

ABSTRACT

Title of Thesis: MULTI-FEATURE ANALYSIS OF EEG SIGNAL ON SEIZURE PATTERNS AND DEEP NEURAL STRUCTURES FOR PREDICTION OF EPILEPTIC SEIZURES

Xinyuan Ma, Master of Science, 2020

Thesis Directed By: Professor Robert Wayne Newcomb, Department of Electrical and Computer Engineering

This work investigates EEG signal processing and seizure prediction based on deep learning architectures. The research includes two major parts. In the first part we use wavelet decomposition to process the signals and extract signal features from the time frequency bands. The second part examines the machine learning model and deep learning architecture we have developed for seizure pattern analysis. In our design, the extracted feature maps are processed as image inputs into our convolutional neural network (CNN) model. We proposed a combined CNN-LSTM model to directly process the EEG signals with layers functioning as feature extractors. In cross validation testing, our CNN feature model can reach an accuracy of 96% and our CNN-LSTM model could reach an accuracy of 98%. We also proposed a matching network architecture which employs two parallel multilayer channels to improve sensitivity.

MULTI-FEATURE ANALYSIS OF EEG SIGNAL ON SEIZURE PATTERNS
AND DEEP NEURAL STRUCTURES FOR PREDICTION OF EPILEPTIC
SEIZURES

by

Xinyuan Ma

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2020

Advisory Committee:
Professor Robert W. Newcomb, Chair
Professor Raj Shekhar
Professor Armand M. Makowski

© Copyright by
Xinyuan Ma
2020

Acknowledgements

I want to thank everyone who has given me help to make this thesis to be public.

I owe my deepest gratitude to my parents along this process. Without the unconditional love and support from them, I could not have completed this thesis. I thank the sacrifices my parents have made along the way. I'm still on the way to get it back to them. I'm grateful for their encouragement and caring during this period and I hope I will do my best to achieve further accomplishments to make them proud of me.

I also want to thank my advisor Prof. Newcomb. He has given me the research training which no one could have ever given. I want to thank my advisor Dr. Raj Shekhar. It is in his lab I got the original research idea and dataset and developed this project. I thank Prof. Makowski for his meticulous effort in commenting on the important details of my thesis. I appreciate the help from the administration staff who has coordinated the document processing and the professors in our school who have provided me with guidance in completing the thesis.

Table of Contents

Acknowledgements.....	ii
Table of Contents	iii
Chapter 1: Introduction	1
1.1 The Seizure Prediction Study.....	1
1.2 Related Research.....	2
1.3 Our Contributions	4
Chapter 2: Discrete Wavelet Decomposition and Feature Extraction	7
2.1 Introduction to the Dataset.....	7
2.2 Discrete Wavelet Decomposition	9
2.3 Feature Extraction.....	13
Chapter 3: Spatial and Temporal Network Structures for EEG Signal Classification	16
3.1 Fully Connected Neural Networks.....	16
3.2 Convolutional Neural Network for Spatial Signal Inputs.....	18
3.3 Recurrent Neural Network for Temporal and Sequential Signal Inputs.....	21
3.4 LSTM EEG Classification Structure	24
Chapter 4: Combined Convolutional Neural Network and LSTM for EEG Seizure Prediction	26
4.1 Combined CNN-LSTM network	26
4.2 Matching Network Architecture for EEG Epoch Testing.....	28
Chapter 5: Experimental Analysis and Comparative Evaluations.....	36
5.1 Implemented Dataset Illustration	36
5.2 CNN and CNN Feature Model Comparison and Determination	36
5.3 CNN-LSTM Model Structure Determination.....	39
5.4 CNN-LSTM and CNN Comparative Experiments on 10 Patient Cases.....	41
Chapter 6: Conclusions and Future Work.....	44
Appendix I	46
Appendix II.....	47
Bibliography	48

Chapter 1: Introduction

1.1 The Seizure Prediction Study

The Electroencephalogram (EEG) signal monitors the complex electrical behavior of the brain. The electrical impulses between brain cells are extended to the surface of the scalp so that the signals are measured through electrodes placed on the scalp. The EEG signals are analyzed through the following waves: Delta waves ($< 4Hz$), Theta waves ($4Hz - 8Hz$), Alpha waves ($8Hz - 12Hz$) and Beta waves ($12Hz - 30Hz$). Each frequency band focuses on the electrical behaviors in different regions of the brain. For example, the Beta waves are predominant in the behaviors of the frontal portion of the brain while the Alpha waves mainly occur in the posterior region. The distinguishable feature of the multi-channel EEG signal makes it an ideal tool to explore different brain activities, especially abnormal symptoms in the brain [1].

Seizure is a central nervous system disorder that derives from aberrations in electrical brain activities. Recurrent and unpredictable seizures can damage the nervous system and even result in death. As one of the most effective ways to analyze scalp electrical signals, EEG signals with multiple channels monitoring different regions of the brain have significant uses in seizure studies [2]. The characteristics of EEG signals vary largely from patient to patient, hence, the seizure patterns from patient to patient usually differ as well. The variability of seizure patterns among patients increases the difficulty of seizure recognition.

Brain activities are complicated and highly random, and the primary indicator of the brain's electrical behaviors—EEG signals—are non-Gaussian and nonstationary. For

seizure analysis, instead of studying the EEG signals from either purely time or frequency domains, researchers have found that a time-frequency (TF) analysis could provide a method to extract features that outperforms conventional studies [3]. To learn the representations of EEG signals from a TF approach, automatic EEG signal classification has a significant advantage in the sheer scale of cases it could process. Further, studies have shown its high and increasing accuracies with respect to different classification algorithms. The learning process usually involves raw data processing, feature extraction, model learning, and final prediction. This process is illustrated in Figure 1.1.

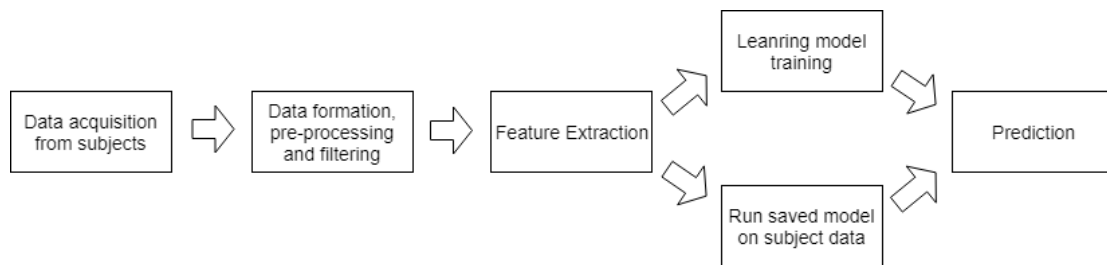


Figure 1.1 EEG Signal Prediction Model Training

1.2 Related Research

Subha et al. explored EEG signal analysis methods, with an emphasis on time-frequency based approaches. In the time domain, linear prediction (LP) and independent component analysis turn out to be effective tools for signal extraction by reducing input signal dimension [1]. For time frequency methods, wavelet transforms demonstrate significant performance, while both continuous and discrete transforms have useful applications respectively. Other methods including higher order statistics, state space reconstruction, correlation dimension, and entropy approaches have also been used.

Subasi et al. proposed a discrete wavelet transform (DWT) strategy followed by dimension reduction algorithms applied directly on the decomposed signals [4]. The results show high testing accuracy with the DWT process. They also conducted experiments comparing principle component analysis (PCA), independent component analysis (ICA), and linear discriminant analysis (LDA) methods. Their general normal EEG signal classification rate for simple classes reached as high as 98% in certain experiments.

Instead of studying the signals directly on the decomposed bands based on DWT, Liu and his colleagues developed a multi-feature extraction strategy from the sub-bands from decomposition. This extraction strategy explores the EEG signal in different key perspectives including fluctuation, relative amplitudes, energy distribution, and variation. The results give high accuracy with 19 out of 21 testing cases above 90% [5].

For classifiers, Bashivan and his colleagues developed a recurrent convolutional neural networks method for seizure classification. They introduced a 2D mapping for the 3D coordinates of the electrodes placed on the patient scalp. Then they use the mapping as the input to convolutional neural network (CNN) models. With cubic interpolation, the mapping is turned into an image for classification. The ImageNet by Krizhevsky is a neural architecture employed with long-short term memory units (LSTM) at the final layer. The classifier performs at a high sensitivity of over 85% which is significantly higher than the results obtained by traditional classifiers [6] [18].

On the deep learning architecture side, the human learning process has inspired the idea of taking small training samples to learn a problem, a mechanism in which the matching network conducts few shot or one-shot learning. Oriol and colleagues proposed an

architecture by matching the features from embedding functions through an attention mechanism. The results are encouraging on alphabet image classification [19]. Their image classification performance could range from 60% to 98% for certain image groups, and with a large quantity of training samples.

1.3 Our Contributions

The study of EEG seizure detection faces difficulties on several fronts. The current works focus on patient-specific detection rather than on generic seizure detection. Although specific training and classification make the algorithms more efficient, the application of the detection algorithms is limited. Tests have shown that the classifier trained for one patient performs much less efficiently on another patient. Another difficulty lies in the debate over feature extraction strategies. There are multiple approaches to EEG signal feature extraction, from time frequency approach to use of higher order statistics. However, there is no clear evidence as to which feature extraction combination could represent the most relevant information to seizure patterns. Hence, study of the automatic feature learning, selection, and alignment strategy for seizure detection is in high demand. Another problem is that the seizure data sets are usually imbalanced in terms of the seizure-to-normal phase ratios, as most patient cases have only several minutes of seizure onset duration over the course of hours of monitoring. For cases in which the seizure samples are sparse, a well-designed, specific learning architecture has yet to be developed.

In this work, our three main contributions are:

- a) We introduce a discrete wavelet transform-based feature extraction strategy.

From the decomposed bands on interested frequency range, we design multiple

feature vectors for all channels of the EEG signal. This feature alignment combined with convolutional neural network models achieves high performance in comparative experiments.

- b) We designed a combined CNN-LSTM model for EEG feature extraction and seizure prediction. A convolutional neural network-based feature extractor is proposed to extract distinguishable features from convolutional operations. A 1D sliding filter window is introduced to the convolution layers, and the preserved temporal information from the CNN layers is fed into the LSTM layer for epoch prediction. This approach aims at reducing the complexity and blindness of selecting and computing features from background knowledge and signal processing techniques.
- c) We propose a matching network learning architecture to implement reinforcement learning for seizure prediction based on a feature extractor and deep neural network channels. Within this architecture, the neural networks from each channel are used to conduct metric learning to compare epoch similarities. Through the metric learning process the performance is significantly improved. The networks are synthesized by the attention model to give final distribution.

In the following chapters of this thesis, Chapter 2 introduces the EEG signal dataset that we use and illustrates our wavelet-based feature extraction strategy and feature selection mechanism. Chapter 3 proposes the construction of the CNN and LSTM models and their alignment with the feature maps. Chapter 4 introduces the design of our combined CNN-LSTM model and the matching network architecture in reinforcing

the performance of prediction. Chapter 5 gives the results of experiments and analyzes the comparative advantages of the models. Chapter 6 contains conclusions and ideas for future work.

Chapter 2: Discrete Wavelet Decomposition and Feature Extraction

2.1 Introduction to the Dataset

EEG measures the electrical activity of the brain. By taking the difference of potentials between electrodes, each channel has a signal that tracks the scalp electricity, triggering as continuous voltage variations. Hence, EEG captures the overall electrical activities of millions of neurons. During seizure onsets, a group of EEG channels usually perform rhythmic activities or certain patterns of variations. These activities are composed of different frequency components and are usually specific to individuals.

