CLASSIFICATION AND ANALYSIS OF A LARGE MEG DATASET USING CONVOLUTIONAL NEURAL NETWORKS

by

Jon Garry

Submitted in partial fulfillment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
July 2019

In loving memory of Grandma and Papa. You both helped spur my interest in the arts and sciences from a young age.

Table of Contents

List of Tables

List of Figures

Abstract

Convolutional neural networks were used to classify and analyse a large magnetoencephalography (MEG) dataset. Networks were trained to classify between active and baseline intervals recorded during cued button pressing. There were two primary objectives for this study: (1) develop networks that can effectively classify MEG data, and (2) identify the important data features that inform classification. Networks with a simple architecture were trained using sensor and source-localised data. Networks trained with sensor data were also trained using varying amounts of data. The important features within the data were identified by applying different visualisation techniques to trained networks. An ensemble of networks trained using sensor data performed best (average test accuracy 0.974 ± 0.001). It was determined that a dataset containing on the order of hundreds of participants was required for this particular network and task. Visualisation maps highlighted features known to occur during neuromagnetic recordings of cued button pressing.

List of Abbreviations

AEF Auditory Evoked Field.

ANN Artificial Neural Networks.

Cam-CAN Cambridge Centre for Ageing and Neuroscience.

CNN Convolutional Neural Networks.

CT Computed Tomography.

dSPM Dynamical Statistical Parametric Mapping.

ECG Electrocardiogram.

EOG Electrooculogram.

GPU Graphics Processing Unit.

ICA Independent Component Analysis.

MEG Magnetoencephalography.

MRI Magnetic Resonance Imaging.

PSP Post Synaptic Potential.

ReLU Rectified Linear Units.

tSSS Temporal Signal Space Separation.

US Ultrasound.

VEF Visual Evoked Field.

Acknowledgements

I would like to thank my supervisor, Tim Bardouille for all his help, guidance, and general advice throughout the course of my program. Also, thank you for the opportunity to be a part of such a talented and interesting group, I learned so much in such a short period of time.

I would also like to thank my committee members: Thomas Trappenberg and Steven Beyea for bringing their unique perspectives and expertise to this project. Your feedback throughout brought focus to the project and provoked interesting ideas for exploration.

To my friends and family, I want to thank you all for your support. Will, thank you for all the time we spent letting off steam and for the discussions of our shared interests (and shared suffering) over the years. Sarah, your constant support and words of encouragement were essential to my success. Thank you for everything you do.

Last, but certainly not least, I want to thank my parents. I am grateful to you both for all of the support and encouragement you have given me over the years and for your continued support throughout this adventure in continuing education.

Chapter 1

Introduction

Big data in medical imaging has the potential to inform new models in diagnostics. In recent years there has been an exponential increase in the amount of data in many fields, including health care [1]. With appropriately large datasets, approaches found in the field of deep learning can be applied in order to augment and improve currently established methods for diagnosing patients.

Machine learning systems learn patterns that have been extracted from raw data in order to perform some task without the need for explicit programming or instruction sets [2]. However, the performance of these systems heavily relies upon the representation of the input data. Traditional machine learning approaches typically require specialised feature engineering in order to develop an appropriate data representation. The construction of an effective data representation typically requires expert knowledge of the problem domain and results can be highly sensitive to changes within the representation. One solution to feature engineering is to use machine learning to discover the appropriate representation as well as the functional mapping between the representation and the output. Whereas traditional machine learning uses models that act as discriminators in classifying or clustering data, deep learning models generate latent representations of input data. This is referred to as representation learning and it is central to deep learning [3].

Models in deep learning are constructed using artificial neural networks (ANN). ANNs were inspired by early models of brain functioning with the base unit of information being the neuron [3]. A neuron accepts a set of inputs which are multiplied by an associated weight and summed together. This summation is passed through a non-linear transformation known as an activation function before being outputted. These outputs are then passed along to other neurons or as output of the network. By assembling networks of neurons, and by tuning the set of weights within the network via optimisation, an ANN can be thought of as a function approximator that

constructs a functional mapping between some set of inputs and a set of outputs [3]. The layers of neurons within the network are called hidden layers in that they are hidden between the input and output layers. By varying the number of hidden layers, and the number of neurons within each layer, the capacity of a model can be tuned to theoretically approximate any function [3].

A convolutional neural network (CNN) is a type of ANN that was designed for processing images and video. First introduced by LeCun [4] and popularised by Krizhevsky et al. [5], a CNN contains layers that perform the mathematical operation of the convolution on inputs passed to them. Input passed to these layers is convolved with small matrices known as kernels in order to extract specific types of information. When many convolutional layers are stacked together, a nested hierarchy is formed with some layers being responsible for detecting different types of structure within the data [3]. For instance, one layer may detect lines with a specific orientation whereas another layer may detect textures. The combined function of these layers would be to detect a specific type of object within input images. The outputs from the convolutional layers are input into dense neuron layers as found in a standard neural network in order to construct functional relationships. A CNN therefore performs a combination of efficient feature extraction (via weight sharing) and functional approximation using data with topographic structure.

Within the field of medical imaging, recent research has focused on developing networks that perform classification, detection, and segmentation using medical images [1, 6]. These networks are currently being developed for applications such as the detection and segmentation of tumours, the classification of lung nodules into benign and malignant, and classification of heart disease in cardiac imaging [1, 7]. The networks within these studies are designed to work with data from computed tomography (CT), ultrasound (US), and magnetic resonance imaging (MRI) [1, 6, 7, 8]. One imaging modality that has not been used within these types of studies is magnetoencephalography (MEG).

MEG sensitively measures magnetic fields generated by neuronal activity within the brain with a good combination of localisation accuracy and temporal resolution [9]. Medical applications of MEG include classification of patients with multiple

sclerosis [10, 11], Alzheimer's disease [12], and the detection and localisation of pathological activity in patients with epilepsy [13, 14, 15]. MEG has been shown to be an effective tool in localising eloquent cortex in order to guide pre-surgical planning in patients with brain tumours and intractable epilepsy [14, 15]. Given these use-cases, models developed using deep learning have the potential to improve diagnostics and outcomes for patients. As an initial step in this direction, we are interested in developing models of healthy brain functioning constructed using large collections of normative data which could help to delineate between healthy and unhealthy brain activity.

The Cambridge Centre for Ageing and Neuroscience (Cam-CAN) is a large, collaborative research project aimed at investigating the effects of ageing and cognition. As a part of this project, Cam-CAN provides a large open-access dataset that includes demographic, behavioural, and neuroimaging data recorded across a large, healthy adult population [16]. Of specific interest for our research was a set of MEG scans from 700 participants recorded during a cued button pressing task (with auditory and visual cues). We were interested in investigating this data because it contains records from healthy controls with a simple, well-understood task. Thus, we could determine if the data representations learned by the CNN match expectations based on previous literature. These data provided a collection of normative data measured during an understood and well-established activity across a diverse population.

The MEG correlates of cued-button pressing have been well-studied. Auditory and visual cues both generate a reproducible complex of magnetic field deflections over approximately 300 ms following the cue. The deflections, termed the auditory and visual evoked fields (AEF and VEF respectively) [17], are clearly shown by averaging MEG data recorded from a number of repetitions of stimulus presentation. The VEF and AEF are observed bilaterally on sensors over the occipital and temporal cortex, respectively. As well, bursts of magnetic field deflections centred on 10 Hz are observed on occipital sensors when no stimulus is occurring. These "alpha" bursts are suppressed for up to one second when a visual stimulus occurs [18].

The act of button pressing also generates a reproducible complex of magnetic field deflections that are most clearly revealed by averaging MEG data over a number of

repetitions of button pressings. This evoked field includes a pre-movement component observed on sensors over the contralateral primary motor cortex (M1), known as the readiness field, and a combination of components that are localised to M1 and primary somatosensory cortex (S1) and the broader sensorimotor network, in the approximately 300 ms after movement [19]. As well, bursts of magnetic field deflections centred on 10 Hz and 20 Hz are observed on bilateral sensors over M1/S1 in the absence of stimulus or movement. These "mu" (∼8-12 Hz) and "beta" (∼15-30 Hz) bursts are suppressed for about 750 ms following a button press [18, 20]. At 750 ms to 2000 ms, beta bursts are more common than in the pre-movement interval, such that more beta band activity is measured in this interval, termed the beta rebound.

When a simple button press is cued by auditory and visual stimulus, we expect that all of these responses will sum, with very little change to the spatiotemporal dynamics. The only exception is that the pre-movement readiness field should occur over a substantially shorter period, since it does not begin until the cue is received. Furthermore, there are likely additional responses involved in integrating the processes associated with the cue and required response. Responses associated with these higher-level cognitive processes are not considered in this study.

Whereas the data used in this study contain activity from a well known task, the intention of this research is to pave the way for the development of networks that can identify pathological activity within a clinical setting. Eventually, we want to be able to not only identify pathological activity, but also provide a means of identifying the regions responsible for it. Ideally, trained networks would provide high quality classification and visualisation techniques would allow for localisation.

In this paper we present a CNN designed to classify between MEG measurements recorded during active and baseline intervals. Given successes in other medical imaging studies, we believe that a CNN is an appropriate choice of network for this task and should therefore produce high quality classification results when classifying sensor measurement and source-estimated records. Although MEG measurements typically have low SNR, there is underlying structure that is consistent across sets of channels and times which a CNN can learn. With enough training data, we hypothesise that a CNN will learn to effectively classify MEG records by extracting this underlying structure. We examine and compare the performance of networks that have been trained

using minimally processed sensor records and source-localised data. We investigate network performance in terms of dataset size by varying the number of participants included in training. We also present visualisation and attribution methods in order to reveal the features that these networks extract. We hypothesise that these features will relate back to what is previously known about the task of cued button pressing. Specifically, we hypothesise that the visualisation of the CNN will reveal a sensitivity to occipital and temporal activity (due to the cue) and motor-related activity (due to the button press).

Chapter 2

Methods

2.1 Participants & Paradigm

The datasets used with the networks in this project were derived from magnetoencephalography (MEG) measurements recorded during the second, or core cognitive neuroscience stage, of the Cam-CAN study [16]. This stage included 700 individuals with a distribution of 100 participants within each 10-year age bracket (18-87 years of age). Participants completed a series of sessions in which structural and functional magnetic resonance imaging (MRI), along with the MEG measurements were collected.

MEG data were acquired from 306 channels (102 magnetometers and 204 planar gradiometers) at a sampling rate of 1000 Hz. Inline band-pass filtering between 0.03 and 330 Hz was applied using a 306-channel Vectorview system (Elekta Neuromag, Helsinki, Finland). Digitisation of anatomical landmarks (i.e., fiducial points; nasion and left/right preauricular points) as well as additional points on the scalp was also performed for registration of MEG and MRI coordinate systems. Head position was monitored continuously, and electrooculogram (EOG) and electrocardiogram (ECG) were recorded concurrently along with stimulus/response event markers. T1-weighted magnetic resonance images (MRI) were acquired using the 3T Siemens Tim Trio system with a 32-channel head coil.

Participants performed the task of pressing a button using their right index finger after the presentation of visual, auditory, or combined audio-visual stimuli. The onset time of these stimuli was randomised between two and 26 seconds within each trial in order to prevent anticipatory effects. This temporal separation between button presses also meant that most magnetic field deflections due to the previous button press would have completed with sufficient time before the next cue occurred. This ensured that the MEG data of the current trial was not contaminated by the activity recorded within the preceding trial.

For the purposes of the binary classification scheme for this project, one-second MEG data segments centred on the button press within each trial served as a record belonging to the "Active" class. Participant response times were 300 ms on average (across all participants). Thus, the active records included both cue- and motor-related activity. One-second MEG data segments ending 700 ms prior to the button press were extracted from each trial and provided the "Baseline" class records.

2.2 Data Preparation

2.2.1 Data Pre-processing

Prior to constructing datasets of sensor-level and source-estimated records appropriate for use with a CNN, there were two sets of pre-processing pipelines applied to the data. One set of pre-processing tasks was performed by the Cam-CAN group prior to the public release of the data [21]. The other pre-processing was applied by our lab [22].

The pre-processing performed by the Cam-CAN group included the application of temporal signal space separation (tSSS) [23]. This technique was used to remove noise introduced by external EM sources, perform head movement corrections, and provide a virtual data transformation to a common head position for each dataset. The tSSS process also allowed for the reconstruction of missing or corrupted MEG channels. The virtual data transformations were especially important for our project because they allowed for a straightforward aggregation of data across participants, giving consistent sensor location with respect to the head across all participants.

The set of pre-processing tasks performed in-house by our group included splitting up each participant's dataset into individual trials and applying independent component analysis (ICA) [24]. Using the event markers within the MEG data, the time series for each participant was separated into trials and synchronised to the time of each button press. Each trial contained a total of 3.4 seconds of measurements with a pre-stimulus (baseline) period of 1.7 seconds in duration. Trials were excluded if poor task performance occurred (button press occurred more than one second after cue) or if the button press occurred within three seconds of the previous button press. This second exclusion criteria ensured that all records derived from the pre-stimulus

interval, which provided records for the baseline class, were not contaminated by field deflections due to task performance. In particular, this criterion avoided the inclusion in the baseline of a prolonged increase in the magnitude of the centrally-generated beta rhythm in the roughly three seconds following a movement, termed the post-movement beta rebound.

The automated FastICA routine was used to remove artefacts by decomposing the data into individual components [25, 26]. Those data components with amplitudes and phases similar to those of the electrooculogram and electrocardiogram signals were removed [27]. Trials with signal amplitudes that exceeded 5 pT (magnetometers) or 400 pT/cm (gradiometers) were also excluded. The remaining components were reconstructed to form the artefact-removed datasets. For each participant, these processes resulted in MEG trial data as a tensor with dimensions [# Trials, # Sensors, # Time Samples].

MEG data were averaged across all trials and participants to generate the grand-average evoked field data for cued button pressing. These data were visualised to reveal the average magnetic field deflections associated with the task. It was expected that these field deflections would match the features based upon past literature.

2.2.2 Sensor Record Processing

A set of processing tasks were performed in order to prepare the processed MEG sensor data for use with a CNN. Specifically, individual trials were split up into class records and the dimensionality of these records were reduced in order to reduce computational complexity of the networks used.

The number of dimensions were reduced by first choosing only the magnetometer channels thereby reducing the number of sensors in each dataset from 306 to 102. Magnetometers were chosen instead of planar gradiometers for ease of interpretation of the resultant field topographies. Since we expected the activity of interest to be within a relatively low frequency range, a low-pass filter of 40 Hz was applied to all trials. Each trial was also down-sampled to one quarter of the number of time samples (i.e. 250 Hz).

With all of the processing applied, each trial was split up into baseline and active intervals. The baseline interval included data acquired between 1.7 and 0.7 seconds

prior to the button press and the active interval was taken from between -0.5 and 0.5 seconds around the button press. Thus, each participant trial provided two classified records: one record labelled as baseline in the case of the data extracted from the interval prior to button press, and one active class record from the interval around the button press. All records were scaled to unit normal by subtracting the mean and dividing by the standard deviation across each sensor.

After all processing tasks were complete, there were 75,396 records across 605 participants with an even distribution between the active and baseline classes. Each record was essentially represented as an image with each row representing an MEG sensor and each column representing a time sample. The dimensions of each record contained 102 magnetometer channels and 250 time samples.

2.2.3 Source Localised Record Processing

Experimentation with classifying MEG data was also performed on representations of the sensor data transformed into source space. (i.e., estimated as current flow at specific locations in the brain, rather than magnetic field deflection outside of the head). Estimated data in source space has the advantage of providing more spatially-specific data, although there is the potential for inaccuracy in the source data due to the estimation process. Source localisation was performed using the same cleaned data that was used to process the sensor records. Similar to the sensor record processing described above, a 40 Hz low-pass filter was applied to the data and each trial was down-sampled to 250 Hz. Unlike the sensor record processing, all sensors (magnetometers and gradiometers) were kept in order to provide more accurate source estimations.

A boundary element model based upon each participant's MRI was used to provide an accurate model for source localisation. Accurate MEG/MRI co-registration was performed manually [22] and used in these calculations. The method of dynamical Statistical Parametric Mapping (dSPM) [28] was used to generate time courses of source estimates at 11,656 vertices over the cortex. The number of source estimates was reduced by taking single vertices from each of 68 regions of interest (FreeSurfer aparc cortical parcellation) [29]. Specifically, the vertex associated with the centre of mass of each anatomical region was used.

Upon completing the source transformations, each trial was split up to form active and baseline records. The same intervals as used in the sensor records were used with 1.7 to 0.7 seconds prior to button press forming the baseline class, and the interval 0.5 seconds before and after button press serving as the active class. As with the sensor-level records, each record was scaled to unit normal using the mean and standard deviation across each region. After processing was completed there were a total of 72,518 source-level records from 582 participants. Participants were excluded from this process if no valid epochs were available after the pre-processing step. Further exclusions were made if MRI/MEG registration could not be performed, if a valid boundary element model could not be constructed, or if anatomically prescribed dipole locations could not be determined. Each record consisted of 68 anatomical regions by 250 time samples.

2.2.4 Training, Validation, and Testing Subsets

For classification experiments using both the sensor and source-estimated records, the processed datasets were split into training, validation, and testing subsets. All subsets were constructed by randomly sampling on the basis of participants. This sampling was performed such that all records from a specific participant only existed within a single subset in order to avoid data leakage. For example, if a participant was selected for the validation subset, their records would only exist within the validation subset. The largest subset was used for training the networks and was an 80% portion of the available participants. The validation subset contained a random sampling of 5% of the participants and was used to guide training and inform hyper-parameter tuning. The testing dataset contained the remaining 15% of the data and served as an unbiased estimate of the classification accuracy of fully trained networks.

