of the 306 available sensors are selected with their sensor label on the y-axis and the stimuli marked with green, vertical lines labeled in the top axis. The visualization in Figure A.1 provides an overview of the structure of the data, but does not give a clear view of how the magnetic field varies during the recording. To inspect the variation in-depth, the MNE software package provides a function to select specific sensors. When applying this function on the sensor *MEG0111* a detailed view of the measured fields is displayed which are depicted in Figure A.2.

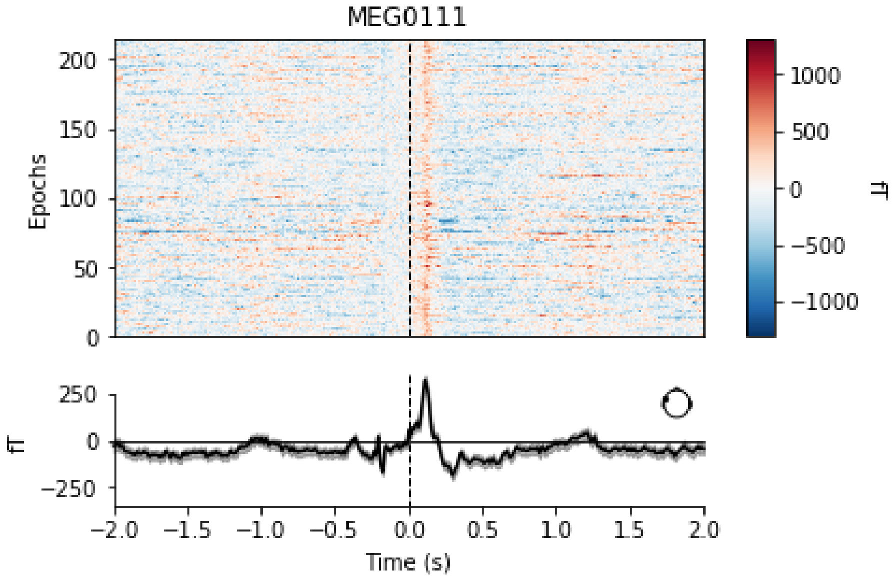


Figure A.2: In-depth visualization of the recorded magnetic field in sensor MEG0111. The top diagram shows the strength of the magnetic field for all epochs. The bottom diagram displays the magnetic field for all epochs superposed. The position of sensor MEG0111 is marked in the head to the upper right in the lower diagram.

The top diagram shows the magnetic field for all epochs during each time step of the recordings for the participant. The black-dashed vertical line marks the stimulus and the brain's response appears as an increase in the magnetic field after a few milliseconds. The bottom diagram displays the sensor's magnetic fields for all epochs superposed with the solid, black line representing the average field. In addition, the MNE provides a head object marked with the sensor location which can be used to draw conclusions about what brain regions caused the magnetic activity.

In addition to the recorded MEG data, two lists have been delivered from Medtech West. The first list contains the participants' MSNA inhibition value. A positive inhibition value reflects a response which inhibits the MSNA increment and a larger value means a better inhibition ability [1]. A negative inhibition value represents a response which does not inhibit the MSNA increment and a large negative value means a poorer inhibition capability [1]. Inhibitors and non-inhibitors are defined as persons with measured MSNA inhibition values above respective below 30 % [1]. In the list for the 20 participants, 50 % were inhibitors and 50 % were non-inhibitors [1].

The second list contains MEG sensors positioned close to brain regions where the change in beta power after an applied stimulus, i.e. the beta rebound, is anticipated to be strong. Identification of the sensors was performed with visual inspection in the MNE software environment and the list consists of two sensor groups: decent and peak. The decent sensors cover a wider area and are relatively strongly activated and the peak sensors cover a smaller area and are the most strongly activated. The approximate sensor location are illustrated in Figure A.3. The information presented about the second list was obtained through an email conversation with a researcher at MedTech West and is accounted for in Appendix C.

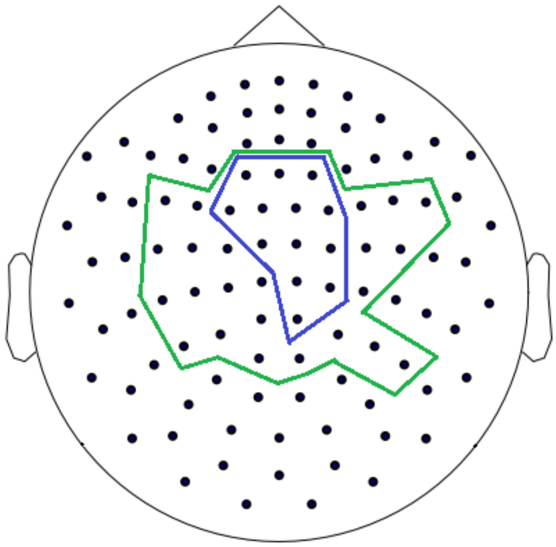


Figure A.3: Illustration of the areas covered by the decent sensors (green) and peak sensors (blue). Note that the sensor locations are approximate and might differ for each test subject.