We would like to give a brief introduction to the EEG signal monitoring of the seizure patients first. For example, Figure 2.1 is a segment of the monitoring record of a patient experiencing seizure onset. In this recording, the seizure starts at 17 seconds from the beginning and behaves a rhythmic waving and significant fluctuation in channels from FP1-F7 to P3-O1. This seizure onset lasts 44 seconds with a similar pattern.

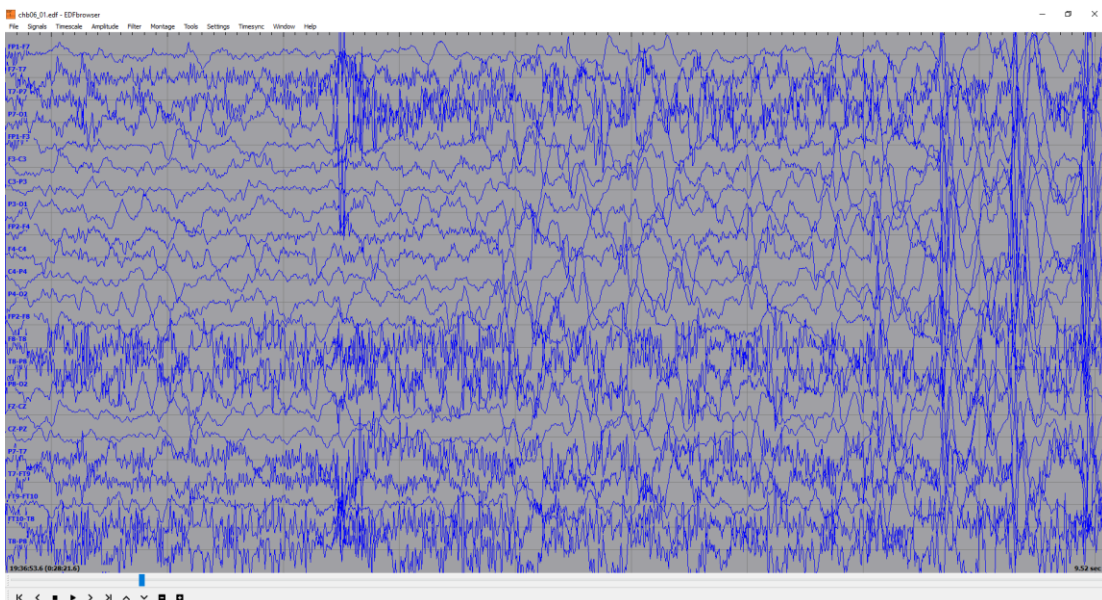


Figure 2.1 Seizure onset EEG Epoch of Patient 1

Seizures in different patients usually behave in different manners. Figure 2.2 shows the EEG signals of another patient with the seizure onset record. The onset is more drastic with spike-like behaviors. It begins with a rise in fluctuation magnitude in channels from FP1-F7 to P3-O1 and CZ-PZ to FT10-T8. The pre-ictal fluctuation stabilizes for a period, and then most channels begin to show significant spike magnitudes.

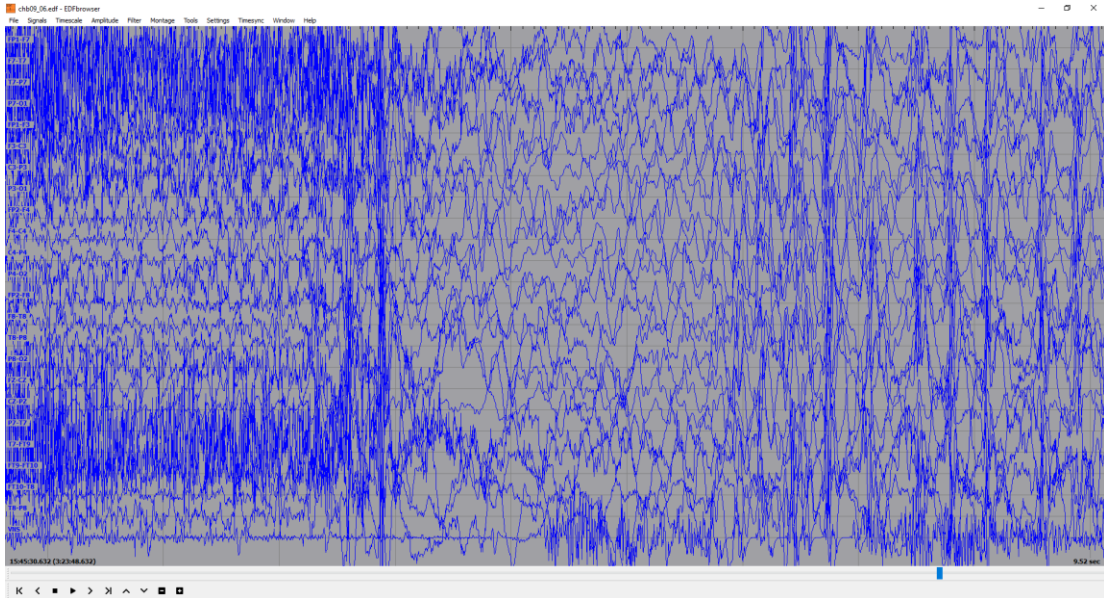


Figure 2.2 Seizure onset EEG Epoch of Patient 2

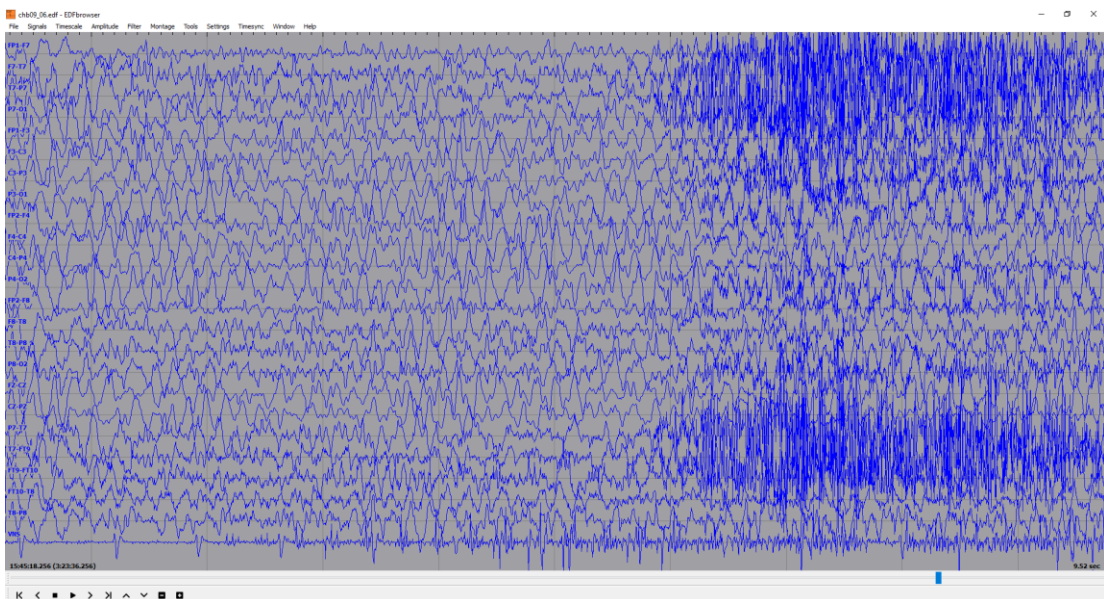


Figure 2.3 Pre-ictal EEG epoch of Patient 2

If we investigate the pre-ictal phase of this seizure onset, as shown in Figure 2.3, the signal frequency rise is distinguishable. The more stationary normal phase behaves rhythmically compared to the seizure phase.

The database we use here is the CHB-MIT scalp EEG database [20] [21]. A total of 24 patient cases with seizure onsets were recorded. The data set contains 844 hours of continuously recorded EEG and 163 seizure onsets. The lengths of seizures usually range from 30 seconds to 1 minute. The sampling frequency is 256 Hz for all channels. The notations from FP1 to P8 represent each electrode placed on the scalp, and the 23 channels analyzed show the voltage differences between different electrodes. The arrangement of the channels is illustrated in Appendix I.

2.2 Discrete Wavelet Decomposition

In traditional Fourier analysis, a periodic and wide-band signal that has high frequency sampling and a long observational period to maintain good resolution in the low frequencies is assumed. Taking the process one step further, the wavelet transform (WT) theory uses signal analysis based on varying scales in the time and frequency domain. It correlates the signal with a dictionary of waveforms that are concentrated in the time and frequency domains. Its ability to extract information for transient signals has outperformed Fourier transforms (FT) in many applications [7] [8].

The WT is described in the terms of its basic functions, called wavelet or mother wavelet. The variable for frequency ω in FT is replaced by scale factor a (which represents the expansion in frequency domain) and the variable for displacement in time is represented by translation factor b . The main characteristic of WT is that it uses a variable window to scan the frequency spectrum, increasing the temporal resolution

of the analysis. For a single analysis, the wavelets based on a mother wavelet ψ are represented by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \quad (2.1)$$

where a and b are the scale and translation parameters, respectively.

The discrete wavelet transform (DWT) is obtained by discretizing the scale and translation parameters of WT. Its waveforms are expressed as:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{a_0^j}} \psi\left(\frac{t - b_0 a_0^j}{a_0^j}\right) \quad (2.2)$$

where $\psi_{j,k}$ shape the wavelet bases and j, k are integer parameters. The form we use in this work is based on powers of 2 scale parameter, which takes $a_0 = 2$ and $b_0 = k$, and the function turns into:

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}} \psi(2^{-j}t - k) \quad (2.3)$$

The DWT makes use of the information redundancy of wavelet transform to shape the time frequency bands. In practice, in many cases it is more efficient to conduct feature extraction at interested frequency ranges from DWT instead of dealing with wavelet transformed images.

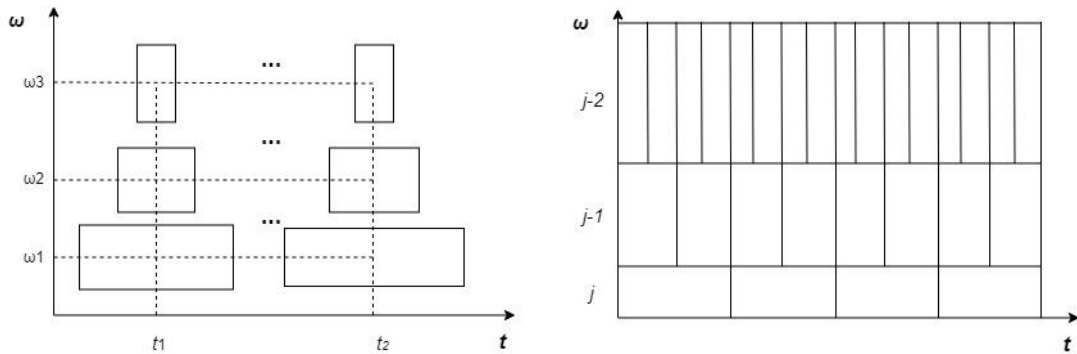


Figure 2.4 Continuous Wavelet Windows (left) and Discrete Wavelet Windows (right)

In WT each wavelet could be treated as a 2D observing window in the time frequency space. When it comes to DWT, the windows are assigned with certain sizes and positions as illustrated in Figure 2.4. The discrete windows fill the whole space. Hence, analysis in separated bands is possible.

