Resting State EEG Classification for Motor Learning Skills Using Echo State Networks

by

Hang Yuan

Bachelor Thesis in Computer Science

Prof. Dr. Herbert Jaeger
Name and title of the supervisor

Date of Submission: May 11, 2017

Jacobs University — Focus Area Mobility

With my signature, I certify that this thesis has been written by me using only the indicates resources and materials. Where I have presented data and results, the data and results are complete, genuine, and have been obtained by me unless otherwise acknowledged; where my results derive from computer programs, these computer programs have been written by me unless otherwise acknowledged. I further confirm that this thesis has not been submitted, either in part or as a whole, for any other academic degree at this or another institution.

Signature　　　　　　　　　　　　　　　　　　　　　　Jacobs University Bremen, May 11, 2017

Abstract

Electroencephalogram (EEG) records the electrical activities from the scalp surface via electrodes. As a modern medical imaging technique, it has been proven to be useful in many different fields. Clinical diagnosis, psychotherapy, Brain-Computer Interfaces (BCIs) and the pharmaceutical industry all have benefited from the insights that one can glean from EEG measurements.

However, there exist various difficulties such as uniqueness of individuals, large volume of data and influences of artifacts that prevent us from extracting useful information from those measurements, and thus more involved analytical tools are needed. Recurrent Neural Networks (RNNs) are particularly suitable for dealing with EEG because RNNs can capture the critical spatiotemporal characteristics that EEG contains.

In this project, we successfully applied Echo State Networks (ESNs), to classify the people's Motor Learning Skills (MLS), given the resting state EEG recording, and confirm that hypothesis

need to fill in after discussion

Contents

Abbreviations

BCIs	Brain-Computer Interfaces
DL	Deep Learning
EEG	Electroencephalogram
ESNs	Echo State Networks
FNNs	Feedforward Neural Networks
LDA	Linear Discriminant Analysis
MARA	Multiple Artifact Rejection Algorithm
MC	Memory Capacity
MLP	Multilayer Perceptron
MLS	Motor Learning Skills
MSE	Mean Square Error
OH	Old high performers
OL	Old low performers
RNNs	Recurrent Neural Networks
SVMs	Support Vector Machines
YH	Young high performers

1 Introduction

Given the long history of EEG studies, we have already decoded its relationships with a few brain processes like one's motor learning, motor imagery performance and even intelligence [1] [2][3]. Pursuing this line of inquiry this guided research plans to investigate if there exists a correlation between EEG signal and subjects' MLS, a latent variable that we will introduce more formally later.

OK to use abbervi-ation in ab-stract?

ESNs [4] are the more engineering favored reservoir computing method that was independently discovered with Liquid State Machines [5], which concern more the computational neuroscience's perspectives. We are mainly interested in the engineering problems, and thus solely touch on ESNs. ESNs are a type of RNNs, which have a few notable advantages over traditional methods for a sequence learning task (EEG classification) [6]. Static methods like Support Vector Machines (SVMs) and Feed-Forward Neural Networks (FNNs) have achieved excellent results on numerous learning tasks without explicitly modeling sequentiality. They can even combine inputs within a window of time frame for a model to encode the time dependency. Nevertheless, these static models cannot answer the questions about the events that occur outside the binned time steps. That's where RNNs come to rescue. RNNs are a kind of neural networks whose units form directed cycles. The input is of the form $(x^1, x^2, \ldots, x^T)$ and the corresponding labels for each time step is of the form $(y^1, y^2, \ldots, y^T)$, where T is the number of time steps we have.

It is empirically difficult to train RNNs mainly due to vanishing gradient and exploding gradient [7]. Standard methods like backpropagation through time and real time recurrent learning suffer from the vanishing gradients since they both use the error gradient taken from the objective function, and the gradient values become very small already after several steps.

ESNs give us an easy solution to the above issues and yet maintaining RNNs' power as we desire. ESNs are constructed using random weights for internal connections, which are fixed throughout the training process. It suffices to have a linear readout function for the output weights on the network responses which are simulated in the training. Because of ESNs' simplicity, we can focus more on the understanding of the nonlinear dynamical systems which we train the models on.

So far, there have been several successful applications on using RNNs or ESNs on EEG data analysis. Epileptic seizures disorder is a popular application of such techniques. We can build a warning system for the patients to be informed about an upcoming episode using EEG classification. Furthermore, the doctors can use the EEG model for epileptic activities to evaluate the treatment effectiveness [8] [9]. RNNs have also been used to detect Alzheimer's disease early on and other neurological degeneration [10][11].

In this guided research, we have two objectives: one is of biological interest, finding out the association between resting state EEG and MLS, and the other is of engineering interest: investigating the limitations and effectiveness of ESNs on EEG like data.

Outline: in section 2, we present the theoretical background for both ESNs and EEG, which is followed up with a compact description of the motivation of this guided research in section 3. Then, in section 4 and 5 we delineate what we plan to do and how to interpret the results. Finally, we present the projected schedule in section 6.

2 Theoretical Frameworks

2.1 Echo State Networks (ESNs)

We now introduce the general architecture of ESNs. In this section, we mainly follow the notations from [4][12] to keep things consistent. ESNs are mostly used for temporal supervised learning. We will present the setup in discrete time domain, denoting each time step as $n = 1, 2, 3, \ldots, T$, which leads the input signal to be $\mathbf{u}(n) \in \mathbb{R}^K := (u_1(n), \ldots, u_K(n))^T$ and the teacher signal to be $\mathbf{y}(n) \in \mathbb{R}^L := (u_1(n), \ldots, u_L(n))^T$.

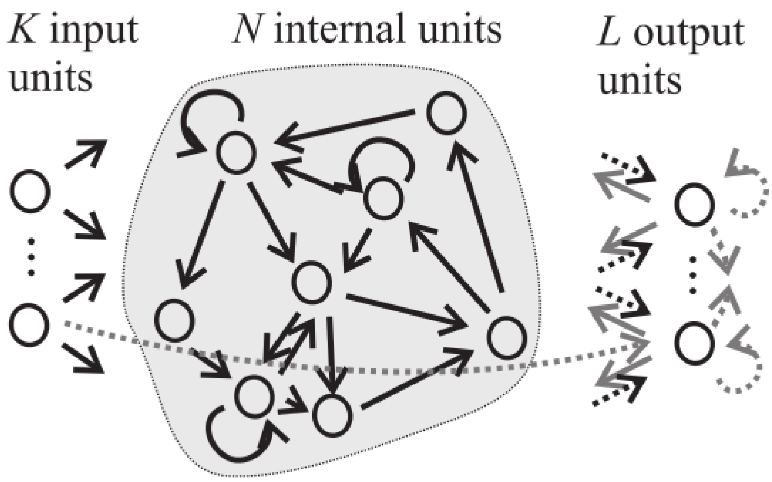

Figure 1: A basic ESNs architecture taken from [4]. The network consists of three layers, an input layer of size K, an internal reservoir of size N and an output layer of size L.

As shown in figure 1, the three different layers in the network have intermediate connections with each other and sometimes they can even have feedback projections onto themselves. The black lines are the necessary connections and the dotted ones are the optional connections. A typical minimal network graph will require three kinds of connections whose weights are expressed as: the input to the reservoir $\mathbf{W^{in}}$, the internal connections among the units of the reservoir $\mathbf{W}$ and the reservoir to the output units $\mathbf{W^{out}}$. Optional input-to-output connections will increase the performance slightly at the cost of longer training time [13] and the optional output feedback loops to the internal units or to the output units themselves are used in signal simulation or to increase memory span[14]. For the rest of the discussion, we will stick with the minimal setup with a back projection from output to the internal units, and express the back projection weights as $\mathbf{W^{back}}$.