2.3 CNN Architecture

The CNN architecture for this study was developed with simplicity and efficiency in mind. It was systematically designed with the smallest number of components required to effectively classify MEG records. By developing networks in this way, training could be performed quickly and analysis of the trained networks was simplified.

The CNN design consisted of four layers: two convolutional layers, one fully-connected layer, and a softmax classification layer. The network architecture for this study is shown in Figure 2.1. The first convolutional layer contained eight kernels that were convolved with the input MEG records. This meant that the input data was convolved with eight different kernels producing eight feature maps as output for this layer. The feature maps produced by this layer, as well as by the second layer, had the same dimensions as the input data because convolutions were performed using zero padding with a stride of one. The dimensions of the first layer kernels were 8x16 (channels x time samples). While typical convolutional kernels used in image classification have dimensions of 5x5 or 3x3, 8x16 was chosen in order to capture the structure within the MEG data associated with button-pressing which is expected to occur on the timescale of tens of milliseconds. Furthermore, the kernel spanned 8 channels due to the fact that MEG data is spatially less resolved than image data. The output feature maps from the first convolutional layer were used as input into the second convolutional layer. The second layer performed convolutions using 16 kernels with a more standard 3x3 dimensionality. A standard set of dimensions were used in the second layer because, unlike the first layer, we had no *a priori* justification for a different kernel dimensionality.

Following the two convolutional layers, the outputs were flattened and connected to a fully-connected layer, which contained 64 neurons for sensor records and 32 neurons for source-estimated records. Each of these first three layers used standard rectified linear units (ReLU) for non-linear activation. Finally, a two neuron layer with softmax activation transformed the two class neuron values in order to output a probability distribution across the active and baseline classes.

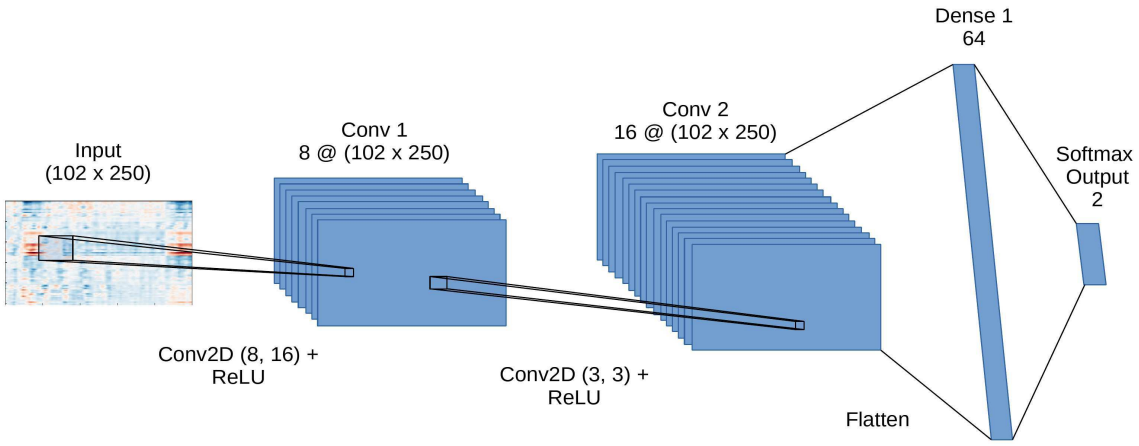

Figure 2.1: The architecture used to classify the 2D MEG sensor record with dimensions containing 102 channels by 250 time samples. The network contained four layers: 2 convolutional layers, a dense layer containing 64 neurons, and a softmax classification layer containing 2 neurons. This same design was used to classify the source-estimated records but with dimensions of 68 sources by 250 time samples propagated through the network and a dense layer containing 32 neurons.

Batch normalisation was also employed within each layer prior to passing the weighted sums to the ReLU activation functions. Batch normalisation is a technique that normalises layer outputs across training batches in order to provide gradient stability between layers, and to speed up network training [30].

In order to prevent over-fitting to the data, the network used dropout regularisation in which a proportion of network weights were randomly excluded during each training step. The fully-connected layer excluded 50% of neurons during each training step. Within each of the convolutional layers, spatial dropout was used to exclude 25% of the feature maps at each step.

2.4 Training Methodology

Binary cross entropy was the loss function that provided a measure of the difference between the predicted and target class distributions during training. For each record evaluation, the binary cross entropy was calculated as follows:

$$J = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}), \tag{2.1}$$

where y was the ground truth class label and $\hat{y}$ was the network predicted label.

The AdaGrad algorithm was used to minimise the loss function by updating the network weights during training [31]. AdaGrad is a gradient-based optimisation algorithm that minimises a function via iterative updates using gradient calculations and an adaptive learning rate strategy. It performed these updates using batches of input records during each training step (i.e. mini-batch gradient descent).

At each training step, batches of training records were used by the optimiser to evaluate the loss function, calculate the loss function gradient using backpropagation, and compute the network weight adjustments. Batches of 25 records were used at each training step when training using the sensor records whereas networks trained on the source-estimated records used batch sizes of 50 records. Each batch was made up of a random sampling of records from the training dataset. A training epoch was said to occur when sufficient training steps had taken place such that all of the available training data had been used to train the network. This meant that in the case of the sensor records, with 60,310 training records and a batch size of 25, a single epoch occurred after 2,413 training steps. The process of training the networks was typically repeated for several epochs until the network had sufficiently converged. Convergence, described in more detail below, was determined using an approach that considered a combination of classification accuracy and loss function values.

After each epoch, the classification accuracy of the network was calculated using the training and validation datasets. The classification accuracy was simply the proportion of correctly classified records to the total number of records within a given dataset. The classification accuracy values along with the cross entropy loss values over both datasets were recorded. These values were used to guide training, inform the training routine when to save network weights, and to allow for offline analysis of network performance.

Since gradient-based minimisation approaches cannot guarantee convergence, a heuristic approach was employed for obtaining optimally trained networks. During training, the weights of the network were saved if the validation accuracy surpassed that of all previous training epochs. Optimal networks were then chosen based upon a combination of largest validation accuracy achieved and smallest associated cross entropy values. In this way, the best networks had a combination of sufficiently large validation accuracy values and loss function values within the neighbourhood of a

minimum.

2.5 Characterising Network Performance with All Available Data

When gauging the range of performance one can expect from networks that have a specific architecture and that were trained upon a specific dataset, analysis must be performed using a collection of networks. Ensembles of 10 identical networks were trained using all of the available data in order to characterise the performance of our network design. One ensemble was trained using the sensor record data and a second ensemble was trained with the source localised data. All networks were trained for 50 training epochs while the classification accuracy and cross entropy values were calculated using the training and validation datasets.

After training was completed the classification accuracy and cross entropy values were averaged across the ensembles at each epoch. These values were plotted with the standard deviations in order to assess expected network performance and variability.

2.6 Characterising Network Performance with Limited Data

Due to the poor performance of networks trained using source localised records (see Figure 3.2), further investigations were limited to the sensor data only. Datasets with varying numbers of participants were constructed by randomly sampling from the original sensor records to characterise network performance in terms of dataset size. Sampling was performed on the basis of participant such that all of a participant's records were taken to avoid leakage. Like the original record set that contained all records, these datasets were split up into subsets consisting of training, validation, and testing records. Table 2.1 lists the number of participants within each dataset along with the number of participants that were split up into each subset.

Table 2.1: Datasets were generated with a varying number of participants. Each dataset was randomly sampled from the original set of MEG records and then randomly split up into training, validation, and testing subsets.

Total	Training (80%)	Validation (5%)	Testing (15%)
20	16	1	3
40	32	2	6
60	48	3	9
80	64	4	12
100	80	5	15
200	160	10	30
300	240	15	45
400	320	20	60
500	400	25	75
600	480	30	90

Ensembles of 100 identical networks were trained using each dataset for a maximum of 15 epochs. Due to the initially smaller dataset sizes and a constant model capacity (number kernels, fully-connected neurons were not altered), a higher degree of variability in performance was expected due to over-fitting. As such, a larger number of networks were included in each ensemble in order to better capture this variability within the analysis. Training was also limited to 15 epochs in order to limit the degree to which the networks would over-fit. As described in Section 2.5, classification accuracy was calculated using the training and validation subsets, and was saved for offline analysis. When training was completed, classification accuracy was also calculated using the optimally trained weights from each network along with the test subset of the sensor records. For each dataset, the training, validation, and testing classification accuracies were averaged and plotted as a function of the number of participants included in each dataset. The standard deviations calculated across each ensemble and classification accuracy type served as a measure of the spread in values.

2.7 Network Visualisation & Attribution

In the past, neural networks had been criticised for being difficult to interpret when compared to models developed using more traditional machine learning techniques.

However, there exist a number of methods that can be employed in order to investigate what features these networks extract from input data and identify the features most informative for classification.

In the case of models developed specifically using CNN architectures, extracted features can be observed directly by examining kernels and the feature maps they produce. Feature maps are constructed by convolving trained kernels with input data. Using these maps, one can examine the learned representations of the data that are propagated down through the network.

Recently, researchers in computer vision and image recognition have developed further methods for investigating and visualising trained networks. Some of these methods include activation mapping [32, 33], saliency mapping [32], and occlusion mapping [34]. Although not restricted exclusively to CNNs, activation mapping is a method of visualising the types of inputs that a specific network layer is sensitive to. Saliency and occlusion mapping, on the other hand, are attribution techniques that identify the portions of an input that contribute most to successful classification.

Since these approaches were developed for networks trained with photographs, we had to adapt them for use with networks trained on MEG. Typically maps are generated and analysis is performed on the basis of individual input records or images. Maps generated over a single image provide a readily interpretable and intuitive visualisation of feature importance. In contrast, when these techniques are employed with networks trained on MEG data, an ensemble approach is more appropriate. Like image data, MEG records will contain consistent, underlying structure. Whereas images contain features such as specific shapes that may be scaled, rotated, or occluded (e.g. windows, doors, and roofs within an image inform the presence of a house), features contained within MEG records are in the form of peaks within magnetic fields that occur at particular, reproducible times and on particular, reproducible sets of sensors. Due to the relatively low SNR in MEG measurements, these field deflections must be elucidated by aggregating over many records, which is not required when these techniques are applied to photographs. As such, we have developed analysis strategies that leverage maps generated over an ensemble of input records.

For the purposes of visualisation and attribution analysis, a single network was selected which was fully trained upon MEG sensor records. From the ensemble of 10

identically initialised and trained networks discussed in section 2.5, the best network was chosen. This network was one that exhibited the best performance in terms of classification accuracy and cross entropy loss from the ensemble. The above visualisation and attribution methods were then performed using this network in order to investigate the features specific to MEG that inform model performance.

2.7.1 Trained Kernels

Using the selected network, the trained kernels within the first layer were examined directly by plotting them using a colour palette such that negative values were blue and positive values were red. These plots were visually inspected in order to determine if they contained suggestive, identifiable structure. Specifically we looked for the presence of peaks, their sign, as well as the distance of separation between peaks.

2.7.2 Feature Maps

The eight kernels from the first layer of the trained network were convolved with all of the records from the test dataset in order to generate eight sets of feature maps. A total of 11,396 feature maps were generated for each of the eight kernels (91,168 total feature maps).

A random selection of feature maps were plotted and visually inspected in order to investigate the general effects that each kernel had on individual MEG records. In order to examine the overall 'response' of each kernel, the channel averages were computed over each of the active and baseline classes for the eight sets of feature maps. These map averages were then compared with the grand-average of the (unconvolved) test dataset for the active and baseline classes.

The peaks associated with different events within the active class tend to occur within specific time periods around the button press, but with some variance among participants. The process of averaging can have the effect of flattening out or otherwise smoothing specific components within the signal. For instance, averaging across trials would eliminate sensitivity to the change in alpha bursts associated with the presentation of a visual cue, or similarly, the beta burst suppression associated with button pressing. Signal content related to these bursts would be effectively removed because bursts are not phase-locked to the button press time.

In order to investigate peaks within specific channels and times, peak detection was performed on each feature map, and the spatial and temporal distribution of peaks across all feature maps was studied. Peaks that were greater than or equal to four standard deviations (4σ) were identified and the peak channels and peak times were recorded. Both positive and negative peaks (i.e., local maxima and minima) were included as peaks of interest because large amplitude positive and negative values both represent neuromagnetic signals in MEG data. These sets of channels and times were recorded across all feature maps for each of the eight kernels. From these data, two-dimensional histograms (102 channels x 250 time samples) that counted the number of times a peak was detected for a particular channel and time combination were generated across the set of feature maps for each kernel. Using these histograms we investigated the channels and times that had the largest overall contribution to the data representation that propagates through the network.

2.7.3 Activation Maps

Activation maps can be thought of as input records that maximally activate neurons within a network associated with a specified class. In the case of networks trained with photographs, they represent the types of objects and structures that a network is sensitive to when successfully classifying an object within an image. They are generated by iteratively optimising via gradient ascent within input space in order to maximise the probability associated with a target class [35]. The optimisation process can be initialised using random values or using representative data as seed input.

For the purposes of visualisation, a set of activation maps were generated using the MEG sensor records in the test dataset. These records provided the initial conditions for the optimiser in generating each activation map. A set of 11,136 maps were generated in this way with an even distribution of active and baseline records.

Peak detection and histogram construction was performed as described above in Section 2.7.2. Both the positive and negative large amplitude peaks were detected and went into the construction of the class-specific histograms. The histograms were examined in order to determine which sets of channels and times this method identified as having the most contribution to class-specific probability.

2.7.4 Saliency Maps

Saliency maps provide a representation of which individual data points within a given record have the largest impact on classification. In order to generate these maps, the softmax activation function was removed and the values of the class-specific neurons were computed directly upon processing input. The output values from these neurons can be thought of as a score function with respect to a specific class. By calculating the gradient of the score function with respect to each data point within an input, the importance of each data point was mapped. The data points that provided large gradients were considered to have a large saliency whereas data points associated with small gradient values had a smaller impact on classifying a particular record.

In order to investigate the most frequently salient points, saliency maps were generated across all MEG sensor records within the test dataset. Peak analysis as described above was also performed using the collection of saliency maps. However, since this type of mapping contains positive-definite values, only the positive peaks outside of four standard deviations were identified and recorded.

2.7.5 Occlusion Maps

Occlusion maps provide similar information as saliency maps but by using a different approach. The output of the class-specific neurons were again calculated directly and served as values of a class score function. Occlusion maps were generated by systematically setting portions of input records to zero while recording the change in this score function. Heat maps were generated as a moving window of 2x2 (channels x time samples) dimensions was systematically placed over the input records. As the occlusion window was placed over each region, the score function was recorded. The resulting heat map was re-scaled to the range [0,1]. When covered, those regions most important for classification provided the lowest values within the heat map (i.e. values closest to 0). The least important regions contained the largest heat map values since covering them had little or no impact on the score function.

Similar to the activation and saliency maps, occlusion maps were generated over the sensor records within the test dataset. Because the minimum of the score function infers region importance, the occlusion maps were re-scaled by subtracting 0.5 from

them (i.e. maps were re-scaled to the range [-0.5, 0.5]) prior to performing peak analysis. The channels and times that contained negative peaks outside of four standard deviations were recorded and considered regions important for correct classification.

2.7.6 Topographic Analysis

One interesting feature within the feature map histograms was an observed decrease in peak counts between times before and after button press within the active records. An advantage of constraining feature maps to the same dimensionality as the input sensor records is that phenomena found within feature maps can be mapped back to MEG sensor locations around the head. In this way, topographic maps could be generated using feature maps and feature map histograms. The intention of these topographic maps was to provide insight at an approximately anatomical level.

Based on an observed reduction in the number of peaks occurring after the button press in Active records, the difference in counts between the time ranges of -400 ms to -300 ms before button press, and 152 ms to 252 ms after was calculated across all eight feature map histograms. These calculated differences were then mapped back to the original sensor locations in order to construct topographic maps that highlighted areas where counts decreased most prominently.

Whereas feature maps illustrate which features are directly extracted from input data, activation, saliency, and occlusion maps attempt to identify features of importance via different approaches. In order to examine the common features across these importance mapping techniques, we investigated the peak channels and peak times that were commonly identified by all methods. To facilitate this, topographic maps were generated by mapping the histogram counts within selected time samples onto the associated MEG sensor locations. Time intervals were selected based on the 2D histograms, to investigate the spatial distribution during periods of time that were reliably deemed relevant to the CNN. Histogram values were averaged and projected over the following time ranges with respect to button press: (1) -200 ms to 0 ms, (2) -65 ms to -55 ms, (3) 50 ms to 60 ms, (4) 125 ms to 135 ms, and (5) 250 ms to 400 ms. In terms of known neuromagnetic activity for this task, these intervals encompass cue-related evoked responses, the motor readiness field, the movement-evoked fields in M1/S1, and the suppression of beta and alpha bursts.

There were five topographic maps generated from each of the activation, saliency, and occlusion map histograms. These topographic maps were then compared to one another to identify common peak locations. Topographic plots over these time ranges were also generated using the grand-average of the active records within the test dataset. These grand-average plots identified expected regions of activity with which to compare the features identified by the visualisation and attribution techniques in the context of the expected neuromagnetic responses.