To generate the observing windows, there are wavelet function families that function as bases. Typical wavelets such as Molet wavelet, Haar wavelet, and Daubechies wavelets have been proven to work successfully in their specific application fields. In EEG practice, mother wavelets should be chosen according to the properties of the patient recordings and the application scenarios.

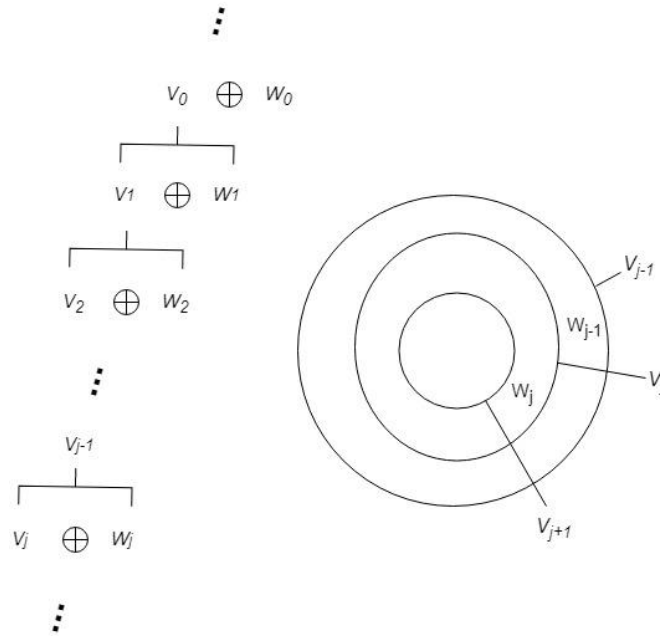


Figure 2.5 Scale Spaces for Wavelet Bases of the Same Mother Function

When the wavelet mother function is determined, the switching of its scale and time translation parameters can be viewed as scaling and moving the functions in the time frequency spaces. In the power 2 discrete wavelet transform we use here, if we define V_j as the scale space of the current function $\psi_{j,k}(t)$, all the time translations of the current function are also in the same scale space. If we shrink the scale of the current function by factor 2 to $\psi_{j+1,k}(t)$, the scale space would be V_{j+1} . From our definition,

we can reason that $V_{j+1} \subset V_j$. We define the space $W_j = V_j - V_{j+1}$, so that there is a sequence of orthogonal spaces.

The frequency spaces of the signals can be viewed as the subspaces in Figure 2.5. And if we define the whole frequency band $(0, \pi)$ as V_0 , the space can be divided into low frequency band $(0, \pi/2)$ as V_1 and high frequency band $(\pi/2, \pi)$ as W_1 .

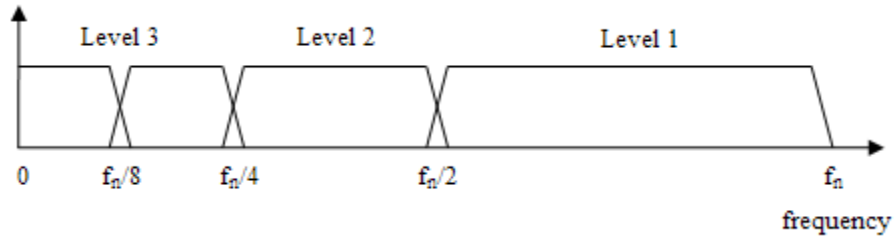


Figure 2.6 Frequency Domain Representation of DWT

We can keep doing the decomposition to the level as required and this division could be denoted as:

$$V_0 = W_0 \oplus V_1 = W_0 \oplus W_1 \oplus V_2 = \dots = W_0 \oplus W_1 \oplus \dots \oplus W_{j-1} \oplus V_j \quad (2.4)$$

Here the high frequency space is W_j . The quality coefficient for the ratio of bandwidth to center frequency remains the same for any j .

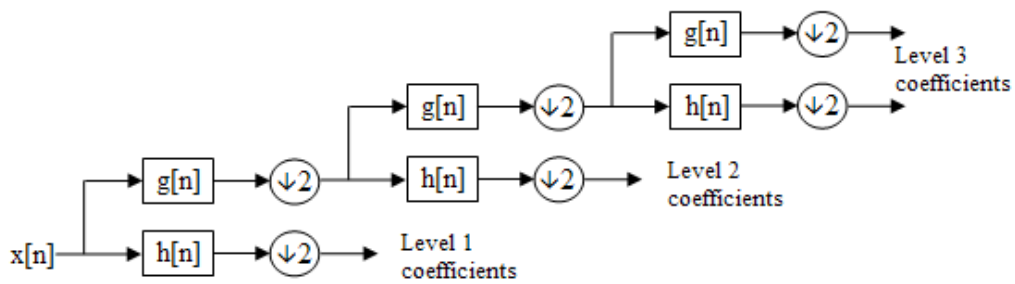


Figure 2.7 Multiresolution Filtering Approach for DWT

And if we treat the decomposition process as a multiresolution filtering process, the low pass and high pass filters remain the same at each scale, since the normalized frequencies are constant. Hence, the discrete wavelet decomposition process could be

implemented by filter banks as shown in Figure 2.7. We employ the multiresolution filtering idea to conduct the decomposition in our approach to process EEG signals.

2.3 Feature Extraction

Major seizures happen from delta to beta waves, from 3 Hz to 29 Hz , in the frequency range of brain waves [9] [10]. From the spectral energy perspective, EEG signals also indicate a redistribution of energy on a set of channels along the process. The change in spectral energy on each channel typically contains a reappearance of frequency components within the $0 - 65\text{ Hz}$ band [11]. The EEG signals we use here are with a sampling rate of 256 Hz , and we apply a 6 scales decomposition to get the approximation coefficient a_6 ($0 - 4\text{ Hz}$) and detail coefficients d_6 ($4 - 8\text{ Hz}$), d_5 ($8 - 16\text{ Hz}$), d_4 ($16 - 32\text{ Hz}$), d_3 ($32 - 64\text{ Hz}$), d_2 ($64 - 128\text{ Hz}$). Figure 2.8 shows the decomposition of two 3 seconds epochs on Patient 10 in our dataset using Daubechies-4 wavelet.

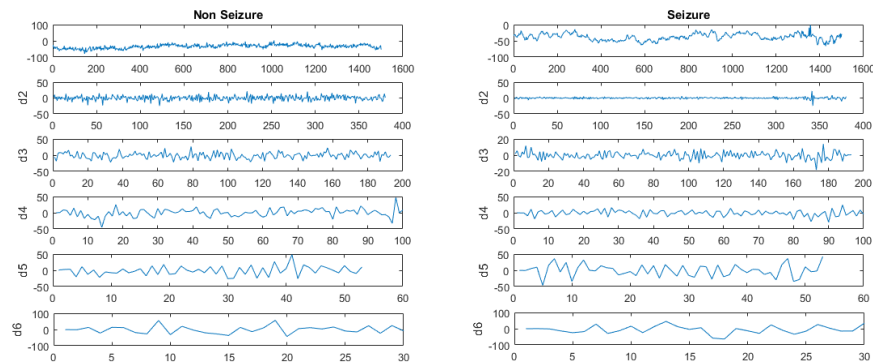


Figure 2.8 Decomposition of Non-Seizure (left) and Seizure (right) Epochs

The features extracted include relative energy, coefficient of variation, fluctuation index, detrended fluctuation index, Shannon entropy, and approximate entropy. They are applied to each channel on selected frequency scales and are then aligned together

to form feature vectors. To introduce a unified notation, from Equation 2.4 to Equation 2.8, l indicates the scale selected, $D_l(i)$ is the detail coefficient of scale l at time index i , and N is the length of vector of each scale.

The relative energy is an indicator of the energy distribution among selected scales.

$$E(l) = \frac{1}{N} \sum_1^N D_l(i)^2 \cdot \tau, E_r(l) = \frac{E(l)}{\sum_1^S E(i)} \quad (2.4)$$

The coefficient of variation is a metric to measure how close the various standard deviations are to the mean value.

$$V(l) = \left(\frac{\sigma(l)}{u(l)}\right)^2, u(l) = \frac{1}{N} \cdot \sum_1^N D_l(i), \sigma(l) = \sqrt{\left(\frac{1}{N}\right) \sum_1^N (D_l(i) - u(l))^2} \quad (2.5)$$

The fluctuation index shows the magnitude of the fluctuation of the signal by comparing adjacent epochs.

$$F(l) = \frac{1}{N} \sum_1^N |D_l(i+1) - D_l(i)| \quad (2.6)$$

The detrended fluctuation index represents the statistical self-affinity of a signal. The time series s segmented into boxes (intervals) with the n_{th} box with the length of $N(n)$.

And the detrended fluctuation is calculated as:

$$DF(l) = \frac{1}{N(n)} \sum_1^{N(n)} |\overline{D_{l,n}} - D_{l,n}(k)|^2 \quad (2.7)$$

Where $\overline{D_{l,n}} = (\sum_{N(n-1)+1}^{N(n)} D_l(i))/N(n)$ and $D_{l,n}(k) = D_{l,N(n-1)+k}$.

Seizure is an abnormal activity of the brain. The Shannon entropy estimator defined below is a disorder indicator measuring how unorganized the signal epoch is.

$$Ent = \frac{\sum Di \cdot \log^*(Di)}{\log(N)}, S = -\log(Ent) \quad (2.8)$$

The EEG signals we use have 23 selected channels, and for each channel we choose 4 scales of frequency bands so that each feature gives a (23, 4) matrix. For vectorization purposes, we align them as column vectors with a length of 92, and stack all 5 feature vectors with respect to the learning models input requirements. For some of our models, the feature vectors are aligned as matrices called feature maps. In our implementation, when feeding the feature vectors into the neural networks, the vectors are normalized by training batches. The methods by which we align the features vectors for our model structures are further illustrated in the next Chapter.

Chapter 3: Spatial and Temporal Network Structures for EEG Signal Classification

3.1 Fully Connected Neural Networks

Our work is based on neural network structures. Different kinds of neural networks applied to different application fields are inspired by the multilayer neural network structure. This type of neural network structure is called “fully connected structure” since it correlates neurons by their connection weights. The multilayer neural network, with its adaptability to different problem dimensionalities, relatively simple structure adjustment operation for fitting requirements, and efficient training costs, has outperformed other traditional classifiers such as linear regressions, kernel estimators, and support vector machines (SVM) [12]. Here we apply a one-hidden-layer neural network to experiment on feature selections at the early stage. The simple fully connected neural network also functions as a method validation for our subsequent models. Since there is no analytical method to determine the number of layers and the number of neurons on each layer, we conduct experiments and compare the results to the experiments from previous methods that have been conducted to design the network for our study [13] [14].

The hidden layer neural network and more sophisticated networks built for specific applications are derived from the basic model of neuron connections. Each neuron in the network works as an activation function of the linear combination of its inputs. As an example, the neuron j in the layer yields an output y_j as:

$$y_j = f\left(\sum w_{ji} \cdot x_i\right) \quad (3.1)$$

where w_{ji} is the weight parameter for the i th input to the j th neuron and x_i is the input vector, while f is the non-linear activation function.

The network generates output, and usually the output is compared with targeted results to indicate the cost of the classification. Minimizing the cost leads to the adjustment of the network parameters, and this optimization process functions as the training process for the network. The cost we use here is the cross-entropy cost:

$$E = -\frac{1}{C} \sum_{n=1}^C [y_n \log(\hat{y}_n) + (1 - y_n) \log(1 - \hat{y}_n)] \quad (3.2)$$

where C is the number of training data classes, y_n is the output for the n th class and \hat{y}_n is the targeted output for the n th class.

