A Convolutional Network on EEG Spectrograms for Sleep Staging

Evgeny Naumov

Master of Science

Computer Science

McGill University

Montreal,Quebec

April 2017

This thesis is submitted to McGill University in partial fulfilment of the requirements of the degree of Master of Science

DEDICATION

To Sophia Davis.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Joelle Pineau, for her excellent instruction and clear guidance. I would like to thank Andrew Lim for the insights on sleep staging he provided, and to the other members of Sunnybrook Hospital with whom I collaborated, and who provided for me the excellent collection of EEG recordings which made my work possible.

ABSTRACT

Sleep is a very important, though as yet poorly understood, aspect of human physiology. In humans, sleep is subdivided into a number of physiologically distinct sleep stages. The accurate determination and labelling of sleep stages from EEG recordings is fundamental in sleep research and clinical practise. This thesis proposes a new technique of using a convolutional neural architecture with spectrograms of EEG data as input to perform sleep stage classification. This technique does not rely on expert features or informed preprocessing of EEG data. This architecture is shown to deliver competitive results when trained on a data set of 120 patients' overnight EEG recordings using strict cross-patient validation. Additionally, the use of bagging is validated as a reliable measure of uncertainty for the architecture's output.

ABRÉGÉ

Le sommeil est un aspect très important, bien que encore peu compris, de la physiologie humaine. Chez l'homme, le sommeil est subdivisé en plusieurs stades de sommeil physiologiquement distincts. La détermination et l'étiquetage précis des stades du sommeil à partir des enregistrements d'EEG sont fondamentaux dans la recherche et la pratique clinique du sommeil. Cette thèse propose une nouvelle architecture neurale profonde pour l'étiquetage automatisé des stades du sommeil à partir de l'EEG. L'architecture utilise une nouvelle technique, des réseaux neuronaux convolutionnels avec des spectrogrammes de données EEG comme entrée, qui ne dépend pas de caractéristiques d'experts ni de prétraitement informé des données EEG. Ce modèle est démontré à fournir des résultats compétitifs quand entrainé sur un ensemble de données d'enregistrements EEG de nuit de 120 patients, en utilisant une validation rigoureuse des patients. En outre, l'utilisation du "bagging" est validée comme une mesure fiable de l'incertitude pour l'architecture.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Sleep

Sleep is a very important, though as yet poorly understood, aspect of human physiology. It is a strict necessity for every person, and insufficient or disturbed sleep can significantly reduce quality of life and productivity [60, 64]. Human sleep can be affected by a number of disorders like sleep apnea, insomnia, periodic limb movement, and many others. These disorders impose a steep financial and human cost on society [29]. For these reasons, the study of sleep is an important area of research.

One facet of sleep science is the study and measurement of a patient's **sleep architecture**. Sleep architecture collectively refers to the details of the duration and succession of the **sleep stages** into which human sleep is divided. Sleep stages are distinct, physiologically distinguishable states of sleep. As a person sleeps, he or she transitions between different sleep stages. Each sleep stage is defined by a set of measurable physiological markers; over the course of sleep, these stages generally follow a predictable pattern of transitions. However, the order of transitions between these stages and the duration of each stage in a particular patient is subject to individual variation, and is potentially affected by the presence of sleep disorders.

Many disorders and physiological processes are associated with a distinct stage of sleep. An example of such a disorder is REM Sleep Behaviour Disorder (RBD), which occurs during the REM (rapid eye movement) stage of sleep. RBD causes sufferers to act out their dreams [11] verbally and physically, and is a common cause of talking and moving about in one's sleep. Night terrors and sleepwalking are two other disorders linked to a particular stage of sleep. Patients with night terrors experience symptoms similar to those which might occur in a panic attack. Despite apparent agitation and complex movements, someone experiencing night terrors is unresponsive to external stimuli. Sleepwalking, also known as somnambulism, involves individuals getting out of bed and possibly performing complex behaviours while asleep. During this activity, they are unresponsive to external stimuli, though their eyes may be open. Sleepwalkers have no recollection of the event afterward [3]. Both night terrors and sleepwalking are believed to be caused by sudden arousals from the slow-wave stages of sleep [13, 57]. Even in the absence of disorders, a host of physiological mechanisms in sleeping individuals is intimately linked with the stage of sleep they are experiencing. Patients' breathing patterns [61] and muscle movements [32] depend on their current sleep stage. The pruning of unnecessary neuronal connections has been found to occur specifically during REM sleep [40]. The above examples are far from an exhaustive list. The different stages of sleep and the details of the sleep architecture are associated with many more distinct neurobiological mechanisms and specific physiological and health outcomes [50, 59, 6, 62, 54, 53, 51, 42, 48]. In order to gain insight into the disorders and processes linked with distinct stages of sleep, and to investigate

the fundamental nature of sleep itself, the identification and measurement of a patient's sleep stages is an important and frequently performed clinical procedure. Having an accurate record of a patient's sleep stages over the course of the night can also help medical professional reach or confirm a diagnosis of a sleep disorder.

1.2 Sleep Staging

The act of creating a record of a patient's transitions between sleep stages is called **sleep staging** (or sleep scoring), and a record of such stages - often coreferenced with other physiological measurements - is called a **hypnogram**. The exact classification of sleep stages, and even their total number, has been subject to scientific debate and revision. Currently, there exist two principal medical standards which enumerate and define the stages of sleep in terms of definite physiological markers. The older of the two is referred to as the **RK** (Rechtschaffen and Kales) standard [49], which defines six distinct stages of sleep: WAKE, REM, S1, S2, S3 and S4, listed in rough order of how difficult it is to rouse a person in that stage. In the literature, stages S3 and S4 are frequently treated as a single SWS (slow-wave sleep) stage owing to their physiological proximity [31]. The RK standard has been superseded by the more recent **AASM** (American Academy of Sleep Medicine) standard [31], which defines only five distinct stages, WAKE, REM, N1, N2, N3. The exact definitions of each stage in terms of physiological markers is slightly altered from those of the RK standard, meaning that results with respect to one set of rules are not compatible with those based on the other. The method developed in this work classifies sleep stages according to the AASM standard.

A common collection of measurements taken expressly for the purpose of identifying a patient's stages of sleep over time is called a **polysomnograph**, or PSG, which consists of a number of component signals. PSGs frequently include [22], but are not limited to:

- an **electromyogram** (EMG), a recording of body movements
- an **electrooculogram** (EOG), a recording of eye movements
- an **electrocardiogram** (ECG), a recording of heart rate
- an **electroencephalogram** (EEG), a recording of brain activity

Not all of these recordings may be present in a PSG. The work presented in this thesis focuses on learning sleep stages from EEG in isolation. The EEG is present in almost all PSGs in the literature, so a method able to perform sleep staging from EEG alone will have the widest applicability.

1.3 EEG

EEG is widely used in the investigation of brain activity. It is recorded via electrodes places on the scalp, and is considered a non-invasive procedure. It is relatively cheap to carry out, and can be done in an outpatient setting, where patients are monitored by EEG equipment in their homes. Outpatient EEG is also widely used in seizure detection, and is referred to as ambulatory EEG.

In part because EEG is so noninvasive, it is limited in the kinds of brain activity it can measure. Only surface brain activity is captured, and it is captured at a low spatial resolution. Nevertheless, sleep stages can be inferred from EEG information alone [45]. As a rule, EEG recordings consist of multiple EEG **channels**, each of which corresponds to a particular electrode location on the

scalp. These electrodes are placed in standard positions, most commonly in accordance with the "10-20" system [33]. The output of each EEG channel is a time series sampled at a high frequency (256 Hz is common). For the purposes of sleep staging, in standard clinical practice the EEG recording is divided into non-overlapping 30-second segments, called **pages**. Each page contains all EEG channels. Each page is then classified independently into one of the five AASM stages by visual evaluation of the EEG and auxiliary signals (if present) by a trained technician. Where a page contains the markers of more than one sleep stage, it is classified according to which stage's markers cover most of the page [31].

1.4 Problem Statement and Objectives

Normally, classification of EEG into sleep stages requires a trained medical practitioner, and is quite time consuming. The objective of this work is to devise **a fully automated machine learning technique for generating a hypnogram from EEG**, meeting a number of requirements crucial in clinical practise.

The method should be robust. It should be tolerant to data noise and variation across patients, technicians, and recording equipment. The algorithm should perform well on data sourced from different technicians and from patients with abnormal sleep patterns. Critically, the **algorithm should work well on data from new patients never seen in training**. In service of this aim, the sleep staging method should not rely on expert preprocessing or feature extraction, and should be instead **learned from human labels and raw EEG signal only** to the maximum extent possible. The method should allow retraining on

new sources of data or patients with particular disorders without adjusting the architecture and hyperparameters. Furthermore, the method should be able to adapt in a way an algorithm with expert features might not. These concepts are elaborated upon in later sections, with examples of expert features are given in Chapter 3. Finally, acknowledging that in medical contexts it is especially important for machine learning algorithms to provide a good estimate of their confidence in their outputs, the method should provide a reliable **confidence estimate** to its user. EEG pages with a low level of confidence can be flagged for manual review by a technician, while high-confidence pages can be accepted.

1.5 Contributions

The task of automatic sleep staging from EEG data has been tackled by many other authors in the literature. A number of diverse machine learning methods have been developed, which are described in detail in Chapter 3. However, none of the methods surveyed simultaneously meet all of the objectives given in the problem statement.

To that end, this thesis presents a deep learning architecture based on a novel use of convolutional neural networks with EEG spectrogram data as input. The key contribution is to **treat the spectrograms as images, and to treat EEG channels as image colour channels**, thereby translating the sleep staging problem into an image classification problem. This use of the spectrogram as input to a convolutional neural networks for sleep stage classification is believed to be a novelty of this work.

The objective of the architecture is to automatically generate reliable hypnograms from EEG. The classifier is a "black box" and requires no tuning or manual preprocessing steps. Once trained, the classifier can operate on variable-channel EEG input of various frequencies. Internally, it first transforms the EEG into two parallel spectrogram representations, one for the low-frequency components of the signal, the other for the high-frequency components. These spectrograms are passed through a convolutional neural network, whose outputs are then concatenated and passed through a fully connected neural network, which outputs its prediction for the sleep stage of the EEG page being processed. This architecture is experimentally validated on a dataset of 110 full-night recordings collected from patients at Sunnybrook Hospital using cross-patient training and cross-validation. Cross-patient validation permits the assessment of the algorithm's ability to generalize. As with any deep architecture, the performance of the architecture presented will degrade when given inputs from a different source than the one which produced the data it was trained on. Examples of different sources include EEG from patients of a different demographic or medical status than the training group, or sourced with different equipment or methodology. However, adaptation to new data sources requires no retuning or readjustment beyond "black box" retraining on data examples from the new source.

The objective of obtaining a confidence estimate for the classifier's output, which is traditionally difficult for neural architectures, is addressed by use of the bagging ensemble technique. Under the bagging scheme, multiple classifiers are trained independently on subsets of the data. To predict the label of an unseen

EEG page, these sub-classifiers' output is added (they are said to **vote**), and the label with the highest total probability is chosen. Bagging is commonly used for improving the accuracy of a classifier's predictions, which is a function it also serves in the presented architecture. This thesis adapts bagging as tool for estimating the classifier's confidence in its output. This confidence estimate is, as intuition might dictate, given by the agreement rate between the bagging sub-classifiers. This thesis rigorously quantifies and validates the use of sub-classifier agreement as a reliable measure of uncertainty for the sleep staging architecture presented.