Propagation steps At each time step, the internal units are updated using:

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W^{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W^{back}}\mathbf{y}(n)) \tag{1}$$

$\mathbf{f}$ is the activation functions for the internal units. Threshold functions, hyperbolic tangent functions or sigmoid functions are all candidates for the activation functions. As for the output, we update them using:

$$\mathbf{y}(n+1) = \mathbf{f_{out}}(\mathbf{W^{out}}\mathbf{x}(n+1)) \tag{2}$$

Training steps We start with some state $\mathbf{x}(0)$ and then use equation (1) to simulate the output signal until T_{max}. We then discard the simulation results until n_{min} when, the network dynamics become stable. From this point on, we assume time 0 is the first time step after n_{min}. The Error function E usually measures the Mean Square Error (MSE) defined as:

$$E(\mathbf{y}, \mathbf{y^{teach}}) = \frac{1}{N_y} \sum_{i=1}^{N_y} \sqrt{\frac{1}{T} \sum_{n=1}^{T} (y_i(n) - y_i^{teacher}(n))^2} \tag{3}$$

It suffices to run a linear regression on the output signal to minimize the MSE between the predictions and the teacher signals. Nevertheless, after we concatenate the simulation outputs into a matrix $\mathbf{X}$, $\mathbf{X}$ is most likely overdetermined because quite often $T > N_x$. N_x is the number of the reservoir units. It follows that we need to make use of a few techniques to solve the system. We now look at two of such methods, ridge regression and pseudoinverse inverse.

Ridge regression yields:

$$\mathbf{W^{out}} = (\mathbf{y^{teach}}\mathbf{X^T}(\mathbf{X}\mathbf{X^T}) + \beta\mathbf{I})^{-1} \tag{4}$$

where $\mathbf{I}$ is an identity matrix and β is a regularization coefficient. Ridge regression in theory works with large datasets. As shown in equation (4), all the elements do not depend on T. The limitation of this method is finite floating point representation when one needs to add a big number with a small one [12]. Schemes like adding similar values or Kahan summation are recommended [15].

Pseudoinverse comes to:

$$\mathbf{W^{out}} = \mathbf{y^{teach}}\mathbf{X}^{+} \tag{5}$$

where $\mathbf{X}^{+}$ is the Moore-Penrose pseudoinverse. This approach is straightforward but the inverse computation is expensive which limits the reservoir size N_x and the number of training data points.

Prediction steps Implant the trained $\mathbf{W^{out}}$ in the readout layer and do the propagation steps for any new input data.

Networks parameters There are three global parameters that define ESNs, namely $(\mathbf{W^{in}}, \mathbf{W}, \alpha)$, where α is the rate. There are other important global parameters for the network: the of internal units, sparsity, spectral radius of $\mathbf{W}$ and scaling of $\mathbf{W^{in}}$ [12]. To achieve satisfying performance, one should consider the above factors in the design phase. We briefly list a few optimization techniques with respect to those parameters:

- Spectral radius: The critical parameter that ensures the effectiveness of ESNs. There are a few assumptions that ESNs approach has, one of which is the echo state property. It entails that

$$\exists E.E = (e_1, \ldots, e_N) \text{ where } e_i : U^{-\mathbb{N}} \implies \mathbb{R} \tag{6}$$

so that for any left-infinite input sequence, the current state is determined by:

$$\mathbf{x}(n) = \mathbf{E}(\ldots, \mathbf{u}(n-1), \mathbf{u}(n)) \tag{7}$$

In plain English, the echo state property implies that given a long enough input sequence, the current state is uniquely defined by the previous history such that sequence and the network state $\mathbf{x}(n)$ should not rely on the information that occurs before the initial state the input [4]. In most cases, $\rho(\mathbf{W}) < 1$ guarantees the echo state property. For the implementation wise, we can first compute the spectral radius of $\mathbf{W}$, and then divide the matrix itself with this value to obtain the unit spectral radius which can be easy to use in the tuning phase.

- Size of the reservoir: In [16], the memory capacity (MC) of an N-unit RNN with linear output to recall an i.i.d. input has been shown to be bounded by N. It makes sense to have $N_x \geq N$ for the minimal setup. On the other hand, only when $T < 1 + N_u + N_x$, will we have a reservoir layer that's too large for the dataset. In general, the bigger the reservoir one uses, the better the performance will be.

- Leaking rate: The significance of this parameter stems from the discretization of the continuous time update, which can be described as:

$$\dot{\mathbf{x}} = -\mathbf{x} + tanh(\mathbf{W^{in}}[\mathbf{1}; \mathbf{u}] + \mathbf{W}\mathbf{x}) \tag{8}$$

where $[-; -]$ represents vector(matrix) wise concatenation. In the context the discrete time, we will have:

$$\frac{\Delta \mathbf{x}}{\Delta t} = \frac{\mathbf{x}(n+1) - \mathbf{x}(n)}{\Delta t} \approx \dot{\mathbf{x}} \tag{9}$$

It becomes clear that the leaky rate α is the transformation piece between the discrete and continuous worlds. Changing α to match up with the change rate of $\mathbf{u}(n)$ and or $\mathbf{y}^{teach}(n)$ is similar to resampling of inputs in order to achieve better performance [17].

2.2 Resting State Electroencephalogram (EEG)

EEG measures the electrical activities from the scalp surface which are recorded via electrodes and other conductive media [18]. The brain has three components: cerebrum, cerebellum and brain stem. EEG is mostly influenced by the activity of the cerebral cortex that is close to the scalp surface. The EEG data records the relative voltage difference between the an electrode and a reference electrode that is usually being placed in the middle of the scalp. There are two reasons why the raw voltage values are not of interest. First, the voltage values will change due to different choices of baseline subtraction. Secondly, the raw values will be hard to analyze because of the individuals' differences that may not play a role in the desired cognitive processes studies. [19]

EEG has three temporal properties: resolution, precision and accuracy. Resolution reflects how many data points are recorded per unit time, precision reflects how certain the measurements are and accuracy reflects the mapping between the timing of the EEG signals and the timing of the actual occurrences of the events [19]. The temporal resolution is determined by the rate of acquisition. It enables one to extract frequency-band-specific features. Furthermore, brain waves are separated into five groups based on their frequency domains. Their corresponding frequency domains are usually associated with:

- γ waves: $freq \in (30, 80)Hz$
- β waves: $freq \in (13, 30)Hz$
- α waves: $freq \in (8, 13)Hz$
- θ waves: $freq \in (4, 8)Hz$
- δ waves: $freq \in (0.5, 4)Hz$

The brain waves from different frequency bands look like:

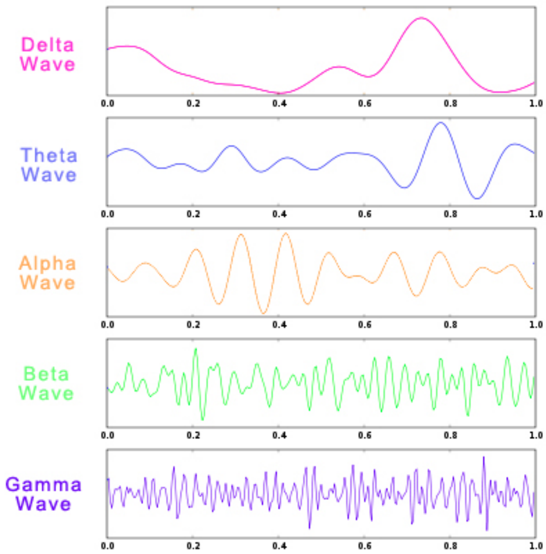

Figure 2: Brain waves visualization based on different frequency bands.[1]

EEG is a good technique to study the brain for a few reasons: This method records the brain dynamics at the time when the cognitive events happen; Secondly, EEG directly measures the brain activity. Changes in voltage potentials are due to neurological behavior at the neuron population level; Lastly, EEG contains rich information and is multidimensional. It not only has the spatiotemporal information, but also frequency, power and phase as features that give us ample knowledge about the internal brain activities. [20]

Properties of EEG features for analytics These are paramount factors to consider when one analyzes EEG signals.

[1] Figure taken from the site: http://www.brainworksneurotherapy.com/what-are-brainwaves

- Noise and signal: EEG signals are noisy and the noises are often hard to discriminate. One will have to find the fine line between removing too much useful information and having noisy data depending the given task. Figure 3 is a good demonstration of the signal and noise relationship.

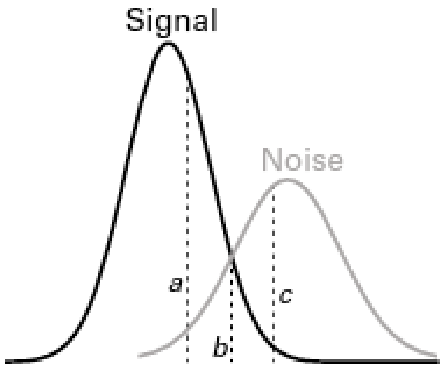

Figure 3: This plot shows the interconnected relationship between signal and noise in EEG (taken from [19]). The x-axis is the degree of data cleaning that we conduct and the y-axis is the leftover for signal and noise after the cleaning. We can see the distribution of signal and noise with respect to the level of preprocessing. Area left to a implies there is little noise left, area between a and b has a mixture of both noise and signal and the area right to c has mostly noise.