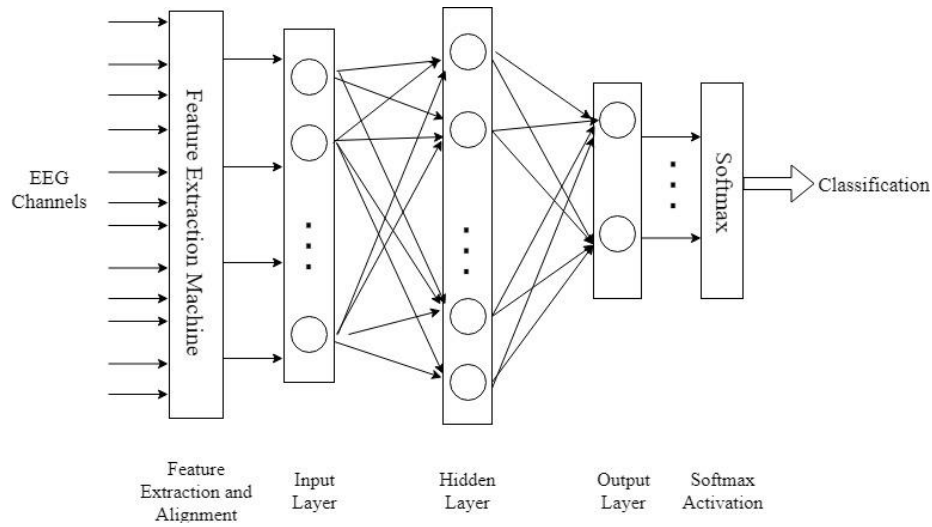


Figure 3.1 Multilayer Neural Network Classifier for DWT Based EEG Features

As shown in Figure 3.1, the EEG epochs are fed into the feature extraction model. The feature extraction model here is the discrete wavelet decomposition model illustrated in Chapter 2. For each epoch, the features are extracted and then these column feature vectors are concatenated as an input vector. The feature vector is fed into the neural network with a hidden layer and the classification results are yielded through softmax

activation. The output is a length 2 vector representing the probability distribution over the seizure and non-seizure classes. For the network training, a cross-entropy cost is applied. In the feature design phase, we use this simple network to test each feature in terms of classification accuracy; the results helped us to determine the five features we would use throughout our experiments (See Chapter 2). This model has a relatively simple structure for making adjustment. Its relatively low training cost saved a great amount of experimentation time. But more importantly, its structure lays the foundation for us to develop more adaptive neural networks to deal with the extracted EEG features.

3.2 Convolutional Neural Network for Spatial Signal Inputs

Convolutional neural networks (CNN) emerge as powerful tools to conduct image related learnings. They have been employed to tackle a variety of real-world problems in identifying objects and powering vision in robots [16]. In our EEG seizure prediction study, we designed CNN models to learn the aligned feature matrices built from the feature vectors to develop seizure prediction machines. And starting from this feature extractor idea, we further applied CNN layers as feature extraction filters to process the EEG signal epochs for better prediction performance.

The architecture of a CNN is based on a sequence of layers. Different from the basic multi-layer neural networks, it operates with 2-D convolution filters to handle images.

The key components of CNNs are:

- Convolution: Convolving previous outputs with 2-D filters.
- Non-linear activation: Non-linear function to activate filter outputs.
- Pooling: Down sampling images to smaller size.

- Fully connected layer: Element-wise weight parameter connection.

A CNN usually operates with typical combinations of the components above. For example, a convolution layer with a non-linear activation followed by a pooling layer is the most significant building module of CNN. This building module would be repeated multiple times to form the CNN layers to the desired depth. At the end of the cycles of convolution, non-linear activation and pooling operations, fully connected layers with activations are added to yield the classification results. Other kinds of layers may be inserted as per the needs of the machine learning tasks, however, they are not necessary for a neural network to be called CNN.

There are various arrangements of layers of CNNs for different tasks. LeNet, proposed by Yann LeCun and his colleagues, laid the foundational framework of CNNs in terms of image classification [17]. The GoogLeNet, incorporated with an inception module, significantly reduced the number of parameters in traditional frameworks while maintaining high performance [18]. The VGGNet is a very deep CNN that showed how the depth of a network could critically determine the performance of the framework [19]. There are other models that have been proposed recently, such as ResNet, DenseNet, etc., which show excellent performance in certain applications [20] [21].

We designed our CNN model with structure and parameters suitable for our EEG feature map size. The model structure is shown in Figure 3.2. Here the activations of the convolution and pooling layers are RELU layers.

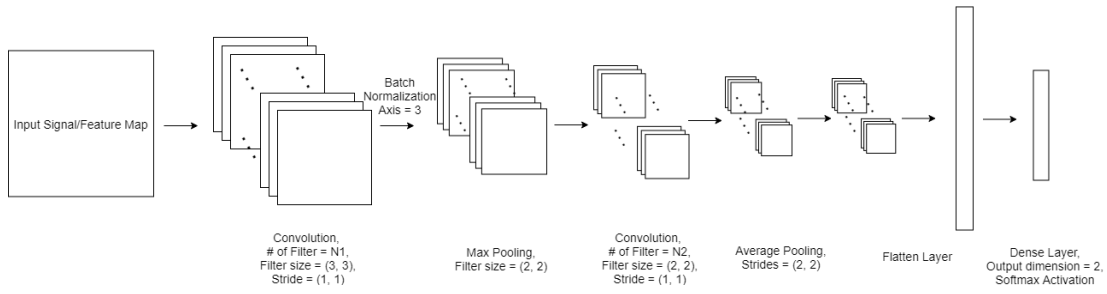


Figure 3.2 Our Design of CNN Model for Seizure Prediction

The filter sizes are chosen to work with both feature maps and raw signal inputs. Moreover, the numbers of filters are chosen in the training experiments as per the requirement of training performance. For input, our model can adapt to two signal input approaches.

The first approach is designed for raw signal input. We process the 23-channel signal into epochs. If the epoch length is 3 seconds, with the sampling frequency of the CHB-MIT dataset at 256 Hz, our input epoch size would be (23, 768). This input would be fed into our model (See Figure 3.2) and train the network through batches.

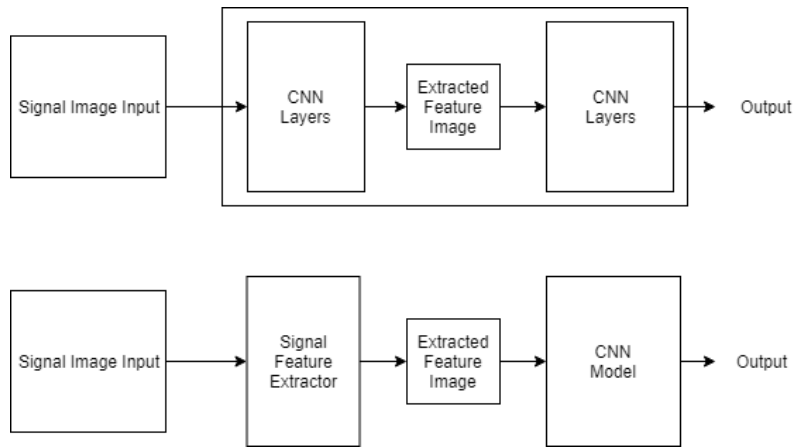


Figure 3.3 CNN and CNN with Signal Feature Extractor for EEG Signals

For the feature map approach (described in Chapter 2) using the EEG signal, we have 23 channels, and for each channel we select four of the decomposed frequency bands. For each band we have five features, hence, the input feature map size is (23, 20).

Figure 3.3 illustrates how these two approaches form two paradigms for CNN model seizure prediction.

In this chapter, we also propose our own CNN frameworks to tackle the seizure classification task. Different from image classifications, the EEG signals are multi-channel nonstationary signals. We applied two approaches: The first was to decompose the signals into multi-channel images with signal processing algorithms. The second was to apply feature extraction techniques to preprocess the signals into images of epochs by rearranging the feature vectors. Because of its convolution and subsampling nature, CNN has a feature extraction ability through multiple layer operations.

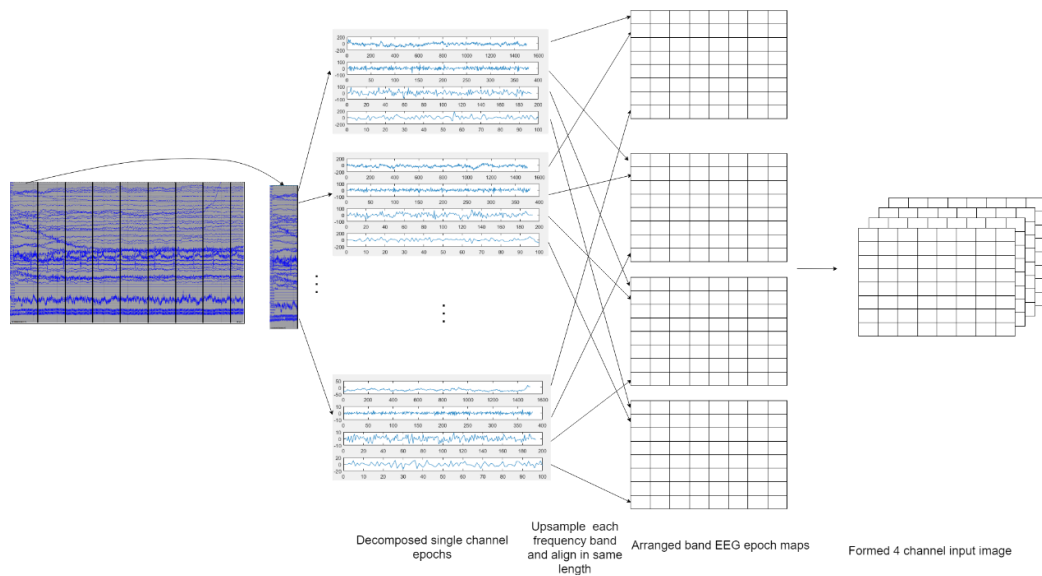


Figure 3.4 The Alignment of Decomposed Frequency Bands

We conducted comparative experiments in Chapter 5 to further analyze the performance of both frameworks.

3.3 Recurrent Neural Network for Temporal and Sequential Signal Inputs

Recurrent neural network (RNN) is a class of artificial neural networks that deals well with sequential data. It has been successfully applied to computational neuroscience

and learning tasks based on time series [15]. Unlike traditional neural networks, RNNs perform the same operation on each element of the sequence and give out an output that is dependent on previous computations.

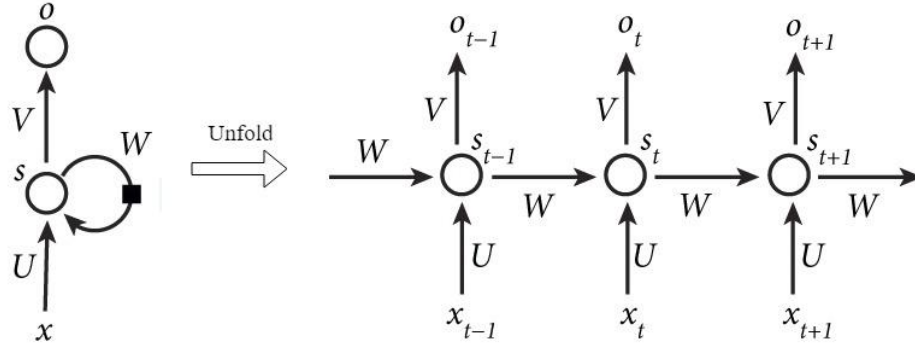


Figure 3.5 Recurrent Neural Network Structure Unfold

The recurrent neural network functions with an inner loop passing hidden states through time steps. For example, at time t , x_t is the input vector, and s_t is the hidden state. The state is obtained from the input and previous state by the relation:

$$s_t = f(Ux_t + Ws_{t-1}) \quad (3.3)$$

Where f is the non-linear activation function. The output o_t usually follows as an activation $o_t = \text{softmax}(Vs_t)$. U, V and W are the unit parameters to be trained.