CHAPTER 2
Technical Background

The goal of this chapter is to describe the basics of neural networks, in particular convolutional neural networks, and explain their properties which are used in the architecture developed in this work. In addition, the concept of a spectrogram, and frequency representation of a signal more broadly, is introduced and motivated. Finally, the concept of bagging, an ensemble method for constructing a classifier from a collection of subclassifiers, is discussed. Bagging is used later in the work as both a measure of uncertainty and as a means to improve classification accuracy.

This chapter assumes a basic familiarity with the ideas of supervised learning. Concepts like training and validation sets, overfitting, regularization, and hypothesis class should be familiar to the reader.

2.1 Neural Networks

This section presents a very basic introduction to neural networks, a machine learning technique which has recently come into prominence. A more comprehensive background with links to other statistical models can be found in [20], while more comprehensive information on types of neural nets and their properties is presented in [21].

While the term "neural network" encompasses a rich variety of architectures, they tend to share the following broad similarities:

- they can be used for both classification and regression
- they have highly variable capacity
- they require a differentiable loss function

As a whole, neural networks define a class of highly non-linear approximators to arbitrary continuous functions over $\mathbb{R}^d$, with d potentially very large. These networks are employed to capture the inscrutable functional dependencies found in situations where learning is done from very raw input. This is in contrast to methods like linear regression, where a very stringent functional form is assumed in advance. This malleability is achieved at comes at a cost of scrutability: the parameters of a neural network correspond to coefficients of either very raw input values (such as the value of a particular pixel in an image), or to the coefficients of complex non-linear transformations of such raw input values. In either case, the precise value of a subset of parameters is generally not very informative, and can even vary substantially between networks trained on the same data, and which approximate the same functions over the data distribution of interest.

This section gives an overview of the particular types of neural architectures relevant to the sleep staging work done in this thesis.

2.1.1 Multilayer Perceptrons

The simplest kind of neural network is the **multilayer perceptron** [24], or MLP for short. The MLP architecture takes $\mathbf{x} \in \mathbb{R}^m$ to a real-valued $\mathbf{y} \in \mathbb{R}^d$. A multilayer perceptron is a non-linear function parameterized by a sequence of **weight matrices** W_l and **bias vectors** $\mathbf{b}_l$

$$\mathbf{W}_l \in \mathbb{M}(z_{l-1}, z_l)$$
$$\mathbf{b}_l \in \mathbb{R}^{z_l}$$
$$l \in \{1, \ldots, L\}$$

Where $\{z_l\}_1^{L-1}$ are hyperparameters, $z_0 = m$ equals the dimensionality of $\mathbf{x}$, and z_L is the dimensionality of $\mathbf{y}$. A further set of hyperparameters is a collection $\{\phi_l : \mathbb{R}^{h_l} \to \mathbb{R}^{h_l}\}_1^L$ of **activation functions**, which must not be linear, except possibly ϕ_L. Common activation functions include the sigmoid function or the ReLU [21, 44], and the development of new activation functions is an active area of research.

The hypothesis class of functions $h : \mathcal{X} \to \mathcal{Y}$ comprising multi-layer perceptrons with the given hyperparameters is then defined recursively as:

$$\mathbb{R}^d = \mathcal{Y} \ni \hat{\mathbf{y}} := \mathbf{z}_L$$
$$\text{for each } l \in \{1, \ldots, L\} : \mathbf{z}_l := \phi_l(\mathbf{W}_l \mathbf{z}_{l-1} + \mathbf{b}_l)$$
$$\mathbf{z}_0 := \mathbf{x} \in \mathcal{X} = \mathbb{R}^m$$

Put into words, the neural network takes the input vector and applies an alternation of affine transformations $(\mathbf{W}\mathbf{x} + \mathbf{b})$ and nonlinearities ϕ_l. The nonlinear nature of the activation functions is critical, since with linear activations the whole network would be equivalent to a single affine transformation. The vector of possible outputs at each stage l is called a **layer**, and stages 1 through $L - 1$ are called **hidden layers**, since their values are neither part of the input nor

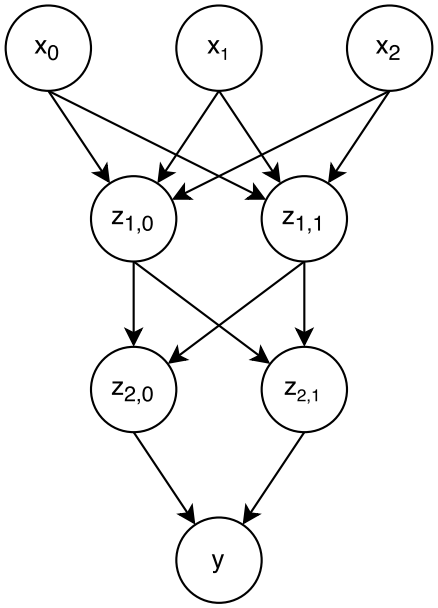

Figure 2–1: The structure of a toy MLP with two hidden layers and scalar output. Here $m = 3, L = 3, z_1 = 2, z_2 = 2, z_3 = d = 1$.

output, but internal to the network's mathematical operation. Layer L is called the **output** layer and layer 0 is called the **input layer**.

Often, a neural net's structure is visualized as a graph like in Figure 2–1. Such a graph depicts the structure of a neural network independently of the particular values of inputs and outputs to it. The layers are visualized as a collection of **nodes**, which correspond to the individual entires of the hidden state vectors, abstracted from their numerical value on any particular input. The parallels between the structural connectivity between nodes in layers and the interconnections of neurons in the human brain inspired the term "neural network" [24]. Continuing this parallel, the $\mathbf{z}_l$ values for a given input are called the network's **activations** or activation vectors.

Despite a fairly straightforward mathematical definition, even with one hidden layer (i.e. $L = 2$), MLPs can approximate any continuous function [30]. It can also be shown that with increasing **depth**, or number of hidden layers, a neural network can approximate exponentially more linear regions in the input function than a neural network with a single layer. **Deep learning** refers to the study and implementation of neural networks with a large number of hidden layers (sometimes thousands [26]). These deep architectures have made advances in a number of fields in recent years. The architecture presented in this thesis is an example of a deep architecture.

Training Neural Networks

Neural networks are trained with **backpropagation** [27]. In backpropagation, the gradient of the loss with respect to each of the network's parameters is computed. In modern implementations, this is done with automatic differentiation software. Subsequently, a **gradient descent** algorithm is used to minimize the training loss. For the large data sets needed to train neural networks, it's not practical to calculate the gradient of the total loss on the training set, so **stochastic gradient descent** methods are used instead. For stochastic gradient descent, at each iteration the gradient is taken with respect to a subset of the training data. These subsets are usually sampled randomly at each iteration, or cycle through the entire training set in a random order. In its most basic form, at each time step, gradient descent updates the parameters of the network by a small step in the direction opposite of the gradient of the loss function with respect

thereto:

$$\theta_{t+1} = \theta_t - \alpha \nabla_\theta \mathcal{L}(\mathbf{x}, \theta)$$

In practise, algorithms more sophisticated than the simple rule above have been developed [35, 58] for training neural networks. These algorithms, however, still fundamentally rely on the availability of the gradient. It is for this reason the loss and activation functions in the definition are required to be almost-everywhere differentiable. Designers of more advanced neural architectures strive to preserve this differentiability, which allows very sophisticated predictors to be trained using a shared numerical machinery of automatic differentiation and advanced gradient descent algorithms.

2.1.2 Convolutional Neural Networks

A common variant of the fully connected neural network is the convolutional neural network (CNN). These networks were pioneered for use in image processing [38, 39]. They are designed to exploit input with spacial locality and possess a degree of translation invariance. These are precisely the properties of natural images, where an object can appear at many places in an image, and pixels corresponding to a fixed object are adjacent to one another.

The CNN is distinguished from an MLP by the connectivity structure of its layers. Layers where each node of a layer is connected through the weight matrix $\mathbf{W}$ to every node in the next layer, as in the MLP, are referred to as **fully connected** layers. In a CNN, however, the lower fully connected layers are replaced with **convolutional layers**. In a convolutional layer, instead of an

activation vector $\mathbf{z}_l$, the output at layer l, $\mathbf{Z}_l$, is a tensor of f_l multidimensional **feature maps**:

$$\mathbf{Z}_{l,j} = [\mathbf{z}_{l,1}, \ldots, \mathbf{z}_{l,f_l}]$$

$$\mathbf{z}_{l,j} \in \mathbb{R}^{n_{l,1}} \times \cdots \times \mathbb{R}^{n_{l,k}}$$

$(n_{l,1}, \ldots, n_{l,k})$ is a hyperparameter called the **shape** of the feature maps. For CNN, k is usually constant for all layers, while $n_{l,k}$ varies with l. In image processing applications, the shape is $(n_{l,1}, n_{l,2})$, corresponding to the x and y dimensions of the image. That means the input and output tensors of the convolutional network will have rank two, with each axis corresponding to a dimension of the image. This is also the shape used in the sleep staging architecture developed in this thesis, presented in Chapter 4.

Each feature map can be thought of as a separate hidden state for the given layer, and carries some distinct aspect of information observed therein. For the method introduced, the number of input feature maps, f_0, is the number of EEG channels. Hidden layers generally have many more feature maps than the input.

Instead of a weight matrix $\mathbf{W}$, each non-input layer in a CNN is associated with f_l distinct **kernels**, which are $k+1$-dimensional tensors $\mathbf{K}_{l,j}$ of size $(u_{l,1}, u_{l,2}, \ldots, u_{l,k}, u_{f_l-1})$. The values u are referred to as the **receptive field size** of the kernel, and represent the support of the convolution operation with the kernel. The activation of feature map j in layer l is generated by "sliding" $\mathbf{K}_{l,j}$ across each feature map of the layer's input and convolving the corresponding slice of the kernel with the input values in each region of that feature map. This can be

expressed formally by the cumbersome expression below:

$$\mathbf{z}_{l,j,i_1,\ldots,i_k} = \phi^l \left(\mathbf{b}_l + \sum_{a=1}^{f_{l-1}} \mathbf{K}_{l,j,a} * \mathbf{z}_{l-1,a,[s_l i_1 - z_{l,1}, s_{l,1} i_1],\ldots,[s_l i_k - z_{l,k}, s_{l,k} i_k]} \right)$$

where $\mathbf{K}_{l,j,a}$ is the k-dimensional slice of $\mathbf{K}_{l,j}$ with the feature map index equal to a, and where the bracketed subscript notation $\mathbf{q}_{\ldots,[a,b],\ldots}$ denotes a sub-array of the given quantity from index a to index b along the given dimension. An additional complication, the **stride** $s_l \in \mathbb{N}^+$ appears in the above equation. If the stride is greater than zero, the kernel "skips" $s_{l,j}$ positions in the input feature maps between successive points in the output map. Layer l layer then has $\prod_j s_{l,j}$ times fewer nodes than layer $l-1$. To downsample further, CNNs often incorporate **pooling layers**, whose action is to sub-sample certain indices of their input, outputting a a single representative value of th3 input values in their receptive field. The most common form of pooling is d-**max pooling** where the output is the maximum value of all of the inputs in the receptive field. For example, when working with rank 2 image data, each point in the output feature map of a max pooling layer is the maximum value of a $d \times d$ patch of the input. In this case, output feature maps contain d^2 times fewer points. Downsampling is important, since the top few layers of a complete CNN architecture are usually fully connected, rather than convolutional. As will be seen below, fully connected layers need inputs of reasonably low dimension to have a manageable number of parameters.