Chapter 3

Results

3.1 Network Performance: Training Epoch Dependence

3.1.1 Sensor Records

The classification accuracy and loss function (cross entropy) values of ensembles of 10 networks trained for up to 50 epochs were calculated and recorded. Plots of the average classification accuracy and average cross entropy from these ensembles are shown in Figure 3.1. The average classification accuracy is shown on the left and the average cross entropy is shown on the right. The metrics calculated over the training subset are shown as blue circles and validation subset values are represented as orange squares. The networks tended to converge quickly with the average validation accuracy reaching a maximum of 0.960 ± 0.001 after only six epochs. Subsequent validation accuracy values tended to fluctuate about this maximum value. Conversely, the average loss function values attained a minimum around the same epoch value before trending towards an increase. The variance in the validation cross entropy among the ensembles also trended towards an increase as training progressed. This observed trend in the validation cross entropy would suggest that network performance did not improve after approximately six training epochs and that networks tended to over-fit to the data after this point. An increase in cross entropy would suggest that the optimisation process was increasing the difference in class probabilities away from the correct labels, essentially increasing the probabilities associated with incorrect labels. The average classification accuracy calculated over the test dataset and using optimal network configurations was 0.974 ± 0.001. Of the 11,396 records within the test dataset, a single network within the ensemble mislabelled 302 records. There was a relatively even split between the active and baseline records with 161 and 141 respectively.

Another point of interest is that the average training accuracy was lower than the

average validation accuracy for the first three training epochs. This is probably due to the fact that dropout was used during training. As a result, training accuracy was calculated using a portion of the network elements after each epoch.

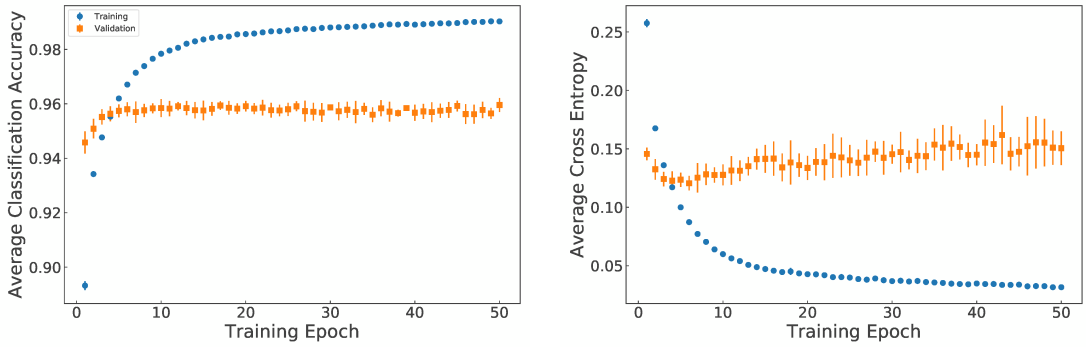

Figure 3.1: The performance of ensembles of 10 CNNs trained over 50 epochs as measured by the average ensemble classification accuracy (left) and the average cross entropy (right). The blue circles indicate the metrics calculated over the training dataset and the orange squares represent the values calculated over the validation set. After only six epochs the average validation accuracy reached a maximum value of 0.960 ± 0.001 before plateauing.

3.1.2 Source Localised Records

The networks trained on the source-localised data did not perform as well as those trained on the sensor-level data. Figure 3.2 shows the average classification accuracy (left) and average cross entropy (right) calculated over the ensembles trained on the source records. As above, the metrics were calculated over the training and validation subsets. The average validation accuracy reached a maximum of 0.814 ± 0.009 after four epochs before slowly decreasing. Similar to the sensor record results, the average cross entropy reached a minimum within the same epoch range before increasing. This behaviour of increasing cross entropy and decreasing validation accuracy (with improvements to training accuracy) would suggest that again the networks were over-fitting to the data but more severely than those trained to the sensor record data. Using the optimally trained source-level networks, the classification accuracy on the test subset was 0.793 ± 0.006.

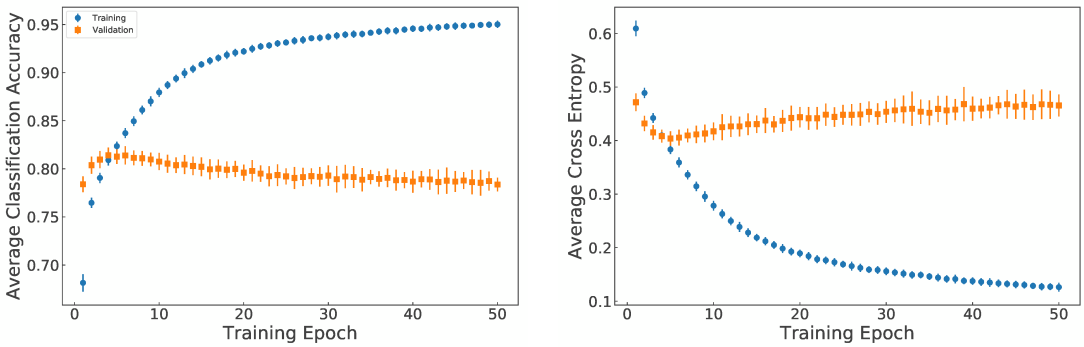

Figure 3.2: Ensembles of 10 CNNs were also trained using source-localised MEG records with the average classification accuracy (left) and average cross entropy (right) being recorded. Metrics calculated using the training data are represented as blue circles and the orange squares represent the values calculated using the validation subset. The networks poorly classified these records with a maximum average validation accuracy of 0.814 ± 0.009 after four epochs. After this point the validation accuracy decreased while the cross entropy increased indicating that the networks tended to over-fit to the data even with a decreased dense layer capacity.

3.2 Network Performance: Dataset Size Dependence

In order to estimate the effect of study sample size on CNN performance, ensembles of 100 networks were trained using datasets that contained increasing numbers of participants. For each dataset of varying size, the optimal networks from each ensemble were used to calculate the classification accuracy on the training, validation, and testing subsets. Figure 3.3 shows the dependence of network performance on the number of participants included within the dataset. The training classification is shown as blue circles, the validation subset is represented by orange squares, and the test classification is shown with green triangles. From the smallest dataset containing 20 participants up to the dataset containing 200 participants, classification accuracy steadily increased suggesting that a substantially better model could be developed as more participants were included in the analysis. For larger datasets, the networks tended to increase accuracy but with a smaller rate of increase suggesting that adding participants does not substantially impact model performance at these samples sizes. The average test accuracy calculated over the ensemble trained using data from 600 participants was 0.965 ± 0.002.

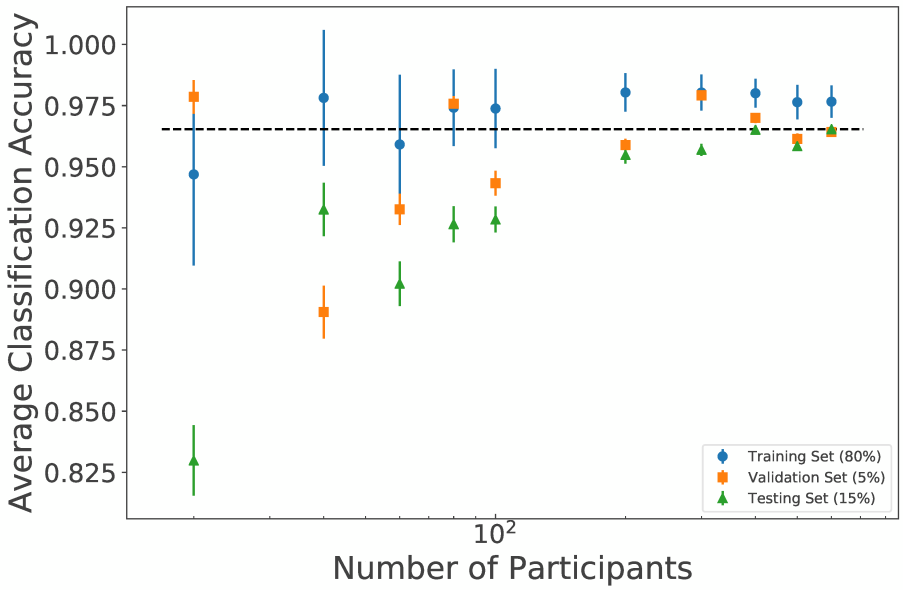

Figure 3.3: Network performance as a function of dataset size. Ensembles of 100 CNNs were trained on datasets containing increasing numbers of participants. Each point represents the ensemble average calculated over the training set (blue circles), validation set (orange squares), and testing set (green triangles) for each dataset size. The average network accuracy showed a steady increase up to and including the dataset containing 200 participants. After this point, average accuracy increased but at a lower rate. The ensemble average over the test subset of the largest dataset was 0.965 ± 0.002

3.3 CNN Analysis: Trained Kernels & Feature Maps

Figure 3.4 shows the grand-average MEG data for the 500 ms prior to and following the button press. Clear magnetic field deflections are evident at -100 ms, -60 ms, 55 ms, 120 ms, and 325 ms. Within each topographic plot, the sets of red and blue maxima are paired and represent a single source of neuromagnetic activity. The positive red maxima represent the components of the magnetic field coming out of the head and the blue maxima represent the field components pointing into the head (with current direction indicated using the right hand rule). The topography of the peaks prior to the button press suggest likely bilateral occipital activity related to the cue. At 55 ms, the topography suggests bilateral central sources, likely related to efferent and afferent sensorimotor activity associated with button pressing. The topographies at 120 ms and 325 ms are less interpretable, likely due to the involvement of multiple

brain regions in later processing.

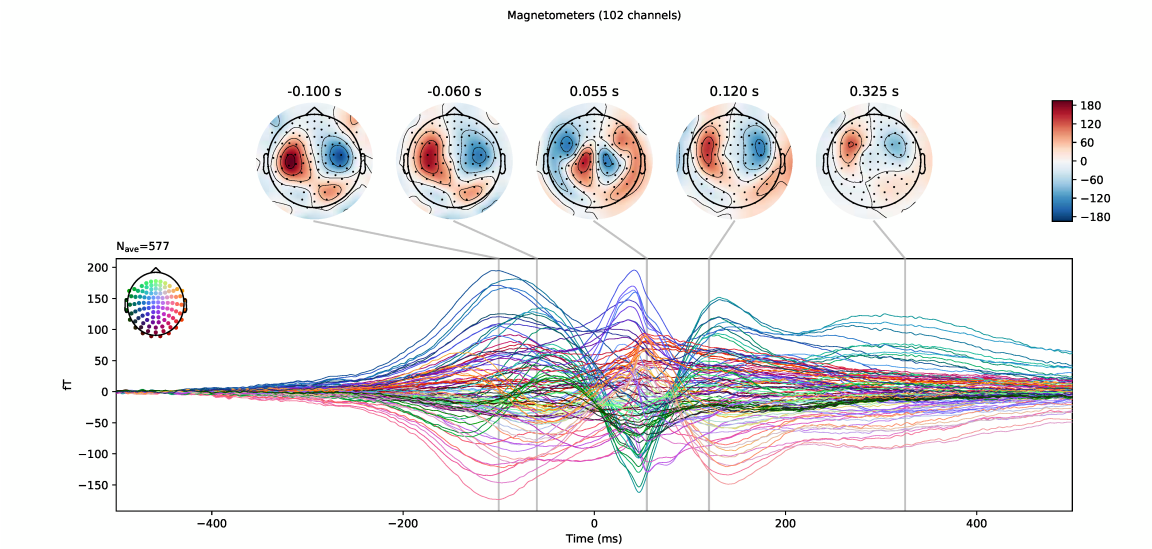

Figure 3.4: The grand-average MEG data for 500 ms before and after the button press. The topographic plots are shown above for locations with evident field deflections. The topographies for the times prior to button press suggest cue related bilateral occipital activity. The topography plotted at 55 ms suggests sensorimotor activity. Topographies plotted for the later times (120 ms and 325 ms) are less interpretable due to a complex of multiple brain regions involved in later processing.

The trained kernels from within the first layer of the best CNN trained using the sensor records were extracted and plotted. Figure 3.5 shows these plots with a colour gradient of positive values in red and negative values in blue. Each of the trained kernels contain alternating positive and negative elements that appear to act as peak detectors with elements separated spatially, temporally, or along both dimensions. This would suggest that kernels appear to be sensitive to patterns within the data along the temporal and/or spatial dimensions. For instance within kernel (II) pairs of peaks with opposite signs can be seen separated temporally within channels one, three, and four, suggesting that this kernel is sensitive to peaks within time. Kernel (V) contains readily identifiable peaks separated spatially between channels four and six. Kernel (I) appears to contain a combination of both of these patterns with temporally separated peaks among the third and seventh channels (8 to 36 ms and 12 or 16 to 40 ms respectively), and spatially separated peaks between these two channels. Other

kernels however, do not appear to contain readily discernible patterns.

The structure contained within these kernels is constructed during training in order to extract specific features from the data. These extracted features go into forming an informative representation of the data with which the network uses to perform classification. Of particular importance for MEG data are the temporal dynamics of the patterns within the kernels. Among the kernels, there is varying temporal separation among peaks with some appearing to be separated by 32 up to 64ms (8 to 16 kernel elements). For example kernels (I) and (II) show this, as well as within channels three and four of kernel (VI). This suggests that some kernels may have a sensitivity to temporal dynamics with a particular associated period. In some cases, these time scales equate to the beta burst frequency range. For example, channels two and five on kernel (IV) have positive and negative peaks separated by 20 ms. These channels will be maximally sensitive to an oscillation with a 40 ms period, which equates to 25 Hz.

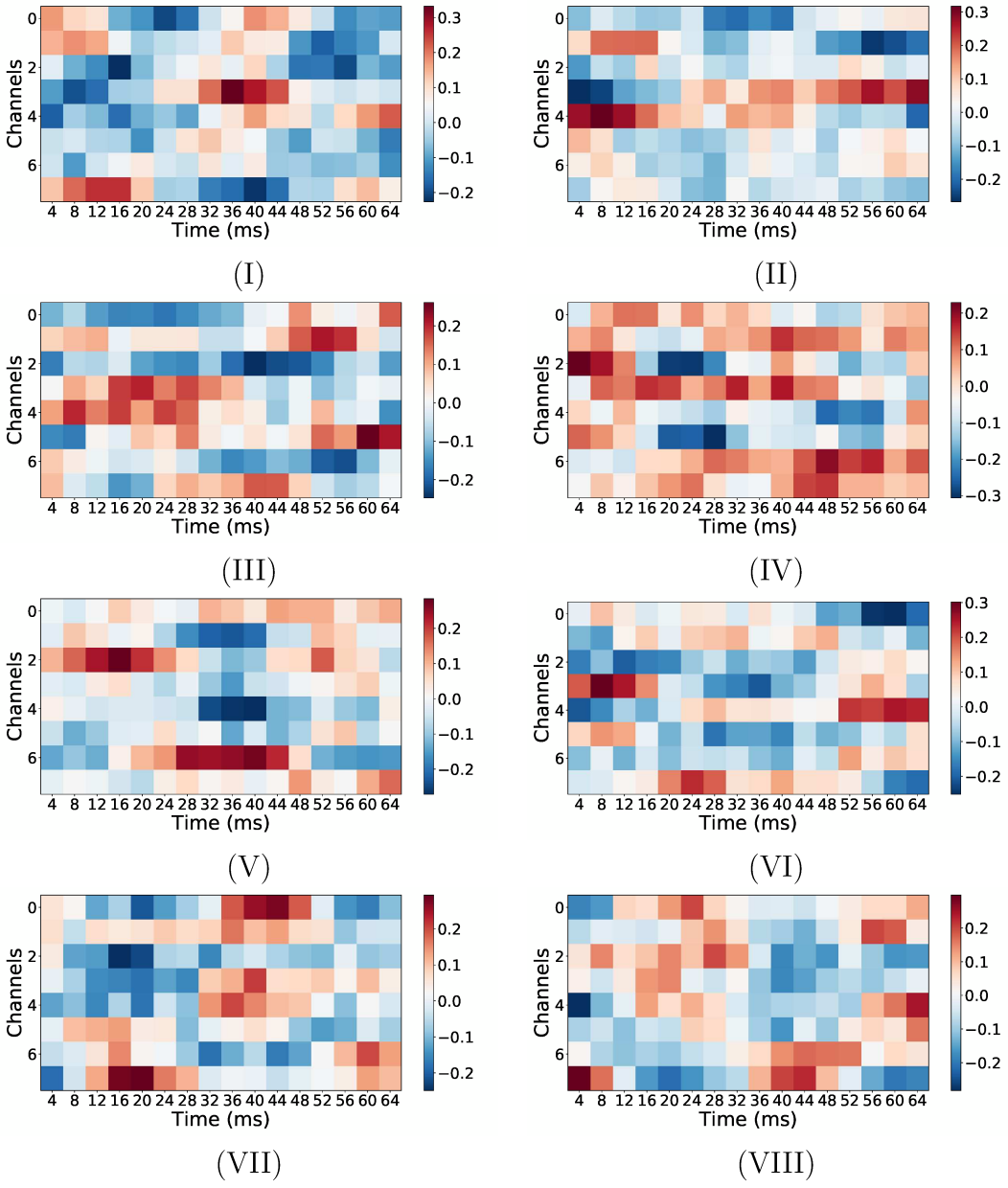

Figure 3.5: The kernels from the first layer of a CNN trained on the sensor record data. Kernel elements are plotted with red positive values and blue negative values. The presence of peaks with alternating signs separated spatially and temporally suggest that kernels are sensitive to peaks within the channels and time samples of the records.