Unlike a traditional deep neural network, RNN shares the same parameters across units. This largely reduces the number of parameters to train for the same size task. The reason that RNNs function well with far fewer parameters lies in its structure, which enables the states to capture information from previous steps. This works significantly well when dealing with input series which have temporal correlations across successive steps, such as time series and natural language sentences.

Although RNN units capture information from previous steps, the mechanism only works effectively within small ranges across temporal steps. When the temporal

duration of the inter-unit dependencies increases, the temporal contingencies would emerge among the input and output sequence span in the long term [16]. A long-short term memory (LSTM) neural network is proposed to solve this problem by introducing gates that control the information passing through [17].

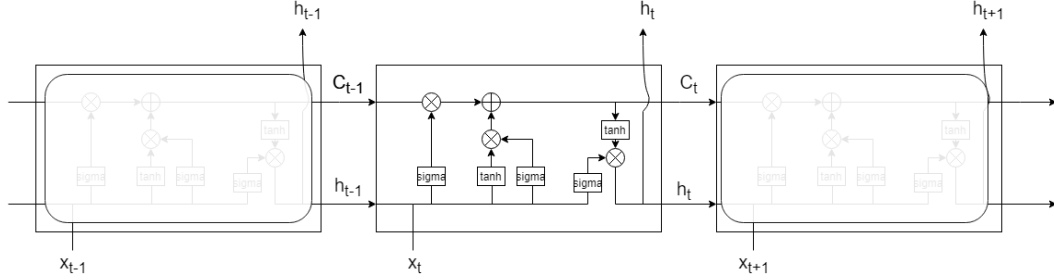


Figure 3.6 Concatenate Long-Short Term Memory Units

There are two classes of states passing through the LSTM units. At time t , the long-term state C_t carries the information that passes through the units without a nonlinear operation, and the unit state h_t outputs the operations within the current unit to the next unit. The forgetting window f_t determines how much of the long-term state should pass through by judging the information from the previous unit state and current input, namely,

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.4)$$

where W_f and b_f are the parameters to be trained of the unit.

To determine the portion to pass through from the short-term unit state, we also have a gate and state given by

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.5)$$

and

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3.6)$$

The new long-term state is then updated

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (3.7)$$

The new output and unit state are from the previous unit state, input and new long-term state, with

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.8)$$

and

$$h_t = o_t \cdot \tanh(C_t) \quad (3.9)$$

3.4 LSTM EEG Classification Structure

Based on the LSTM principles, we build the EEG classification network with LSTM units structured as the units in Figure 3.6. Here we take a fully connected neural network layer to function as the dense layer to take the output of the LSTM layer and form it into a length 2 vector. As illustrated in Figure 3.7, the input vectors could be the feature vectors from the feature extractors or simply vectorized sliced EEG epochs. For example, we can slice 1 epoch into 10 same length pieces along the time axis and vectorize each piece. In our experiments, we always use our LSTM model as a part of our combined model to improve its performance.

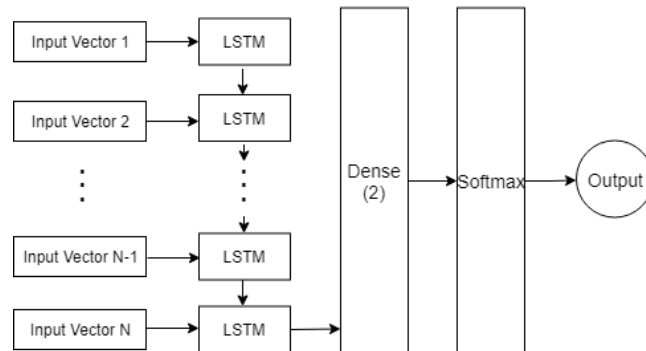


Figure 3.7 LSTM Classification Network

Briefly, we want to further explain the dimensionalities of the input vectors as a preparation for the model proposed in the next Chapter. Taking the feature vector input as an example, in our 23 channels case, within each epoch each channel is decomposed into 4 scales. We select 3 lower detail frequency bands and the approximation band to compute the features. With each band there are 5 features associated. Hence, we have a (23, 4, 5) size feature extracted for one epoch. The details can be found in Figure 3.8. Each input feature vector has length 20, and there are 23 input vectors corresponding to 23 channels.

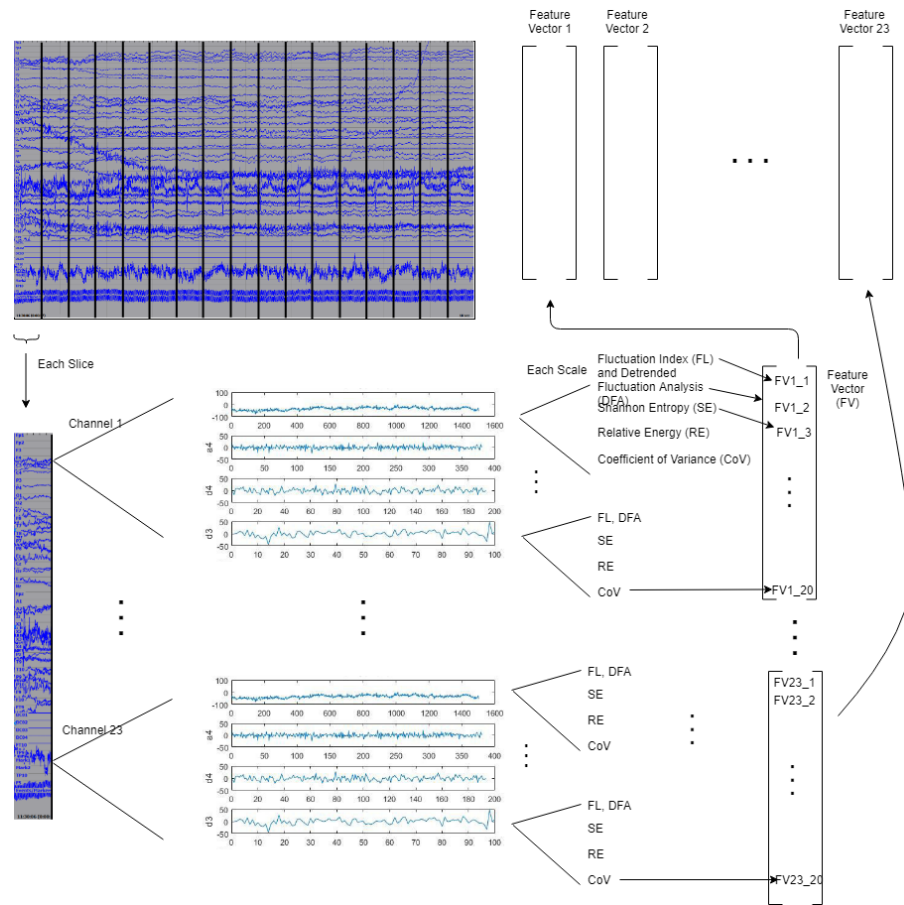


Figure 3.8 Feature Vectors of Channels and Their Alignments

Chapter 4: Combined Convolutional Neural Network and LSTM for EEG Seizure Prediction

4.1 Combined CNN-LSTM network

In building CNN for the EEG seizure analysis, we process the features as images and train the network for classification. The model can reach high performance in terms of testing accuracy. However, the CNN model usually encounters an overfitting problem due to its sophisticated structure. By processing the signals as feature images, the temporal correlations of the EEG epochs are not utilized to distinguish between seizure and normal epochs. Moreover, the training of the CNN model could be very time consuming. For example, our CNN model usually takes more than 40 minutes for one of the ten folds for one patient case. The LSTM model is intended to deal with sequential data. Designing feature extraction layers that preserve the sequential information of the input data would make it possible for LSTM layer to make use of the temporal correlations of the input signals.

Based on the analysis above, in order to improve our method, we designed a model combining CNN and LSTM layers to improve performance from several perspectives.

The structure of this model is shown in Figure 4.1.

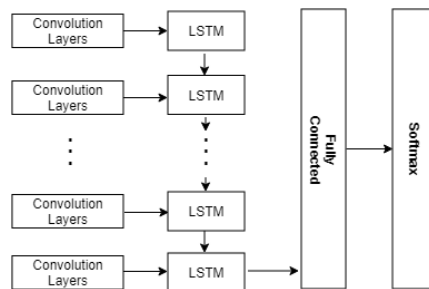


Figure 4.1: CNN-LSTM Architecture

A further illustration of the details of the convolution layer and its connection to the LSTM units is displayed in Figure 4.2. We apply a 1D sliding filter window CNN (1D CNN) which filters the signal input only along the time axis. For each CNN filter, it processes the EEG channel signals as images (2D signal matrix) to yield a vector representing the image features in a temporal order. For example, when we are using a 3-second long epoch, with 23 channels and a 256Hz sampling rate of original data, the size of one input matrix would be (23, 768). The sliding filters function as feature extractors to yield vectorized outputs for the LSTM units.

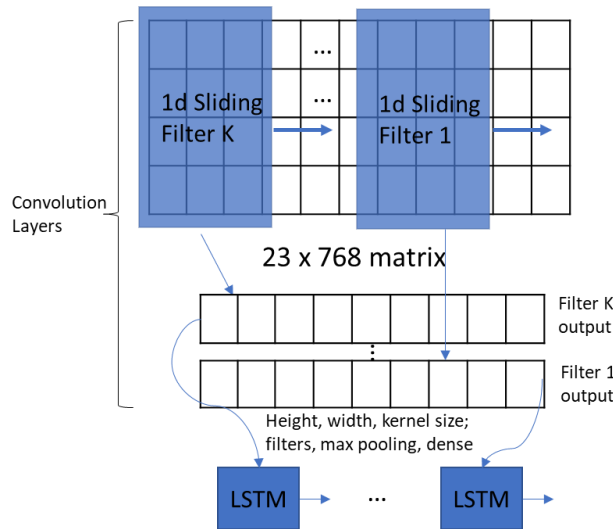


Figure 4.2: Operation of Each 1D Sliding Window

This model is proposed to improve the performance from three aspects. First, the 1D sliding filter window would save a significant amount of training time. Second, it preserves the temporal correlations of the input signal. Third, it has fewer parameters than merely implementing the CNN layer, which would make it less likely to have overfitting problems.

4.2 Matching Network Architecture for EEG Epoch Testing

Deep learning has gained significant success in various tasks but is notorious for its requirement for large training datasets. Not only does it take a substantial amount of time to train the networks, but adjusting the structures could be very costly depending on the training results. Because of the complex patterns of seizures, some non-parametric methods combined with advanced signal processing techniques could perform relatively well in terms of time efficiency. However, these methods have very limited adaptivity [22].

4.2.1 Matching Network Mechanism

In the EEG recordings of seizure patients, the number of seizure onset samples is not large compared to the normal phase. In training across populations in which the samples are relatively affluent, straightforward deep learning networks could be applied directly to learn the signal representations and yield predictions. However, if we inspect a specific patient case, the dataset will typically have a very imbalanced class ratio between seizure and normal phases, thus making it considerably more difficult for the neural network to learn to recognize one class over the other. Hence, developing an architectural mechanism to curate the deep learning model to deal with the imbalanced dataset is a key demand.