- Non-stationarity: EEG signals can change quickly over time.
- Small training sets: Due to the costs of collecting data from subjects, the training sets are oftentimes smaller than ideal.

Preprocessing

- Filtering: One wants to have high-frequency artifacts and low-frequency drifts removed in this step. It is recommended to use high-pass filter at 0.1Hz or 0.5Hz to get rid of the slow drifts [19].
- Spatial filtering: Spatial filtering is needed when you want to localize a result and to eliminate topological features of the data. For instance, if an experiment requires the subjects to conduct some tasks that involve multiple brain regions, it would be otherwise difficult to isolate the active regions without spatial filtering. [19]

It is worth noting that there is a trend in machine learning that tries to construct end-to-end models without any preprocessing of the data nowadays. People often use a class of techniques called Deep Learning (DL). DL methods compose multiple levels of representational simple and nonlinear layer. Each layer learns at different degrees of abstraction. With enough layers, the models can learn how to discriminate important aspects of data from other variations. DL has made lots of advances in solving challenging problems. In the recent decade, DL has produced competition wining approaches in imaging recognition [21][22] and promising methods for sequence learning tasks in natural language processing [23] [24]. However, DL methods are not suitable for EEG data processing, because DL methods usually have many free parameters, and it is simply hard to collect enough EEG data for the training purpose. In robotics, although it is also difficult to obtain real world data, people can simulate the training data in a physics engine because the mechanics of robots' interactions and the real world are well understood. Unfortunately, this is not possible for EEG data yet, since we simply don't understand the brain well enough to have a simulator for the brain's underlying mechanisms.

Artifacts Artifacts stem from numerous places in an EEG study, such as blinks, muscle movements and wire noise. Note that EEG is not an error-free measurement technique and we do not know all sources of error. Fortunately after reasonable preprocessing, most analytics tools are robust enough to the noise leftover. Independent Components Analysis (ICA) is the common choice for artifacts removal. It essentially separates the source into different independent sources. Originally, ICA was meant to solve the blind source separation problem, trying to retrieve the independent sources $\mathbf{s} = (s_1(t), \ldots, s_N(t))$ [25]. The sources $\mathbf{s}$ is being mixed by an unknown matrix $\mathbf{A}$, such that the recorded N mixture $\mathbf{x} = (x_1(t), \ldots, s_N(t))$ has the property that

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{10}$$

ICA in the context of EEG records, separates the data at different electrodes into a sum of various temporally independent components [26], and thus muscle movement, eye blinks and oculomotor activities can generally be detected and rejected. Traditionally, artifacts detection from ICA need human experts to manually remove the corrupted components but this takes lots of time and training. Most of the ICA-based removal techniques require additional recording for a subject's artifact activity. For example using two eye electrodes to record eye movements and then do a correlation analysis to remove artifactual components later on. In this project, we are using a fully automated method Multiple Artifact Rejection Algorithm (MARA) [27]. MARA is a supervised-learning linear approach that learns from over 1k samples labeled by experts. Without using additional recordings, we are still able to automate the artifact removal process.

Classification overview There are five common types of classification algorithms in EEG analytics: linear classifiers, neural networks, nonlinear Bayesian, nearest neighbor and classifiers combinations [28]. Here we will discuss two of them and their corresponding characteristics in EEG classification.

- Linear classifiers: Linear Discriminant Analysis (LDA) and SVMs are the most common linear classifiers in EEG analysis. LDA uses several hyperplanes to separate the data. It is cheap to compute and therefore a good candidate for online learning. SVMs also make use of hyperplane separation but they have a different objective, maximizing the margins between different class planes. SVMs have a few good properties, thanks to the regularization terms: robustness for overfitting and tolerance to the curse-of-dimensionality [29].
- Neural networks: Multilayer Perceptron (MLP) has been widely used in EEG classification [30][31] due to its flexibility to adapt to different problems. However with noisy and non-stationary data like EEG, it is particularly prone to overfitting [32], and thus it needs careful tuning and architecture design. We should pay special attention to Gaussian classifier which is specifically created to process EEG data in BCIs [33] [34]. In this local neural classifier, each unit in the network is a Gaussian

discriminator for each class prototype and if a class has several prototypes, only the nearest one is used. During training, units are pushed towards the EEG samples of the same class and are pushed away from the ones that do not belong to the same class. It has been shown that this kind of architecture is superior to MLP in terms of the rejection efficiency for uncertain samples [34]. As for ESNs, they have been used to construct a fast and reliable method for epileptic seizure detection [8], however, it's not clear how effective ESNs are in EEG classification when being compared with other methods due to the lack of relevant study, nevertheless ESNs should still be a good choice for such a sequence learning task and we would like to explore its limitations in this regard.

3 Motivation

The main objective of this guided research lies in two folds, one is engineering oriented, and the other is physiology oriented:

- Construct a reliable and efficient ESNs to classify the EEG signals.
- Explore if there exists a relationship between resting state EEG and MLS. If such a relationship does exist, what we can say about that?

So far the applications of ESNs being used as an EEG classifier is not extensive, although there have been a few in the past years on sequence learning tasks, classifications of real time moving objects [35] and time series classification for the prediction of dialysis [36]. On this note, we wish to summarize the proposed questions of interests from the engineering perspective:

- How tolerant ESNs are when dealing with noisy and non-stationary data like EEG, and when it is good enough to stop cleaning without compromising useful information.
- How ESNs can best handle time-variant features, more specifically how they deal with the drifting of amplitudes which can be slow and fast at different times?
- As training a classifier for EEG using neural networks is prone to over-fitting, what kinds of tuning need to be done in order to avoid this situation?
- Is it possible to adapt the reservoir distribution somehow such that the model is better suited for EEG classification?

Now we come to the physiological side. So far, stable resting brain activities have been shown to have correlations with personal traits like personality, intelligence and neurological disorder [37][38]. We also know that there is a correlation between event-based computation and the preceding resting state EEG of that event, for instance, the strategy one uses in problem solving [39]. There are more correlations to discover, and correlaton between EEG and MLS is one of them. Merely by using a partial least squares regression model, one can already predict the motor skill acquisition well [40]. It would not surprise us if we can achieve better prediction performance using ESNs. Our hypothesis is that the MLS is related to resting state EEG in both young and old age groups. Along with this line of queries, this guided research is determined to tackle the following questions:

- Given the traits of each individual, what is a plausible definition for the MLS that both makes sense physiologically but will also work well for ESNs?
- What insights can we learn from the response activity in each internal unit of ESNs? Do the responses contain any critical information about the subjects like the biological age or the cognitive ability?
- Does compensation effect exist in old high-performing group?

