1 - Introduction

Patients who have a form of epileptic seizures exhibit excessive inhibition or propagation of neural signals. Signals from these patients can be analyzed with neural networks, an extremely useful tool in understanding what a dataset represents. The accuracy of these models is far greater than those of standard machine learning models. In this laboratory, we explore multiple types of neural networks to understand their inner workings in order to achieve rapid and accurate methods of classifying epileptic seizure data. Creating fast and accurate decoders is vital to predict epileptic episodes and provide the necessary treatment in a timely manner.

---

2 - Methods

2.1 - Algorithm Descriptions and Rationale for Development, and Implementation

2.1.1 - Chaos Networks and FORCE Learning

Chaos networks are echo state networks (sparsely connected, pseudo-randomly organized neural networks that have normally distributed connection weights) that utilize a parameter *g* (apdx: Table 1, #2) to relate the variance of network's weights to the number of neurons in the network. When this parameter is greater than 1, the connection weights in the neural network take on the property of being chaotic, in that the weights are continuously changing. FORCE learning tries to establish arbitrary patterns in the given data by creating a random, concurrent network and adjusting weights based on minimization of the error during each timestep (apdx: Table 1, #1). Chaos implies that the network has a high degree of seemingly random weights. This network was implemented using MATLAB, observing changes to the weights as g changed.

2.1.2 - Forward Propagation

Forward propagation is a technique used in deep neural networks to pass information between layers. Each layer contains a different set of neurons and weights, and information flows forward, from input to output layers, hence the name "forward propagation". This technique entails multiplying an input set with weights and applying a nonlinear transform to gather a set of numbers that will be used as an input to the next layer or simply as the output of the current layer. One common transform is known as the sigmoid function (apdx: Table 3, #1) which fits an input onto an S shaped curve. This network was implemented using MATLAB and a custom neural net class definition (apdx: Code Section) which contains a suite of functions for implementing a neural network including a forward propagation function. The data used to test forward propagation contained firing rates of 95 neurons for an 8 direction reach task executed by a monkey.

2.1.3 - Deep Learning and Autoencoders

Autoencoders are neural networks that learn a representation of a given input dataset, and typically try to solve the problem of data encoding. They utilize linear algebra to recreate data that is similar to the input dataset, in a lower dimensional space. Deep networks utilize multiple layers of autoencoders that have different numbers of neurons. MATLAB has

a deep learning toolkit that can train neural networks given specified parameters such as layer size, weight regularization, sparsity regularization, and this toolkit provides a clean and quick way to analyze data. The parameter of interest in this section was the number and size of layers used in autoencoders.

2.1.4 - Convolutional Neural Networks (CNNs)

CNNs are layered neural networks inspired by the structure of the visual cortex in animals, meaning that many neurons operate on small portions of the input data and work together to build up progressively larger features of the data. These features can then be combined to generate a classification or prediction. The CNNs used in this lab are trained using transfer learning, which enables faster training of a network because only the layers at the end of the network need to be retrained. Using MATLAB, A predefined network trained on images of letters is retrained to classify digits. The last three layers of the network were replaced with a fully connected layer. Classification accuracies were gathered at multiple different training-test data ratios.

2.1.5 - Seizure Detectors

Seizures produce oscillations in EEG data that follow patterns detectable by machine learning algorithms. Specifically, these detectors can utilize features such as signal amplitude, zero crossings, and line length to parse the data and indicate seizure activity. These features can highlight portions of the seizure related EEG data and machine learning models can be trained to detect these significant parts of the data. This detector was run on EEG data which contained three channels sampled at 3 kHz and came from two patients who exhibited seizure conditions. The procedure followed was to train the detector on one patient (learn the three features mentioned before) and test it on the other.