Using the above trained kernels, feature maps were generated by convolving the eight kernels with the sensor records within the test subset. These feature maps were averaged by channel for each of the eight kernels. The feature map averages over the active records tended to show structure similar to that of the grand-average over the same test data records indicating that the kernels are, at least in part, sensitive to

the evoked fields occurring during a cued button-press task. Figure 3.6 compares the grand-average (top) of the test subset for the active (left) and baseline (right) classes to the averages of the feature maps (bottom) generated using one kernel. All plotted time series for the Active records are centred with the button press occurring at time $t = 0$ ms. The feature map averages were calculated using the feature maps generated with kernel (I) shown in Figure 3.5. Peaks prior to and following the button press can be seen within both the grand-average as well as the feature map average. The peaks within the active feature maps tended to have larger amplitudes over some of the channels compared to those within the test record averages. Furthermore, some of these peaks were temporally shifted as compared to the associated grand-average peaks.

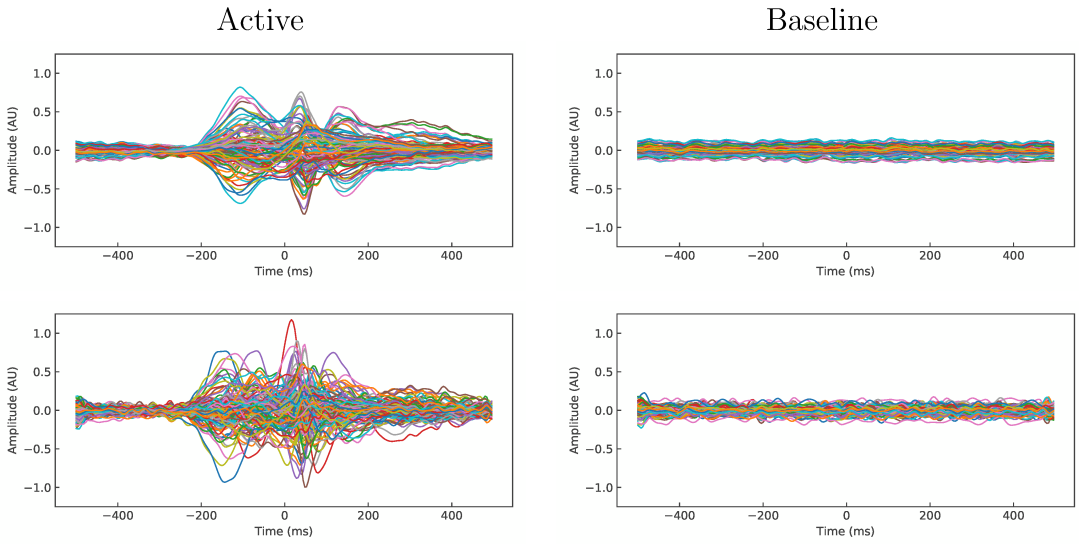

Figure 3.6: A comparison of the channel averages calculated across the active (left) and baseline (right) classes from the test dataset records (top) and the feature maps (bottom) generated using kernel (I) from the trained network. Active plots are centred at the button press at time $t = 0$ ms. Baseline plots are centred at -1200 ms prior to the button press. The peaks associated with stimulus onset as well as button press can be seen within both averages over the active class. Some of the peaks within the feature map averages appear to have larger amplitude than those within the grand-average. Furthermore, some of the peaks are shifted in time with respect to their grand-average counterparts.

Peak analysis was performed in order to investigate where the largest signals tended to exist within the features maps. Histograms that counted the number of

times a peak of 4σ or larger within a set of channels and times were constructed across all sets of feature maps. The left panel of Figure 3.7 contains the histogram constructed from the feature maps produced by kernel (I) in Figure 3.5. Each trace represents the peak counts for a specific channel at each time sample. A key feature observed within this (and other) histograms was an apparent decrease in counts after the button press, in comparison to the interval prior to the button press. This decrease in counts following the button press appeared to match the temporal dynamics of expected beta suppression in S1/M1. Therefore, the difference in counts between intervals before (-400 ms to -300 ms) and after (152 ms to 252 ms) button press were calculated over each channel. The intervals from which the count differences were calculated are shown in blue. These differences were mapped onto the sensor location to construct topographical plots in order to investigate the location with the largest decreases.

The topographical map constructed by mapping the change in counts following the button press to the associated sensor locations is shown on the right side of 3.7. The largest decreases in counts appeared to occur over the central sensors overlying S1/M1, as would be expected with beta suppression.

Along with beta suppression, this particular kernel appeared to be sensitive to features along the edge of sensor space. This could be the result of the representation used for the senor records. Within the sensor record array, sensors were listed by row in an order that matched the naming convention set by the hardware manufacturer. This ordering only partially resembled the spatial relationship among the sensors with groups of sensors clustered together within specific regions of the head. While sets of sensors were listed together in regional groups, there were some spatial discontinuities among these groups.

Reminiscent of the grand-average and feature map average curves shown in Figure 3.6, two peaks were observed prior to button press between -200 ms to 0 ms, and afterwards between 0 ms to 150ms. These appear to be associated with expected peaks related to stimulus onset and the sensorimotor response respectively. Two prominent peaks can be seen within the first few time samples as well as the last few. These peaks are artefacts that were possibly the result of edge effects introduced by zero-padding the convolution operations. These artefacts did not affect our analysis

since important data features were generally contained within a range of time samples away from edges.

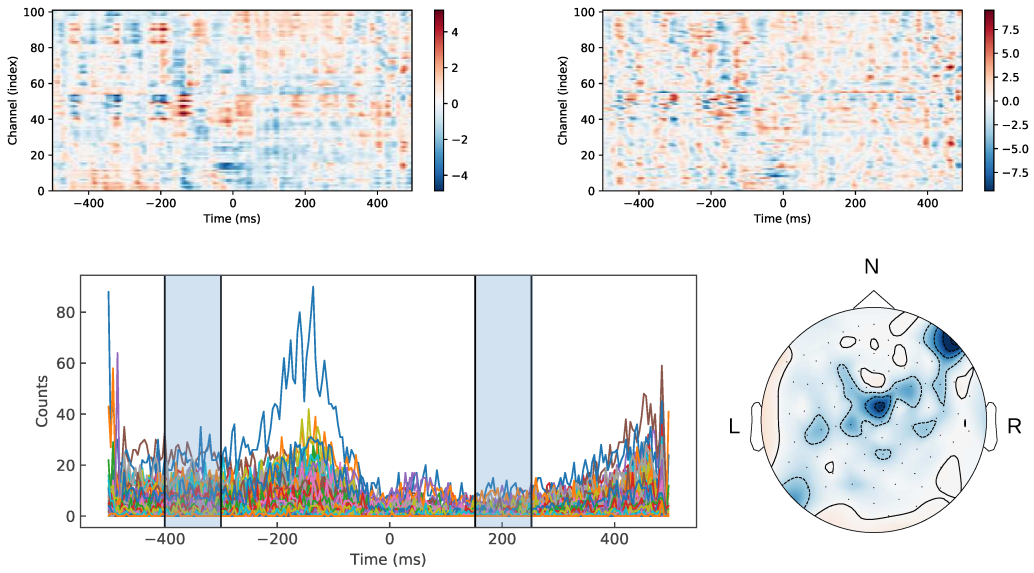

Figure 3.7: Peak analysis of the feature maps show evidence that trained kernels were sensitive to beta rhythms. An example active MEG record is shown (top left) along with a feature map produced by convolving the record with kernel (I) (top right). A histogram of the number of times peaks $\geq 4\sigma$ occurred within feature maps for one kernel is shown (bottom left). An apparent decrease in counts before and after button press can be seen. This difference in counts was calculated for each channel between the intervals shown in blue. The topographical plot (right) shows the sensor location of these differences with large decreases centrally located. The temporal dynamics of the histogram along with the centrally located decrease in counts suggest that the trained kernels were sensitive the phenomenon of beta suppression.

3.4 CNN Analysis: Visualisation & Attribution Maps

Activation maps were generated by optimising input space in order to maximally activate output neurons associated with the active and baseline classes. A set of activation maps were generated using the records from the test dataset as initial conditions for optimisation. Similar to the peak analysis performed with the feature maps, peaks were identified and counted in order to determine the locations that contained the most prominent features (peak values $\geq 4\sigma$).

The histograms shown in Figure 3.8 count the occurrence of peaks within the active

(left) and baseline (right) activation maps. Peaks appear to be sparsely distributed across both classes indicating that only a small number of data points contained large-amplitude peaks. This small number of peaks could be due to the optimiser only tuning a small number of data points in order to maximise output with respect to each class. Within both classes, peaks were primarily grouped into a small subset of channels and times between $t = 0$ ms and $t = 200$ ms. The counts within the baseline histogram are more widely distributed than those within the active histogram. Unexpectedly, there are a large number of peaks within the last time point of the active maps possibly associated with the edge effects introduced by zero padded convolutions. This would suggest that features propagated through the network from active records typically contained this type of edge artefact.

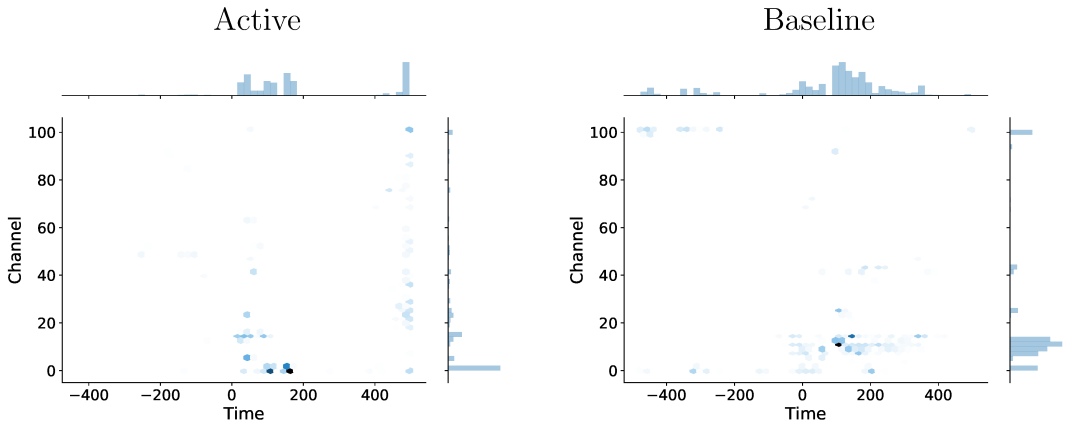

Figure 3.8: Histograms that count the number of peaks $\geq 4\sigma$ within the sets of active (left) and baseline (right) activation maps are shown. Counts appear to be clustered within a small number of channels and mostly occur after $t = 0$ ms for both classes. Counts are more broadly distributed for the baseline class with some peaks occurring before 0 ms.

A set of saliency maps were generated using the sensor records from the test subset. By calculating the gradient of each class neuron with respect to each data point within an input record, the relative importance of each point was mapped. Figure 3.9 shows the peak histograms generated across the active (left) and baseline (right) records. The distribution of peaks is similar across both classes with a large central peak occurring between $t = 50$ ms and $t = 60$ ms. This largest central peak occurred within a temporal location similar to the one found within the feature map histogram shown in Figure 3.7, and furthermore the second peak found within the grand-average

(see Figure 3.4). This central peak is surrounded in time by smaller peaks on either side. These smaller peaks occurred within a small subset of the same channels as the larger peak. There are also a number of regions that contain clusters of peaks prior to button press as well as afterwards. Within the active class, maps appeared to contain more salient data points prior to button press between $t = -200$ ms and $t = 0$ ms. More salient points occurred within a later region (after $t = 200$ ms) within the baseline maps as compared to the active ones. The structure contained within these histograms suggest that the network was sensitive to the expected features that can be seen within the grand-average of the active records, including activity likely related to both the cue ($t < 0$ ms) and the button press ($t > 0$ ms). The similarity between the active and baseline histograms could possibly be due to the fact that the network attempted to identify the occurrence or absence of these features when performing classification.

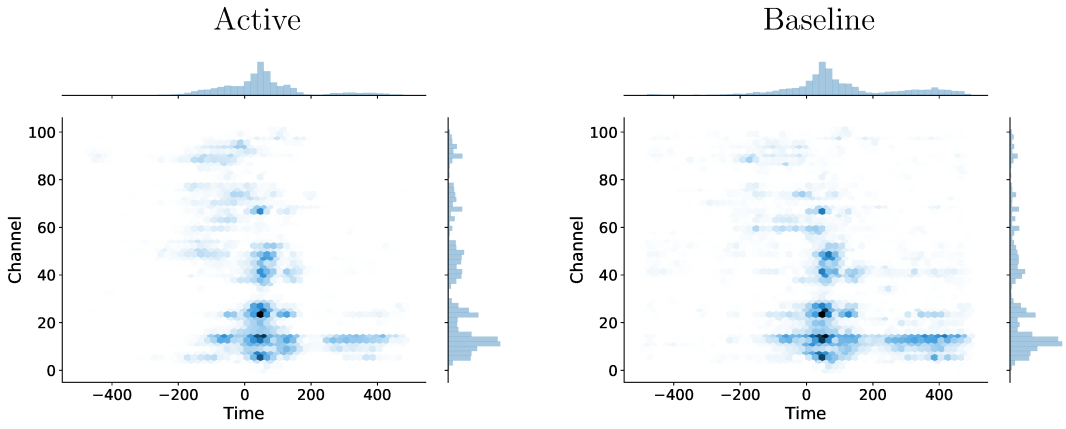

Figure 3.9: Similar to the plots within Figure 3.8, histograms that count the number of times large-amplitude positive peaks occurred within the active (left) and baseline (right) saliency maps is shown. The presence of peaks indicated the importance of particular data points in classifying the input record. The largest number of peaks appear to occur between $t = 0$ ms and $t = 200$ ms. The salient data points are similarly distributed between the active and baseline classes. More peaks occurred prior to 0 ms within the active maps, whereas more peaks occurred after 0 ms within the baseline saliency maps.

Like saliency maps, occlusion maps quantify the relative importance of regions within input records. These maps were constructed by measuring the relative differences in class-specific neuron values as a 2x2 occlusion window systematically covered

the records. As with all other visualisation/attribution maps, the occlusion maps were generated over the test dataset records. Negative values within these maps were associated with a decrease in class-specific neuron values and therefore indicate regions that were important for determining the class membership of a record.

Figure 3.10 shows the histograms resulting from peak analysis performed on the active (left) and baseline (right) occlusion maps. There appears to be some common structure among the saliency and occlusion map histograms. Large central peaks can be observed between $t = 50$ ms to $t = 60$ ms within both the active and baseline histograms. The active histogram also appears to contain fewer pronounced clusters of peaks prior to button press between $t = -200$ ms and t=0 ms. Although there are similar peaks within both histograms, the baseline counts are far more distributed over the channels and times with many more peaks counted after $t = 200$ ms.

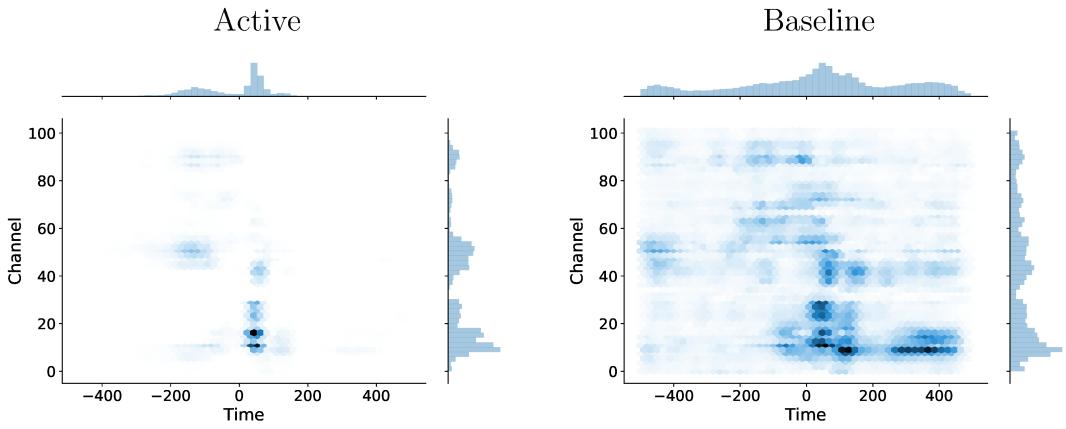

Figure 3.10: Shown are the histograms that counted the number of times negative peaks $\geq 4\sigma$ occurred within the active (left) and baseline (left) occlusion maps. Negative values within the occlusion maps imply that regions were important in determining the class of an input record. The active histogram contains structure that is consistent with that observed within the saliency histograms. Specifically, a large central peak can be observed after button press (between $t = 50$ ms and $t = 60$ ms) as well as less pronounced peaks prior to button press (between $t = -200$ ms and $t = 0$ ms). Compared to the active histogram, the peak counts within the baseline histogram are distributed across the more sets of channels and times.

In order to investigate which brain areas were identified as generating signals relevant for classification, the active class histograms generated by each method were mapped back to the MEG sensor locations associated with each channel. This transformation was performed in order to investigate the locations of important features,

and to compare these locations among the methods used, as well as to the locations of activity that we expect to observe during the task of cued button pressing.

For reference, the top of Figure 3.11 contains a plot of the grand-average over the active test records. Immediately below are the topographic plots containing the average sensor values over five selected time intervals. Similar to the topographies over the grand-average of the MEG data (Figure 3.4), the topographic plots were generated over regions containing evident peaks and highlight regions of activity associated with the cue and button press. Specifically, the first two plots prior to button press show topography that suggests bilateral occipital activity related to the cue. The topographic plot in the middle (50 to 60 ms) suggests sources that are bilateral and likely related to the sensorimotor activity involved in button pressing. The last two plots over 125 to 135 ms and 250 to 400 ms show topographies that are less interpretable with activity that is likely the result of multiple brain regions involved in later processing.