4 Experiments

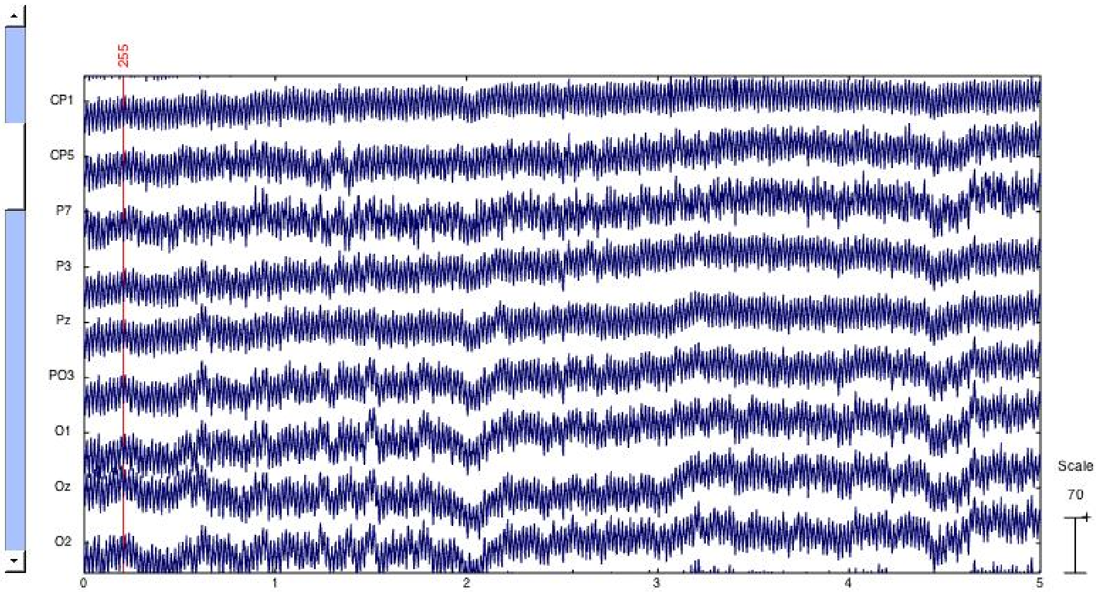

Figure 4: A clip of raw data for 8 channels across five seconds

Data source The EEG data was recorded by Professor Benjamin Godde and his research group for a motor learning study in older adults with 32 channels and at a sampling rate of 2048 Hz. In figure 4, one can see a short clip for the EEG recording for eight channels. Clearly, most of the signals demonstrate some periodic behaviors and on the lower half of the plot, channels like O1, Oz and O2 have more fluctuations within this time frame. The recording machinery is an active electrode system (ActiveTwo, BioSemi, Amsterdam, Netherlands) mounted in an elastic nylon cap [41]. The eventual usable data contains 79 subjects' resting EEG recordings for 80 seconds. There are 30 young subjects and 49 old subjects. The total EEG data is about 1.5 GB. In addition, for each subject, we have a few meta variables, some of which are going to compose MLA later on:

- Biological traits: age and gender

- Age group: a binary class variable to denote if a participant belongs to the young or old group
- Moca: an indicator value for the risk of dementia
- VO_2 peak: peak oxygen consumption in a stationary bicycle task for fitness level measurement
- MVC: max voluntary contraction, the max force between the thumb and index finger for 5 seconds
- PreRMSE and PostRMSE: the motor performance before and after the motor learning training

Since we are building a binary EEG classifier, the model construction pipeline is quite obvious. For this project, we follow the construction diagram in Figure 5. The whole experiment section mainly consists three stages: preprocessing, model building and postprocessing.

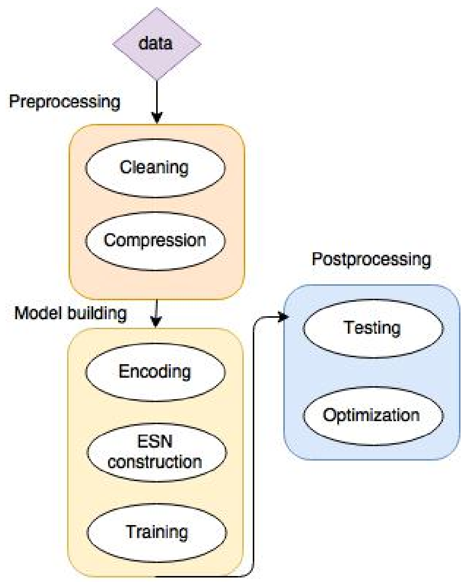

Figure 5: Data processing pipeline

Preprocessing As we already discussed in section 4, two of the main problems that an EEG classifier faces are the curse of dimensionality and low data-noise ratio. Therefore, cleaning and compression stages are necessary. For cleaning, we use a bandpass filter 0.5 - 79 Hz, then we apply ICA, in preparation for artifacts removal.[2]. After obtaining the independent components, we all the components that are marked as artifacts by MARA. Let's look at one example of MARA in our data.

In figure 6, we have 12 independent components of a subject's recording. Component one and component three are marked as artifact with high probability. In the current setup, all components marked as artifacts are removed by default, but if one wants to be more careful, a threshold on probability can be used. In component one, there is a strong impulse in the frontal area which should be mainly due to eye movements, and in component three on the right hemisphere where no brain tissue exists, we also have lots of activities, and thus it is not surprising that these two components are marked as artifacts.

Now we look at the effect of removing these two components. In figure 7 and figure 8, we can compare the differences to have a appreciation of the power of MARA. Notably, at around 11 seconds in figure 7, there exists a clear indication of artifact because we have some sharp amplitude changes which are more severe in the frontal lobe. We know this causation, because channels Fz F3 and F3 all show significant intensity changes. This effect should be attributed to eye movement as those channels are close to the eyes.

[2]We run the runica version ICA in EEGLAB.

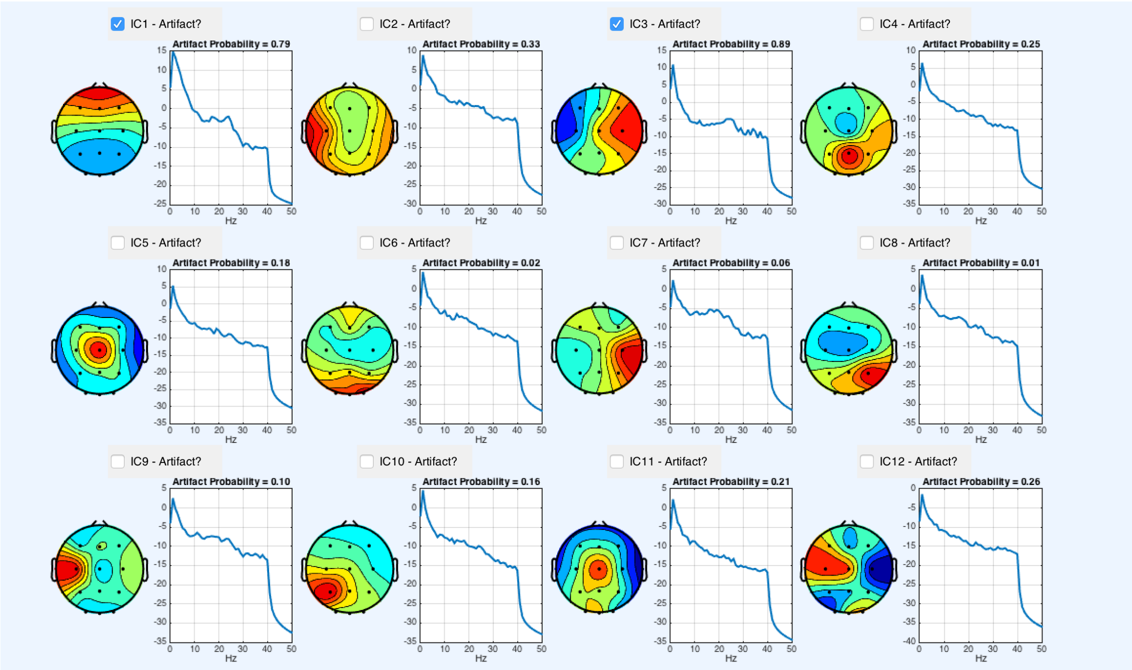

Figure 6: Artifact markers for a sample data stream by MARA.

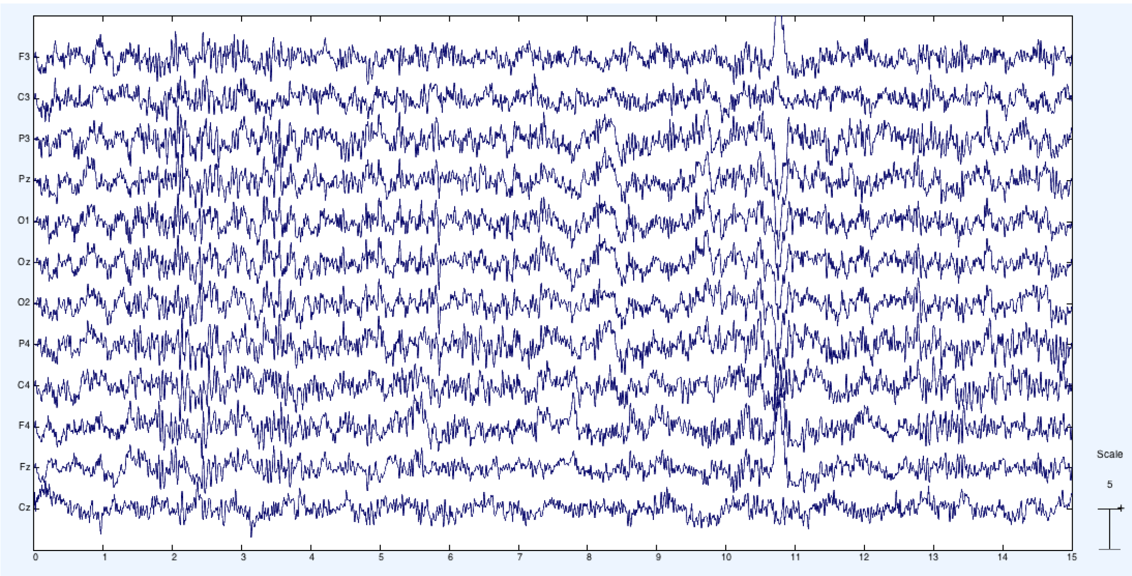

Figure 7: Data stream with corrupted artifacts.