2.1.6 - High Frequency Oscillation Detectors (HFOs)

Seizures are composed of many bands of frequency in their signals. These can be decomposed using tools such as Fourier Analysis to understand its components, but the next step would be to detect a certain specific frequency band. High frequency oscillations are defined as features of EEG data that occur above a rate of 80 Hz. These signals are nontrivial to extract from the dataset as they occur at temporal modalities of 20 to 150 ms and can be difficult to isolate from noise in the signal. Extracting HFOs from EEG data is important because the theories of HFO formation mirror those of seizure formation, allowing seizures to be localized spatially and temporally. Data was filtered to contain only HFOs and used to train an algorithm to detect HFOs from an untouched dataset.

3 - Results

3.1.1 - Chaos Networks and FORCE Learning

When the g parameter is above 1 (specifically 1.5), the network's weights are chaotic, in that they do not converge as time passes on (apdx: Figure 2). This value for g also causes the accuracy of the network to improve, as the mean average error of the network decreases when g

increases (apdx: Table 3). This network also has the property that at a g value of 1.5, a network size of about 380 neurons is the minimum number of neurons that results in an error of ten times that of the default network size of 2000 neurons.

3.1.2 - Deep Learning and Autoencoders

Modifying the size and number of layers of autoencoders changes a network's accuracy. In (apdx: Figure 3) one can see that utilizing a training and test ratio of 60% to 40%, or vice versa, results in the highest accuracy with an autoencoder that contains two layers with ten neurons each. The weights utilized for this type of autoencoder can be visualized (apdx: Figure 4). Utilizing more layers and more neurons results in changes to the accuracy (apdx: Figure 5) of a network, in that adding more layers (Autoencoder 2 in Figure 5) actually exhibited decreased performance when compared to simply adding more neurons to a layer.

---

4 - Discussion

Neural networks are an incredibly useful tool for analyzing signals from the brain because they offer incredible flexibility by having the property of many parameters. Changes to the size and number of layers, what types of transfer functions are used, the rules for transferring information about weights, and many more allow neural networks to learn about a wide range of datasets. Another variant of neural networks, known as reinforcement learning, allows for a type of agent to learn about the dataset by attempting different actions, and these could provide a sense of artificial intelligence to solving this problem. The power of these tools is incredibly valuable to patients with seizure related illnesses, as they can provide a predictive element to administering treatment, and (although potentially expensive to compute) these give more insight onto the origins and propagation of seizure activity. The networks that were explored in this laboratory range from echo state networks, which provide solid performance for sparse networks, to deep convolutional neural networks, which can be powerful enough to map high dimensional datasets into smaller representations. Each of these networks is useful for a different domain, and modifying their parameters can aid a wide variety of patients.

For situations where neural networks are inefficient or superfluous for tasks such as decoding, understanding new ways to pick out features of a dataset is useful for small time domain situations. Both neural networks and decoding are important factors to have in analyzing an epileptic situation, and more work should be done to accelerate the accuracy and efficiency of these algorithms.