Following the grand-average topographic plots are those constructed over the activation maps, the saliency maps, and the occlusion maps. Each set of the topographic plots shows some correspondence with the grand-average activity across the same time intervals. These similarities range from singular peaks as in the case of the activation maps, to collections of different peaks associated with activity and edge artefacts.

The activation maps for each time interval appeared to identify a single loci of strongly activated sensors. This loci is in a different location for each time interval, and sometimes overlaps with an associated channel set within the grand-average. There were no counts during the 125 ms to 135 ms interval, although there is clearly activity in the grand-average at this time. This lack of counts is not surprising, given the sparsity of data in the 2D histogram (see left panel of Figure 3.8). The lack of consistency between the activation maps and the movement-related neuromagnetic signals is perhaps not surprising, given that activation maps often do not result in visualisations that look like any record in the original image set. When applied to photographs, activation maximisation techniques typically result in mappings that contain class-related shapes and textures with varying colours and orientations. Image activation maps represent inputs that maximally activate class-specific neurons, or in

other words, the types of inputs the network is sensitive to. As such, MEG activation maps cannot be considered representative MEG records of the target classes. In the case of these maps, it appears that optimisation was performed by tuning only a small subset of the input data points. So while these activation maps may not represent records belonging to a particular class, they do suggest the importance of a small subset of the possible channels and times.

Upon visual inspection, the saliency maps appear to have the most correspondence to the features observed within the grand-average, as well as our expectations based on prior studies of cued-button pressing. The saliency maps seemed to identify most of the same peaks as the activation maps. Specifically, during -200 to 0 ms, important features in central and occipital sensor sets were observed that align very closely with the sensors activated within the grand-average during this time interval. During -65 to -55 ms, high counts on a small locus of left central sensors was observed, which overlaps with a positive locus in the grand-average. Interestingly, the other strongly activated sensor loci in the grand-average at this time interval show a weaker correspondence in the saliency map, indicating that not all of the field represented in the grand-average is particularly salient for classification by the CNN. During the remaining three time intervals, consistently high saliency is measured on central sensors that overlap with the measured grand-average. Interestingly, the saliency topography matches closely with the feature map topography in Figure 3.7. This suggests some commonality between these visualisation/attribution methods, and points to a possible representation of beta suppression in the saliency map.

Lastly, the topographic plots from the occlusion maps also identified some of the same regions of activity but they also seemed to identify loci of sensors along the edge of the array. This is also true for the saliency map in the -200 to 0 ms time interval, and is uncommon in results generated by standard approaches to analysing neuromagnetic recordings. This result likely represents an artefact of the analysis method. These are termed “edge effects”, as they might be the result of the fact that the sensor order has only a loose relationship to their spatial location. Thus, there are spatial discontinuities in the representation of the records, wherein channel order within the data arrays did not fully correspond with the relative locations of the MEG sensors on the head. Kernels that are selective for sensors (rather than

time) may enhance the saliency at these discontinuities.

It is important to note that the grand-average provides a basis for identifying regions of importance but it may not contain some of the features that visualisation methods are sensitive to. Specifically, the grand-average will not include any signals associated with rhythmic bursts because these bursts are not phase-locked to the button press. However, the results of the feature map visualisation make it clear that the CNN is sensitive to both rhythmic bursts and evoked activity. This may partly explain why the topographic plots show only partial overlap with the grand-average in regions and time intervals.

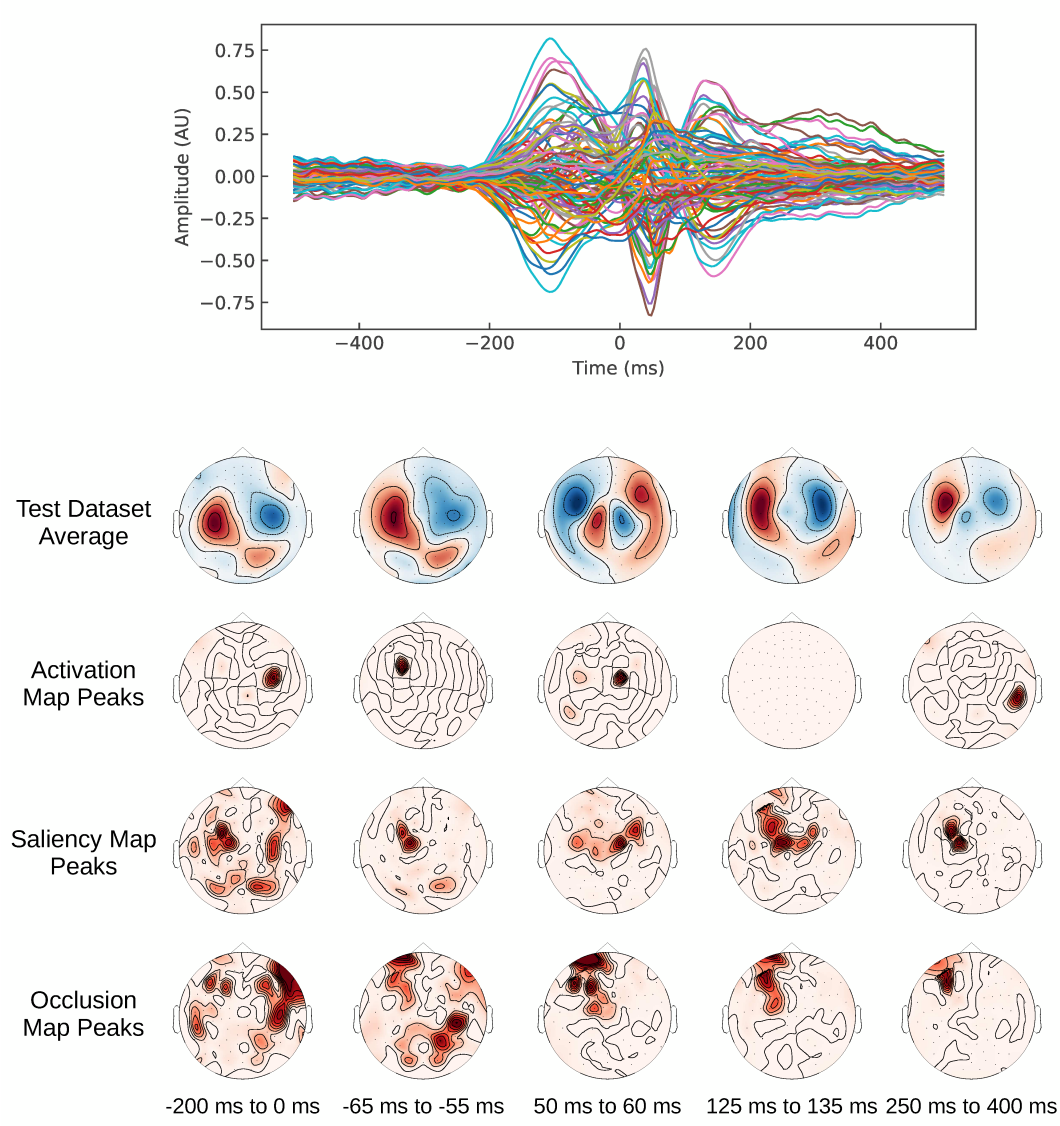

Figure 3.11: Topographic plots generated from mapping the visualisation and attribution histograms to sensor location are shown for five selected intervals. The grand-average over the active test subset records is shown here for reference (top). The topographic plots generated over the grand-average are shown immediately below. Below these are the topographic plots generated over the histograms from each type of visualisation/attribution map. These plots show that some of the regions identified by the visualisation methods correspond with the grand-average activity across the same time intervals. Correspondence ranges from peaks within singular regions (activation maps) to a collection of peaks and edge effects.

Chapter 4

Discussion

4.1 Summary of Main Findings

In this study we developed a CNN that can classify between active and baseline intervals within MEG sensor records with minimal processing. We have shown that high quality classification can be achieved using a relatively small network constructed using only the basic building blocks of CNNs and without specialised architectures. These results suggest that a CNN is a viable choice for constructing models using MEG data. This is an important step within the field because previous research has mostly focused on the application of traditional machine learning techniques to MEG. Any examination of deep learning has primarily been limited to studies with small datasets.

Despite successes using sensor-level records however, networks trained on source-localised records performed poorly. Further research into appropriate data representations and network designs will be required in order improve performance with source-estimated data.

We also studied the effect of dataset size on the performance of our networks. By training ensembles of networks on datasets that contained sensor records from a variable number participants, we showed network performance increased dramatically with dataset size, particularly with smaller datasets (on the order of tens of participants). These dramatic increases were observed for datasets that included up to 200 participants. For datasets larger than 200 participants, performance continued to improve but with less pronounced increases in classification accuracy. As expected, optimal performance was observed when networks were trained using all available data. These results suggest that any future application of deep learning to minimally-processed MEG data should employ datasets containing on the order of hundreds of participants.

Using visualisation and attribution techniques developed within the field of computer vision, we showed that a CNN trained using MEG sensor data can learn the underlying structure within MEG data in order to perform classification. It was shown that one can determine the features that are extracted by a trained network in order to effectively represent the data. This can be achieved by examining the trained kernels and the features maps they produce. With the use of activation, saliency, and occlusion maps, we investigated the specific channels and times within active records that the network focused upon in order to identify records. Our analysis showed that these regions corresponded well with the activity associated with perceiving an auditory and/or visual cue, as well as the motor response elicited during button pressing. These visualisation and attribution techniques offer a means of investigating network training as well as a way of determining which features of brain activity are important for accurate classification.

Importantly, these techniques allow for analysis on a single-trial basis. Traditional analysis of MEG data is performed on aggregates from a dataset in order to elucidate phase-locked activity as well as changes in rhythmic bursting activity. The approach described here allows for the investigation of both. Feature maps from a trained network showed network sensitivity to beta suppression as well as the activity associated with the auditory,visual, and the sensorimotor evoked fields.

4.2 Poor Performance on Source-Estimated Data

Although our CNN design could effectively classify sensor-level records, networks trained on the source-localised data produced poor results. One factor that could contribute to this is the fact that source localisation methods provide an estimate of the current flow within different regions of the brain. These estimates can be effected by factors including the model that was used to facilitate the calculations as well as the specific mathematical method employed. While one can achieve relative agreement among localisation methods when dealing with a well-understood paradigm, variability can exist among the results of various methods. For example, the beamformer spatial filter used to estimate current flow has a tendency to attenuate spatially separated sources that are highly correlated [36]. If any such activity was important for classification at the sensor level, then it will be attenuated in the

source-estimated data, which would negatively impact on classification accuracy.

Another likely challenge for the source-estimated model is the spatial representation of the data. Although these estimates are calculated over a large collection of vertices, a reduction in the number of vertices is required to form practical representations. Thus, the number of sources is reduced from 10,000's to less than 100 by taking the source estimate at the centre of anatomically defined regions of interest. The source activity at each region is represented in the same way that MEG sensors are represented in the sensor-level CNN. This reduction in the number of channels makes it convenient to investigate source activity in terms of projections onto models and aggregate calculations. Importantly, at the sensor level there is a rough spatial correlation between sensor index and sensor position, such that adjacent rows in the record tend to represent data from nearby sensors. This spatial correlation is lost in the source-estimated representation, when reducing 3-dimensional positions to rows in the record. Likely, this is problematic for training the kernels. Thus, there is a loss of information, in terms of representing these records effectively as input photographs.

In order to fully exploit the advantages of source space estimates and the ability to extract features using convolutions, a different representation is required. As stated, we trained networks directly on arrays that contain the time courses over each region. Better results might be attained by using a four-dimensional representation of the data. This representation could include three dimensions to represent the spatial projection of activity onto the brain with temporal dynamics over the fourth dimension. This particular representation of the data would be much more computationally intensive, as it would mean using the original 10,000's of vertices. To address this, records could be restricted to particular regions of interest (e.g., ranges in x, y, and z) for tasks involving more localised activity. However, we specifically wished to avoid feature engineering (such as region selection) in this study. With sufficient time and CPU access, a four-dimensional CNN would be an interesting future direction for this study.

4.3 Representation Learning Versus Feature Engineering

One main advantage of using deep learning is that feature engineering is not required in order to develop viable and effective models. Guided by probability, neural networks learn representations of the data automatically in order to accomplish a task. Feature engineering employed in more traditional machine learning applications, on the other hand, requires expert-level knowledge of a problem area and model predictions can be sensitive to the choice of representation. Furthermore, these representations may fail to capture the more subtle factors of variability within the data that is required to effectively represent class membership. The representations generated by a deep learner empower models that are both more robust and allow us to gain insight into highly complex systems. When provided with enough data, and a network with an appropriate capacity, deep learning models can be developed to solve problems within highly complicated subject areas.

Using various visualisation and attribution methods, we showed that these representations can be examined both to ensure that networks are performing correctly, and in an exploratory manner to reveal which features in the data are important for classification. Our results provide evidence that visualisation and attribution methods are effective at identifying features of the neuromagnetic recordings that are previously known to be generated during cued button pressing. This is an important validation of the proposed CNN before its utilisation in more challenging problems (i.e., more complicated cognitive tasks). Our results do not provide clear evidence that these visualisation and attribution methods can identify previously unknown features within MEG data. However, this possibility is difficult to rule out entirely, as these unknown features may be subtle or non-obvious. This points to one of the challenges of visualisation and attribution, which is that it still falls to a human to interpret the maps. It is also worth noting that the network we employed was shallow. Our choice of model allowed rapid development toward a proof of concept but possibly limited the interpretability of its results. Perhaps a more specialised, deeper CNN implementation, can readily identify more features within these data as well as any hierarchical relationships that exist between features.

4.4 Relative Value of Visualisation and Attribution Techniques

Among the visualisation and attribution methods presented, saliency maps appeared to provide the most informative results. While the active and baseline saliency histograms were very similar, there were some noticeable differences between these and the activation and occlusion histograms. The activation histograms tended to show sparse regions of activation and produced results that were difficult to interpret. These differences were likely due to the optimisation process that produced them. The occlusion histograms showed similar structure to the saliency histograms but with importance being more spread out over the baseline records. This intuitively suggests that specific regions were more important for classifying active records as compared to the baseline ones.

Visual inspection of the topographic plots showed that the saliency maps tended to more accurately capture the regions of activation as demonstrated within the evoked field. Furthermore, the late component represented in the saliency maps (approximately 200-400 ms) may represent a sensitivity to suppression of cortical rhythms.

By calculating the change in the output with respect each data point, the resulting mappings provided a reasonably intuitive representation of feature importance. One thing to note about the implementation shown in this study is that the absolute values of the gradients were propagated through the network to generate the maps. This meant that small changes in data points that had a positive impact on class score were represented as well as those with a negative effect. Other gradient-based techniques can be employed that use only positively impactful gradient information [37].

The gradient-based saliency method identified more of the expected regions than those sparsely identified using input activation and did so with fewer edge effects than what was observed in maps generated via occlusion. The activation maps contained limited regions with relevant activation. This was probably due to the fact that the input could be tuned at arbitrary data points in order to make patterns that give high probability for a specific class. In our examples, it appears that the activation maps were generated by maximising activity over a small subset of sensors and times.

The occlusion maps lacked the specificity of the saliency maps and attributed importance to unrelated sensors. They were generated by systematically setting a grid

of data points to zero using a 2x2 moving window. At certain locations within the records the occluder would cover both informative regions as well as non-informative ones at the same time. A decrease in the score function would still be observed when this occurred. When this occurs in photographs, a gradient of gradually changing values can be seen in the resulting occlusion maps. However, due to the spatial discontinuities within the representation of the MEG sensor records, importance can be attributed to sensors that were spatially unrelated. These regions where the occluder overlapped with spatial discontinuities were likely the cause of the edge effects observed within the topographic plots.

4.5 Application to New Datasets

Following a few guidelines, the methods we have demonstrated can be used to train and analyse networks using new data. First, an appropriately large dataset is required in order to take advantage of deep neural networks. In the case of our simplified, shallow model, and the particular binary classification task we used, we found that a dataset that contained records from hundreds of participants was essential for accurate classification. This was equivalent to having a dataset on the order of tens of thousands of individual records with an even distribution of records between the classes. The reason for requiring such a large dataset is two-fold: more training data means better classification accuracy, and the dataset should be divided into subsets in order to guide training and test model performance.

To ensure that networks train correctly, and to avoid over-fitting to the data, classification accuracy as well as loss function evaluations should be considered. These metrics should be calculated using the validation subset of the data after each training epoch. Training of a network should proceed until the network has converged to a classification accuracy maximum and an associated loss function minimum. With this combination of metrics, the resulting models should generalise to new data.

4.6 Future Steps for Clinical Applications

A number of steps must be taken in order to prepare the technologies from this study for use in a clinical setting. Ideally, a system employing these networks would provide

an informative model of healthy and pathological brain activity for a given paradigm. Such a network would provide high quality classification for diagnosis, and analysis using some form of visualisation and attribution methods would localise regions of the brain presenting the activity of interest.

Future studies will be required in order to develop network architectures that improve classification accuracy using MEG data and provide more refined visualisations. For instance, a deeper network with more convolutional layers may allow for the construction of a hierarchy among the learned features. This could improve the overall performance of the network as well as the interpretability of the learned data representations.

Further research will also be required in order to frame the application of these networks within the context of current diagnostic practices. Application to a specific paradigm and a comparison to current standards of care will be required. For example, the use of simulated MEG data could provide the ground truth with which the performance of the developing technologies can be compared to the current standards. With careful consideration these technologies have the potential to enhance clinical diagnostics.