The operation of a convolutional net can be made clearer with a diagram, seen in Figure 2–2. An important thing to note is the locality of information

flow. Because the kernel was bounded in each dimension by a (usually small) size u, locations in the input layer which are distant from a particular node do not contribute to it. This is in contrast with an MLP, where each node of the input has a potentially non-zero weight connecting it to each node in the layer above.

This endows convolutional networks with two important properties different from those of an MLP. First, the number of parameters does not by necessity scale with the size of the input. Second, the number of parameters for a given size input can be made much smaller than for an MLP. Consider an input image of $50 \times 50 = 2500$ pixels. With a hidden layer of size 100, an MLP with just a single layer would need $2500 \times 100 = 2,500,000$ parameters, a very large number. In contrast, a convolutional neural net with 5 layers, receptive fields of 5×5, and 32 feature maps per layer has $5 \times 5 \times 5 \times 32 = 4000$ parameters. Furthermore, because the same kernel is applied at at every location in the input, the convolutional architecture learns translation-invariant structure.

The reduction in the number of parameters is not free, however: it imposes a strong assumption on the types of input distributions the convolutional hypothesis class will learn to represent well. Where data is not dominated by spatially local relations, the spatially local kernel will not have access to useful information, and learning will fail. For this reason, a CNN would perform much worse at learning to recognize faces if, for example, each image's pixels were first scrambled by a random permutation chosen in advance. On the other hand, an MLP's performance would not change. However, translation invariance and the ability to handle very high dimensional input with local structure do make the convolutional

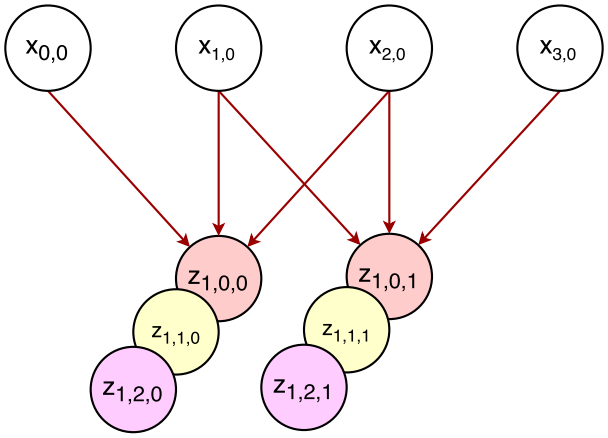

Figure 2–2: The structure of a CNN with 1-dimensional input ($k = 1$). The colours represent feature maps in the first hidden layer, of which there are three. The input $\mathbf{x}$ is one-dimensional with one feature map. Each triplet of red arrows represents the same set of weights, convolved with three input nodes at a time to produce the value of the corresponding red hidden node. The weights are free to differ between colours.

architecture very well adapted to the task of learning from natural images, which are representations of the spatially local and translation-invariant objects found in our world [36]. However, images should not be seen as the only use case for convolutional nets: any data distribution possessing locality and translation invariance is a candidate for convolutional learning, and as I argue later, EEG recordings in spectrogram form are a good example of such data.

Figure 2–3 shows what the structure of a slightly more realistic, albeit still very small, complete CNN might look like. In the literature, a variety of very large convolutional neural networks with thousands of layers have been experimented with.

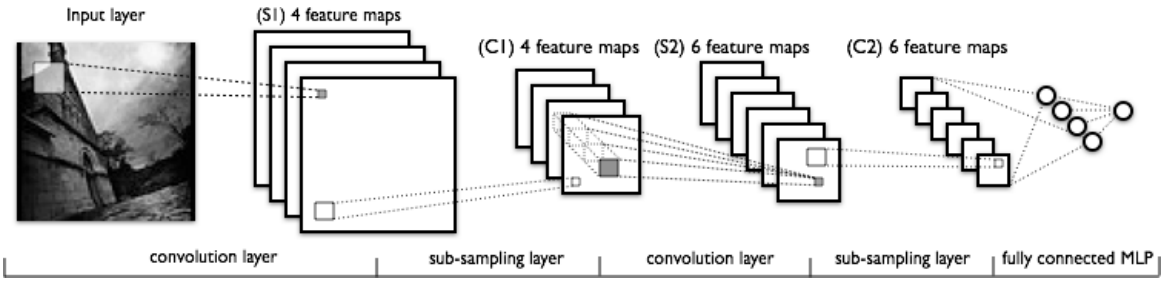

Figure 2–3: Schematic of a convolutional neural network, including subsampling layers. Note that at the uppermost layers the feature maps are laid out in one dimension and connected to a fully connected network. Taken with permission from `deeplearning.net`

2.2 Bagging

When trained on some finite training set $\mathcal{T} = \{(\mathbf{x}, \mathbf{y})_i\}_i$, most algorithms are likely to converge on a suboptimal hypothesis, owing to either lack of training data or local minima in the training algorithm. However, this can be remedied by training a number of partially independent classifiers. These are then combined into an **ensemble** classifier by aggregating their predictions [20]. Aggregate classifiers produced in this way improve on the performance of their constituents if two conditions are satisfied [18]:

- each classifier in the ensemble performs better than random chance, and
- classifiers in the ensemble make at least partially independent errors.

The above conditions are quite generous. Consequently, there exist many methods by which such ensembles may actually be constructed. One of the most basic and widely used such methods, and the one used in the work presented in this thesis, is **bagging**.

Under the bagging scheme, multiple instances of the training set of the same size as the original are created by sampling the training set with replacement [12].

These samples are called **bootstrap** samples. A new classifier is then trained from scratch on each of the bootstrap samples. The prediction of this aggregate classifier is defined as a combination of the outputs of each of the individual classifiers, usually obtained either by voting (for discrete labels) or by averaging (for regression-like outputs). In the language of bias-variance trade-off, the bagged classifier has lower variance while leaving bias unchanged [8, 20]. Therefore, bagging is a universally useful tool wherever the cost of retraining the classifier multiple times is not too steep. Another useful property of bagging which can be shown mathematically is that from a Bayesian perspective, training a classifier over multiple bootstrap samples in approximately equivalent to sampling from the posterior distribution over classifiers on the training set [20]. In particular, this is the case if one assumes a Dirichlet prior over the class distribution, which is a quite natural assumption. From this, it follows that the degree of agreement among the constituent classifiers on any label can be interpreted as a measure of uncertainty in the prediction, even when the component classifiers are deterministic.

2.3 Spectrograms

The final major technical concept used in this thesis comes not from the domain of machine learning, but rather from signal processing. Signal processing deals with time series, or time-ordered sequences of real numbers corresponding to some possibly noisy measurement. EEG is a prime example of a time series. Often, the structure of a time series is better revealed in the **frequency domain**. The frequency domain is a representation of a time series where the independent axis is not time, but frequency. The value of the frequency representation at each

frequency ω is the phase and amplitude of the component corresponding to that frequency in the original signal. In the context of signal processing, the original time series is referred to as being in the **time domain**. Mathematically, the time and frequency domains are related by the **Fourier transform**. The Fourier transform has a large number of variants, depending on whether the input signal is square-integrable, periodic, discretized, or continuous, and furthermore depending on the dimensionality of the signal. Below is the basic mathematical form for continuous signals of finite energy in one dimension,

$$\mathcal{F}(f)(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} \mathrm{d}x$$
$$\mathcal{F}^{-1}(g)(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g(\omega) e^{i\omega x} \mathrm{d}\omega$$

where the second line gives the inverse of the transform. Several other conventions exist where the factor of 2π is on the forward transform, or where the transformed variable is frequency ξ rather than angular frequency $\omega = 2\pi\xi$, in which case the factor of 2π moves from being a coefficient to being in the exponent. These conventions are equivalent up to rescaling, and are usually a numerical implementation detail.

It should be noted that the Fourier transform outputs complex values, and is defined both for positive and negative frequencies. For real input signals, the value of the transform at each negative frequency $-\omega$ is the complex conjugate of the transform at ω, which allows a more compact representation in some implementations. The Fourier transform is an invertible transformation, but its output, being complex and defined on negative frequencies, is often not the most

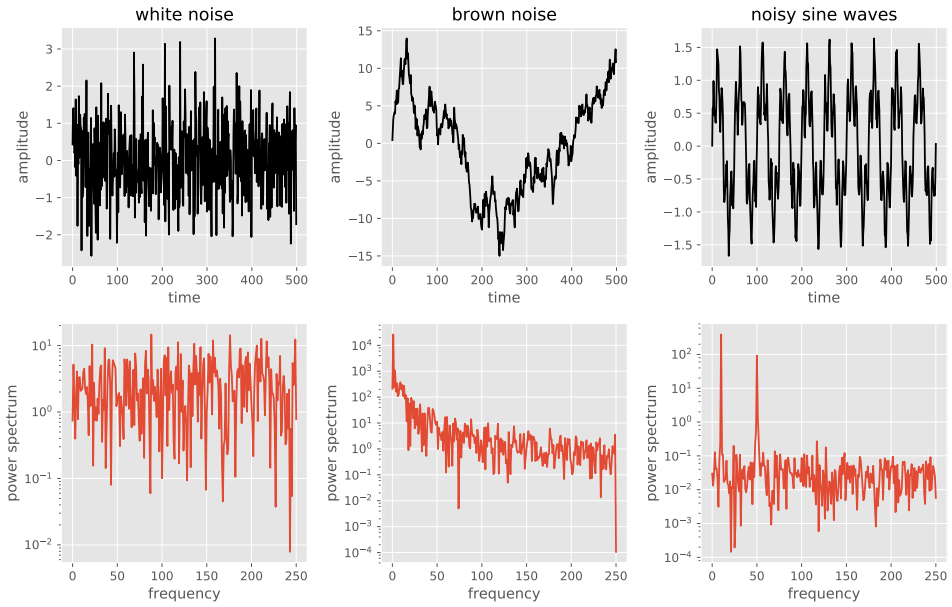

Figure 2–4: Some timeseries signals (top plots) and their corresponding power spectra (below).

intuitive way to interpret the frequency composition of a signal. For this reason, a derived quantity, the **power spectrum** or **energy spectrum**, depending on whether the signal is stationary or square integrable respectively, is often used instead. The one-sided power spectrum is real valued and is defined over the positive reals. As the power spectrum discards phase information, is no longer invertible. It represents the power (or energy for square integrable signals), carried by the signal at a particular frequency. The power spectrum for a finite energy real signal is given as $P(f)(\omega \geq 0) = 2|\mathcal{F}(f)(\omega)|^2$ and is illustrated in Figure 2–4 for various signal types. The factor of two comes from the fact that the Fourier values at negative frequencies, whose square modulus at $-\omega$, for real signals, equal to that at ω, is included in the value of $P(f)(\omega)$.