BIOMEDE 517 - Neural Engineering - Labs 10 and 11 - Dr. Chestek

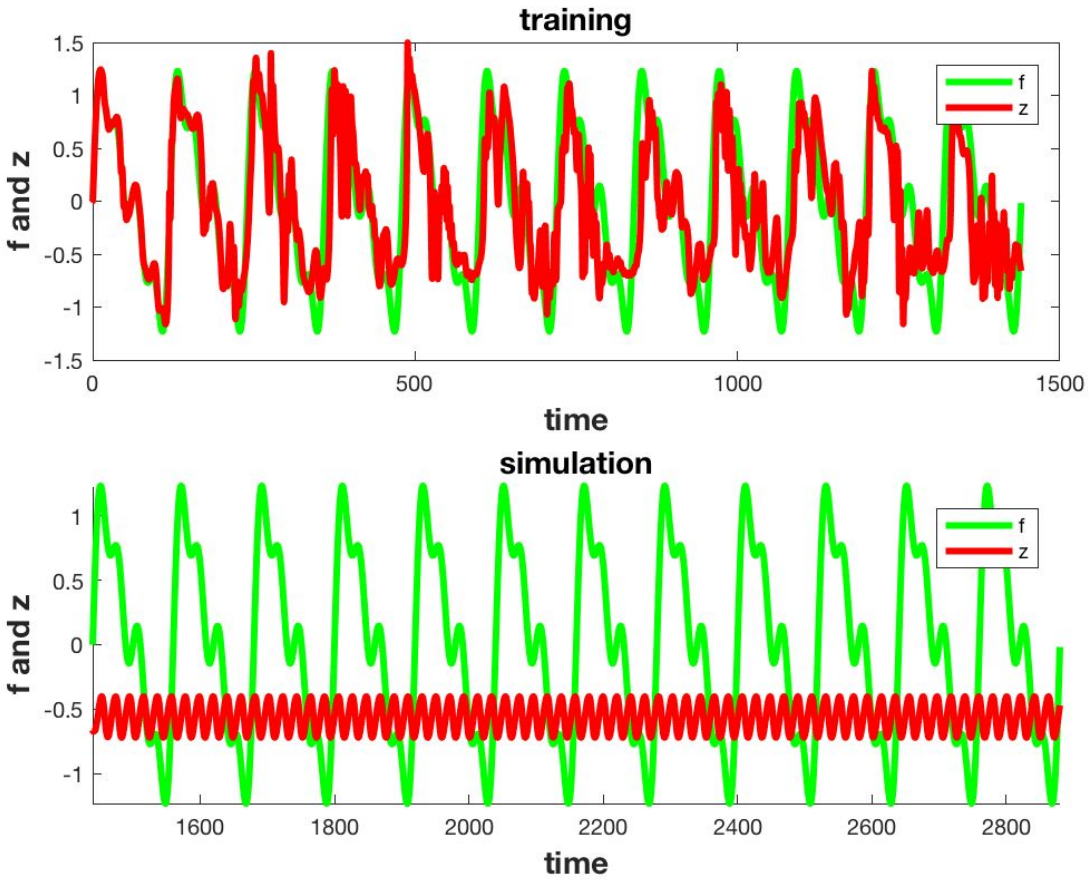
Appendix

Kushal Jaligama

Table 1 - Chaos Networks and FORCE Learning Equations and Results

#	Equation Name	Equation
1	FORCE Learning Process	1. Create random, recurrent network $y_j(t+\Delta t) = M_{ij}\vec{r}_i(t) = M_{ij}\tanh(y_i(t))$ 2. Construct random output readout $z(t) = \vec{w}^T\vec{r}(t)$ 3. Simulate timestep and calculate error $e(t) = z(t) - f(t)$ 4. Adjust weights based on the error $w(t+\Delta t) = \vec{w}(t) - e(t)\vec{P}(t)\vec{r}(t)$
2	*g* Parameter	Zero mean: $\mu = 0$ Variance: $\sigma^2 = \frac{g^2}{N}$ $g > 1 \rightarrow$ Chaotic firing