B Magnetoencephalography Data Details

The metadata of the test subjects who participated in MedTech West's MEG recordings is listed in Table B.1 where each individual is identified by the patient ID. The MSNA inhibition value is a measurement of the person's ability to inhibit the increased MSNA activity when the index finger has been stimulated by an electric pulse. In addition to the MSNA inhibition value, the number of epochs for each pulse stimulation is presented together with the total amount of active sensors during the MEG recording.

Table B.1: Information about the MEG data retrieved from MedTech West.

Patient ID	**MSNA Inhibition Value** [%]	**Number of Epochs (Pulse 1)**	**Number of Epochs (Pulse 2)**	**Number of Epochs (Pulse 3)**	**Number of Sensors**
0075	-1.23	21 513	21 816	21 816	303
0076	75,57	22 032	22 032	21 726	306
0102	42,69	21 726	21 420	21 420	306
0137	11,78	21 681	21 384	21 384	297
0140	1,5	20 930	20 631	20 332	299
0158	85	21 888	21 584	21 584	304
0178	11,37	21 816	21 816	21 816	303
0179	7,9	21 442	21 140	20 536	302
0263	53,87	22 032	22 032	22 302	306
0266	44,51	22 032	22 032	22 032	306
0267	48,2	21 726	22 032	21 726	306
0283	58,64	22 032	22 032	22 032	306
0284	16,45	22 032	22 338	21 726	306
0293	30,01	21 960	21 960	21 655	305
0294	9,37	21 672	21 672	21 672	301
0295	-21,32	22 032	22 032	22 032	306
0299	83,74	21 960	21 960	21 960	305
0300	13,83	21 655	21 960	21 960	305
0301	-132,47	21 513	21 210	21 210	303

C Communication with Neuroscientist

An email conversation from May 12, 2020, with Justin F. Schneiderman, senior lecturer at the University of Gothenburg and MEG researcher at MedTech West, is presented below. The email was sent with questions regarding the beta wave and how the neural oscillation was related to the recordings in the provided data.

1. **Q** – The beta wave is generated when the brain is aroused and actively engaged in mental activities. However, is beta rebound short for the post-motor beta rebound (PMBR)? If no, is there a paper with a good explanation? When we search after information, we only retrieve PMBR explanations...

A – Not quite - the power in beta increases during arousal and active engagement... but the absolute power of beta varies all the time (as with almost all neural oscillations). That is why we always plot relative changes in beta power (dB or similar), normalized to the baseline (which we specify as being the pre-stimulus interval). This doesn't guarantee that beta changed in power because of non-stimulus related events (mental processes or other), but it's generally how things are done. And what you mean by 'short' or 'long' is unclear... and perhaps not terribly relevant, either. I suggest you limit the explanation for what beta 'means' and simply indicate something along the lines of what we did in Bushra's manuscript: it's modulated by mental activity, motor control, somatosensory input, etc, and then describe 'beta rebound' as being what we use to define the increase in beta power that we see (after the short-lived decrease) after the stimulus.

2. **Q** – We revisited Bushra's PhD thesis and she wrote that there was a strong correlation between the Rolandic beta rebound and the MSNA and therefore it could serve as a biomarker for the MSNA response profile. Would it be incorrect to say that the Rolandic region is a part of the motor cortex? (we mostly find articles about Rolandic epilepsy rather than an explanation about the region itself). For the peak and decent sensors, are they located around this region?

A – The Rolandic fissure (aka Central sulcus) is what separates the frontal and parietal lobes of the brain. The Rolandic area then includes the fissure and the 2 sulci around it–in front of it is the motor cortex, and behind it is the somatosensory cortex. In other words, the motor cortex is a part of the Rolandic area. The reason we used such a 'large' area is because we expected activity in both motor and somatosensory areas in response to the stimulus. And yes, the 'peak' and 'decent' sensors are roughly located above/around the Rolandic areas... that is to say, I picked them because they are most sensitive to activations from the Rolandic area, but that is not to say that they did not pick up signals from other parts of the brain.

3. **Q** – The last question is about the recorded magnetic field for all sensors. Is the measured fields in our data recorded only from the beta wave or from all waves combined? If the field recorded in all sensors are all waveforms combined, did you use some special MNE function to only see the beta rebound for the sensors listed

in the decent and peak list? If the field is strictly from the beta wave, we assume the decent and peak lists includes sensors where the activity is increased?