Some more mathematical complications arise where a signal is sampled at discrete points in time, which is the case for digital signal. It has to be assumed that the maximum frequency for which the frequency representation is non-zero is less than half the sampling frequency. Otherwise, the sampled signal will be **aliased**. Aliasing is a process whereby high frequency components become indistinguishable from lower-frequency ones and are added together in the resultant transform. As an example, consider sampling a sine wave with a frequency of 1 Hz with samples one second apart: the samples will fall on zeroes of the sine-wave, and the sampled signal will be indistinguishable from a zero signal. For this reason, signals are **bandpass filtered** before being digitized. Bandpass filtering is a process that attenuates high frequencies, so that they will not alias to lower frequencies when sampled. When applied to discrete and finite time series, the Fourier transform is known as the **discrete Fourier transform** (DFT), often referred to as the **FFT**, which stands for Fast Fourier Transform, and is the name of a particular DFT algorithm. The discrete Fourier transform acts on vectors of real numbers representing the signal at discrete points in time, and produces a vector of equal length giving the signal's frequency representation at a set of discrete frequencies. The DFT can be represented by a unitary matrix multiplication.

Often, it is interesting to see how the frequency composition of a signal evolves over time. This is not possible with a regular Fourier transform, since all time information is integrated out to leave a pure frequency representation. Instead, a **spectrogram** is used. A spectrogram is a mixed time-frequency representation of the signal's power spectrum. It is calculated by performing a

discrete Fourier transform on (possibly overlapping) subsegments of the time signal. The spectrogram's time axis contains a sequence of time **bins**, and its second axis is frequency. At each (bin, frequency) pair, the spectrogram gives the spectral power of the signal at that frequency for that bin. The spectrogram is usually plotted as a colormap, where each point's colour intensity corresponds to the spectral power at that point. It's this colour map representation that inspired this work's use of spectrograms as image inputs to a CNN.

To generate a spectrogram, the time series segment in each bin is usually first multiplied by a **windowing function**, which is a function that tapers off to zero on either side of the trace segment. This is done to suppress artifacts. To recover the information lost by the windowing, there is usually some overlap between successive time bins. One of the most common spectrogram techniques is the Welch method [63], which uses 50% overlapping time bins and the Hann window function [10]. A modification of this approach using the same Hann window but with an overlap of 75% is used in the EEG classification method. Figure 2–5 provides a basic illustration of a spectrogram.

The more time points of the original signal each time bin includes (i.e. decreasing the time resolution), the finer the frequency resolution becomes. Conversely, the finer the time resolution, the fewer the points in each bin, which means frequency information becomes less defined. In general, the product of the time and frequency resolutions of the Fourier transform is bounded below according to the uncertainty principle [28]. In the discretized case, this can be intuited in terms of linear algebra. Since DFT is an invertible linear transformation, the number

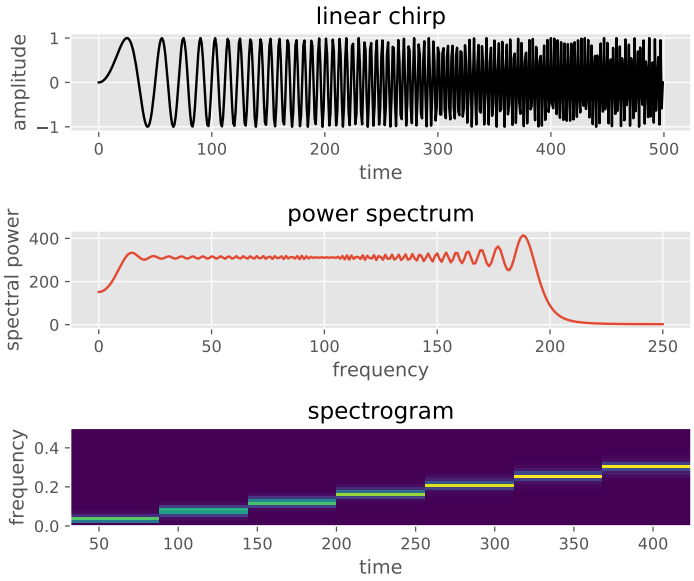

Figure 2–5: A signal of linearly increasing frequency (a **linear chirp**). The power spectrum does not reveal the time-varying structure, which is made clear in the spectrogram.

of independent frequency components that can be distinguished is equal to its rank, which is in turn equal to the dimension of the input space. In the case of the spectrogram, the dimension of the input space is just the number of time points in each bin, since the DFT acts on each time bin separately.

CHAPTER 3
Review of Literature on Sleep Staging and Deep Methods for EEG

The automated generation of hypnograms is a task well studied in the literature, and a diverse variety of architectures has been developed for this purpose. It is also the case that deep convolutional architectures have been used to learn from EEG data for tasks other than sleep staging. This chapter discusses the method presented in this thesis in the context of the existing literature. The application of a deep convolutional architecture to the task of sleep staging from EEG is believed to be a novelty of the work in this thesis.

3.1 Convolutional Architectures for EEG in the Literature

The use of convolution for EEG is not a novelty of this work. One-dimensional convolutions have been used on EEG data for the purposes of signal classification recognition of rhythm stimuli [56], EEG pattern detection for brain-computer interfaces [14], seizure detection [43], and more. The use of Fourier transforms together with convolution is also known in the literature. Indeed both [56] and [14] use a Fourier transformation layer within the neural architecture. However, the convolutional network trained over this data was still one-dimensional.

Convolutional architectures which work explicitly with two-dimensional time-frequency representation of sleep data are also known. Bashivan et al. [7] use a two-dimensional convolutional architecture to learn from EEG features. In fact, the window size and layer structure of the architecture presented in this work

was inspired directly by [7]. However, the input data in [7] is not in the form of a spectrogram: the frequency components are the channels of the input, while the x, y coordinates are mapped to physical locations on the brain. This is in contrast to the spectrogram as used in this work, where electrode locations are the channels. The use of a spectrogram representation in that form as input to a two-dimensional convolution is believed to be novel.

3.2 Automated Sleep Staging Methods in the Literature

At a high level, automated sleep staging methods can be grouped into two types: methods using expert features, and methods, like the architecture in this thesis, that use raw data as input. In this context, expert features are considered a collection of a fixed number of hand-engineered markers that are not learned from data, and which are extracted from the data in advance of any machine learning method being applied. Another way machine learning methods can be grouped is by what input data they take. Many methods in the literature use polysomnograms containing signals other than the EEG. Finally, sleep staging methods can be classified based on whether they use deep learning or not. Methods not using deep learning are referred to in this chapter as *classical* methods. Below is a survey of some modern methods in the literature. It is not exhaustive, but is representative of the current performance of automated methods various types of models and training methodologies. Owing to their preponderance in the literature, methods training on hypnograms generated according to the older RK standard are included.

A large variety of classical models and algorithms have been used to attack the sleep classification problem. Luo and Min [41] use a conditional random field model explicitly designed for training on the patient on which the prediction is to be carried out, thereby adapting to the patient's idiosyncrasies. They use four patients in their experiments, which is a somewhat limited experimental scope. The performance they obtain is reported only as aggregated accuracy across all sleep stages, and is 0.83. A different model using support vector machines is presented by Gudmundsson et al. [23]. They classify into four sleep stages, and their best set of features, based on Hjorth complexity parameters, attains an average accuracy of 0.81. Pan et al. [46] use a Hidden Markov Model, and use EMG and EOG as input signals in addition to simple EEG. Thirteen spectral features were selected from among these three signals. The feature vector of each 30 second epoch was taken to be the average of spectral features of each two-second segment in that epoch.

An interesting set of expert features is used by Acharya et al. [2], who use a Gaussian mixture model as the classifier. The features they use are higher-order spectrum features, which are nonlinear functions of the Fourier transform evaluated at pairs of frequency points. They try various combinations of these features, and validate their method by 10-fold cross validation on a data set of approximately 40 patients' overnight EEG recordings. Their results stand out in achieving an excellent performance on N1. However, their N1 stage does not appear to be at all underrepresented like it is in other works, suggesting a possible

nonstandard nature of the recordings on hand which enables this high level of performance.

In addition to the classical models described above, several deep models are also known in the literature. A neural network architecture for hypnogram generation was created by Ebrahimi et al. [19]. Their work uses a set of 12 expert features extracted from the data derived from wavelet transform coefficients. The REM and N1 stages were merged into one in their experiments. Their work did not appear to use cross-patient validation, and certain steps of their experimental methodology remain unclear. A similar work, also using an MLP on top of expert features derived from wavelet coefficients, is presented by Sinha [52]. A limitation of this work, which achieves high accuracy, is that he classifies only between three sleep categories: WAKE, REM and SS, the latter being the aggregate of all other stages. It is also interesting to note that the model considers 2-second pages, and appears to classify at this granularity.

A deep model that uses no expert features is presented by Längkvist et al. [37]. This architecture uses a Deep Belief Network (DBN) to automatically learn features from sleep data. Their most successful architecture uses an HMM over the features learned by the DBN to make the final sleep stage prediction, creating a combination of deep learning for feature extraction and classical learning for prediction.

In addition to the purely academic approaches described above, several proprietary commercial sleep stage classification packages exist. The most well known is probably Somnolyzer. The software's web page claims to have "scores

work	standard	signals used	notes	validation	metric	WAKE	REM	S1/N1	S2/N2	N3/SWS
this thesis	AASM	EEG	-	cross-patient	F1	0.84	0.87	0.45	0.86	0.90
Längkvist et al. [37]	RK	EEG	transitional pages removed	cross-patient	F1	0.78	0.78	0.37	0.76	0.84
Ebrahimi et al. [19]	RK	EEG	bad pages removed	mixed-patient	Acc.	0.99	0.92		0.89	0.95
Pan et al. [46]	RK	EEG,EMG,EOG	-	cross-patient	Acc.	0.89	0.90	0.34	0.82	0.95
Acharya et al. [2]	RK	EEG	-	mixed-patient	F1	0.88	0.94	0.89	0.75	0.96
Anderer et al. [5]	AASM	EEG,EMG,EOG	-	-	Acc.	0.87	0.92	0.60	0.82	0.82

Table 3–1: Summary of performance metrics across select literature surveyed.

are indistinguishable from those of a human scorer" [1]. This system was developed by Anderer et al. [4] to classify according to the RK standard and uses a number of expert features and the EOG, EEG and EMG channels to perform the classification. That work reports an overall agreement with human raters of 79%. The software was later updated to classify according to the AASM standard, and Anderer et al. [5] found an agreement with human experts equal 82% for the AASM version. These numbers are the same as the human interrater agreement rate, suggesting the limitation in evaluating is performance is the accuracy of the human labels. One of the advantages of this software is that it explicitly identifies many of the physiological features such as sleep spindles or K complexes in the signals.

3.3 Summary of Literature Performance

The best performance for the most relevant methods discussed above is presented in Table 3–1, alongside the median cross-validation result obtained by the multi-page bagged architecture presented in this thesis. This table attempts to bring the performance of various methods into as comparable a form as possible. However, the different training and validation methodologies make it difficult to make a final judgment. Ebrahimi et al. [19] and Sinha [52] do not appear to use cross-patient validation and use a restricted number of sleep stages, while some

methods, like [41], are explicitly designed for training on the same patient as for prediction. I believe the method presented in this thesis is a strong competitor given its lack of need for expert features and explicit cross-patient validation. Its performance on N2 is particularly strong.

With that said, the thorough evaluations by [4] and [5] suggest that the fundamental limitation in evaluating a model's accuracy beyond 80% or so is the inconsistency of human labelers. Therefore, caution should be taken when looking at results of accuracy of 90% or more.

CHAPTER 4
Experimental Methodology

This chapter presents the experimental methodology and details of the architecture used for the work in this thesis. First, a brief high-level overview of the architecture is presented, with the aim of motivating the particular architectural choices used in the work. Then, a description of the dataset used is presented. Subsequently, the specifics of the convolutional neural architecture used to implement the classifier are given. Finally, the specifics of the training procedure are explained.