Figure 1 - Training, Test, and Weight Values for g = 0.9



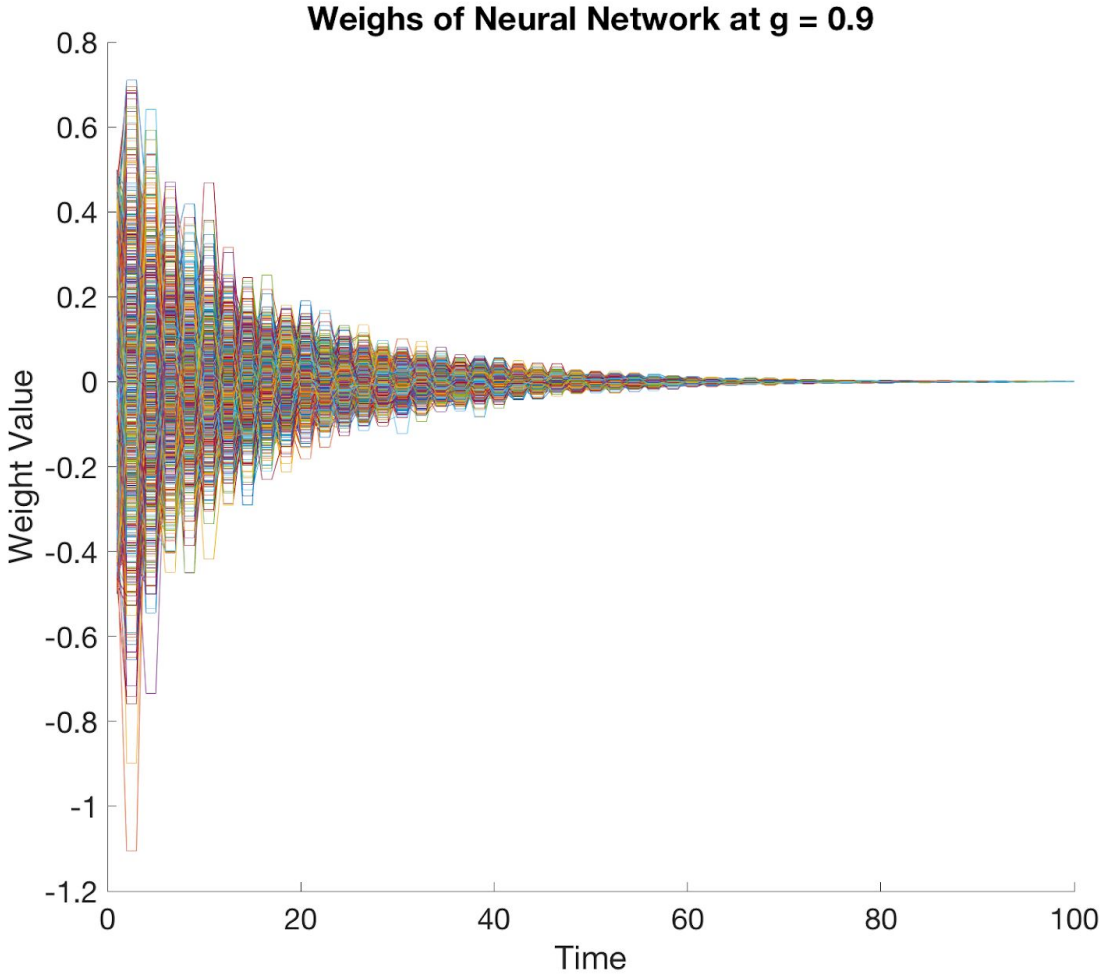
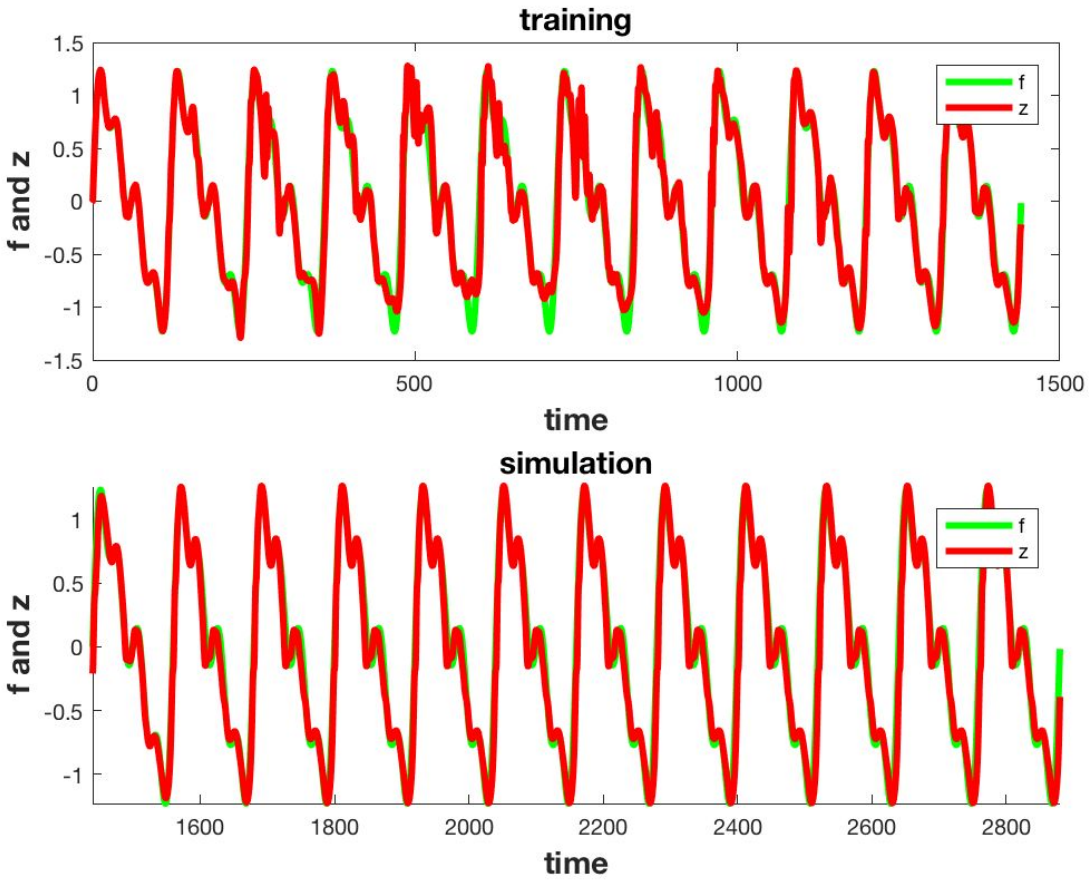