Chapter 5

Supplemental Material: Background

5.1 Magnetoencephalography (MEG)

MEG is a non-invasive neuroimaging technique that measures the magnetic fields generated by neuronal activity within the brain. Neurons are the base information processing units within the brain. They send signals called action potentials along their axons to other neurons in response to potential differences across their membranes. Action potentials occur when these potential differences, caused by the flow of sodium and potassium ions, reaches a specific threshold [38]. When action potentials are sent along an axon, these ions build up within the synaptic cleft between the axon of the sender and the dendrites of a receiver neuron. This build up and subsequent flow of ions across cell membranes generates post synaptic potentials (PSPs) [39]. It is these slowly fluctuating PSPs that generate magnetic fields that can be measured using MEG [15]. In order to generate a large enough magnetic field change to be measured using current MEG technology, synchrony among PSPs from 10,000's to 100,000's of neurons is required, which equates to about a 1-2 cm diameter patch of cortex.

For the purpose of localisation, the neuronal sources that generate MEG signals are modelled as infinitesimally small current elements, referred to as "current dipoles" [9]. Using a simplified spherically symmetric conductor to model the head, the magnetic field generated by a current dipole is given by [9]:

$$\mathbf{B}_z = \frac{\mu_0}{4\pi} \frac{\mathbf{Q} \times \hat{\boldsymbol{r}}}{r^2} \cdot \hat{\mathbf{z}}, \tag{5.1}$$

with permeability of free space μ_0 and the current source $\mathbf{Q}$ located with respect to some origin at $\mathbf{r}_Q$. The vector $\boldsymbol{r}$ represents the distance between a point $\mathbf{r}$ outside of the conductor and the current source $\mathbf{r}_Q$ such that:

$$\boldsymbol{r} = (\mathbf{r} - \mathbf{r}_Q); \quad r = |\mathbf{r} - \mathbf{r}_Q|; \quad \hat{\boldsymbol{r}} = \frac{\boldsymbol{r}}{r}. \tag{5.2}$$

The vector $\hat{\mathbf{z}}$ is the unit vector that points along the axis of the sensor. This expression shows that MEG sensors specifically measure the tangential component of the primary current. The measured magnetic field $\mathbf{B}_z$ falls off with the inverse square of the distance between the sensor and the source. Therefore, MEG sensors are more sensitive to sources near the surface of the cortex.

5.2 Supervised Machine Learning

Machine learning is the study of algorithms and statistical models that computer systems use in order to learn to perform tasks. These algorithms are designed such that models can be developed without the need for sets of instructions or explicit direction from an operator. Instead, model parameters are tuned in order to optimise a performance criterion using data or past experience [2]. Models developed using machine learning can be predictive, descriptive, or both.

The goal of any machine learning algorithm is to learn to perform some task. Machine learning tasks can be described in terms of how a system should process a record. Most machine learning approaches fall into two main categories: supervised and unsupervised learning [2, 3, 40]. These categories are determined by how an algorithm processes datasets during the training process. A dataset is a collection of records and each record is a collection of features which have been measured from some object or event. More specifically, a record can be represented as a vector $\mathbf{x} \in \mathbb{R}^n$ with a set of elements x_i as features. These features can be anything from raw sensor measurements, location coordinates, or specially engineered features such as averages, median values, or variances.

Models developed using unsupervised learning come from a class of algorithms that process datasets with the objective of learning useful properties or discovering unknown structure within the data. These types of algorithms can be used to construct what are called generative models. The objective of a generative model is to represent the entire probability distribution that generated the data [40]. Other types of unsupervised learning algorithms perform clustering in which a dataset is divided up into subsets of records that have similar or related properties.

Unlike unsupervised learning, supervised learning involves training models on a dataset that contains properties or structure that is known beforehand. Supervised

learning algorithms process datasets containing records such that each record is associated with a label or a target class. These associations between record and target are known *a priori* and guide the development of a model. The learning is said to be supervised in that a target is provided to the model by an instructor in order to guide the training process. Some examples of supervised machine learning tasks include regression, machine translation, and of particular importance to this project, classification. Classification is the task of specifying to which of c categories some input belongs.

Formally, a model that is trained to perform classification produces the functional mapping $f : \mathbb{R}^n \rightarrow \{1, ..., c\}$. During training, a model randomly observes input records $\mathbf{x}$ and the associated class value encoded as $y = f(\mathbf{x})$. For each record, the model learns to produce a class prediction $\hat{y}$ by estimating the probability $p(y|\mathbf{x})$ [3]. Within other classification schemes, such as the approach used within this project, the probability distribution $P(y = j|\mathbf{x})$ over j classes is produced by the model instead.

In order to guide training and provide a metric for performance, the classification accuracy of a model is calculated. The classification accuracy is simply the proportion of correct class predictions within a given dataset. For the purposes of training and testing models, datasets can be divided up into two or three subsets: training, validation, and testing. The training set is the largest subset of the data and is used to directly train a model. A validation set is used to calculate classification accuracy during training at different stages in order to gauge the progress of training. Although not always used, a separate test subset can also be used to perform offline testing after the completion of training. The classification accuracy is calculated using each dataset for differing purposes. The classification accuracy calculated over the test set serves as an unbiased measure of model performance because it was not used during the process of training. Although other training methodologies such as k-folds cross validation can use all of the data during training, the dataset within this project was split up into train, validate, and test subsets.

5.3 Deep Learning with Artificial Neural Networks

Machine learning systems learn the patterns that have been extracted from raw data. The performance of these systems heavily relies upon the representation of the input

data. Each piece of information included in the data representation is known as a feature and it is the selection of features that directly affects outcomes [3]. In order to achieve optimal performance, the collection of features within a representation must adequately capture or separate out the factors of variability within the data.

Traditional machine learning approaches require specialised feature engineering in order to develop an appropriate representation of the data. Raw data can contain too many dimensions (i.e. "the curse of dimensionality") for traditional models to be tractable. In cases such as these, feature engineering is required in order to reduce the dimensionality of the data while preserving the factors of variability within data. In other cases, the separation of classes may not be obvious in the current feature space without some additional transformation. This transformation (which could very well be non-linear) is required in order to make the problem tenable for a traditional machine learning approach. Effective feature engineering requires expert knowledge of the problem domain in order to understand the important patterns within the data so that an informed representation can be developed [3]. Feature engineering tasks can include data transformations, statistical approximations, or the calculation of other aggregate data. The performance of these systems can vary widely depending on the method of feature engineering. For instance, one might find that the median value over a set of measurements produces a better way to discriminate among a set of classes as compared to the mode or mean values.

In contrast to this, deep learning techniques avoid this problem by performing what is known as representation learning [3]. Deep learning builds up complicated representations of the data from simpler ones using artificial neural networks. These networks are inspired by functioning in the brain with the neuron forming the basic unit. A neuron is a mathematical construct that takes a set of inputs, multiplies them by a set of weights, sums them, and in some cases, a scalar known as a bias is added. The output of a neuron is then put through a non-linear transformation known as an activation function before being passed along as input into other neurons or as network output. The most commonly used activation function is known as the rectifier and neurons that use it are referred to as rectified linear units (ReLU). The operation they perform is simply expressed as follows:

$$f(x) = max(0, x) \tag{5.3}$$

where the value of an input x is returned if is greater than zero and zero is returned for all negative values. Networks are developed by constructing layers of neurons that can be interconnected in various configurations. These networks are called deep because model complexity is increased as a factor of the number of layers that make up the network.

In order to train a neural network to perform a supervised task, a function that expresses the difference between network output and target class is required. This function is called the loss function and a basic one is the mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|^2. \tag{5.4}$$

This function represents the average difference between the predicted class and target class for each record with n records. With an expression of the difference between network prediction and target label, training can be framed within the context of optimisation.

Neural network training typically employs some variant of the gradient descent algorithm that tunes the network parameters in order to find a loss function minimum. This process therefore minimises the difference between the predicted and target labels. If the loss function is expressed as a function of the set of network weights $E(\theta)$, then the standard gradient descent algorithm can written as:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta E(\theta), \tag{5.5}$$

where each future weight value is made up of the current weight, adjusted along the direction of the negative gradient. The magnitude of each step is dictated by a learning rate η. The gradient of the loss function with respect to the network weights ($\nabla_\theta E(\theta)$) is determined using the backpropagation algorithm. In simple terms, this iterative algorithm works in two main stages: first the loss function is evaluated by forward propagation of an input example, and secondly, the chain rule is used to propagate backwards calculating the activation function derivatives over the weights within each successive layer.

5.4 Convolutional Neural Networks (CNN)

The basic building blocks of a CNN typically include three stages: convolutions, nonlinear activation functions, and pooling [3]. The convolutions in these networks sparsely extract features from inputs while the activation functions facilitate the formation of nonlinear interactions between input and output. In some networks, pooling is introduced in order to reduce computational complexity and to allow for representations that are invariant to translations within the input. The idea being that it is more important to determine the presence of particular features and not the precise location of them.

For the purposes of classifying MEG data, a CNN using the first two basic building blocks was designed. Pooling was not incorporated because we required more specificity in the times that events occurred within each record. We further required feature maps to possess the same dimensionality as the input records. This was important because we wanted the feature maps produced by the first layer to contain one-to-one correspondence with input MEG records. This simplified investigation of feature maps and allowed for direct comparison to topological sensor location.

5.5 CNN Visualisation and Attribution Examples

For the purposes of demonstration, a VGG16 network [35] that was trained on the ImageNet 1000 [41] dataset was used. Figure 5.1 shows an image of a cat on the left along with a square grid of images that illustrate the visualisation and attribution techniques that were used in this project. The features maps that were examined in this study were generated by convolving input records with the first layer kernels of a trained network. Within the example shown in Figure 5.1, some of the kernels appeared to act as edge detectors. Features such as the shape of the cat's head and body can be clearly observed in the example image. Features such as these were propagated through the network in order to form an informed representation of the data. Activation maps are generated by tuning input space in order to maximise output with respect to a specific class. The resulting maps typically contain an assortment of features with varying colours, rotations, and scalings that the network is sensitive to when determining a particular class. In the case of the example shown

in the top-right corner of Figure 5.1, objects resembling cat eyes, noses, and tufts of fur are clearly visible. The bright coloured pixels within the example saliency map highlight the portions of the input image that contributed most to classification. In the case of the example, the network appeared to be most sensitive to the pixels surrounding the cat's head. A similar result is shown in the example occlusion map. The blue patches within the map indicate minimum values of the output as portions of the image were systematically set to zero. Similar to the saliency map, these blue minima correspond with the cat's head and also with a region around the tail.

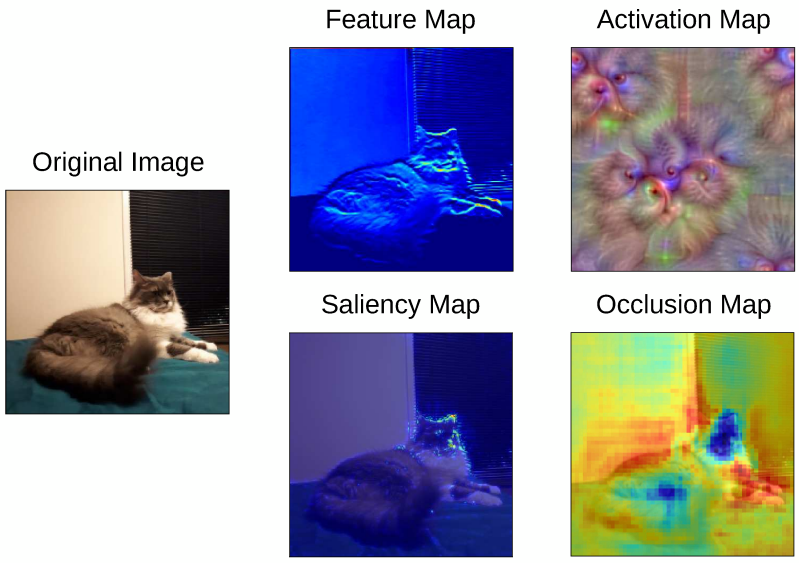

Figure 5.1: Examples of visualisation and attribution techniques used with a VGG16 network that was trained on the ImageNet dataset. This dataset contained over 14 million images over 1000 classes. The original image (left) was convolved with a first layer kernel to produce the feature map shown (top-centre). This particular kernel appeared to act as an edge detector. Activation maximisation over the "persian cat" class was used to generate the activation map shown (top-right). The activation map shows the network was sensitive to shapes that resemble cat eyes, cat noses, and tufts of cat fur. The saliency map (bottom-centre) was generated by calculating the gradients of the class score with respect to each pixel. The bright coloured pixels represent data points with the largest gradients. The pixels around the ears and other features represent the pixels that most contributed to classification. The occlusion map (bottom-right) was generated by recording the changes in the class score function as 2 px by 2 px regions were systematically set to zero. The blue regions within the occlusion map suggest that the pixels on and around the head and portions of the tail were important for performing classification.

Chapter 6

Supplemental Material: Additional Results

The figures in this section show results relating to the feature maps from all eight first layer kernels of a network trained using the sensor records. These plots are shown to demonstrate the consistency of the results across all of the feature maps. The active class plots were centred at the button press whereas the baseline class plots were centred at -1200 ms prior to button press. Figure 6.1 shows the grand-average over the active and baseline records from the test dataset as a basis for comparison.

Figures 6.2 and 6.3 show the channel averages over the eight sets of feature maps, highlighting kernel I to IV in figure 6.2 and kernel V to VIII in figure 6.3. The overall structure within the active class averages show peaks consistent with those shown in the record grand-average. The overall amplitude of these peaks tended to increase across all feature maps. As well, the amplitudes of peaks are different for different kernels, indicating changing patterns of sensitivity across kernels for different peaks in the data. For example, kernel V shows equal sensitivity to the peaks at -100 ms and 50 ms, while kernel VIII appears to be more sensitive to the earlier peak than the later. Changing sensitivity to different components is likely due to differences in how each kernel act as channel selectors.

It is also clear that the peak times were also shifted across some channels. For example, the first peak in the grand-average feature map for kernel I has a minimum at approximately -150 ms. The neuromagnetic peak is clearly at -100 ms based on Figure 6.1. Importantly, many of the kernels (including kernel I) act as "edge-detectors" in time, where a row of the kernel includes a strong negative and a strong positive separated by some period. An edge detector of this type would convert the start of a neuromagnetic peak (where there is a large rate of change) into a peak in the feature map.

Figure 6.4 (kernels I-IV) and Figure 6.5 (kernels V-VIII) show the histograms (right) that count the number of times peaks greater than or equal to 4σ occurred

within the sets of feature maps (across all records). As shown previously for one kernel, we observe a decrease in the number of counts following the button press, as compared to the pre-movement interval, in many kernels. The difference in counts between the intervals -400 to -300 ms before button press and 152 to 252 ms after button press was calculated for each. These differences were mapped to sensor location and shown in the topographic plots on the left. Similar to what was previously shown in Section 3.3, the decrease in counts tended to occur around the centrally located sensors. The decrease in counts and the central location of these decreases suggests that many kernels were sensitive to beta burst activity.

As expected by the variable compositions of the kernels in the first layer, there is a changing sensitivity to different components of the MEG data across kernels. Due to the non-ideal spatial ordering of the sensors into records, this variability is somewhat difficult to interpret. Clearly, a representation of the data that maintains the spatial relationship of the sensors would be preferred for visualisation purposes. In the next chapter, we explore a CNN applied to three-dimensional records, wherein two dimensions account for a flattened spatial representation of the sensors and the third dimension accounts for time.

6.1 Sensor Data Statistics

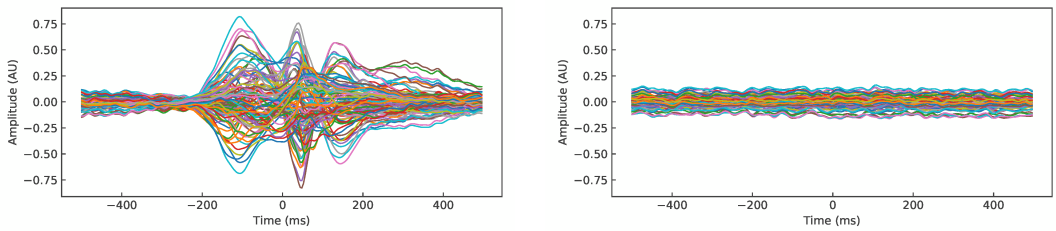

Figure 6.1: Channel averages calculated over the active records (left) and baseline records (right) from the test dataset. Each trace represents a single channel. Active records were centred at the button press at $t = 0$ ms and baseline records were centred at $t = -1200$ ms prior to button press.