The architecture described below and all of the experiments were implemented in the Python programming language, using the SciPy numerical libraries for core functionality [34]. The neural architecture was implemented using the Keras toolkit, using the Theano backend for GPU acceleration [15, 9]. The cross-validation routines were implemented using Scikit Learn [47].

4.1 Architecture Description

The fundamental innovation in the architecture used is the **conversion of a time series into a two-dimensional image representation using a spectrogram transformation**. As described in the technical background section, the spectrogram takes a Fourier transform of subsegments of the signal. On transforming the spectrogram into an image, each new pixel along the vertical extent of the output image is made to correspond to the spectral frequency

divisions of the spectrogram. Thus the height of the output image in pixels is equal to the number of frequencies in the spectrogram representation. Each new pixel along the horizontal extent corresponds to the spectrogram's time bins, which makes the width of the image in pixels equal to the number of time bins in the spectrogram. This value is proportional to the length of the signal, and is inversely proportional to the height. The numerical "brightness" value of each pixel of this two dimensional image is then equal to the output value of the spectrogram at the particular time and frequency corresponding to that pixel.

The second fundamental innovation insight is that **EEG channels can be treated like the colour channels of an image**. The time series for each EEG channel is transformed independently into a monochromatic image through the spectrogram method described above. These monochromatic images are combined into a multi-channel image by stacking them in the same way as the individual colour channels (i.e. red, blue) are stacked to create a composite colour image. The great utility of this technique is that a multi-channel EEG signal can then be processed by a standard image oriented CNN pipeline. These pipelines have been studied and optimized extensively in the machine learning literature. This work does not propose innovations to these CNN methods.

The spectrogram transformation described above is not only computationally useful, but finds justification as a form of low level feature extraction as follows. First, the features of a given sleep stage can occur at any point in the page without altering that page's classification. Therefore a model which is invariant in time is desirable. Second, the defining features of each sleep stage correspond

to time-localized patterns with a particular spectral profile, and should be readily distinguishable on a spectrogram with sufficient resolution. Finally, many of these patterns are defined by a spectral shape more than a particular frequency, and can actually occur at different frequencies across different patients. In a spectrogram, such a difference in base frequency is captured as a translation of the pattern along the frequency axis. Therefore, some measure of frequency invariance in the model is also desirable.

Putting the above together, it was conjectured that the mature and powerful techniques behind CNNs could be used on spectrograms directly for this classification task. The idea of treating spectrograms like images, and thus treating the problem as an image classification problem, motivates almost all of the architectural decisions below. From the image processing perspective, the architecture is very standard, and unless specific explanation is given, architectural decisions can be considered as being in line with standard literature on image processing.

4.2 Architecture Details

With the general motivation of the previous section in mind, this section gives the specifics of the spectrogram parameters and neural connectivity used to obtain the final results in this work.

4.2.1 Spectrogram Parameters

In line with the idea of treating a spectrogram like an image, parameters for the spectrogram were chosen with a few criteria in mind. One, the height in pixels height and width of the dominant spectral patterns should be about equal, so that interesting detail is not compressed along either axis. Two, there should

spectrogram	downsample factor	window	overlap	time bins	freq. bins
high frequency	1	128	96	57	65
low frequency	15	24	18	13	18

Table 4–1: Spectrogram parameters used. Window size and overlap given in number of samples.

be overlap between successive time bins, so that spectral features are smooth and detailed in pixel space. I found it was not possible to achieve the above criteria with a single spectrogram. For this reason, the signal was split into high and low frequency components, and two separate spectrogram inputs were used. This split significantly improved performance in preliminary testing.

Specifically, each 30 second page of the normalized EEG traces sampled at 64 Hz (thus capturing signals up to a frequency of 32 Hz) was transformed into two spectrogram representations: one for higher frequencies (1 Hz to 32 Hz) and the other for lower frequencies (0.36 Hz to 4 Hz), according to the parameters in Table 4–1. Because a neural network expects normalized input, the logarithm of the raw spectrogram values was taken.

A Hann windowing function was used for both types of spectrogram, and the type of spectrogram was for spectral density rather than total energy, as implemented in SciPy's `scipy.signal.spectrogram` function. The log spectrogram was then used as input to the CNN in the manner of an image. Each EEG channel's spectrogram was treated as a colour channel.The order of EEG channels was arbitrarily chosen, but was kept constant throughout.This allowed the network the possibility of differentiating between different areas of the brain (which correspond to particular channels according to the 10-20 system).

4.2.2 Neural Architecture

The architecture below largely follows established practises for a simple image classification CNN, sized appropriately for the given input. One notable difference is that the number of feature maps in the convolutional layers and the number of nodes in the fully connected layers are smaller than one would find in an image classification stack. This is because the number of distinct spectral features in EEG is much smaller (i.e. spectrograms are all much more alike and homogeneous) than in a natural image. Therefore, the model complexity must be correspondingly lower to prevent overfitting.

The basic convolutional architecture used for the single-page predictor is summarized in Table 4–2. The high and low frequency spectrograms were passed through separate convolutional layer stacks which were joined with a sequence of fully connected layers, topped with a five-node `softmax` activation ($\texttt{softmax}(\mathbf{x})_i := \frac{e^{x_i}}{\sum_j e^{x_j}}$) predicting the stage of sleep for the page.

The use of a small convolutional kernel dimension was inspired by the work by Bashivan et al. [7]. Exponential Linear Unit activations, as introduced by Clevert et al. [16], were used as fully connected layer activations. The convolutional activations were ReLUs [44].

4.3 Description of the Dataset

The training data used in the final results consists of raw EEG recordings which were obtained at the clinical neurophysiology laboratory at Sunnybrook Health Sciences Centre, Toronto, Canada according to American Academy of Sleep Medicine (AASM) guidelines [31] using a Grael HD PSG amplifier (Compumedics,

inputs	
low freq.	high freq.
	conv (3×3)
	conv (3×3)
	max pool (2×2)
conv (3×3)	conv (3×3)
conv (3×3)	conv (3×3)
max pool (2×2)	max pool (2×2)
fc (24)	
fc (24)	
fc (6: softmax)	
output	

Table 4–2: Main neural architecture used. Fully connected layers denoted by "fc". Parenthesized numbers are kernel dimension for convolutional layers, downsample factor for pool layers and number of hidden nodes for fully connected layers.

Victoria, Australia). These recordings were manually scored according to AASM guidelines by registered polysomnography technologists to produce the training hypnograms.

To generate a dataset of largely normal recordings containing adequate amounts of all sleep stages, 116 consecutive recordings were selected among those obtained between 2009 and 2015 meeting the following criteria: total sleep time more than 240 minutes, sleep efficiency more than 80%, apnea-hypopnea index (AHI) less than 5, periodic limb movement index less than 5, respiratory disturbance index (RDI) less than 5, oxygen nadir over 90%, %N3 sleep over 15%, %N1 sleep $< 10\%$, and %REM sleep $> 15\%$. The median age of the participants was 29, with interquartile range [23-35]. There were 23 male and 93 female participants. All electrodes were referenced to the FPZ electrode of the 10-20 system.

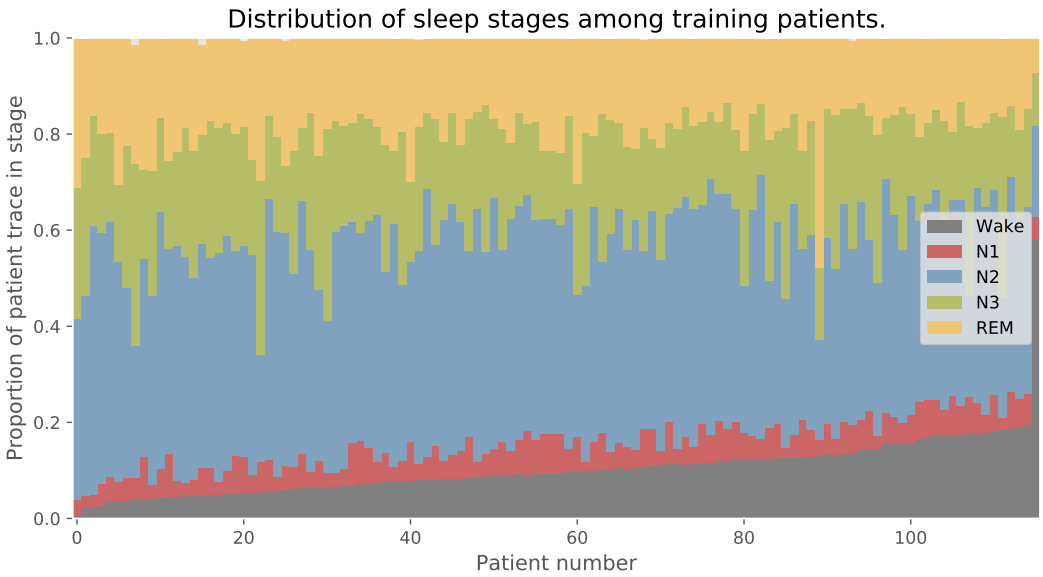

Figure 4–1: Distribution of sleep stage labels among training patients. The precise proportion by total number of given stage in the training set is: Wake=12.4%, REM=19.0%, N1=5.5%, N2=42.5%, N3 = 20.5%

Since the stated aim of the algorithm is to be able to handle data with minimal preprocessing, the EEG recording dataset was not filtered according to any measure of recording quality, with the exception of eliminating 8 traces (excluded from the count of 116 given above) in which the signal was totally corrupted and did not correspond to valid EEG activity. With the exception of the above omission, no artifact detection or removal was performed on the remaining traces. The sleep stage breakdown of the training patients is shown in. Figure 4–1.

4.4 Data Preprocessing

Seven EEG channels, the same across all recordings, were used from each EEG trace. These channels were A1, A3, C3, C4, F3, F4, O1, in that order.

The original recordings were sampled at 256 Hz, but the records had been low-pass filtered at the source to include only frequencies below 30 Hz. For this reason, the recordings were downsampled to 64 Hz before use in order to save computational cost.

Each trace was, as a whole, numerically normalized to zero mean and unity standard deviation in the time domain, in accordance with standard practise of training neural models. The sleep stage labels were **1-hot encoded**. That is, each sleep stage, in addition to the "unknown" sleep stage, was associated with a unique index from 0 to 5. Then the label of each page was represented as a length 6 vector of all zeroes except for a 1 at the index corresponding to the page's sleep stage. The "unknown" entries were all given a weight of zero during training, and were thus used as a numerical convenience rather than learnable output.

4.5 Training Methodology

One of the fundamental objectives of this work is to train a classifier that work well on unseen patients. For this reason, the training and validation sets were **strictly separated at the patient level** at all times. No patient's data was ever used in both the training and the validation sets. 3-fold cross-validation was used for training. There was no set-aside training set, and the final results reported in the following chapter are the aggregate results on the validation folds.

To generate each data sample during training, the following procedure was used. First, a patient was samples uniformly at random from the set of training patients for the fold. Subsequently, an EEG page was sampled from this patient uniformly at random. As a form of regularization, noise equivalent to adding

$\mathcal{N}(0, 0.2^2)$ in time domain was added to training samples sample. This noise was resampled independently for each sample For further regularization, and following standard practise in the literature, 0.5 dropout [55] was added under every fully connected layer except the output layer during training.