After removing the eye movement component based on MARA, and we obtain the data in figure 8, which has much more brain-like oscillations around 11 seconds.

Furthermore, we normalize each input channel by

$$\bar{u}_i(t) = \frac{u_i(t) - mean(u_i)}{max(u_i) - min(u_i)} i \in [1, 12]$$

We understand this normalization approach for EEG data comparison across subjects is not optimal. This channel based method is adopted because EEG amplitude is also

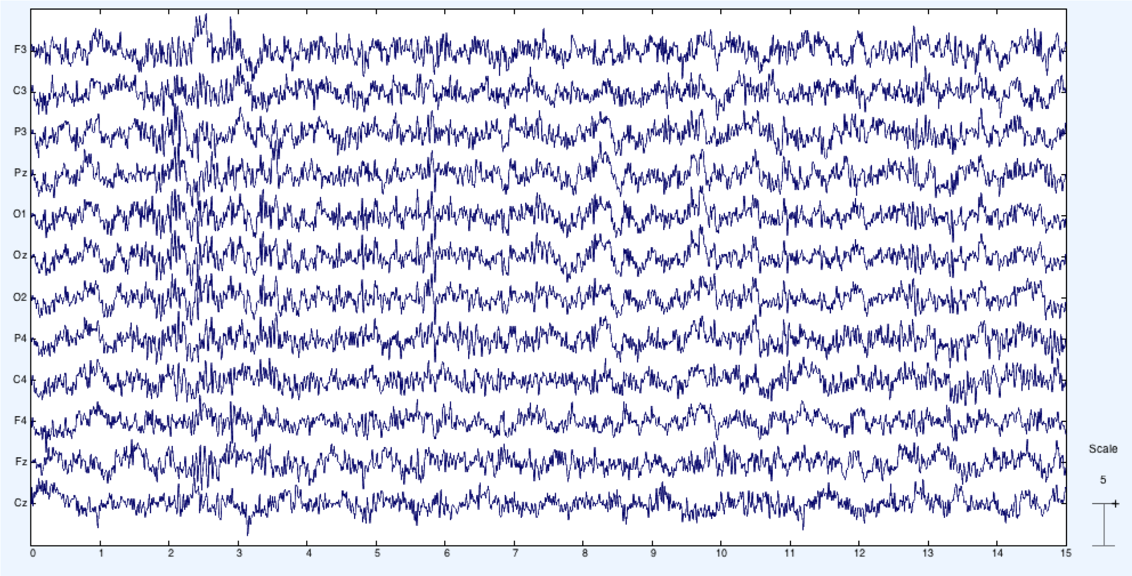

Figure 8: Data stream with artifacts removed using ICA and MARA.

influenced by age, gender and race. It is hard to just normalize each channel within one single subject without jeopardizing target-related information. A more careful approach is to test the statistical difference of EEG amplitudes with respect to the three factors mentioned above and then do a parametric normalization if the differences are significant.

Then, we remove the subjects whose EEG recordings are strongly influenced by artifacts and prior caffeine intake. Finally, we get rid of the relative MLS outliers which leaves us with 77 training samples.

In the compression stage, we choose only 12 electrodes (F3 C3 P3 Pz O1 Oz O2 P4 C4 F4 Fz Cz), because there are overlaps among electrode recordings. We use the 10-20 electrode placement system in our project, which uses numbers and letters to denote the brain regions [42]. The letters stand for the general areas: F(frontal), C(central), P(parietal) and O(occipital). The odd numbers mean represent the electrodes on the left hemisphere, and the even numbers represent the ones on the right. The exact locations of these electrodes on the scalp are shown in figure 9. Then we will down sample the frequency of EEG recordings from 2048 Hz to 256 Hz. Recordings at extremely high frequencies do not have the relevant information about brain processes. We denote each electrode's recording by $c_i.i \in [1, 12]$. All the preprocessing is done in Matlab using EEGLAB toolbox [43].

Model Building

- Encoding : After finished preprocessing the data, we define the relative MLS as

$$MLS = \frac{PreRMSE - PostRMSE}{PreRMSE}$$

The relative MLS is the main criterion that we use to label the subjects. In figure 8, we have the age (y axis) versus relative MLS (x axis) scatter plot. For the relative MLS, the smaller the value is, the more one's motor skills can improve after training. We have three groups of subjects: yellow dots old high performing, blue dots

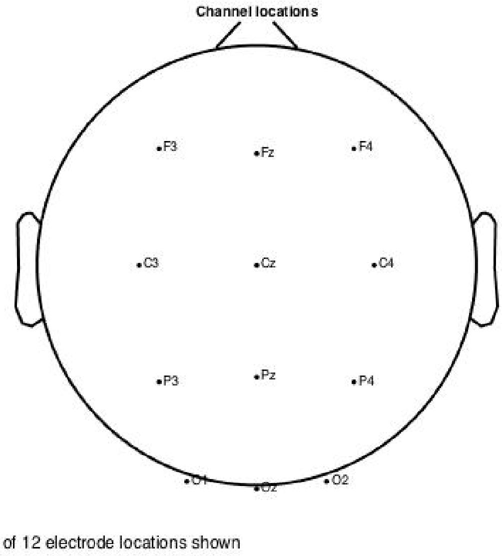

Figure 9: Locations of 12 selected electrode channels.

old low performing and purple dots young high performing. All young subjects are considered high performing and among the old subjects, we use the median to differentiate the high performing and low performing subjects. So after the encoding, the number of subjects in different groups is shown in table 1.

Group name	Subject count
Young High (YH)	28
Old Low (OL)	23
Old High (OH)	24

Table 1: Subject counts in different group after encoding.

- ESN construction: We use 150 internal units as the starting point. We also use leaky neurons to update. Then we construct the teacher signal, a vector of size two by one, one element being one and the other being zero depending on which MLS group this subject belongs to. In the predication phase, we first exploit the inputs $\mathbf{u}$ by using the updates rules defined in equation (2), and obtain $\hat{\mathbf{y}}$ of size 2 by T. The class of $\mathbf{u}$ thus becomes:

$$class_{idx} = max_{rowidx} \frac{\sum_i^T \mathbf{y_{1i}}}{\sum_i^T \mathbf{y_{2i}}}$$

- Training: We use the ridge regression with regularization which is discussed in section 2.1. In our experiment we train three binary classifier in total: young high performers vs old high performers, young high performers vs old low performers and old high performers vs old low performers.

Post-processing As we know, EEG classification is prone to over-fitting, so after the model's been constructed, we fine tune washout threshold, spectral radius, regularization

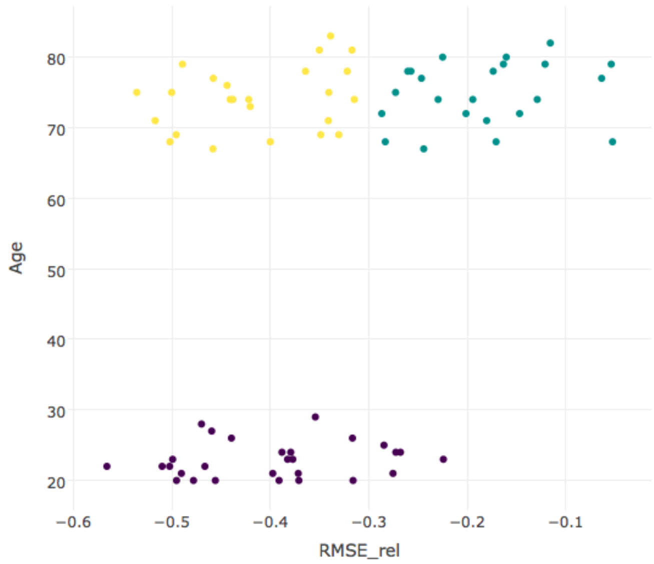

Figure 10: The scatter plot of age vs the relative motor training improvement is shown. The y-axis is for age and the x-axis is for the relative motor training improvement. The smaller the x value is, the more one has improved after motor training. Yellow dots are old high performers; blue dots are old low performers; purple dots are young high performers.

and the leaking rate to get better results [12]. These four parameters are mostly dealt with and the other parameters are disregarded in this experiment.