A – The field we record is a sum of all oscillations and evoked activations (and whatever else is going on in the brain). We used spectral decomposition (time-frequency plots) to help us visualize how oscillatory activity changed (in power) wrt to the stimulus–that is where we first saw the 'interesting' beta rebound that we later found correlated so well with MSNA. The figures we present in Bushra's paper are mostly in the 'source space': the data from ALL sensors is combined to estimate the activations generated in different parts of the brain. As I said above, the decent and peak sensors are the ones whose signals are most likely to be dominated by activity in the Rolandic area. They should therefore have sampled the beta rebound generated in the Rolandic area well, but they still pick up other signals (both in other frequency bands, but also from other brain regions).

D A Note on the LSTM Accuracy

As stated in Section 3.5, the ANNs were optimized to be used for binary classification. Only necessary modifications were made to the networks when they were used for multiclass classification. This worked well for the MLP, which was able to reach high classification accuracies without any further modifications. However, the same was not true for the LSTM network. To test whether the LSTM classification accuracy could be improved, the number of neurons in the two hidden layers were increased. The highest validation accuracy was reached with 800 neurons, the same number as was used for the MLP. The training, validation and test accuracies obtained with the updated LSTM architecture are presented in Figure D.1. Note that the test accuracies are much higher than the ones achieved using 260 neurons in each hidden layer, Figure 4.6. It is worth noting that the same modification did not increase the validation or test accuracies on categories A and B (or $\widetilde{A}$ and $\widetilde{B}$). It is also worth noting that even though the accuracies evaluated on data from category X could be significantly improved, the MLP still achieved higher accuracies for all classifications that were performed in this thesis.

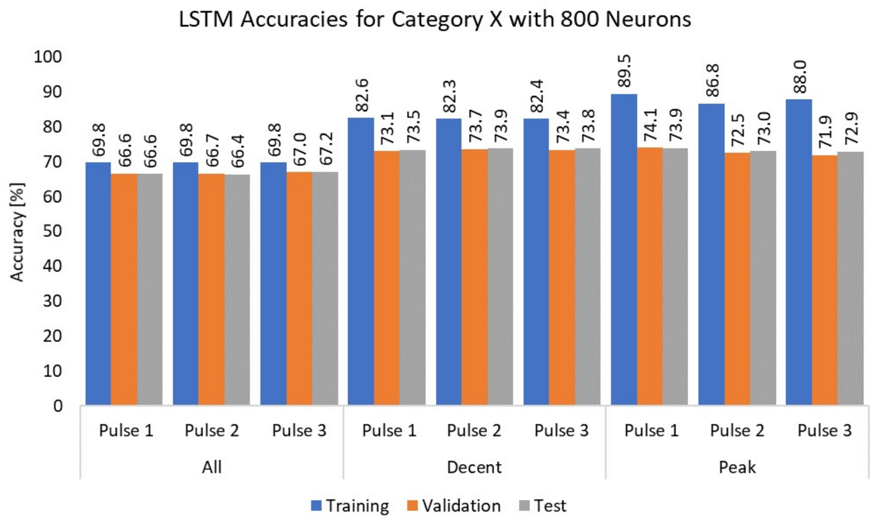
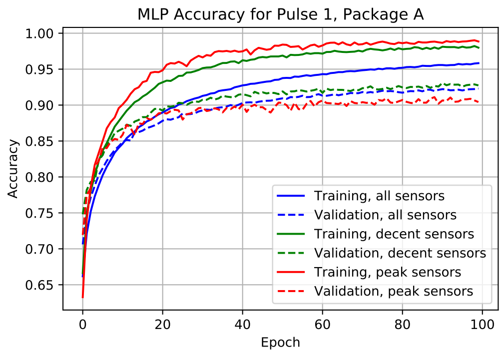


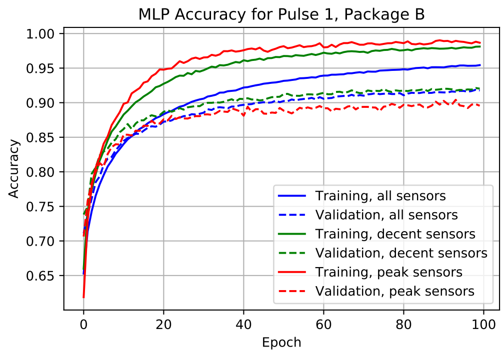
Figure D.1: Training, validation and test accuracies for the LSTM network after 20 epochs of training on category X, with 800 neurons in the hidden layers. The example serves to show that the multiclass LSTM accuracy can be significantly improved by tuning hyperparameters.