The specifics of the training follow the standard practise in the literature. The entire system was trained end to end using the Adam optimizer [35] under the categorical cross-entropy loss $\mathcal{L}(\hat{y}, y) = \sum_i y_i \log \hat{y}_i$. The learning rate was scheduled to decrease every time training failed to make progress in decreasing validation loss for a number of consecutive epochs, this number exponentially dependent on the number of times learning rate had already been reduced to that point. Training proceeded through 3 such learning rate reductions. To address class balance, the loss for each training page was weighted inversely proportionally to the probability of its label being chosen according to the sampling scheme described above. Finally, in order to recover the sleep label from the output `softmax` vector, the argmax was taken, returning the index with the largest entry as the prediction.

4.5.1 Bagging

To create a bagged ensemble of classifiers, the training procedure was repeated on subsets of the training data resampled with replacement. For the purposes of cross validation, the cross-validation splits were carried out before the resampling for bagging, and the entire validation split was used to evaluate the bagging ensemble's performance. Within each cross-validation fold, 10 classifiers were trained within each bag. It should be noted that training on some bootstrap

samples failed. That is, the training loss never moved below the performance of random guessing. These failed iterations were excluded from the bag and were not retrained. Therefore, some bags contained fewer than 10 subclassifiers.

4.5.2 Looking at Multiple Pages

When predicting the stages of sleep, a human technician might look at the pages that precede and follow the page to be classified to make a more informed decision. For this reason, the architecture was also trained using the concatenation of the previous, current and next pages' spectrograms as input. The architecture for this multi-page training was the same as for the single page case other than the addition of an additional fully connected layer before the output. This multi-window architecture is depicted in Table 4–3. Importantly, neural weights were shared between the convolutional stacks operating on each of the pages. Therefore, the number of parameters was not significantly larger than in the single page case. It is important to note that This architecture was trained from random initialization, and weights from single-page training were not transferred.

previous page		page to predict		next page	
low freq.	high freq.	low freq.	high freq.	low freq.	high freq.
	conv (3×3)		conv (3×3)		conv (3×3)
	conv (3×3)		conv (3×3)		conv (3×3)
	max pool (2×2)		max pool (2×2)		max pool (2×2)
conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)
conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)	conv (3×3)
mp (3×3)	mp (3×3)	mp (3×3)	mp (3×3)	mp (3×3)	mp (3×3)
	fc (24)		fc (24)		fc (24)
	fc (24)		fc (24)		fc (24)
			fc (24)		
			fc (6: softmax)		
			output		

Table 4–3: Multi-page neural architecture. Equivalent to three copies of the single page architecture with an additional fully connected layer on top. Each of these copies takes as input the preceding, current, and following page, where the current page is the one to be classified. Weights between convolutional stacks are shared.

CHAPTER 5
Results

This chapter presents the performance of the architecture developed in the previous section. Various use cases are considered, from full automation to partial automation based on the predictor's own confidence in its predictions. The degree to which the architecture's prediction of its uncertainty, based on bagging, can be trusted is also validated.

5.1 Set of Classifiers Considered

The performance of the architecture is considered in terms of four principal classification modes based around from the architecture defined in and trained according to the protocol in the previous chapter. These classifiers are named: UNBAGGED, BAGGED, UNANIMOUS, and ORACLE. These classifiers are defined as follows:

- UNBAGGED is the classifier obtained by training a single instantiation of the neural achitecture defined in the previous section according to the training protocol given in the previous section.

- BAGGED is the classifier obtained by applying ensemble procedure described in section 4.5.1. The output of BAGGED is **the majority prediction**, which is defined mathematically as

$$\underset{i}{\operatorname{argmax}} \left(\sum_i \mathbf{y}_i \right)$$

where $\mathbf{y}_i$ is the `softmax` output of classifier i, where i ranges over the indices of the classifiers in the bag.

• UNANIMOUS uses the same ensemble as the bagging classifier, but its output is only valid when all the sub-classifiers are in individual agreement, i.e.:

$$\text{UNANIMOUS}(\mathbf{x}) = \begin{cases} \hat{\mathbf{y}} \text{ if all sub-classifiers output } \hat{\mathbf{y}} \\ \text{undefined otherwise} \end{cases}$$

Thus results for this classifier omit any pages for which sub-classifiers disagree. We also define a variant of this classifier, UNANIMOUS-k, where the prediction is defined wherever all but k classifiers in the bag agree. This is well defined and unique as long as k is less than half of the number of classifiers in the bag.

• ORACLE is defined as the classifier obtained when the pages UNANIMOUS is undefined on are given to an oracle and labelled perfectly. This is analogous to the algorithm asking a human expert for a label.

These four classifiers are considered for both the single-page and multi-page variants of the neural architecture, as defined in the previous chapter.

5.2 Basic Performance

This section quantifies the performance of the four classification modes. The measure of performance used is the **F1 score**, which is a metric suitable for unbalanced data sets, as it takes into account both the precision and the recall of a classifier. The F1 score is defined mathematically, for each output label, as

$$\text{F1} := 2\,\frac{\text{precision} + \text{recall}}{\text{precision} \cdot \text{recall}}$$

For a given output label, precision is the ratio of the number of true positive predictions to the total number of predictions of that label. Meanwhile, recall is the ratio of the number of true positive predictions to the number of positive instances in the sample.

The performance of the UNBAGGED and BAGGED classifiers can be seen in in figure Figure 5–1. Each boxplot corresponds to the distribution of cross-validated F1 scores across all of the patients in the training set. The blue line of the boxplot corresponds to the median. The performance of the best variants can be seen to have median F1 scores of over 0.8 for the Wake and REM stages, and of over 0.9 for the N3 stage. N2 performance is more modest, at just under 0.8, while N1 performance is the poorest, at just under 0.5 F1 score, with a fairly wide spread. N1 is traditionally the most difficult stage to classify, as will be seen in the next chapter, where these results are compared with what is found in the literature.

It is apparent that unbagged single-page performance leaves a lot to be desired, and that the multi-page classifier performs much better than the single-page variant. This is particularly true of the performance on N1. N1 is the most underrepresented sleep stage, which is often indicative of difficulty of classification. Furthermore, N1 is a transitional stage between Wake and N2, and is defined in terms of markers of those stages in the AASM standard [31]. For that reason, it stands to reason that having access to neighbouring pages would lead to an improvement in predictive performance on N1. It should be noted that the F1 scores in Figure 5–1 are not weighted in the same way the respective stages were during training. If one applies such a weighting to the calculation of the F1 score,

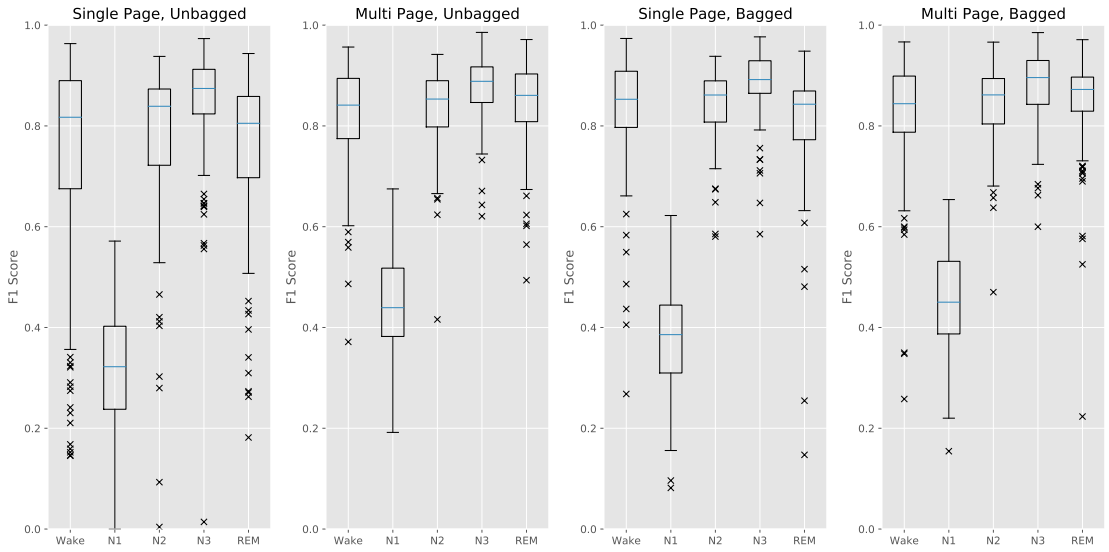

Figure 5–1: The performance of the UNBAGGED and BAGGED classifiers for single-page and multi-page classification.

N1 performance goes up to around 0.7 at the expense of the other stages, owing to the fact that N1 recall is much better than its precision. This suggests that the relative importance of stages as encoded by the training weights carries over to the predictor's behaviour, which is in general a desirable property.

Bagging also leads to significant improvements in the single-page classifier, but its effect is much less pronounced in the multi-page case, and in fact can be seen to drag down the performance of the poorly-performing outliers. This suggests two things. First, because bagging is a variance reduction technique, errors amplified by its are likely from consistent bias in the predictors. Therefore, it is likely the patients whose performance decreases with bagging contain sleep stage markers from a markedly different distribution than the rest of the patients. Second, because bagging leads to such modest gains in the multi-page case, the multi-page

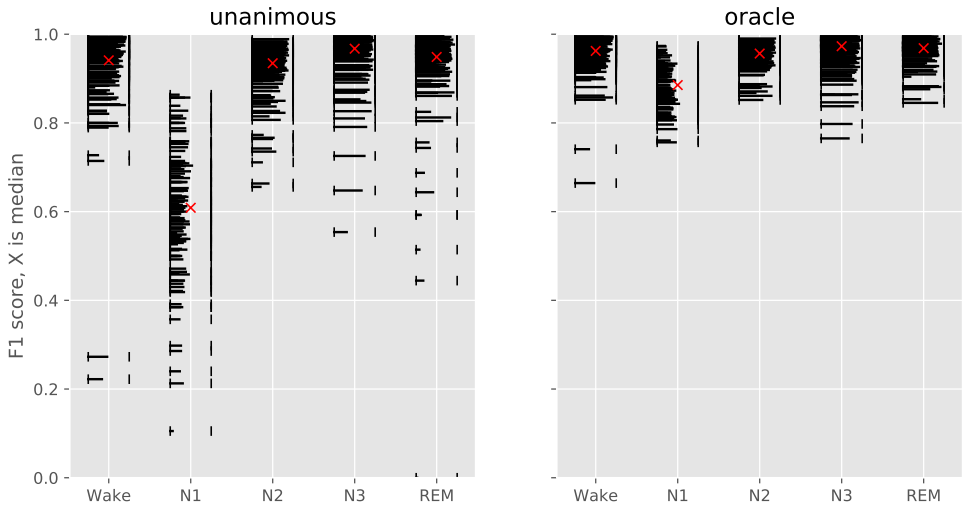

Figure 5–2: The performance of the UNANIMOUS and ORACLE classifiers. The horizontal lines represent, for that patient and for that stage, the ratio of the number of times that stage was predicted unanimously to *the number of times that stage was predicted by the bagged classifier*, and should be interpreted as filling in between 0 on the left and 1 on the right.

architecture's hypothesis class provides an intrinsically good, low-variance fit to the data distribution at hand. Considering that the performance of the architecture is quite strong, suggesting a low bias, the architecture can be judged as a good choice for the sleep stage prediction task with spectrogram inputs.