Human beings learn things in a way that they can recognize similar objects after only having seen several examples. Think about babies learning to recognize cups: the babies could recognize other cups by just seeing the outlines of several cups shown to them by educators. From the machine learning perspective, this procedure can be called “few sample learning”: an intelligent agent learns to recognize a class of objects by

having a very limited number of examples as training data. This few sample learning or named few-shot learning is rising as a major topic in the machine learning field.

To achieve the efficacy of few sample learning, Vinyals and colleagues proposed a one-shot learning model using deep learning feature extraction and vector comparison to perform the task [22]. With a similar approach to tackle this kind of problem, Koch and his colleague introduced a Siamese network for alphabet learnings [23]. Of their work, the most significant attribute of the models is the hierarchical design of using deep neural networks as embedding functions and metric learning operations on top of the embedding functions in the feature space. We refine the model architectures to a matching network architecture and further develop it to perform reinforcement learning on our seizure prediction problem.

The basic idea is to use embedding functions to lift the input images into the feature space and conduct metric learning for feature similarity comparisons. As depicted in Figure 4.3, g_θ and f_θ are the embedding functions for the labeled data input and the testing data input, respectively. The embedding functions are machine learning functions, especially deep neural networks for image or matrix inputs. For one testing input, the extracted test image is compared by a metric comparison mechanism with the extracted labeled images from each class. The comparison mechanism is developed to weigh the similarities between the test image and the labeled images in their learned feature space. The comparing results are synthesized as probability distributions among classes to yield the output as predictions.

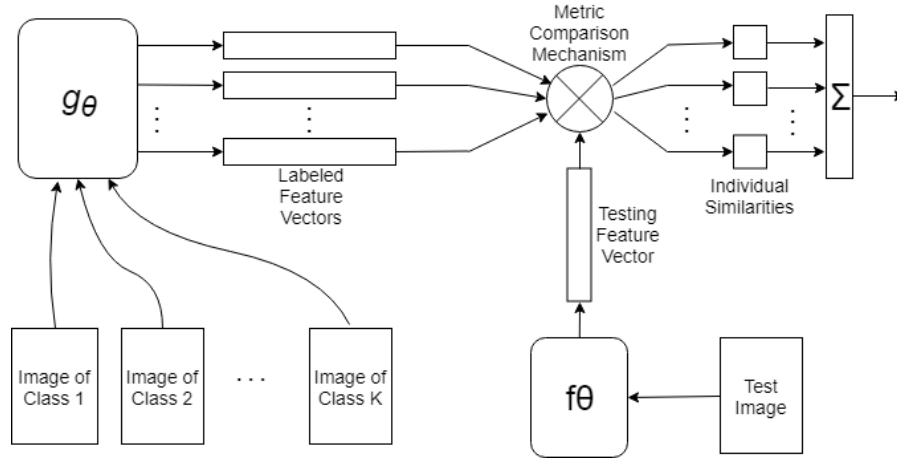


Figure 4.3 Matching Network Architecture

An illustrative example of how the architecture works with a specific case is the Siamese alphabet learning. The goal is to learn to recognize a set of alphabets containing various characters in different languages. With each character, there are several handwritten images used as a training set. By proposing a model based on the architecture we described in Figure 4.3, Koch introduced the Siamese network, which achieved satisfying results with very few training examples in each case.

4.2.2 Matching Network Architecture for Seizure Predictions

In solving the EEG seizure prediction problem, we introduce a two-channel matching network architecture to yield improved performance. The basic idea is to train two parallel networks to incorporate them into our matching architecture and use the incorporated model to yield similarity comparisons between testing and training epochs. With this comparison mechanism, the seizure epoch prediction procedure could be performed as the metric comparison between the test epoch and a set of labeled epochs. Before illustrating the details of the functioning mechanism of our matching network architecture, we first need to define the dataset. The training dataset S is composed of data with the following label:

$$S = \{(x_1, y_1), \dots, (x_n, y_n)\} \quad (4.1)$$

where x_i and y_i are respectively the i^{th} epoch data and its corresponding label. From S , we can pair any two elements in S to formulate our matching network dataset:

$$S' = \left\{ \left((x_i, y_i), (x_j, y_j) \right) \right\} \quad (4.2)$$

where (x_i, y_i) and (x_j, y_j) are any pairs of epoch data and label from set S . In set S' there are a certain number of these pairs.

We sample the seizure epochs as well as the non-seizure epochs from our raw data and make pairs according to training requirements to form the dataset as described in Equation 4.2. The formation of this dataset could help us perform reinforcement learning on top of the two-channel architecture of our matching network model.

Our basic idea is based on the methodology described in Figure 4.1. The design of the embedding functions f_θ and g_θ is from the models we applied on the epoch classification phase. We can use the combined CNN-LSTM network described in Section 4.1 on both channels to build our model. The CNN layers shape the feature map and the LSTM layer outputs the feature vector for similarity comparison.

The training of the CNN-LSTM channels would take time. We also propose a signal feature extraction approach in our model. As illustrated in Figure 4.2, on each channel the seizure epoch is fed into the feature module. The module filters out certain frequency bands and computes features on the selected bands to generate feature vectors. The feature vectors are normalized and interpolated to align as feature maps. For each feature map, the vectors are sorted by frequency scales, and the features in each scale are fed into a particular LSTM cell. For example, if we selected 4 decomposed frequency scales from our DWT, then feature vectors computed from all

scales would be concatenated. The operation details are the same as those described in Chapter 3 Section 3.4.

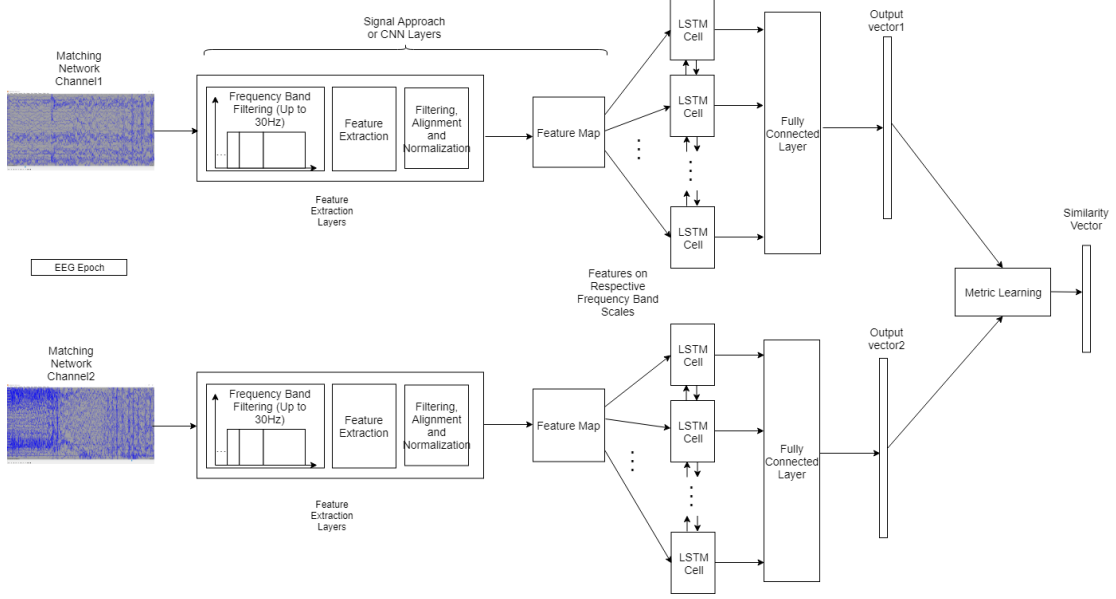


Figure 4.4 Matching Network Model for EEG Seizure Similarities

We have placed an attention mechanism on the LSTM layer to adjust the weights on each scale to optimize the training process. After the LSTM layer, the outputs are fed into a fully connected layer and then flattened into a vector by the layer. This vector is run through a metric comparison module with the other vector that is generated by the second channel, and the similarity between these two epochs is obtained.

For the LSTM layer, we use the common notation for LSTM to illustrate our model [24]. In our expression, LSTM represents an LSTM layer. The x_i is the i_{th} input vector of the i_{th} scale, for the k_{th} LSTM cell. Hence, the intermediate variables on one direction is computed as

$$h_k, c_k = LSTM(h_{k-1}, c_{k-1}, r_k) \quad (4.3)$$

where r_k is the synthesized input defined as:

$$r_k = \sum_i a(x_i, h_{k-1}) \cdot x_i \quad (4.4)$$

and the attention parameter $a(x_i, h_{k-1})$ is defined by the equation:

$$a(x_i, h_{k-1}) = \frac{e^{x_i^T \cdot h_{k-1}}}{\sum_i e^{x_i^T \cdot h_{k-1}}}. \quad (4.5)$$

The attention mechanism assigns the weights on each input, which is the feature vector on each scale. This procedure adjusts the influence of each scale on the output, respectively.

4.2.3 Implementation of the Matching Network Model

Here we use our CNN-LSTM channels to explain how the implementation works. We can break down the implementation of our matching network architecture into two stages. In the first stage, we train the CNN-LSTM network on the training data, and we put two of the same trained networks in parallel, as described in Figure 4.2. Our metric learning method applied here compares the Euclidean distance between the output vector 1 and output vector 2. This stage functions as a feature extraction operation for both channels to compare vector similarities.

Once we have obtained the vector similarity comparison mechanism, we come to the second stage to operate our model. The intention of this stage is to compare the distance between the selected testing epoch with all the labeled seizure epochs from the training dataset. In this operation, we use one channel for the testing epoch and one channel for the training seizure epochs. We first fix a testing epoch to feed it into channel 1 and from that channel it yields an output vector. Then, for the training dataset that has N seizure epochs, we loop over these N seizure epochs to feed them into channel 2 and compare the output vectors one at a time with the output vector from channel 1. By this

operation, we get the distances to the N labeled seizure epochs from our fixed testing epoch. And we implement this operation for all testing epochs. We use a $N \times M$ matrix to store the distance values, where the j^{th} element of the i^{th} column contains the distance to the j^{th} labeled seizure epoch from our i^{th} testing epoch. After this step we get the distance distributions of the testing epochs to the labeled seizure epochs, by operating the epochs at their extracted feature space. Figure 4.3 is an example from patient 6. We plot the histogram for the distances from one normal testing epoch to all the labeled seizure epochs, where normal epoch is labeled 0 and seizure epoch is labeled 1.

The general distributions of the histograms are as follows: In terms of the distance metric, the testing epoch with true label 0 has a dense distribution in the far side (mostly right 1/3 side) as in Figure 4.3. The epoch with true label 1 has a dense distribution at the near side (mostly left 1/3 side). Hence, from our experiments, we propose 4 control parameters for the statistical analysis on the distributions to further improve predicting performance. Division line parameter indicates the division position we assign on the histogram of the testing epoch on the distance metric axis. In the testing epoch case, it has a minimal distance and a maximal distance to the labeled epochs, and their difference is called full range.