E Complementary Results

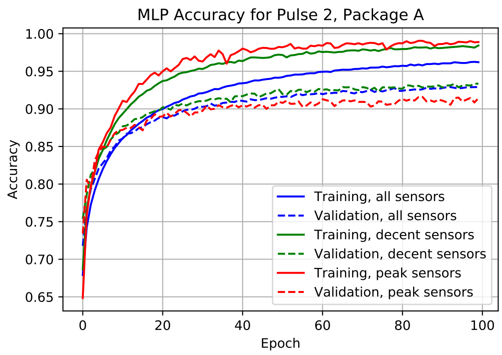
As mentioned in Section 3.6, the training and validation accuracies for the respective data packages were plotted as a function of epoch. However, only the accuracies after the final epoch of training are of importance for the aim of this thesis. As such, only these results were presented in the main report. For the interested reader, the accuracies as a function of epoch are presented below. Observe that, unfortunately, there is a typo in all plot titles. The word "Package" should be exchanged for "Category" to be consistent with the rest of the thesis.



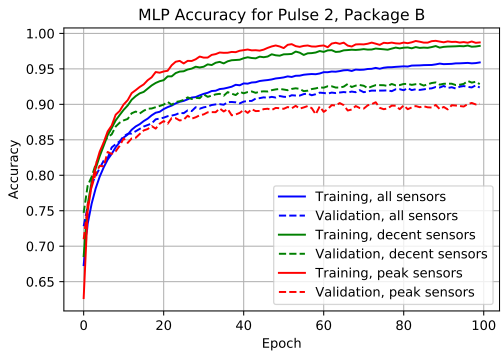
(a)



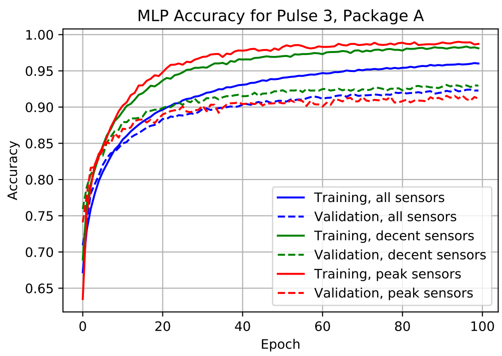
(d)



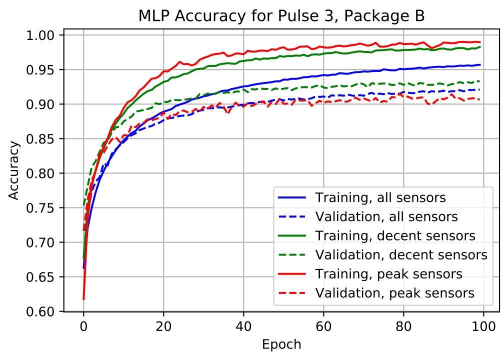
(b)



(e)

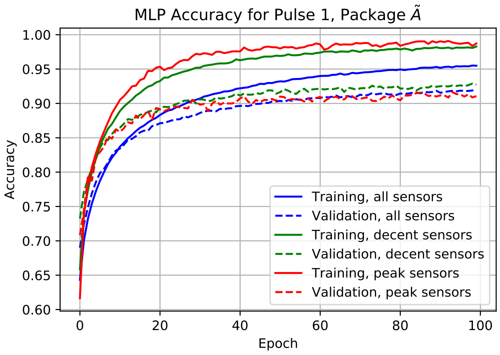


(c)

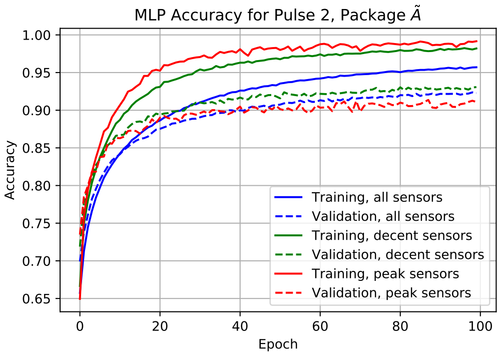


(f)

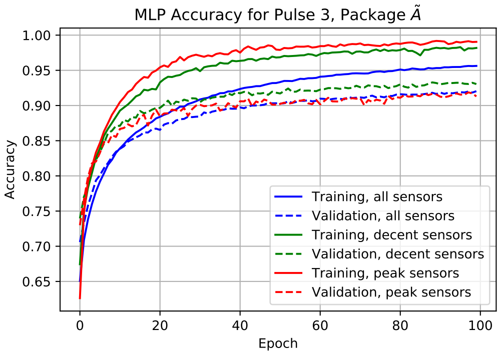
Figure E.1: Training and validation accuracies as a function of epoch for pulse 1, 2 and 3 for (a-c) category A and (d-f) category B with the MLP network. The network was trained using the hyperparameters in Table 4.1.