Weighs of Neural Network at g = 0.9
Weight Value
Time
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0
20
40
60
80
100

Figure 2 - Training, Test, and Weight Values for g = 1.5



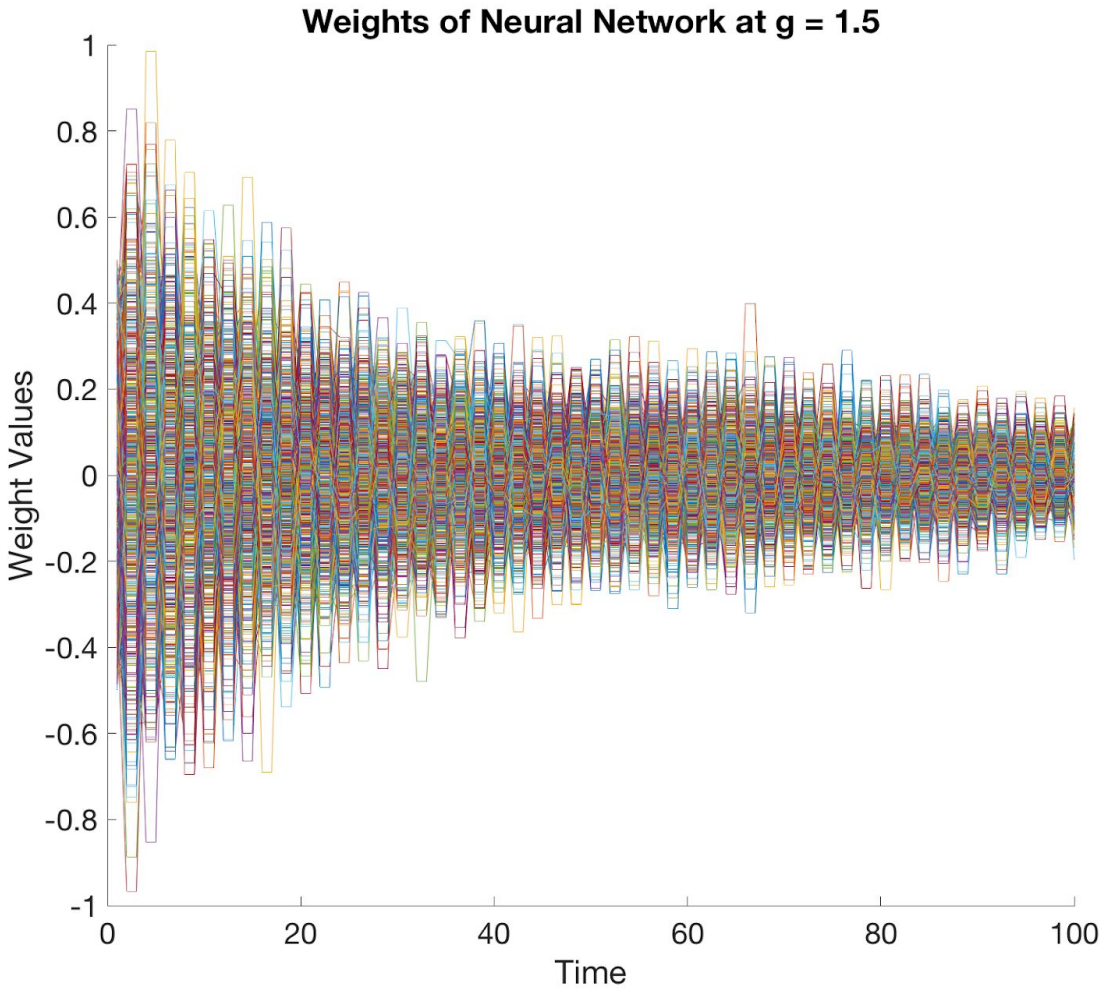


Table 2 - Mean Error of Echo State Network As G Changes

G Value	Training Error	Testing Error
0.9	0.333	0.785
1.0	0.194	0.837
1.1	0.14	0.722
1.2	0.0242	0.799
1.3	0.0122	0.0152
1.4	0.00966	0.0995
1.5	0.00472	0.0318