Stratified three-fold cross-validation In the evaluation phase, we use stratified three-fold cross-validation. Conventional cross-validation in our experiment setup wouldn't work well because we only have two classes and a small number of samples in the order of tens, and therefore in the cross-validation phase mere random allocation of the data samples to different folds might lead to a situation where one class is only present in the testing but not in the training phase which will surely leads to bad performance. A good evaluation method should have both low bias and low variance and k-fold stratified cross-validation is general better than the regular version [44]. We only have 77 subjects and three groups, but we are constructing three mutual binary classifiers among these groups, which gives around 50 subjects for each model. Hence we use the stratified three-fold cross-validation scheme to evaluate the model results. The author also finds that the variations of errors between different folds and randomization differ a lot, and hence we always use the Mersenne Twister method with seed 0 [3] to tackle this instability.

Parameter selection The washout threshold stays reasonably stable when other parameters are changing, so we pick the washout threshold first. The deciding factor is when we move back the threshold, the classification result does not fluctuate too much, and at the same time that the internal neuron responses remain relatively stable. The first value of such a point should be the washout threshold. In our experiment, we found out 550 is a good enough washout threshold.

[3] In Matlab, this can be set by 'rng('default')'.

The parameter values for the YH vs OL performers and YH vs OH are quite similar so we just showcase the validation error behavior on the YH vs OL model.

After washout threshold is being set, we first do a sparse parameters manual search to locate the range of the values for the later systematic tunning, then we do a grid search on the leaky rate α and the spectral radius ρ. The final step is to increase the number of internal units to improve accuracy. In this project, the number of units actually has a strong influence on the performance due to the complexity of brain data.

Even with network of size 1000, we are still not overfitting yet as you can see from figure 11. On the x-axis, we having regularization term starting from 0.0000001 to 0.01. For each iteration we multiply the regularization term by 5. Neither the training or testing error benefits from more regularization. We also found out that the bias term scale does not make a difference here, so we will just use one.

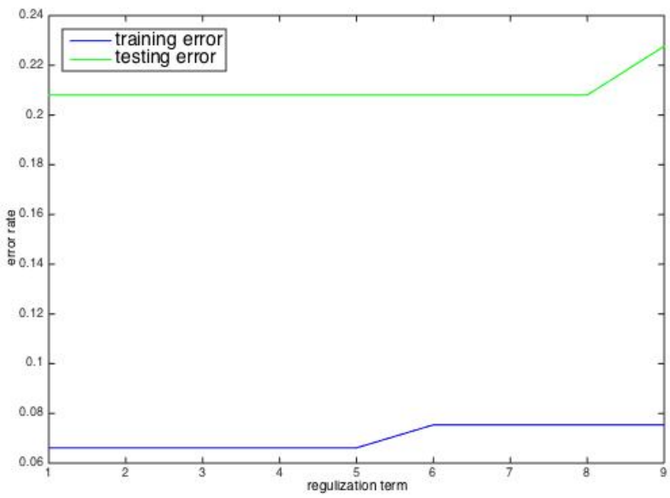

Figure 11: Training and testing error with respect to regularization term

5 Results

5.1 Network dynamics

In figure 12, we have two sample network outputs for both the right and wrong classifications. The simulated class signals are the network outputs and the true class signals are the ground truths. Clearly, for the correctly classified output on the left, the range for the network output span wider between 0 and 1, however for the wrongly classified output, the range for the network output is more narrow and the two simulated class signals swap the leading position more frequently.

We then take a look at the interval units dynamics. In figure 15, we can observe all inputs vary between -0.5 and 0.5, and it is hard to find a correlation between the inputs and interval units just by eyes. As for the internal units dynamics in figure 16, different units have different ranges and show different frequencies as well, for instance, unit 42 seems

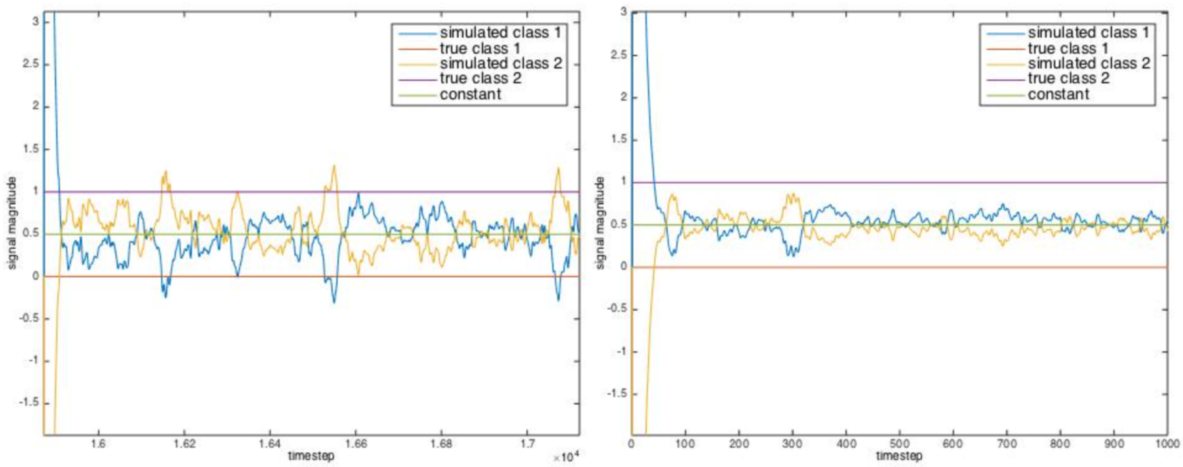

Figure 12: The left chart is a sample network output that has been correctly classified, and the right chart is a sample network output that has been wrongly classified

to change faster than unit 34. The amplitudes of the internals units are still narrow, not making use of the full range between -1 and 1 either.

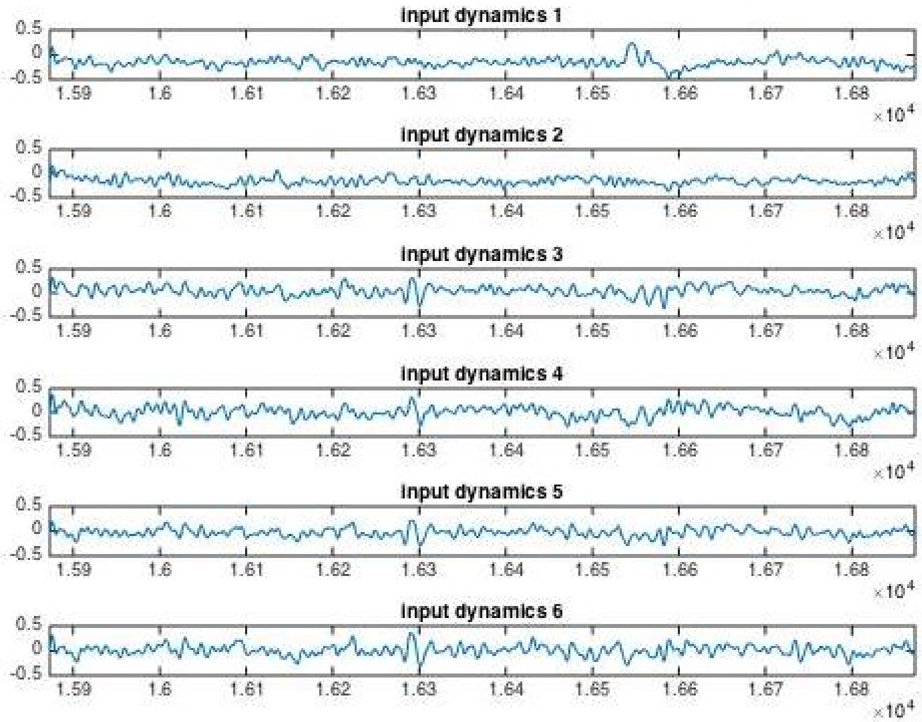

Figure 13: Sample input dynamics for a correctly classified example.