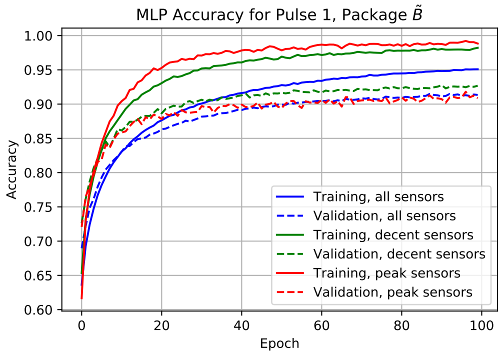
(a)



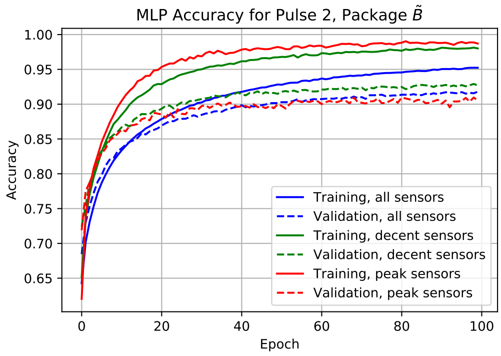
(b)



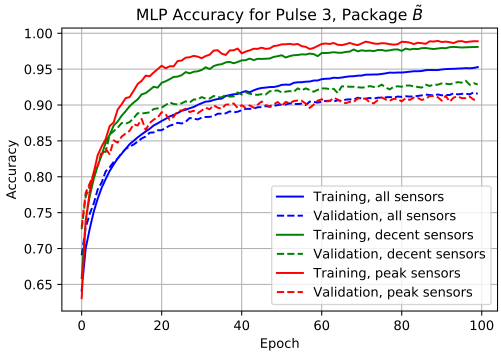
(c)



(d)

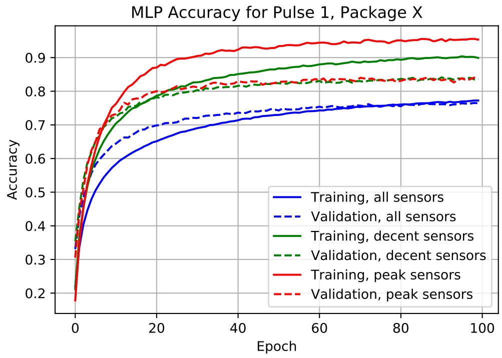


(e)

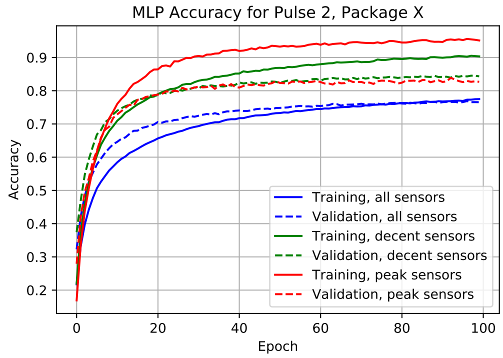


(f)

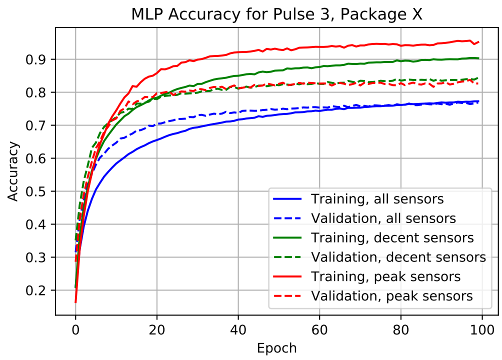
Figure E.2: Training and validation accuracies as a function of epoch for pulse 1, 2 and 3 for (a-c) category $\tilde{A}$ and (d-f) category $\tilde{B}$ with the MLP network. The network was trained using the hyperparameters in Table 4.1.