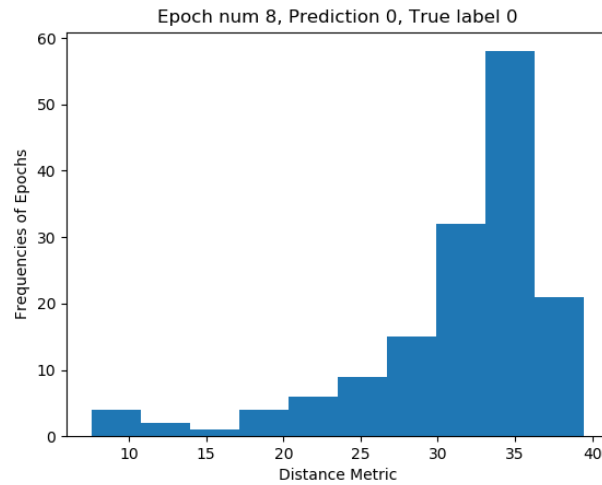


Figure 4.5 Histogram of the Distance Distributions of Epoch 8, Patient 6

The value of division line parameter is the division position minus minimal distance value divided by full range. The integration threshold parameter is associated with the division line parameter, which is the number of frequency counts in the histogram that are below the division position value divided by total number. The control line parameter is the division line parameter on the full range of all testing epochs. And its integration threshold parameter is defined the same as the one of the division line parameter. Once the parameters are set, we conduct our matching network experiments on the testing epochs to update the predicting results. For each predicted normal epoch, when both integration thresholds are exceeded, we predict the epoch as seizure. The detailed settings of the parameters are listed for experiments in Chapter 5.

Chapter 5: Experimental Analysis and Comparative Evaluations

5.1 Implemented Dataset Illustration

The database we use here is the CHB-MIT scalp EEG database. Its description can be found in Chapter 2. In experimenting with this database, we processed the patient files by pairing the seizure and normal epochs according to a predefined ratio to form training and testing datasets. The database provides each patient with a sequence of files, and each file contains the data of a 1hr-length monitoring. We select all the files with seizure onsets from the patient to form the dataset. The monitoring data is segmented into epochs of 3 seconds. We pick all the seizure epochs from this data, pairing normal epochs with the seizure epochs by a 9:1 ratio, which his accomplished by evenly sampling normal epochs along the time axis from the same original file.

5.2 CNN and CNN Feature Model Comparison and Determination

We build our CNN model with the parameters illustrated in Table 5.1, the parameters of which is also used as the classifier of the CNN feature model.

Table 5.1 CNN Model Parameter Settings

Layers	Settings
1	Zero Padding 2D (Strides = (1, 1))
2	Convolutional 2D (64, Filter Size = (3, 3), Strides = (1, 1))
3	Batch Normalization (Axis = 3, Activation('RELU'))
4	Max Pooling 2D (Filter Size = (2, 2))
5	Convolutional 2D (16, (2, 2), Strides = (1, 1))
6	Average Pooling 2D (Strides = (2, 2), Activation('RELU'))
7	Flatten Layer (Single 1D vector output)
8	Dense (Output Dimension = 2)
9	Output Activation ('Softmax')

The structure of our CNN feature model is as designed in Chapter 3. We trained our CNN model and CNN feature model on 10 patient datasets. On each patient dataset we conducted a 10-fold cross validation training and testing. Each fold we apply a 50-epoch (50 training iteration) training, with a batch size of 10. The overall results are shown in Figure 5.1.

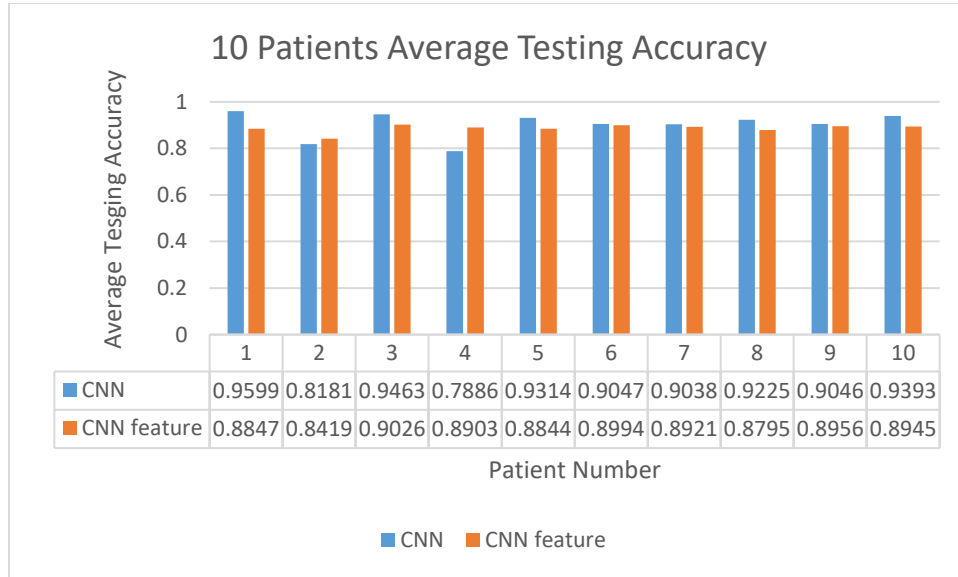


Figure 5.1 Testing Results for 10 Patients Average Accuracy

By comparing the results from the patients, we found that the performance of the two models have a different behavior on specific folds. For example, in Figure 5.2 for patient 5 (the 11th bar represents the mean value), on the folds where both models reach higher accuracy than average, the CNN feature model has a better performance than on the other folds compared to itself. This could be because these folds have a higher ratio of normal epochs. Hence the high specificity model yields higher accuracies than on the folds which contain more seizure epochs.

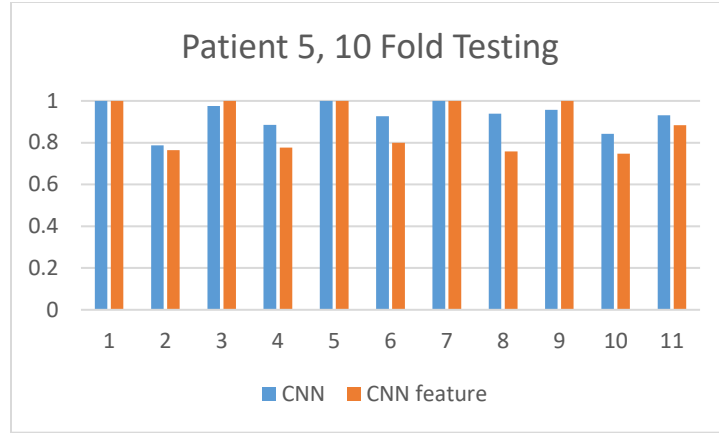


Figure 5.2 Patient 5 Testing Results on Respective Models

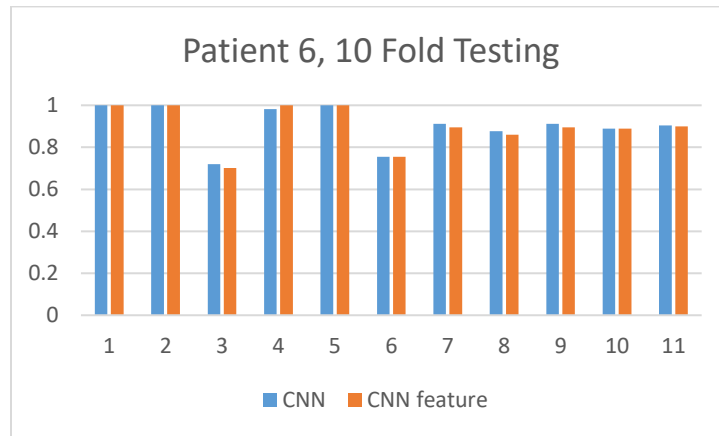


Figure 5.3 Patient 6 Testing Results on Respective Models

The configurations of the PC we use for the training is in Appendix II. We apply batch training with batch size 10 to train the model, and for each fold we setup 50 epochs (training iterations). As an example, in the feature extraction process, for patient 6 the feature extraction machine takes 7 min 38 sec to extract the features from the dataset. We show the overall training time comparisons in the following table.

Table 5.2 Training Time of 10-Fold Cross Validation on One Patient Dataset

	1	2	3	4	5	6	7	8	9	10
CNN	44:04	44:54	46:05	45:52	46:48	45:47	45:18	44:02	44:02	44:09
CNN feature	08:57	08:53	08:49	08:51	08:54	08:54	08:53	09:00	09:01	08:58

5.3 CNN-LSTM Model Structure Determination

For the training process, we compare our constructions of convolutional and LSTM combined model designs. As in Table 5.3, the layer settings are listed for 4 constructs to conduct comparative experiments. We use our notations in the table to simplify expressions. Conv1D represents a 1-dimensional sliding filter convolutional neural network, with the first parameter for the number of filters, second parameter for kernel size (filter window width), and one stride parameter for step size. LSTM layer has two parameters, which are the number of units and the output vector dimension. The default setting of the output of the LSTM layer is to return the last output of the sequence. The intermediate activation layers are set as RELU and final output activation layers are set as Softmax. Dense layer is a fully connected layer shaping the vector into desired dimensions.

Table 5.3 Model Layer Settings

Layer Setting Parameters			
Construct 1	Construct 2	Construct 3	Construct 4
Conv1D (32, 32, strides=2)	Conv1D (32, 32, strides=2)	Conv1D (32, 32, strides=2)	Conv1D (16, 32, strides=2)
Activation('RELU')	Activation('RELU')	BatchNormalization(axis=2)	Activation('relu')
LSTM (32, 64)	Conv1D (16, 32, strides=1)	Activation('relu')	Conv1D (8, 32, strides=2)
Dense (2)	Activation('relu')	Conv1D (16, 32, strides=1)	Activation('relu')
Activation('softmax')	LSTM (16, 64)	Activation('relu')	Conv1D (8, 16, strides=1)
	Dense (2)	LSTM (64)	Activation('relu')
	Activation('softmax')	Dense (2)	LSTM (64)
		Activation('softmax')	Dense (2)
			Activation('softmax')

We apply an Adam optimizer with learning rate = 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, decay rate = 0.01 during the training process [25]. For the loss function we use binary cross entropy. The training processes of the listed constructs are shown in Figure 5.1. We added a batch normalization layer in construct 3. The training accuracy curve has a clear tendency to adjust at each epoch, which gives a higher probability of breaking out from stagnation in training. Construct 1 tends to reach high training accuracy after 100 epochs of training. The convergence process is slow for this construct. Comparing construct 2 and 3, their training processes are similar at the first 40 epochs. The batch normalization layer breaks through the early stagnation and reaches a higher accuracy.