5.3 The UNANIMOUS and ORACLE Classifiers

Given the significant performance gain obtained by using the multi-page classifier, the UNANIMOUS and ORACLE classifiers will be considered only for the multi-page case. In Figure 5–2 we can see the performance of the these two classifiers. The left-hand plot shows the same F1 scatter as in Figure 5–1, but *only considering the pages for which the sub-classifiers were in unanimous agreement*. The horizontal lines indicate the proportion of pages for the given

patient and the given sleep stage for which the sub-classifiers were in unanimous agreement and hence UNANIMOUS was defined, as a proportion of the number of times that stage was predicted by BAGGED. We can see that performance increases across the board, even for N1, and that for the most well-predicted stages, such as N2 and N3, the degree of unanimity appears to positively correlate with predictive accuracy. This positive correlation is an indicator that agreement between classifiers in the bag does give a measure of model uncertainty. The wide spread on the N1 is consistent with the stage having a lower precision, where ambiguous non-N1 cases are likely to be classified as N1.

The right-hand column of Figure 5–2 shows the approximate F1 scores of the hypothetical ORACLE classifier. The shown figures are given according to the following formula:

$$\mathrm{F1}_o = \mathrm{F1}_u(\text{proportion unanimous}) + (1 - \text{proportion unanimous})$$

where $\mathrm{F1}_o$ is the approximate F1 score of the oracle classifier and $\mathrm{F1}_u$ is the F1 score of the UNANIMOUS classifier. It can be seen that median performance inches up only slightly toward 1 when compared to UNANIMOUS for all stages except N1 owing to the already strong performance of UNANIMOUS on those stages. However, the spread for all the stages decreases significantly. This suggests that UNANIMOUS is able to discern patients on which it would perform poorly and refers them to the oracle consistently. This is also the case for N1, though for that stage UNANIMOUS refers almost all cases to the oracle. Given that N1 is the least prevalent stage (as shown in Fig 4–1), this may be feasible to handle in the clinical workflow. It is

interesting to note that when the classifier is trained without sample weighting to correct for the underrepresentation of the N1 stage, the correlation between the degree of unanimity and oracle performance for N1 becomes strongly negative. This means the classifier's unanimous predictions of N1 are low quality.

5.4 Bagging as a Measure of Uncertainty

To further validate the use of the degree of unanimity among the sub-classifiers of BAGGED as a measure of classifier certainty, one can look at the performance of BAGGED vs. the number of sub-classifiers in agreement with the aggregate vote. This result can be seen in Figure 5–3. The top plot shows the accuracy, broken down by sleep stage, of the output of BAGGED vs. the number of constituent sub-classifiers that agreed with the output of BAGGED. We can see a clear linear relationship upwards for all stages from the point at which 3 of six classifiers agree. This strongly indicates that the agreement among the sub-classifiers is representative of the certainty BAGGED has in its result. The bottom graph shows the number of times a particular number of sub-classifiers agreed with the majority, segregated by sleep stage and aggregated across the validation patients. It is interesting to note that N1's performance as a function of agreement is similar to that of the other stages. Instead, the way in which it stands out is that classifiers in the bag vote for N1 unanimously much more rarely than they do for other stages.

It is natural to consider the above result in the context of the ORACLE-k classifier. In particular, we can choose a minimum number of classifiers in agreement with the final vote as a cutoff for accepting the classifier's result, and

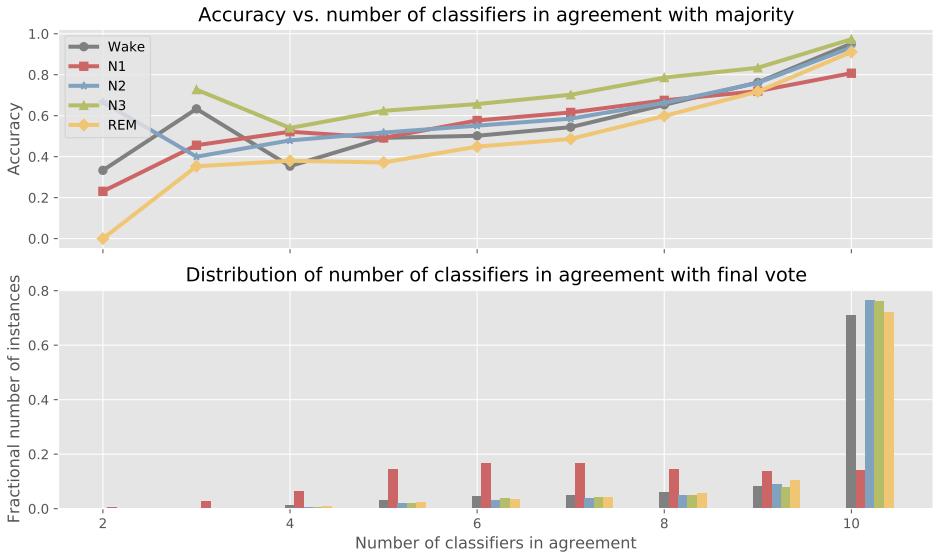

Figure 5–3: The top of these two plots shows the proportion of BAGGED predictions that were correct vs. the number of individual classifiers which agreed with that prediction, segregated by sleep stage. The bottom plot shows the total number of instances for each combination of number of classifiers in agreement and sleep stage.

refer the data to a human otherwise. In this situation, we're interested in the accuracy we obstain vs. the amount of effort a human has to perform. This is illustrated in Figure 5–4. The accuracy is taken as the independent variable, and the proportion of labels done by the human oracle is the response variable, with results broken down by sleep stage. Each marker on the curve corresponds to a different agreement cutoff k required to accept the classifier's result. We can see that for most stages, the accuracy increases with k more rapidly than the proportion of labels done by the oracle. In fact, for all stages but N1, we can attain 0.95 accuracy.

An important caveat in interpeting the results above is that the accuracy values obtained by the classifier are higher than the interrater agreement rate.

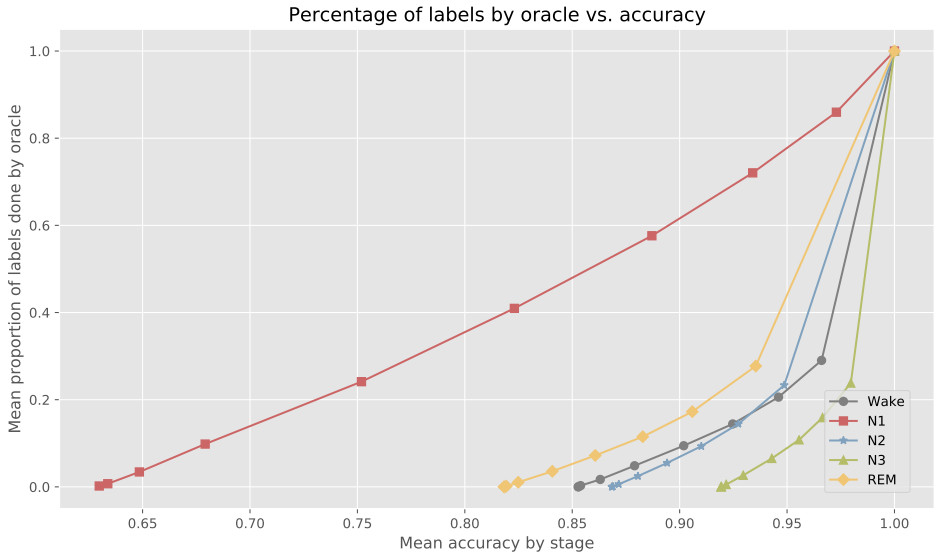

Figure 5–4: For the ORACLE-k classifier, the accuracy attained, by stage, vs. the proportion of labels done by the oracle. The top right is oracle only labelling at perfect accuracy. The leftmost point is the performance of BAGGED. Each point in the middle corresponds to a value of k, i.e. a cutoff of at most k classifiers in the bag disagreeing with the final vote.

With the exception of N1, we see we can get most of the way toward oracle performance with only a fraction of the labels done by the oracle, consistent with the observation that the highest performing stages also had the highest degree of unanimity.

CHAPTER 6
Discussion and Conclusion

6.1 Summary of Contributions

This thesis presented a deep learning approach to automated sleep staging using convolutional neural networks over spectrograms created from multi-channel EEG recordings. The CNN architecture was trained and validated on a dataset of around 120 patients' EEG recordings collected in a hospital setting. It proved capable of attaining competitive performance without the use of hand-crafted features. It is believed this is a novel approach for sleep staging, and an argument has been presented for its strengths relative to other approaches found in the literature.

Furthermore, it was shown that the use of bagging, in addition to improving the accuracy of the method, could be used as a reliable gauge of its confidence in its predictions. Thus it is a suitable candidate for clinical usage where a human technician labels only those EEG pages. It further explored and validated the use of a bagged ensemble classifier to not only reduce the variability in its predictions, but to quantify its degree of confidence of the predicted label.

6.2 Observations on Classifier Performance

On the whole, the performance of the architecture developed in this work is good enough that its refinement and application in the field are plausible goals. However, a number of deficiencies should be remedied before its use can be

seriously considered. It is evident that the predictor still struggles with N1 classification. It is conjectured that N1's definition in terms of WAKE characteristics on a patient-to-patient basis, as well as its substantial underrepresentation, are explanations for the poor performance. On the other hand, when using BAGGED, REM, N2 and N3, and WAKE are classified with very high accuracy, further approaching unity under the UNANIMOUS or ORACLE paradigms.

6.3 Inter-rater Agreement

An important caveat must be taken into account when interpreting numerically the classifier's performance. The dataset used in this paper contains only a single set of sleep stage labels per file, the vast majority of which come from a single technician. A study by Danker-hopfe et al. [17] found that under the 2007 AASM standard, inter-rater agreement for a cohort of 72 majority healthy patients stood at 0.82 (Cohen's kappa of 0.76). That study found that, perhaps unsurprisingly, it was the N1 stage that had the lowest inter-rater agreement (Cohen's $\kappa = 0.46$). This results suggests that the classifier's performance is at its useful limit, even for the N1 stage, when training on labels generated by a single technician. This is a limiting factor in accurately assessing the performance of the architecture described in this thesis. While the classifier's ability to reproduce a single rater's results is a good benchmark for the its suitability for the task and ability to learn a good data representation, to achieve truly reliable and potentially superhuman classification, data scored by two or more raters will be necessary in training.

6.4 Future Directions

The architecture presented in this thesis can be seen as a proof of concept for the proposed, novel method of treating spectrograms likes images. I showed that applying standard CNN based image processing techniques to spectrograms for the purpose of sleep stage classification is a viable technique. However, a lot of work must be done before this method can be used in practise.

6.4.1 Architectural Changes

Most importantly, the method must be expanded to accommodate non-EEG recordings where available. Indeed, one of the strengths of deep methods is the ability to integrate information from different sources in what amounts to being a single, complex nonlinear transformation. Thus the incorporation of other commonly found signals, such as EMG, EKG or other time-series readouts should be straightforward. For many lower-frequency such signals, the use of the convolution would likely be neither necessary nor useful.

It is important to note that the classifier's performance should degrade gracefully in the absence of any of the signals presented above to be maximally useful. In general, for clinical practise, it is ideal that a packaged piece of software incorporating a trained version of the architecture should be transferable between different sites and different data collection methodologies and available signals. It should not need much, or any training to be able to make predictions. This would put it on the same footing as current commercial solutions. At the same time, by virtue of the fact that a deep learning architecture is used, if training data is

available at a new site, the ability to fine tune the weights post installation is a significant advantage.