(a)

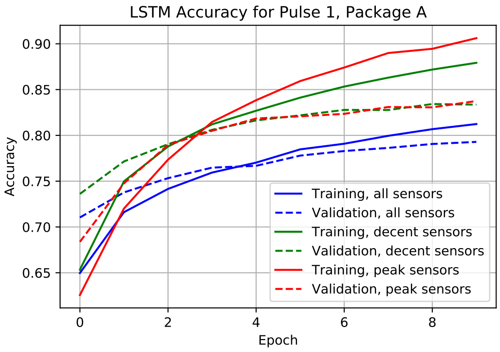


(b)

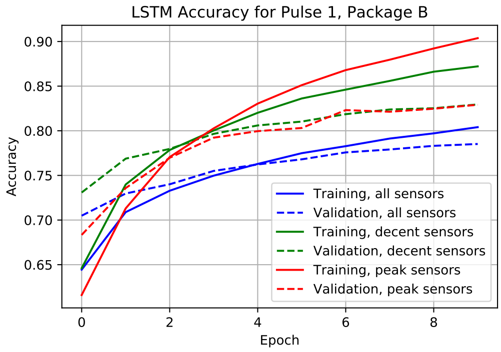


(c)

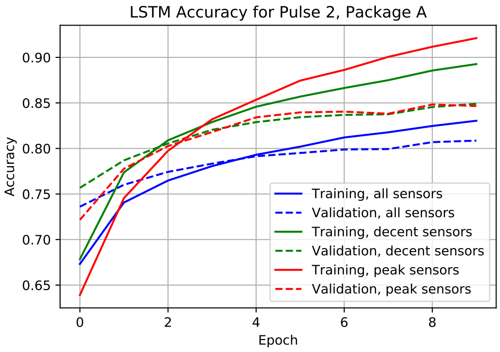
Figure E.3: Training and validation accuracies as a function of epoch for (a) pulse 1, (b) pulse 2 and (c) pulse 3 for category X with the MLP network. The network was trained using the hyperparameters in Table 4.1, with some modifications to allow for multiclass classification, see Section 3.6.



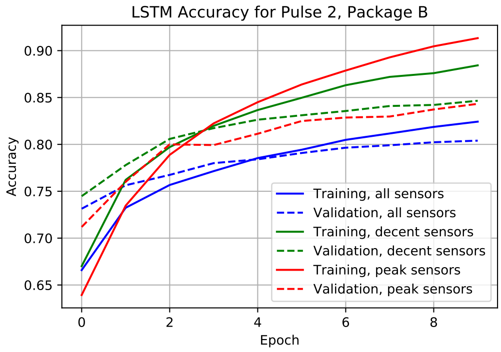
(a)



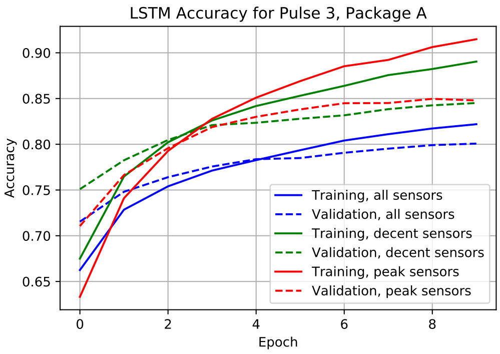
(d)



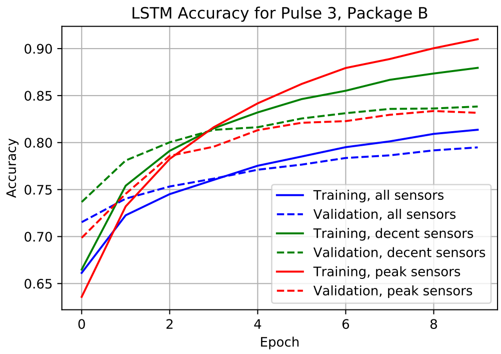
(b)



(e)

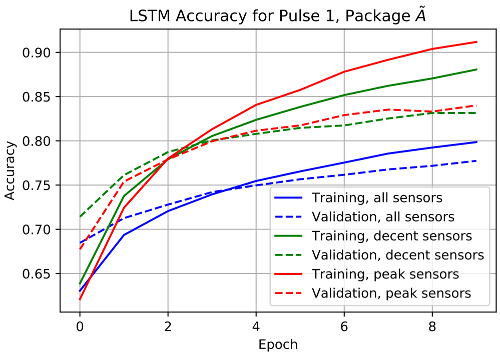


(c)

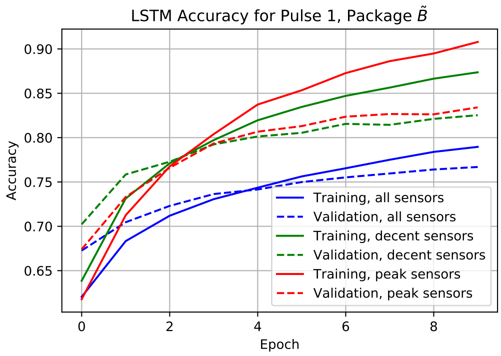


(f)

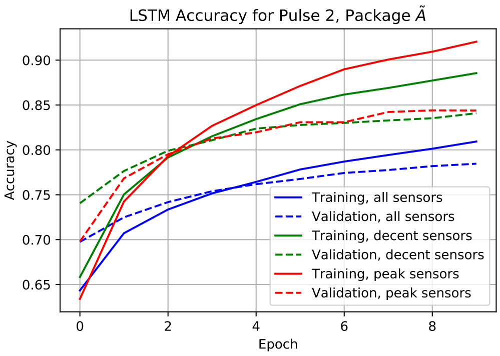
Figure E.4: Training and validation accuracies as a function of epoch for pulse 1, 2 and 3 for (a-c) category A and (d-f) category B with the LSTM network. The network was trained using the hyperparameters in Table 4.2.