6.2 Feature Map Statistics

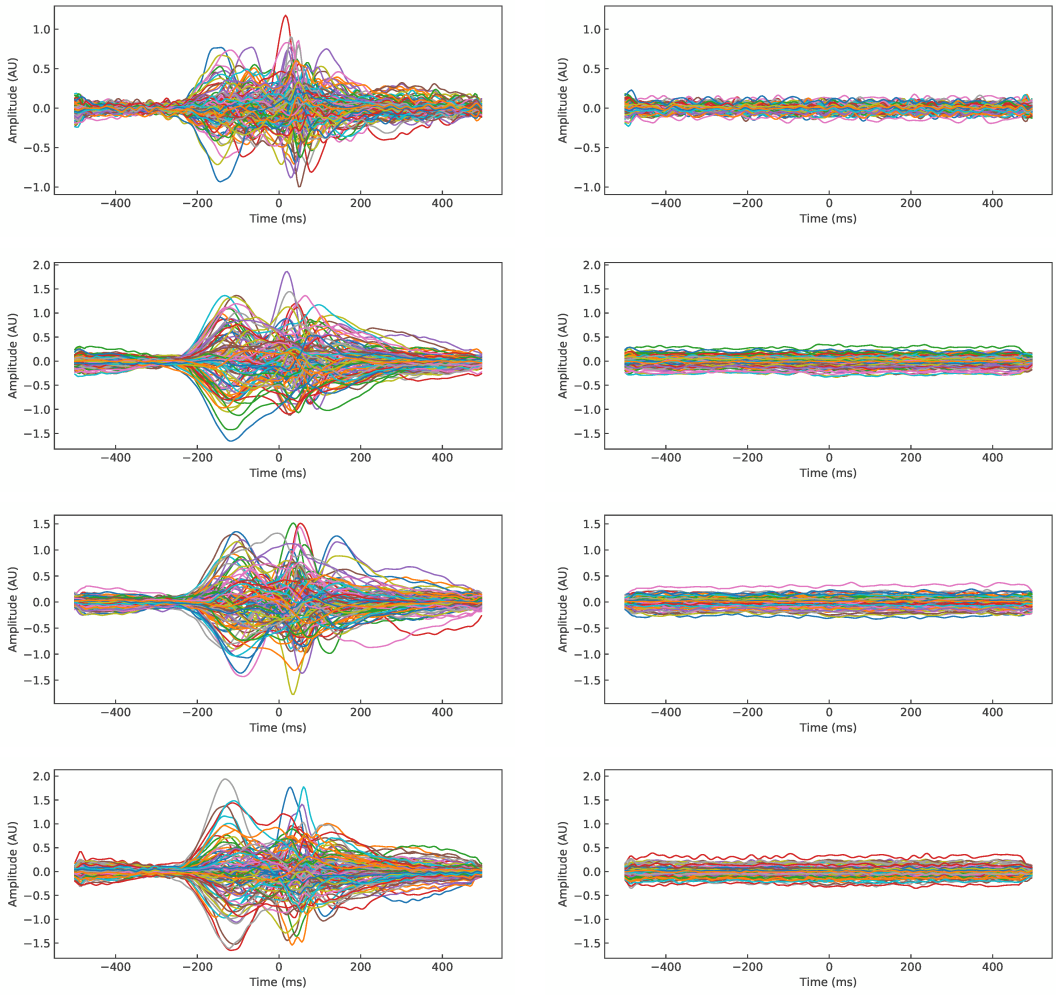

Figure 6.2: Channel averages calculated over the feature maps produced by kernels I-IV on the active (left) and baseline (right) records. The averages over the active class show structure consistent with the grand-average but with an increase in amplitude and a shift in some of the peak times.

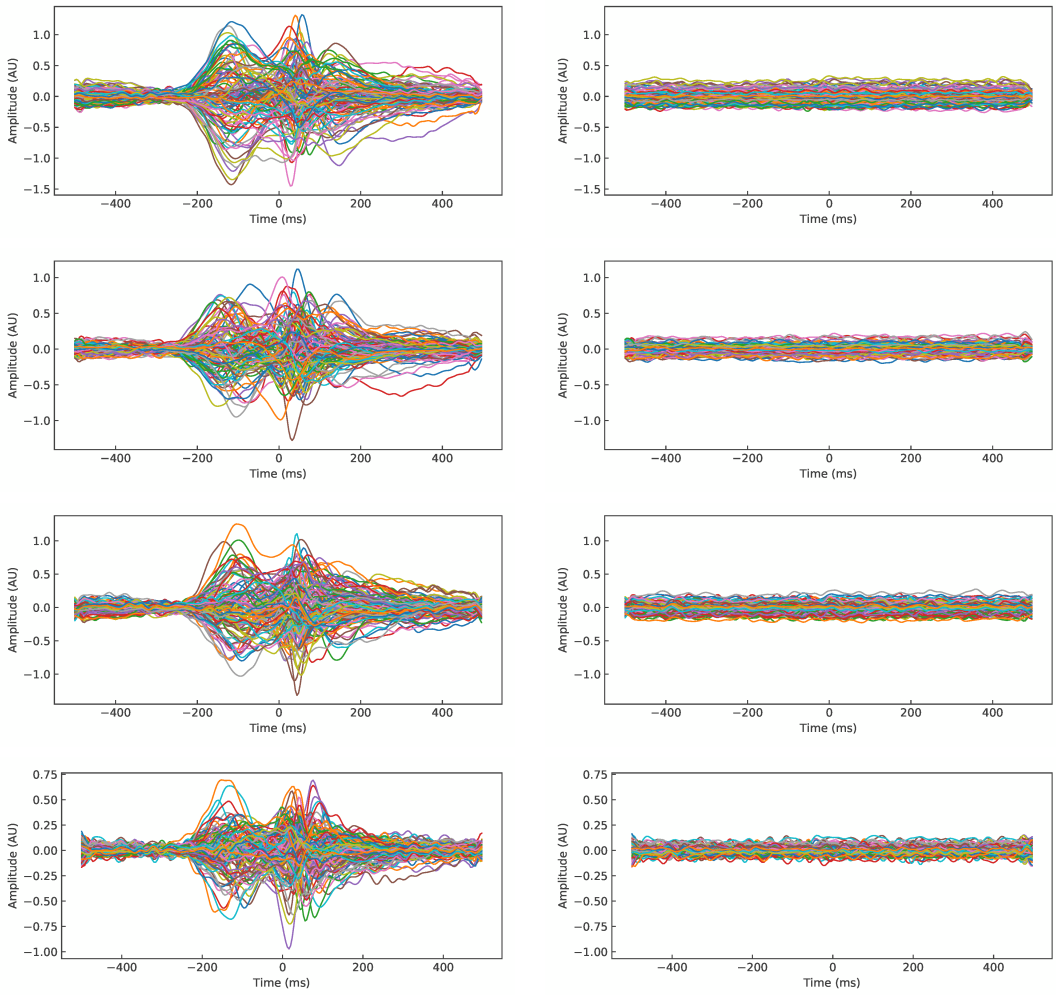

Figure 6.3: The same plots as shown in Figure 6.2 but over the feature maps constructed from kernels V-VII.

6.3 Feature Map Peak Analysis

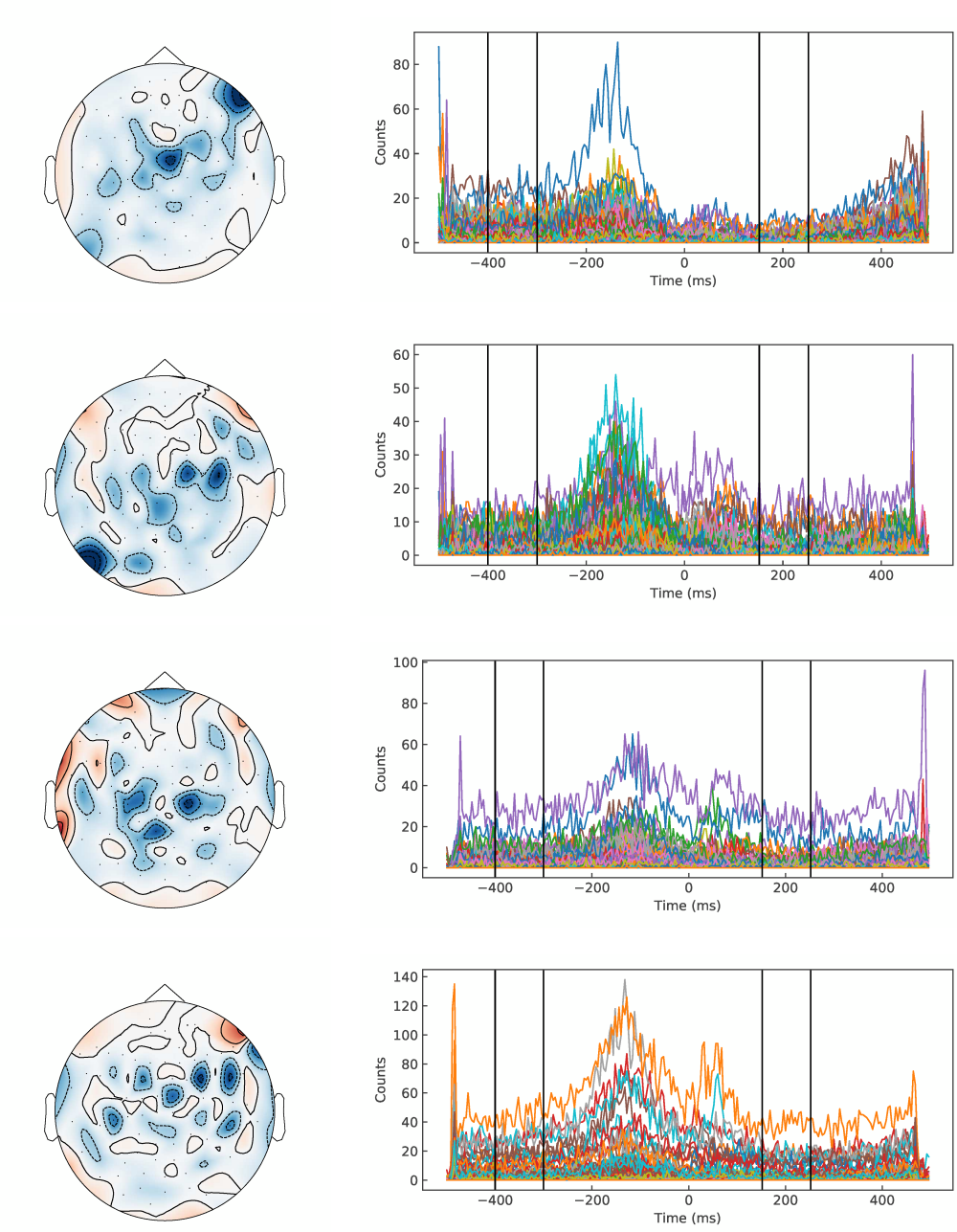

Figure 6.4: Topographic plots (left) show the difference in counts between −400 ms to −300 ms prior to button press and 152 ms to 252 ms after button press within the feature map histograms. These histograms counted the number of times a peak $\geq 4\sigma$ occurred within each channel and time. These decreases tended to occur under the central sensors which is consistent with beta suppression. Histograms and topographic plots shown were calculated over the feature maps from kernels I-IV.

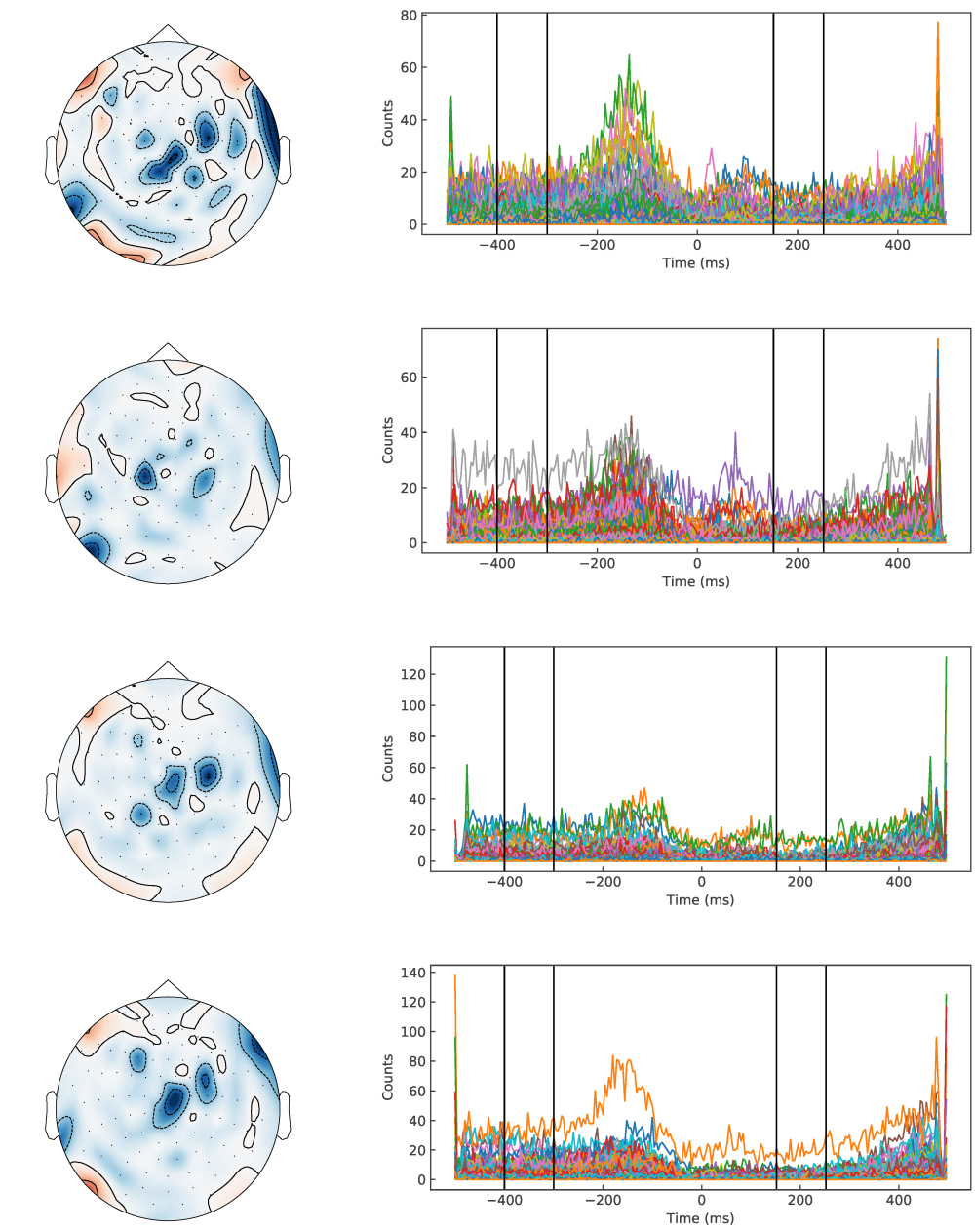

Figure 6.5: The same plots as shown in Figure 6.4 but for kernels V-VIII.

Chapter 7

Supplemental Material: 3D Sensor Representation

7.1 3D Record Transformation

Expression of the spatial relationships among sensors were limited within the two-dimensional records that were used throughout this study. Within each record, sensors were represented by rows with the order determined by the manufacturer sensor naming convention. In this way, sensors were grouped together in related regions but discontinuities existed among them. Visualisation showed possible sensitivity to these discontinuities manifested as the edge artefacts seen in Figure 3.11.

In order to overcome some of these limitations and to better utilise the ability of a CNN to extract spatial structure from data, a three-dimensional representation was constructed and tested. Using the pre-processed records from the dataset described in Section 2.2.2, channel values were mapped back to associated sensor locations. The intermediate values between sensors in the grid were estimated using linear interpolation. The new records contained three dimensions for 42 x positions, 42 y positions, and 250 time samples. Figure 7.1 shows the arrangement of sensors within the left plot along with an example of a time sample with interpolated values on the right. All values outside the bounds of the MEG helmet were set to zero. The records within the new representation therefore were three-dimensional tensors, which contained two-dimensional topographies over time.

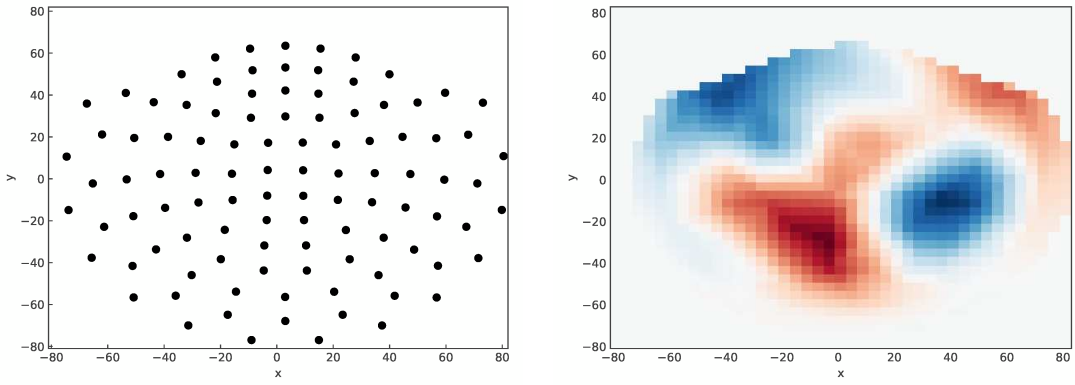

Figure 7.1: The 102 magnetometer locations were represented on a two-dimensional grid using Cartesian coordinates (left plot). For each record, the sensor values were mapped to these locations for each time sample. The intermediate values among the sensors were estimated using linear interpolation (right plot).

7.2 3D CNN Architecture

The network architecture described in Section 2.1 was minimally altered in order to use the new data representation. While using the same basic elements as the original design, this new network architecture was adapted in order to account for the additional dimension in the data. A schematic of this adapted network design is shown in Figure 7.2. Specifically, the two convolutional layers were modified to use 3D kernels to perform the convolution operations. The convolutions were performed with zero padding to maintain upstream dimensionality and resulted in 3D feature maps being generated and propagated through the network. Like the 2D sensor and source networks, the first convolutional layer kernel contained 16 elements along the temporal dimension to account for the slow temporal dynamics of the MEG signals. Spatially the kernel contained eight elements for each dimension. This number of elements was chosen for consistency with the original design and based on the expected spatial resolution of the topographic representations. The second convolutional layer employed a kernel with three elements within each dimension.

Due to the increase in the number of dimensions there was an associated increase in computational complexity and memory requirements. In order to make computation more manageable (reduce memory usage and overall training time), a maximum pooling layer was added that reduced the number of elements within each dimension

by half. To accomplish this a (2 x 2 x 2) moving window was used to select the maximum values as it was swept over adjacent elements within each 3D feature map passed in from the second convolutional layer.