Even considering EEG in isolation, training with something like "EEG channel dropout", where a subset of EEG channels is used in each sample, should make the model more capable of handling the varying collections of EEG channels available at each clinical deployment site. More fundamentally, the input to the neural stack should be restructured so that location of EEG electrodes is reflected in its structure. Currently, EEG channels are ordered as layers arbitrarily, which actually complicates training with respect to extra or missing EEG channels. Perhaps instead of using a one-dimensional "stack" of EEG channels analogous to colour channels, a two-dimensional grid corresponding to positions on the scalp can be used. In this case the input would be a four-dimensional tensor, which while more complex should still be within the scope of standard deep learning tools.

Finally, a straightforward architectural improvement would be to use a recurrent neural structure on top of the convolutional and fully connected layers. I conjecture this can replace the multi-window technique and be both faster and more accurate, by keeping a much longer context around each page to be predicted.

Methodological Improvements

No matter how good the architecture is, it will not be used in practise if clinicians lack the confidence in its results. The bagging confidence method was developed for this reason. In order for the architecture to be truly viable, the architecture must be trained on a larger collection of patients from different

hospitals, whose measurements were taken by different technicians with different tools. Only when the performance of the program on a totally new set of data is quantified will it be reasonable to have confidence that it can function as a drop-in tool.

With regard to the inter-rater agreement problem, by combining the labels of multiple experts, the accuracy and the confidence model can be refined significantly, and truly superhuman performance can in principle be attained, as it has been in image classification [25]. In order to achieve this, a training set with multiple independent labels per page is desirable. Such a data set would need to be created for the sole purpose of training the architecture, and would represent a significant investment, given the high cost of human labelling of sleep pages.

I believe that such an investment, however, is manifestly justified. This is because I believe that a variant of the architecture presented in this work, if developed with care, will ultimately be able to surpass human performance with confidence.

References

[1] Somnolyzer website. http://www.usa.philips.com/healthcare/product/HC1076888/somnolyzer-24x7-sleep-scoring-software. Accessed: 2017-08-13.

[2] Acharya, U. R., Chua, E. C.-P., Chua, K. C., Min, L. C., and Tamura, T. Analysis and automatic identification of sleep stages using higher order spectra. *International journal of neural systems*, 20(06):509–521, 2010.

[3] Agargün, M. Y. Sleep disorders: diagnosis, management, and treatment. a handbook for clinicians. *Acta Psychiatrica Scandinavica*, 107(4):320–320, 2003.

[4] Anderer, P., Gruber, G., Parapatics, S., Woertz, M., Miazhynskaia, T., Klösch, G., Saletu, B., Zeitlhofer, J., Barbanoj, M. J., Danker-Hopfe, H., et al. An e-health solution for automatic sleep classification according to rechtschaffen and kales: validation study of the somnolyzer 24× 7 utilizing the siesta database. *Neuropsychobiology*, 51(3):115–133, 2005.

[5] Anderer, P., Moreau, A., Woertz, M., Ross, M., Gruber, G., Parapatics, S., Loretz, E., Heller, E., Schmidt, A., Boeck, M., et al. Computer-assisted sleep classification according to the standard of the american academy of sleep medicine: validation study of the aasm version of the somnolyzer 24× 7. *Neuropsychobiology*, 62(4):250–264, 2010.

[6] Barsky, M. M., Tucker, M. A., and Stickgold, R. Rem sleep enhancement of probabilistic classification learning is sensitive to subsequent interference. *Neurobiology of learning and memory*, 122:63–68, 2015.

[7] Bashivan, P., Rish, I., Yeasin, M., and Codella, N. Learning representations from eeg with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*, 2015.

[8] Bauer, E. and Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2): 105–139, 1999.

[9] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[10] Blackman, R. B. and Tukey, J. W. The measurement of power spectra from the point of view of communications engineeringpart i. *Bell System Technical Journal*, 37(1):185–282, 1958.

[11] Boeve, B. F. Rem sleep behavior disorder. *Annals of the New York Academy of Sciences*, 1184(1):15–54, 2010.

[12] Breiman, L. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[13] Broughton, R. J. Sleep disorders: disorders of arousal? *Science*, 159(3819): 1070–1078, 1968.

[14] Cecotti, H. and Graeser, A. Convolutional neural network with embedded fourier transform for eeg classification. In *Pattern Recognition, 2008. ICPR*

2008. 19th International Conference on, pages 1–4. IEEE, 2008.

[15] Chollet, F. Keras. `https://github.com/fchollet/keras`, 2015.

[16] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[17] Danker-hopfe, H., Anderer, P., Zeitlhofer, J., Boeck, M., Dorn, H., Gruber, G., Heller, E., Loretz, E., Moser, D., Parapatics, S., et al. Interrater reliability for sleep scoring according to the rechtschaffen & kales and the new aasm standard. *Journal of sleep research*, 18(1):74–84, 2009.

[18] Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[19] Ebrahimi, F., Mikaeili, M., Estrada, E., and Nazeran, H. Automatic sleep stage classification based on eeg signals by using neural networks and wavelet packet coefficients. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1151–1154. IEEE, 2008.

[20] Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

[21] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[22] Grigg-Damberger, M. Why a polysomnogram should become part of the diagnostic evaluation of stroke and transient ischemic attack. *Journal of clinical neurophysiology*, 23(1):21–38, 2006.

[23] Gudmundsson, S., Runarsson, T. P., and Sigurdsson, S. Automatic sleep staging using support vector machines with posterior probability estimates. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, volume 2, pages 366–372. IEEE, 2005.

[24] Haykin, S. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998. ISBN 0132733501.

[25] He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[26] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[27] Hecht-Nielsen, R. et al. Theory of the backpropagation neural network. *Neural Networks*, 1(Supplement-1):445–448, 1988.

[28] Hill, M. The uncertainty principle for fourier transforms on the real line. *University of Chicago*, 2013.

[29] Hillman, D. R., Murphy, A. S., Antic, R., and Pezzullo, L. The economic cost of sleep disorders. *SLEEP-NEW YORK THEN WESTCHESTER-*, 29(3):299, 2006.

[30] Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[31] Iber, C. et al. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications.* American Academy of Sleep Medicine, 2007.

[32] Jacobson, A., Kales, A., Lehmann, D., and Hoedemaker, F. Muscle tonus in human subjects during sleep and dreaming. *Experimental Neurology*, 10(5): 418–424, 1964.

[33] Jasper, H. H. The ten twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375, 1958.

[34] Jones, E., Oliphant, T., Peterson, P., et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

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

[36] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[37] Längkvist, M., Karlsson, L., and Loutfi, A. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012:5, 2012.

[38] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. Face recognition: A convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, 1997.

[39] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324,

1998.

[40] Li, W., Ma, L., Yang, G., and Gan, W.-B. Rem sleep selectively prunes and maintains new synapses in development and learning. *Nature Neuroscience*, 2017.

[41] Luo, G. and Min, W. Subject-adaptive real-time sleep stage classification based on conditional random field. In *AMIA Annual Symposium proceedings*, volume 2007, page 488. American Medical Informatics Association, 2007.

[42] Mander, B. A., Rao, V., Lu, B., Saletin, J. M., Lindquist, J. R., Ancoli-Israel, S., Jagust, W., and Walker, M. P. Prefrontal atrophy, disrupted nrem slow waves and impaired hippocampal-dependent memory in aging. *Nature neuroscience*, 16(3):357–364, 2013.

[43] Mirowski, P. W., LeCun, Y., Madhavan, D., and Kuzniecky, R. Comparing svm and convolutional networks for epileptic seizure prediction from intracranial eeg. In *Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on*, pages 244–249. IEEE, 2008.

[44] Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[45] Nunez, P. L. and Srinivasan, R. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[46] Pan, S.-T., Kuo, C.-E., Zeng, J.-H., Liang, S.-F., et al. A transition-constrained discrete hidden markov model for automatic sleep staging. *Biomed. Eng. Online*, 11:52, 2012.

[47] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct): 2825–2830, 2011.

[48] Rao, M. N., Blackwell, T., Redline, S., Stefanick, M. L., Ancoli-Israel, S., Stone, K. L., in Men (MrOS) Study Group, O. F., et al. Association between sleep architecture and measures of body composition. *Sleep*, 32(4):483–90, 2009.

[49] Rechtschaffen, A. and Kales, A. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. *US Government Printing Office, US Public Health Service*, 1968.

[50] Saper, C. B., Fuller, P. M., Pedersen, N. P., Lu, J., and Scammell, T. E. Sleep state switching. *Neuron*, 68(6):1023–1042, 2010.

[51] Sassin, J., Parker, D., Mace, J., Gotlin, R., Johnson, L., and Rossman, L. Human growth hormone release: relation to slow-wave sleep and sleep-waking cycles. *Science*, 165(3892):513–515, 1969.

[52] Sinha, R. K. Artificial neural network and wavelet based automated detection of sleep spindles, rem sleep and wake states. *Journal of medical systems*, 32 (4):291–299, 2008.

[53] Smagula, S. F., Stone, K. L., Redline, S., Ancoli-Israel, S., Barrett-Connor, E., Lane, N. E., Orwoll, E. S., and Cauley, J. A. Actigraphy-and polysomnography-measured sleep disturbances, inflammation, and mortality among older men. *Psychosomatic medicine*, 2016.

[54] Song, Y., Blackwell, T., Yaffe, K., Ancoli-Israel, S., Redline, S., and Stone, K. L. Relationships between sleep stages and changes in cognitive function in older men: the mros sleep study. *Sleep*, 38(3):411–421, 2015.

[55] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[56] Stober, S., Cameron, D. J., and Grahn, J. A. Using convolutional neural networks to recognize rhythm stimuli from electroencephalography recordings. In *Advances in neural information processing systems*, pages 1449–1457, 2014.

[57] Szelenberger, W., Niemcewicz, S., and DAbrowska, A. J. Sleepwalking and night terrors: psychopathological and psychophysiological correlates. *International Review of Psychiatry*, 17(4):263–270, 2005.

[58] Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[59] van der Helm, E., Yao, J., Dutt, S., Rao, V., Saletin, J. M., and Walker, M. P. Rem sleep depotentiates amygdala activity to previous emotional experiences. *Current Biology*, 21(23):2029–2032, 2011.

[60] Van Dongen, H. P., Maislin, G., Mullington, J. M., and Dinges, D. F. The cumulative cost of additional wakefulness: dose-response effects on neurobehavioral functions and sleep physiology from chronic sleep restriction and total sleep deprivation. *SLEEP-NEW YORK THEN WESTCHESTER-*, 26(2): 117–129, 2003.

[61] Wains, S. A., El-Chami, M., Lin, H. S., and Mateika, J. H. Impact of arousal threshold and respiratory effort on the duration of breathing events across sleep stage and time of night. *Respir Physiol Neurobiol*, 237:35–41, Mar 2017.

[62] Walker, M. P., Liston, C., Hobson, J. A., and Stickgold, R. Cognitive flexibility across the sleep–wake cycle: Rem-sleep enhancement of anagram problem solving. *Cognitive Brain Research*, 14(3):317–324, 2002.

[63] Welch, P. D. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[64] Young, T., Peppard, P. E., and Gottlieb, D. J. Epidemiology of obstructive sleep apnea: a population health perspective. *American journal of respiratory and critical care medicine*, 165(9):1217–1239, 2002.