5.2 Model performance

The parameters used are $\rho = 0.4$ and $\alpha = 0.1$ for OH vs YH on a network whose internal units is of size 1500, and the parameters for the OL vs YH model is quite similar just

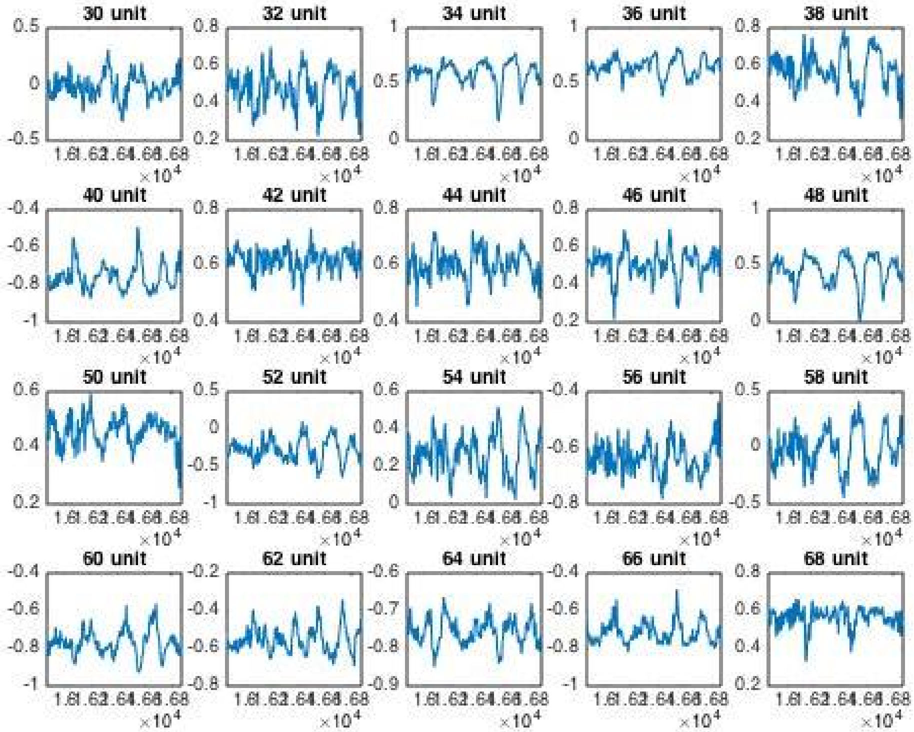

Figure 14: Sample network dynamics for a correctly classified example.

changing ρ to 0.1. However, for the OH vs OL, the author could not find a good enough parameter to make the testing error better than random chance. Regularization is set to zero for all models.

Model type	Training Error	Testing Error
model a: OH vs YH	0.0947	0.2070
model b: OL vs YH	0.0659	0.2092
model c: OH vs OL	0.1667	random chance

Table 2: Training and testing errors for different models.

From table 2, one can see model a and model b achieve similar performances, albeit model c is hopeless.

Confusion matrix is constructed by summing up for target and output labels across our stratified three-fold classification. Let us look at the matrix for model a first, there exists a systematic preference towards YH as 37.5% for OH are misclassified sa YH. The accuracy for OH (88.2%) and the recall for YH (93.1%) are reasonably good, however the accuracy for YH $(75.0\%$)and recall for OH (62.5%) are worse off. The confusion matrix for model b follows a similar pattern, there exists a systematic bias from OL to YH, given that not a single YH has been misclassified as OL but about 20.8% of OL being misclassified as YH. Both accuracy for OL and the recall for YH are perfect but the accuracy for YH (72.5%) and recall(54.2%) for OL are less satisfying. As for model c, no strong bias is observed in the misclassified samples.

Interestingly, If we use model b to test on the OH group, model b will predict that all sub-

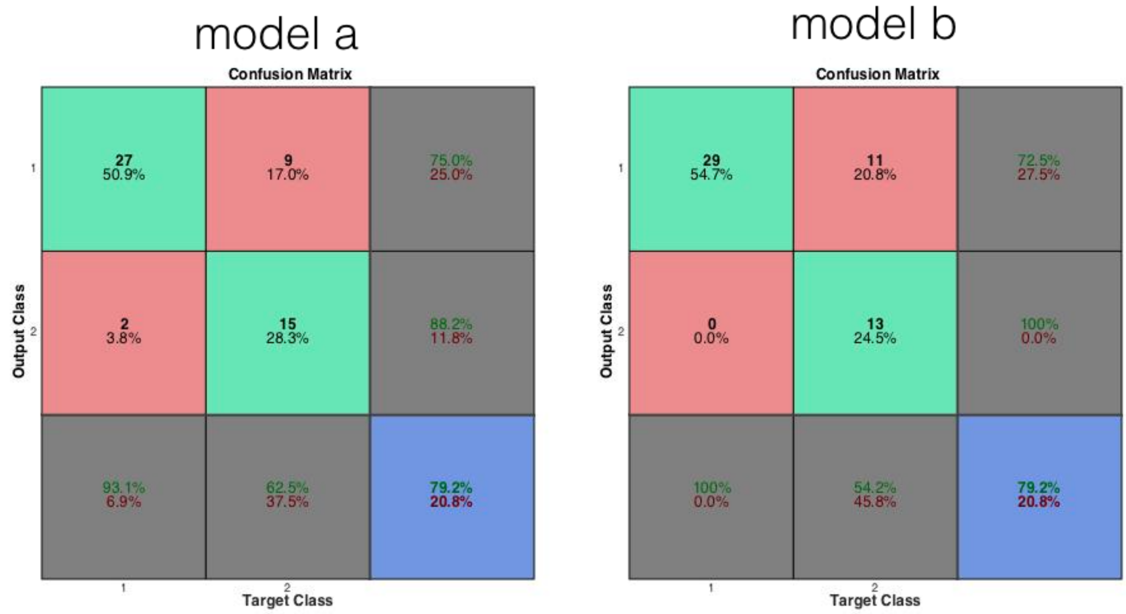

Figure 15: Confusion Matrix for model a and b. For model a, class 1 is YH, and class 2 is OH. For model b, class 1 is YH, and class 2 is OL. The green texts are marginal correctness percentages and the red ones are the error rates.

jects in OH belong to YH. If we use model a to test on OL group, model a will categorize all subjects in OL into OH.

6 Discussion

Engineering analysis As usual, we first talk about the engineering side and then go to the neurological side. Model a and b, which classify between young and old subjects, perform reasonably well, meaning, being able to separate between the YH vs OL and YH vs LH, which human experts are not able to do. So ESNs can indeed pick up some spatiotemporal features that EEG contains.

Now let us look at pitfalls and possible extensions. There exists an obvious gap between the training and testing error. Regularization term does not help with the performance much when the network has 1000 internal units based on figure 11, in which neither the training or the testing error improves with respect to the regularization coefficient. This could imply that the network is still underfitting. This is not too surprising, when one considers the amount of data we have. Each time series has 15k timesteps after down-sampling, and we have 12 channels for each series. For each model, there are approximately 50 subjects to work with, and we using three-fold cross-validation which means around 30 subjects are used for training, which leads to 15k * 12 * 30 = 5.4M number of data points for training.

When we look at the network output in figure 12, the network decision is quite uncertain when dealing with complex data like EEG. Although for the rightly classified data, we can see the amplitude of the output is greater, and leading position of the two competing class signals switches less frequently when being compared with the one that is wrongly classified.

From both figure 13 and figure 14, we notice that inputs are not “well normalized” in a sense that the inputs are not making use of the full range between -1 and 1. We already discuss this issue in section 2.2 and possible extensions by using parametric normalization in order not to lose target-related information. As a consequence of the current implementation, the network will surely find it easier to classify the ones with strongly amplitudes, hence for the ones that have small amplitudes, the internal units are not making use of the full range between -1 and 1 either. Therefore, a more complicated normalization routine should be able to ameliorate the classification performance.

Finally, we have come to discuss to the bias encountered in the confusion matrix in model a and b. Particularly in model b, all the misclassified subjects actually belong to OL, and YH can be perfectly separated. One possible explanation might be, the uneven distribution of training data, given we have 28 YHs and only 23 OLs. So the networks will by default bias towards YH.