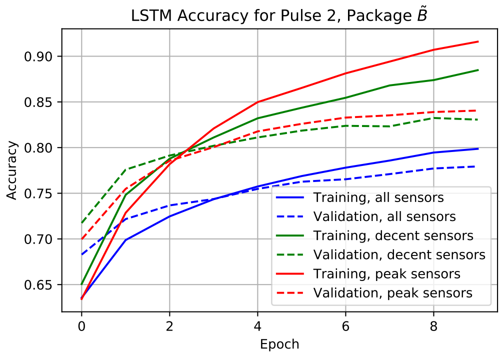
(a)



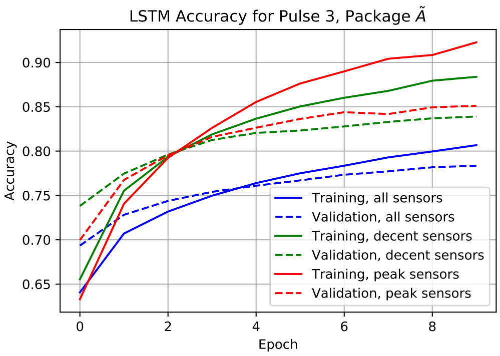
(d)



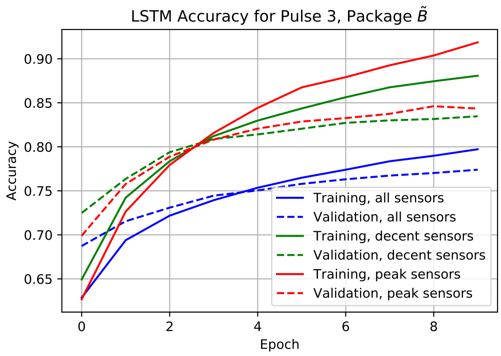
(b)



(e)

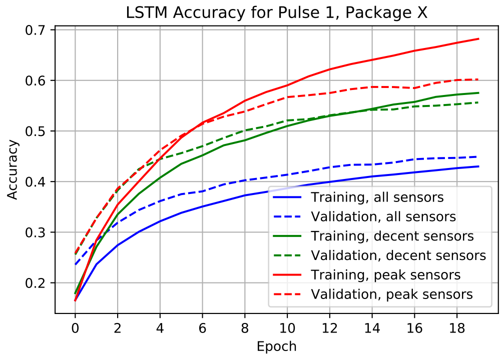


(c)

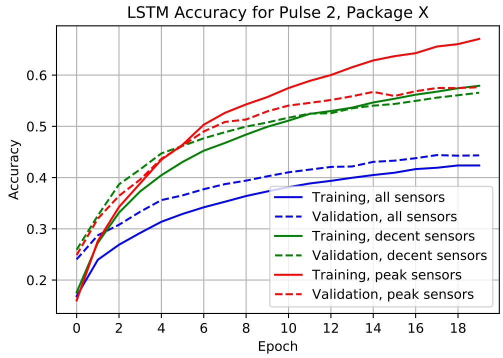


(f)

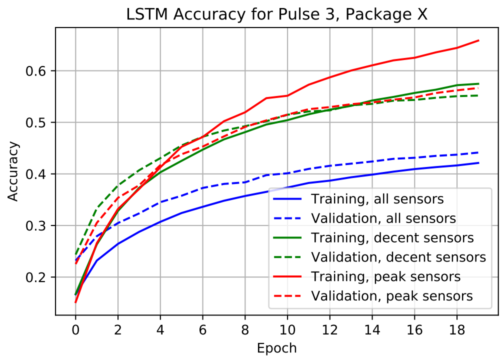
Figure E.5: Training and validation accuracies as a function of epoch for pulse 1, 2 and 3 for (a-c) category $\tilde{A}$ and (d-f) category $\tilde{B}$ with the LSTM network. The network was trained using the hyperparameters in Table 4.2.