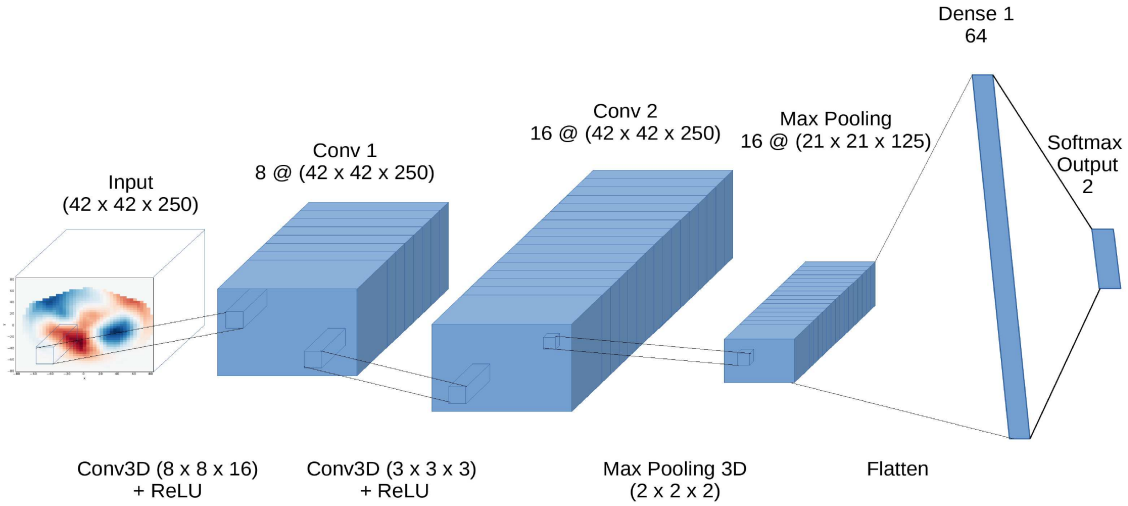

Figure 7.2: A schematic of the network used to train on 3D sensor records. The first two layers performed 3D convolutions in order to generate feature maps with the same dimensions. A maximum pooling layer was placed between the second convolutional layer and the dense layer in order to reduce computational complexity and lower memory requirements. The output from the maximum pooling layer was then flattened and passed as input to the dense layer. The last layer of the network was a softmax layer containing two neurons whose output represented the probability distributions across the active and baseline classes for a given record.

7.3 Network Performance

In order to gauge the performance of this network design, an ensemble of 10 networks was trained using records with this new data representation for 25 epochs. As described in Section 2.2.4, this dataset was also randomly split into training, validation, and testing subsets. The training subset was used to train the networks, the validation subset was used for online testing, and the test subset was used offline to test the fully trained networks.

The average classification accuracy along with the average cross entropy values are shown in Figure 7.3. The metrics calculated over the training subset are represented as blue circles and the validation subset values are represented by orange squares.

The performance of the ensemble was comparable to that of the ensemble trained on the 2D sensor records. The largest average validation accuracy reached 0.964 ± 0.007 compared to 0.960 ± 0.002 achieved by the 2D networks. The 3D sensor networks attained the largest validation accuracy (and smallest associated cross entropy) after 10 training epochs. Similar to the behaviour observed with 2D network training, the validation accuracy did not improve after further training. The average cross entropy values (and ensemble standard deviation) began to increase with additional training suggesting that the networks started to over-fit to the data. The classification accuracy calculated on the test subset was also comparable to the 2D value with the best ensemble accuracy reaching 0.0962 ± 0.002 (3D) versus 0.974 ± 0.001 (2D).

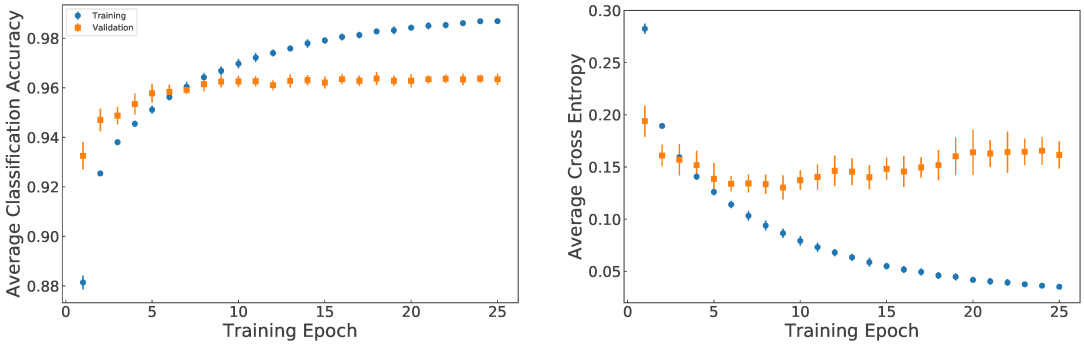

Figure 7.3: The performance an ensemble of 10 CNN trained over 25 epochs as measured by the average ensemble classification accuracy (left) and the average cross entropy (right). These networks were trained using the dataset that contained sensor records in a 3D representation. The blue circles indicate the metrics calculated over the training dataset and the orange squares represent the values calculated over the validation set. After 10 epochs the average validation accuracy reached a maximum value of 0.962 ± 0.002 before plateauing. The loss function values increased after this point along with the variance among the ensemble values, indicating that no further performance improvements were made.

Training networks using a 3D representation of the data may provide advantages such as more readily interpretable feature maps and perhaps visualisation and attribution maps that do not contain edge artefacts. Indeed, we would expect that the visualisation would be more informative about the underlying MEG data that is most important for classification, as compared to the 2-D model. Given the parity in performance between the 2-D and 3-D models, We would recommend pursuing the 3-D model in future studies. This analysis was outside of the scope of this study.

Along with these potential advantages, using this representation comes with additional costs in terms of computing and storage resources that affect all facets of the training, testing, and analysis pipeline. In addition to the standard MEG data preprocessing, and sensor record preparation, additional time was needed to interpolate values between sensors at each time point. This required an additional 1.3 hours of processing time to perform across all 2D sensor records. The additional dimension also meant an increase in data storage with a file size of 146 GB compared to the 15 GB file required to store the 2D sensor records. Furthermore, the amount of time required to train these networks was also increased. For comparison, a 3D network using two graphics processing units (GPUs) required 17.7 ± 0.2 hours on average in order to train for 25 epochs. A 2D network using a single GPU required 45.0 ± 0.6 minutes on average to train for the same number of epochs.

Chapter 8

Conclusion

Using an appropriately large dataset, deep learning models such as convolutional neural networks (CNNs) can be developed to be applied to neuroimaging. In this study we designed a CNN to classify between active and baseline intervals within a large, open-access MEG dataset. We experimented with networks using different representations of the data. These representations included 2D sensor records, source-localised records, and 3D sensor records. The networks trained using source-estimated records did not perform well, probably due to the loss of dimensionality during the localisation process. The networks trained using the sensor records provided the highest quality classification. The 3D representation of the sensor records provided networks with comparable results to the 2D counterparts but with increased overhead. Due to the additional dimension there was an associated increase in computational complexity when training the model along with increased memory and file size requirements.

Visualisation and attribution techniques were used to analyse a fully trained network. Feature maps generated by convolving input with a trained kernel provided a means of investigating the features the network extracted from the data in order to form a representation of the data. The visualisation and attribution techniques found within the field of computer vision provided a means of identifying the portions of the data that informed classification. It was shown that the important data features identified by these techniques could be related back to the activity within brain regions associated with cued button pressing. These results suggest that these types of analyses are not only useful for investigating network training and performance, but have the potential to be used in a research capacity. For instance, regions of activity involved in tasks that are not well understood could be investigated through visualisation and attribution mapping.

There are a number of future directions that can explored in order to expand upon this study. The most obvious next step would be to extend the current network

implementation to perform multi-label classification using the demographic data provided by Cam-CAN along with the MEG data. Previous studies have established significant age-related differences in these data [22] and a study could investigate whether or not a CNN is sensitive to this variability. The network design could be further developed using deeper networks or networks with specialised architectures in order to improve upon the performance as well as data representation constructed by the network. Another extension of this work could include classifying MEG data that was recorded during the performance of different tasks. The ability to classify among different tasks would help to explore the general applicability of CNNs to MEG data as well as the visualisation and attribution methods demonstrated here. With generalisability and further improvements to architecture and analysis techniques, these networks could be developed for use in a clinical setting for augmenting current diagnostic techniques.

Bibliography

[1] E. Klang, "Deep learning and medical imaging," *Journal of Thoracic Disease*, vol. 10, no. 3, pp. 1325–1328, 2018.

[2] E. Alpaydn, *Introduction to Machine Learning*. Cambridge, MA: MIT Press, 3rd ed., 2014.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[4] Y. LeCun, "Generalization and network design strategies," in *Connectionism in Perspective*, pp. 143–155, Elsevier Science, 1989.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, pp. 1097–1105, Curran Associates, Inc., jan 2012.

[6] E. A. Yoshiko Ariji , Motoki Fukuda , Yoshitaka Kise , Michihito Nozawa , Yudai Yanashita , Hiroshi Fujita , Akitoshi Katsumata, "Contrast-enhanced CT image assessment of cervical lymph node metastasis in oral cancer patients using a deep learning system of artificial intelligence Yoshiko," *Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology*, vol. 127, no. 5, pp. 458–463, 2018.

[7] T. A. Retson, A. H. Besser, S. Sall, D. Golden, and A. Hsiao, "Machine learning and deep neural networks in thoracic and cardiovascular imaging," *Journal of Thoracic Imaging*, vol. 34, no. 3, pp. 192–201, 2019.

[8] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122–1131.e9, 2018.

[9] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography theory, instrumentation, and applications to noninvasive studies of the working human brain," *Reviews of Modern Physics*, vol. 65, no. 2, pp. 413–497, 1993.

[10] S. C.J., T. P., V. D. E., v. S. E.C.W., H. A., C. J. Stam, P. Tewarie, E. Van Dellen, E. C. W. van Straaten, A. Hillebrand, and P. Van Mieghem, "The trees

and the forest: Characterization of complex brain networks with minimum spanning trees.," *International Journal of Psychophysiology*, vol. 92, no. 3, pp. 129–138, 2014.

[11] M. M. Schoonheim, J. J. G. Geurts, D. Landi, L. Douw, M. L. van der Meer, H. Vrenken, C. H. Polman, F. Barkhof, and C. J. Stam, "Functional connectivity changes in multiple sclerosis patients: A graph analytical study of MEG resting state data," *Human Brain Mapping*, vol. 34, no. 1, pp. 52–61, 2013.

[12] D. S. Bassett and E. T. Bullmore, "Human brain networks in health and disease," *Current Opinion in Neurology*, vol. 22, no. 4, pp. 340–347, 2009.

[13] J. Vrba and S. E. Robinson, "Signal processing in magnetoencephalography," *Methods*, vol. 25, no. 2, pp. 249–271, 2001.

[14] H. E. Kirsch, S. E. Robinson, M. Mantle, and S. Nagarajan, "Automated localization of magnetoencephalographic interictal spikes by adaptive spatial filtering," *Clinical Neurophysiology*, vol. 117, no. 10, pp. 2264–2271, 2006.

[15] P. Hansen, M. Kringelbach, and R. Salmelin, eds., *MEG: An introduction to methods*. New York, NY: Oxford University Press, 2010.

[16] M. A. Shafto, L. K. Tyler, M. Dixon, J. R. Taylor, J. B. Rowe, R. Cusack, A. J. Calder, W. D. Marslen-Wilson, J. Duncan, T. Dalgleish, R. N. Henson, C. Brayne, E. Bullmore, K. Campbell, T. Cheung, S. Davis, L. Geerligs, R. Kievit, A. McCarrey, D. Price, D. Samu, M. Treder, K. Tsvetanov, N. Williams, L. Bates, T. Emery, S. Erzinçlioglu, A. Gadie, S. Gerbase, S. Georgieva, C. Hanley, B. Parkin, D. Troy, J. Allen, G. Amery, L. Amunts, A. Barcroft, A. Castle, C. Dias, J. Dowrick, M. Fair, H. Fisher, A. Goulding, A. Grewal, G. Hale, A. Hilton, F. Johnson, P. Johnston, T. Kavanagh-Williamson, M. Kwasniewska, A. McMinn, K. Norman, J. Penrose, F. Roby, D. Rowland, J. Sargeant, M. Squire, B. Stevens, A. Stoddart, C. Stone, T. Thompson, O. Yazlik, D. Barnes, J. Hillman, J. Mitchell, L. Villis, and F. E. Matthews, "The Cambridge Centre for Ageing and Neuroscience (Cam-CAN) study protocol: A cross-sectional, lifespan, multidisciplinary examination of healthy cognitive ageing," *BMC Neurology*, vol. 14, no. 1, 2014.

[17] R. Sharma, E. W. Pang, I. Mohamed, B. Chu, A. Hunjan, A. Ochi, S. Holowka, W. Gaetz, S. Chuang, O. C. Snead, and H. Otsubo, "Magnetoencephalography in children: Routine clinical protocol for intractable epilepsy at the hospital for sick children," *International Congress Series*, vol. 1300, pp. 685–688, 2007.

[18] G. Pfurtscheller and F. H. Lopes, "Event-related EEG / MEG synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, pp. 1842–1857, 1999.

[19] D. Cheyne, L. Bakhtazad, and W. Gaetz, "Spatiotemporal mapping of cortical activity accompanying voluntary movements using an event-related beamforming approach," *Human Brain Mapping*, vol. 27, no. 3, pp. 213–229, 2006.

[20] M. T. Jurkiewicz, W. C. Gaetz, A. C. Bostan, and D. Cheyne, "Post-movement beta rebound is generated in motor cortex: Evidence from neuromagnetic recordings," *NeuroImage*, vol. 32, no. 3, pp. 1281–1289, 2006.

[21] D. Price, L. K. Tyler, R. Neto Henriques, K. L. Campbell, N. Williams, M. Treder, J. R. Taylor, C. Brayne, E. T. Bullmore, A. C. Calder, R. Cusack, T. Dalgleish, J. Duncan, F. E. Matthews, W. D. Marslen-Wilson, J. B. Rowe, M. A. Shafto, T. Cheung, S. Davis, L. Geerligs, R. Kievit, A. McCarrey, A. Mustafa, D. Samu, K. A. Tsvetanov, J. van Belle, L. Bates, T. Emery, S. Erzinglioglu, A. Gadie, S. Gerbase, S. Georgieva, C. Hanley, B. Parkin, D. Troy, T. Auer, M. Correia, L. Gao, E. Green, J. Allen, G. Amery, L. Amunts, A. Barcroft, A. Castle, C. Dias, J. Dowrick, M. Fair, H. Fisher, A. Goulding, A. Grewal, G. Hale, A. Hilton, F. Johnson, P. Johnston, T. Kavanagh-Williamson, M. Kwasniewska, A. McMinn, K. Norman, J. Penrose, F. Roby, D. Rowland, J. Sargeant, M. Squire, B. Stevens, A. Stoddart, C. Stone, T. Thompson, O. Yazlik, D. Barnes, M. Dixon, J. Hillman, J. Mitchell, L. Villis, and R. N. A. Henson, "Age-related delay in visual and auditory evoked responses is mediated by white- and grey-matter differences," *Nature Communications*, vol. 8, no. May 2016, p. 15671, 2017.

[22] T. Bardouille and L. Bailey, "Evidence for age-related changes in sensorimotor neuromagnetic responses during cued button pressing in a large open-access dataset," *NeuroImage*, vol. 193, no. March, pp. 25–34, 2019.

[23] S. Taulu and R. Hari, "Removal of magnetoencephalographic artifacts with temporal signal-space separation: demonstration with single-trial auditory-evoked responses," *Human Brain Mapping*, vol. 30, no. 5, pp. 1524–1534, 2009.

[24] P. Comon, "Independent component analysis, A new concept?," *Signal Processing*, 1994.

[25] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, 2000.

[26] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis," *NeuroImage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[27] J. Dammers, M. Schiek, F. Boers, C. Silex, M. Zvyagintsev, U. Pietrzyk, and K. Mathiak, "Integration of amplitude and phase statistics for complete artifact removal in independent components of neuromagnetic recordings," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 10, pp. 2353–2362, 2008.

[28] A. M. Dale, A. K. Liu, B. R. Fischl, R. L. Buckner, J. W. Belliveau, J. D. Lewine, E. Halgren, and S. Louis, "Neurotechnique Mapping : Combining fMRI and MEG for High-Resolution Imaging of Cortical Activity," *Neuron*, vol. 26, pp. 55–67, 2000.

[29] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany, "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest.," *NeuroImage*, vol. 31, no. 3, pp. 968–80, 2006.

[30] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning*, (Lille, France), 2015.

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Jmlr*, vol. 12, pp. 1–40, 2011.

[32] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014.

[33] J. Yosinki, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding Neural Networks Through Deep Visualization," in *Deep Learning Workshop*, (Lille, France), 31st International Conference on Machine Learning, 2015.

[34] M. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV*, pp. 818–833, Spinger International Publishing, 2014.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[36] B. D. V. Veen, W. V. Drongelen, M. Yuchtman, and A. Suzuki, "Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. IEEE Transactions on," *Biomedical Engineering*, vol. 44, no. 9, pp. 867–880, 1997.

[37] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[38] D. Purves, G. J. Augustine, and D. Fitzpatrick, *Neuroscience*. Sunderland, MA: Sinauer Associates, INC, 5th ed., 2012.

[39] M. F. Bear, B. W. Connors, and M. A. Paradiso, *Neuroscience: Exploring the brain*. Philidelphia, PA: Wolters Kluwer, 4th ed., 2016.

[40] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.