Neuroscience analysis The hypothesis of the neuroscience question is, there exists two groups in OH where one subgroup's brains are more like those of YH, whereas the other subgroup's brains are more like those of OL, and there are some compensation effects happening that enable the second group to still perform well.

Based on the classification results, we can say that something is different between the young and old groups that enables model a and model b to perform well. The difference between OH and OL is hard to tell, since model c cannot perform better than random chance.

Another observation is, if we use model b to do a test on OH, the network will classify all the subjects to YH, which is to say that the network thinks the brain activity of the OH subjects are more similar to the ones of YH as compared to OL. On the other hand, model a (trained on YH and OH) classifies all OL to be OH, which indicates that indeed OH are similar to YH, and little or no compensation effects exist.

will discuss this with Godde before submission

7 Conclusion and future work

Previously ESNs have been shown useful for epileptic seizure detection. This project is yet another successful application of using ESNs on EEG data. We also observe evidence that disproves the existence compensation effects motor learning skills in the older subjects.

Furthermore, this guided research is also an attempt for correlation analysis in the realm of end-to-end machine learning techniques. Machine learning techniques are powerful, and they will become even more so, if we understand what is going on behind the scene, in addition to merely focusing on the model performance without much understanding.

Possible future works include constructing the gradient target value for MLS instead of using binary classification. Currently there is no evidence for the existence of compensating effects. This might be due to the the old subjects are too similar to each other, so the compensating effects are harder to detect. By using gradient target value or extreme values (very good performers in OH and really bad performers in OL), compensation, if really exists might, play a bigger role with respect to MLS.

One could also further develop general statistical frameworks of how to construct different classifiers and by analyzing their decision boundary for hypothesis testing and explore their mathematical nature.

8 Acknowledgments

I first want to thank Professor Herbert Jaeger, who gave me all the necessary guidance and who always pulled me back from dark corners of echo state networks. It simply has been incredible to work with him. I also want to thank Professor Benjamin Godde for his generosity in giving me the EEG data to with on and also for his patience in answering all my physiological queries.

References

[1] Rui Zhang, Dezhong Yao, Pedro A Valdés-Sosa, Fali Li, Peiyang Li, Tao Zhang, Teng Ma, Yongjie Li, and Peng Xu. Efficient resting-state EEG network facilitates motor imagery performance. *Journal of Neural Engineering*, 12(6):066024, 2015.

[2] Ozan Özdenizci, Mustafa Yalçın, Ahmetcan Erdoğan, Volkan Patoğlu, Moritz Grosse-Wentrup, and Müjdat Cetin. Resting-state EEG correlates of motor learning performance in a force-field adaptation task. In *Signal Processing and Communication Application Conference (SIU), 2016 24th*, pages 2253–2256. IEEE, 2016.

[3] M. Doppelmayr, W. Klimesch, W. Stadler, D. Pllhuber, and C. Heine. EEG alpha power and intelligence. *Intelligence*, 30(3):289 – 302, 2002.

[4] Herbert Jaeger. The echo state approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[5] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[6] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on Neural Networks*, 5(2):157–166, 1994.

[8] Pieter Buteneers, Benjamin Schrauwen, David Verstraeten, and Dirk Stroobandt. Real-time epileptic seizure detection on intra-cranial rat data using reservoir computing. In *International Conference on Neural Information Processing*, pages 56–63. Springer, 2008.

[9] Mohammad Ali Naderi and Homayoun Mahdavi-Nasab. Analysis and classification of EEG signals using spectral analysis and recurrent neural networks. In *Biomedical Engineering (ICBME), 2010 17th Iranian Conference of*, pages 1–4. IEEE, 2010.

[10] AA Petrosian, DV Prokhorov, W Lajara-Nanson, and RB Schiffer. Recurrent neural network-based approach for early recognition of alzheimer's disease in eeg. *Clinical Neurophysiology*, 112(8):1378–1387, 2001.

[11] Elif Derya Übeyli. Multiclass support vector machines for diagnosis of erythemato-squamous diseases. *Expert Systems with Applications*, 35(4):1733–1740, 2008.

[12] Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade*, pages 659–686. Springer, 2012.

[13] David Verstraeten. *Reservoir Computing: computation with dynamical systems*. PhD thesis, 2009.

[14] Wolfgang Maass, Prashant Joshi, and Eduardo D Sontag. Computational aspects of feedback in neural circuits. *PLoS Comput Biol*, 3(1):e165, 2007.

[15] W. Kahan. Further remarks on reducing truncation errors. *Commun. ACM*, 8(1):40–, January 1965.

[16] Herbert Jaeger. *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001.

[17] Benjamin Schrauwen, Jeroen Defour, David Verstraeten, and Jan Van Campenhout. The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In *International Conference on Artificial Neural Networks*, pages 471–479. Springer, 2007.

[18] Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[19] Mike X Cohen. *Analyzing neural time series data: theory and practice*. MIT Press, 2014.

[20] Michael X Cohen. It‘s about time. *Approaches and Assumptions in Human Neuroscience*, 2011.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[22] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013.

[23] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *arXiv preprint arXiv:1406.3676*, 2014.

[24] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.

[25] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[26] Tzyy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin J Mckeown, Vicente Iragui, and Terrence J Sejnowski. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology*, 37(2):163–178, 2000.

[27] Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic classification of artifactual ica-components for artifact removal in eeg signals. *Behavioral and Brain Functions*, 7(1):30, 2011.

[28] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4(2):R1, 2007.

[29] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2(2):1–13, 2000.

[30] Charles W Anderson and Zlatko Sijercic. Classification of EEG signals from four subjects during five mental tasks. In *Solving Engineering Problems with Neural Networks: proceedings of the conference on engineering applications in neural networks (EANN96)*, pages 407–414. Turkey, 1996.

[31] Ramaswamy Palaniappan. Brain computer interface design using band powers extracted during mental tasks. In *Neural Engineering, 2005. Conference Proceedings. 2nd International IEEE EMBS Conference on*, pages 321–324. IEEE, 2005.

[32] Divya Balakrishnan and Sadasivan Puthusserypady. Multilayer perceptrons for the classification of brain computer interface data. In *Bioengineering Conference, 2005. Proceedings of the IEEE 31st Annual Northeast*, pages 118–119. IEEE, 2005.

[33] Jd R Millan, Frederic Renkens, Josep Mourino, and Wulfram Gerstner. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51(6):1026–1033, 2004.

[34] J del R Millán, Josep Mourino, Fabio Babiloni, Febo Cincotti, Markus Varsta, and Jukka Heikkonen. Local neural classifier for EEG-based recognition of mental tasks. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 632–636. IEEE, 2000.

[35] Abu Farzan Mitul, Md Jubayer Alam Rabin, Muhammad Rakeeb, Abdullah Al Mamun Khan, GM Sultan Mahmud Rana, AbuShahab Mollah, and Md Hafizur Rahman. Classification of real time moving object using echo state network. In *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*, pages 1–6. IEEE, 2013.

[36] Femke Ongenae, Stijn Van Looy, David Verstraeten, Thierry Verplancke, Dominique Benoit, Filip De Turck, Tom Dhaene, Benjamin Schrauwen, and Johan Decruyenaere. Time series classification for the prediction of dialysis in critically ill patients using echo statenetworks. *Engineering Applications of Artificial Intelligence*, 26(3):984–996, 2013.

[37] Robert W Thatcher, Duane North, and C Biver. EEG and intelligence: relations between EEG coherence, EEG phase delay and power. *Clinical neurophysiology*, 116(9):2129–2141, 2005.

[38] Richard J Davidson. Affective neuroscience and psychophysiology: toward a synthesis. *Psychophysiology*, 40(5):655–665, 2003.

[39] John Kounios, Jessica I Fleck, Deborah L Green, Lisa Payne, Jennifer L Stevenson, Edward M Bowden, and Mark Jung-Beeman. The origins of insight in resting-state brain activity. *Neuropsychologia*, 46(1):281–291, 2008.

[40] Jennifer Wu, Ramesh Srinivasan, Arshdeep Kaur, and Steven C Cramer. Resting-state cortical connectivity predicts motor skill acquisition. *NeuroImage*, 91:84–90, 2014.

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

[42] George H Klem, Hans Otto Lüders, HH Jasper, C Elger, et al. The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, 52(3):3–6, 1999.

[43] Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, 2004.

[44] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1137–1145. Stanford, CA, 1995.