(a)

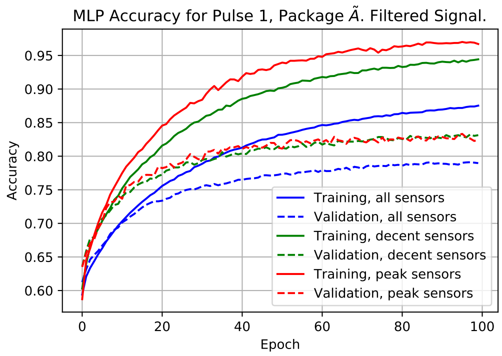


(b)

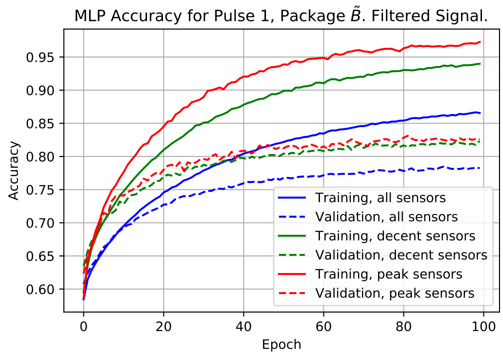


(c)

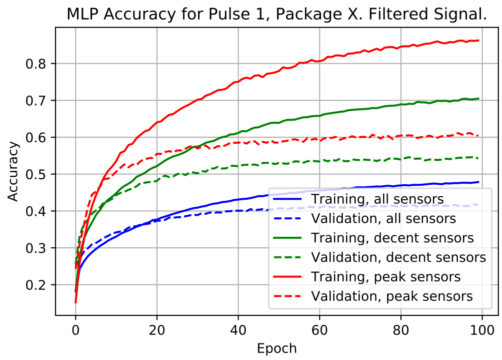
Figure E.6: Training and validation accuracies as a function of epoch for (a) pulse 1, (b) pulse 2 and (c) pulse 3 for category X with the LSTM network. The network was trained using the hyperparameters in Table 4.2, with some modifications to allow for multiclass classification, see Section 3.6. Additionally, since no apparent overfitting was observed, 20 epochs were used.



(a)

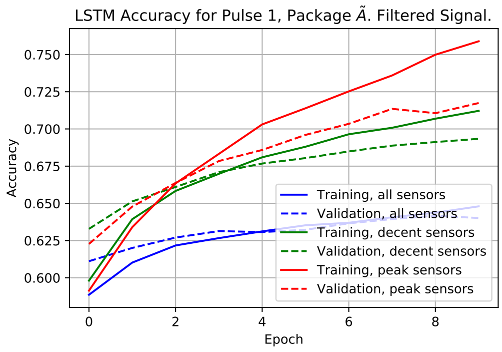


(b)

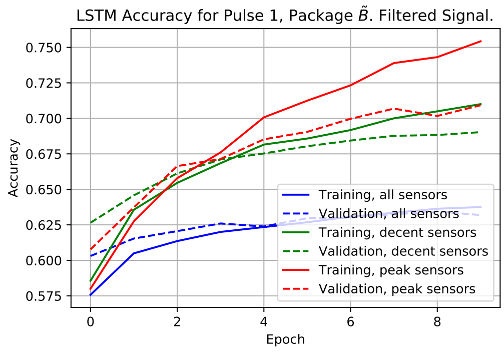


(c)

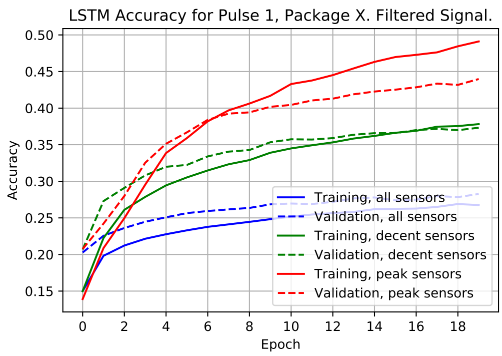
Figure E.7: Training and validation accuracies as a function of epoch after the signal has been filtered with the autoencoder. (a) Category $\tilde{A}$, (b) category $\tilde{B}$ and (c) category X for pulse 1 with the MLP network. The network was trained using the hyperparameters in Table 4.1, with some modifications made for category X to allow for multiclass classification, see Section 3.6.



(a)



(b)



(c)

Figure E.8: Training and validation accuracies as a function of epoch after the signal has been filtered with the autoencoder. (a) Category $\tilde{A}$, (b) category $\tilde{B}$ and (c) category X for pulse 1 with the LSTM network. The network was trained using the hyperparameters in Table 4.2, with some modifications made for category X to allow for multiclass classification, see Section 3.6. Additionally, since no apparent overfitting was observed, 20 epochs were used for category X.