Table 3 - Forward Propagation

#	Name of Nonlinear Transform	Equation
1	Sigmoid Function	$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}.$

Figure 3 - Autoencoder Classification Accuracy

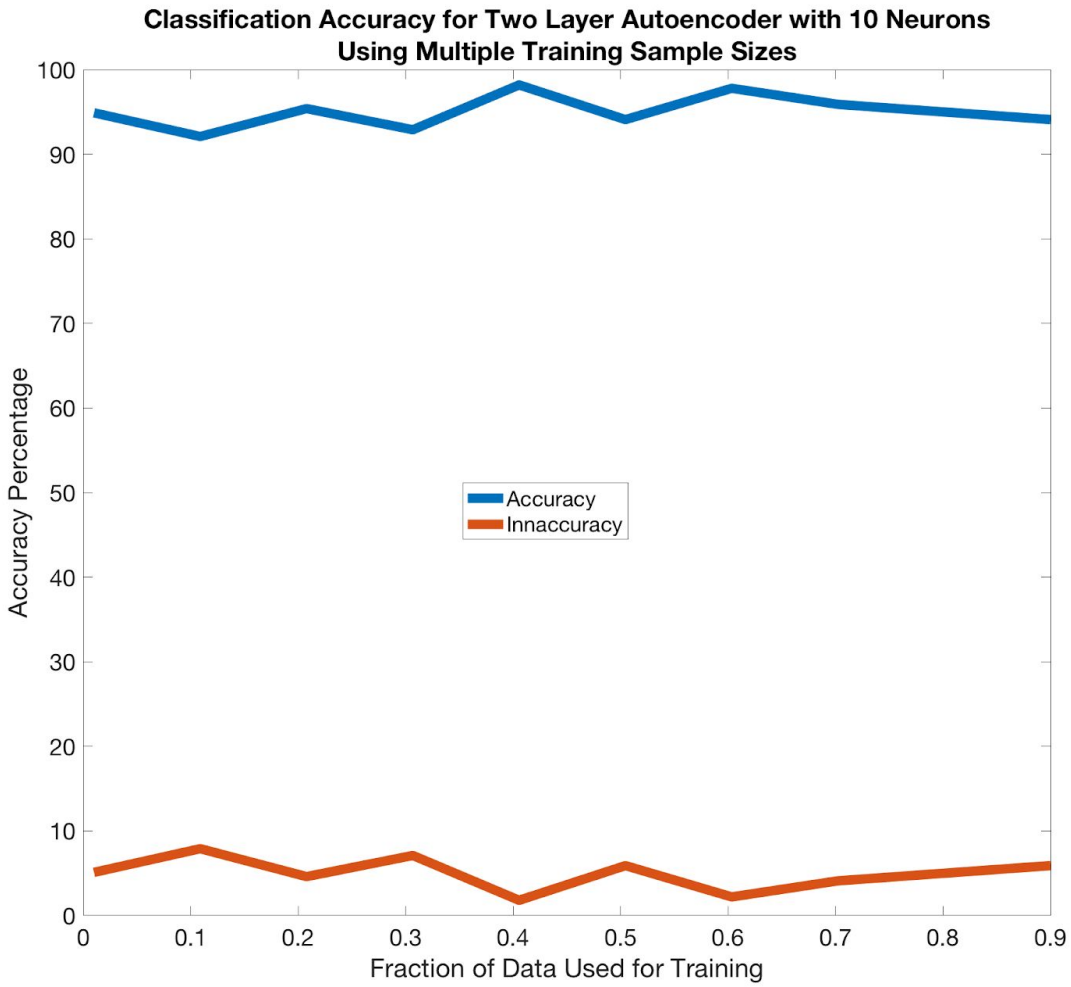


Figure 4 - Autoencoder Weights

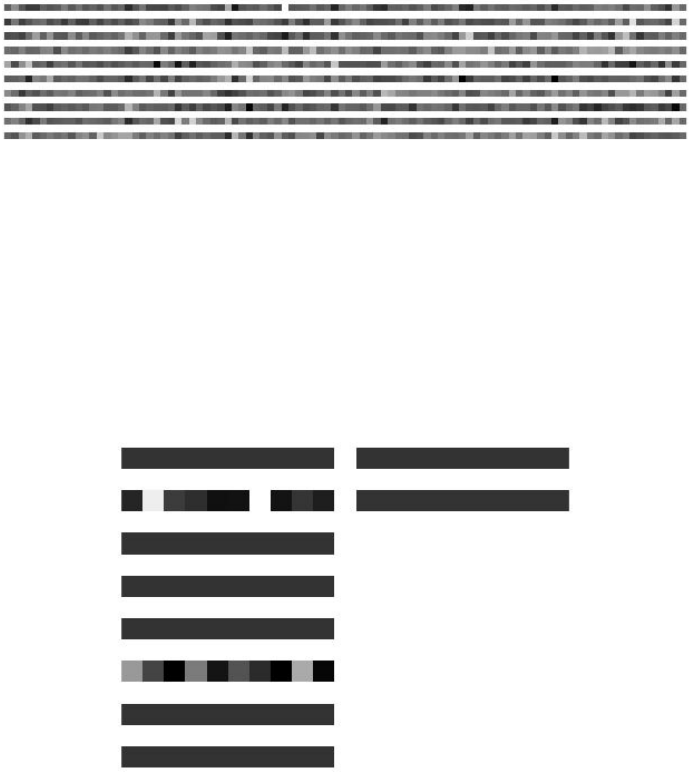


Figure 5 - Multiple Autoencoder Classification Accuracy