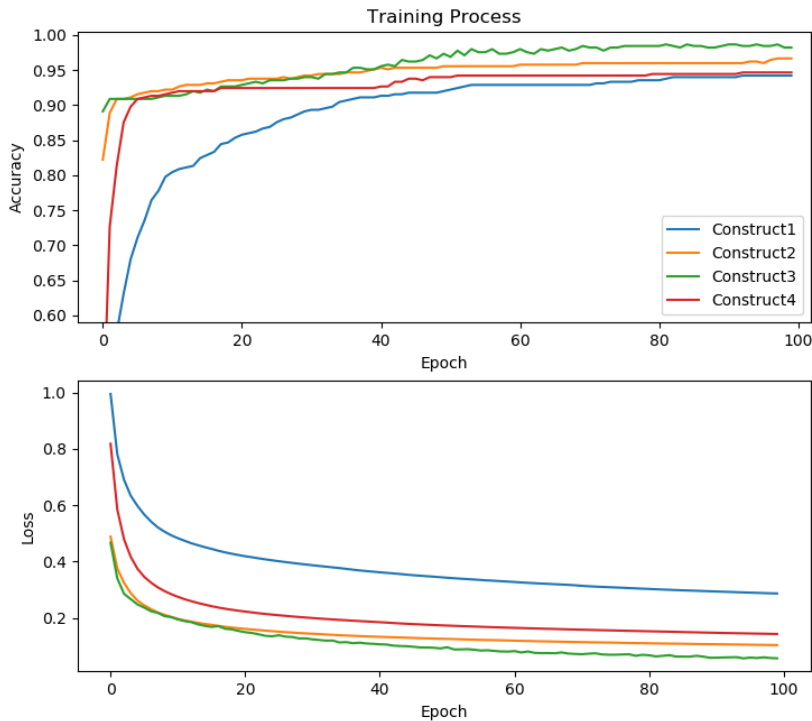


Figure 5.3 Training Process of Respective Model Constructions

5.4 CNN-LSTM and CNN Comparative Experiments on 10 Patient Cases

The training setup for our CNN-LSTM model (construct 3) is the same as for our CNN model. We train the model using a 10-fold cross validation strategy on each patient, and then obtain the overall accuracy from the mean value of the 10 folds results. As illustrated in Table 5.4, the comparison between the CNN-LSTM model and CNN model are listed in terms of accuracies. We have observed a significant improvement in the results from the CNN to the CNN-LSTM model. On average, the CNN-LSTM model has a 2.3% higher testing accuracy than the CNN model. The training accuracy is also higher than the CNN model on average. Further, the training time of the CNN-LSTM model is less than half of the CNN model since we use a 1D sliding filter window. For example, for patient 6, the CNN model takes 44 minutes to train each fold while our CNN-LSTM model takes 19 minutes.

Table 5.4 Training and Testing Accuracies of CNN-LSTM and CNN Model

Patient Number	CNN		CNN-LSTM	
	Training Accuracy	Testing Accuracy	Training Accuracy	Testing Accuracy
1	96.98%	88.53%	99.09%	98.00%
2	98.09%	90.90%	98.25%	83.43%
3	99.11%	94.63%	99.13%	97.20%
4	86.94%	78.86%	95.38%	80.12%
5	97.22%	93.14%	96.08%	96.34%
6	98.12%	90.47%	97.46%	89.21%
7	98.69%	90.38%	98.78%	94.38%
8	96.34%	92.25%	95.91%	92.94%
9	98.83%	90.46%	99.01%	95.79%
10	98.21%	93.93%	98.92%	96.97%

We apply a matching network with division line parameter 0.5 and integration threshold at 0.7. The control line parameter is 0.45, and the control integration threshold

is 0.8 for the final prediction. From the perspective of detecting sensitivities and specificities, we found on average a higher sensitivity for the CNN-LSTM and CNN model on the patient cases tested.

Table 5.5 Sensitivities for Patients Before and After Matching Network Operation

Patient	1	2	3	4	5	6	7	8	9	10
Before	88.06%	59.21%	87.55%	43.76%	62.46%	47.23%	61.60%	66.28%	83.50%	77.79%
After	93.55%	66.67%	95.45%	47.06%	72.97%	48.23%	93.34%	89.15%	86.36%	99.31%

The improvement in sensitivity achieved by our matching network for each patient case can be seen in Table 5.5. The sensitivity improvement varies among cases. On average the sensitivity improved by 16.92%.

Table 5.6 Sensitivities and Specificities of CNN-LSTM and CNN Model

Patient Number	Number of Epochs	CNN		CNN-LSTM	
		Sensitivity	Specificity	Sensitivity	Specificity
1	1598	63.01%	99.38%	93.55%	99.14%
2	204	19.84%	96.83%	66.67%	95.24%
3	1322	57.75%	99.07%	95.45%	98.31%
4	1214	29.41%	89.70%	47.06%	88.71%
5	1643	39.92%	99.36%	72.97%	98.69%
6	567	13.24%	99.59%	48.23%	98.61%
7	545	20.95%	98.91%	93.34%	98.16%
8	2581	54.75%	97.16%	89.15%	96.89%
9	906	11.55%	100.00%	86.36%	97.57%
10	1450	50.19%	98.72%	99.31%	99.09%

It is important to point out that for patient cases with smaller sizes, such as patient 2 and 6, when they are tested for sensitivity and specificity they have relatively lower performance than other cases. This could be caused by the sparseness of the seizure epochs in the dataset. For example, in the patient 2 dataset, there are only 21 seizure

epochs. For analyzing sensitivity and specificity, the quantity of seizure epochs in tested folds is relatively limited. Hence, the results are not as good as might be found in larger datasets. A way to further test the case is to use smaller epoch length. For example, if the epoch length is 1 second, then the dataset would be 2 times larger, hence the testing results could be more stable in terms of testing folds. We also compare our model performance with reference methods as shown in Table 5.7.

Table 5.7 Comparison with Other Approaches on CHB-MIT Benchmark Dataset

Method	Accuracy	Specificity	Sensitivity
Lima et al. [26]	80.30%	86.85%	73.74%
Magosso et al. [27]	65.92%	83.34%	48.50%
Acharya et al. [28]	85.00%	88.29%	83.31%
Ubeyli [30]	84.60%	88.58%	80.62%
Our work	92.44%	97.04%	79.21%

Chapter 6: Conclusions and Future Work

We designed a combined CNN-LSTM model for EEG seizure prediction and explored its performance with respect to other methods. Our model performed significantly higher in terms of testing accuracy, sensitivity, specificity, and training time. Our CNN-based feature map model could reach a high performance with great training time saving. We proposed a metric learning inspired matching network architecture to explore post-processing after the deep neural network training process and the statistics indicate promising improvements. Our future work will focus on advancing in the following areas:

- a) Develop a fitting method for matching network metric distance histograms to simulate typical statistical distributions. Currently we have developed a metric learning architecture to evaluate training results from intermediate layer output, however, we need a fitting method to be able to analytically compare the histograms.
- b) Design an automatic algorithm for matching network validation. The statistical results of the learned model showed clear difference between seizure and non-seizure epochs in terms of metric distance distributions. We need to explore the distribution behaviors of the epochs compared to the labeled samples so that a self-adjusting algorithm could be developed to distinguish between classes.
- c) Improve the method to generate time-frequency image maps as inputs. We are dealing with EEG signals from the image approach. We have seen the clear improvement by processing multi-channel signals as images. We are dedicated to find better methods to learn the signal images so that the training could be

improved with more efficiency in terms of computing cost. That will give us a powerful tool to develop more comprehensive seizure detection systems.

Appendix I

The channels are the electrical potential differences between two electrodes on the scalp. The channel names are (by order):

FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FP2-F4, F4-C4, C4-P4,
P4-O2, FP2-F8, F8-T8, T8-P8, P8-O2, FZ-CZ, CZ-PZ, P7-T7, T7-FT9, FT9-FT10,
FT10-T8, T8-P8

Appendix II

PC configurations:

Processor: Intel(R) Core i5-7600K CPU @ 3.80GHz

Graphics Card: NVIDIA GeForce GTX 1050

Installed memory: 16.0 GB

System type: 64-bit Operating System, x64-based processor

Bibliography

- [1] Subha, D. Puthankattil, et al. "EEG signal analysis: a survey." *Journal of medical systems* 34.2 (2010): 195-212.
- [2] Shoeb, Ali H., and John V. Guttag. "Application of machine learning to epileptic seizure detection." *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010.
- [3] Boashash, Boualem, and Samir Ouelha. "Automatic signal abnormality detection using time-frequency features and machine learning: A newborn EEG seizure case study." *Knowledge-Based Systems* 106 (2016): 38-50.
- [4] Subasi, Abdulhamit, and M. Ismail Gursoy. "EEG signal classification using PCA, ICA, LDA and support vector machines." *Expert Systems with Applications* 37.12 (2010): 8659-8666.
- [5] Liu, Yinxia, et al. "Automatic seizure detection using wavelet transform and SVM in long-term intracranial EEG." *IEEE transactions on neural systems and rehabilitation engineering* 20.6 (2012): 749-755.
- [6] Bashivan, Pouya, et al. "Learning representations from EEG with deep recurrent-convolutional neural networks." *arXiv preprint arXiv:1511.06448* (2015).
- [7] Phadke, Arun G., and James S. Thorp. *Computer relaying for power systems*. John Wiley & Sons, 2009.
- [8] Mallat, Stéphane. *A wavelet tour of signal processing*. Academic press, 1999.
- [9] Kalayci, Tulga, and Ozcan Ozdamar. "Wavelet preprocessing for automated neural network detection of EEG spikes." *IEEE engineering in medicine and biology magazine* 14.2 (1995): 160-166.

- [10] Saab, M. E., and Jean Gotman. "A system to detect the onset of epileptic seizures in scalp EEG." *Clinical Neurophysiology* 116.2 (2005): 427-442.
- [11] Grewal, Sukhi, and Jean Gotman. "An automatic warning system for epileptic seizures recorded on intracerebral EEGs." *Clinical neurophysiology* 116.10 (2005): 2460-2472.
- [12] Bengio, Yoshua, et al. "Greedy layer-wise training of deep networks." *Advances in neural information processing systems*. 2007.
- [13] Oğulata, Seyfettin Noyan, Cenk Şahin, and Rızvan Erol. "Neural network-based computer-aided diagnosis in classification of primary generalized epilepsy by EEG signals." *Journal of medical systems* 33.2 (2009): 107-112.
- [14] Haykin, Simon S., et al. *Neural networks and learning machines*. Vol. 3. Upper Saddle River, NJ, USA:: Pearson, 2009.
- [15] Lukoševičius, Mantas, and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training." *Computer Science Review* 3.3 (2009): 127-149.
- [16] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks* 5.2 (1994): 157-166.
- [17] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [18] I Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

- [19] Vinyals, Oriol, et al. "Matching networks for one shot learning." *Advances in Neural Information Processing Systems*. 2016.
- [20] Shoeb, Ali Hossam. *Application of machine learning to epileptic seizure onset detection and treatment*. Diss. Massachusetts Institute of Technology, 2009.
- [21] Goldberger, Ary L., et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." *Circulation* 101.23 (2000): e215-e220.
- [22] Vinyals, Oriol, et al. "Matching networks for one shot learning." *Advances in Neural Information Processing Systems*. 2016.
- [23] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." *ICML Deep Learning Workshop*. Vol. 2. 2015.
- [24] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.
- [25] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980(2014)*.
- [26] Lima, Clodoaldo AM, André LV Coelho, and Sandro Chagas. "Automatic EEG signal classification for epilepsy diagnosis with Relevance Vector Machines." *Expert Systems with Applications* 36.6 (2009): 10054-10059.
- [27] Magosso, Elisa, et al. "A wavelet-based energetic approach for the analysis of biomedical signals: Application to the electroencephalogram and electro-oculogram." *Applied Mathematics and Computation* 207.1 (2009): 42-62.

- [28] Acharya, U. Rajendra, et al. "Use of principal component analysis for automatic classification of epileptic EEG activities in wavelet framework." *Expert Systems with Applications* 39.10 (2012): 9072-9078.
- [29] Lima, Clodoaldo AM, and André LV Coelho. "Kernel machines for epilepsy diagnosis via EEG signal classification: A comparative study." *Artificial Intelligence in Medicine* 53.2 (2011): 83-95.
- [30] Übeyli, Elif Derya. "Combined neural network model employing wavelet coefficients for EEG signals classification." *Digital Signal Processing* 19.2 (2009): 297-308.
- [31] Khan, Yusuf Uzzaman, Nidal Rafiuddin, and Omar Farooq. "Automated seizure detection in scalp EEG using multiple wavelet scales." *2012 IEEE International Conference on Signal Processing, Computing and Control*. IEEE, 2012.