**PROJECT REPORT**

**Brain Signal Analysis**

Name: Tanmay Jain

University: Bennett

Mentor: Dr. Rishav Singh

Team Number :11

**Definition**

# Domain Background

The electroencephalogram (EEG) is a bio-signal just like MRI, CT, f-MRI, ECG. It is the recording of the electrical activity of the brain from the scalp. Whenever we do some activities neurons are fired in the brain which generates some current, we use electrodes on the scalp and measure that very tiny current and then amplify it. In various researches, it is found that our brain generates different EEG time-series patterns for different activities like moving hands, legs, eyes opening and closing etc. So if we use machine learning to recognize those patterns then we can easily classify them.

# Problem Statement

Given EEG pattern we have to classify motor imagery of left and right hand.

# Performance Metrics

Accuracy alone is not the way to measure the performance of our project.

Example – Let’s say you have 100 apples out of which 90 are of green and 10 are of red. If we just create a naive classifier that gives only green as an output for every data point then,

Accuracy = 90/100 = 0.9

This is totally useless model and this is called as accuracy paradox. This is where Precision and Recall comes in.

**Dataset Input**

The subjects were right-handed, had normal or corrected-to-normal vision and were paid for participating in the experiments. All volunteers were sitting in an armchair, watching a flat screen monitor placed approximately 1 m away at eye level. For each subject 5 sessions are provided, whereby the first two sessions contain training data without feedback (screening), and the last three sessions were recorded with feedback. Each row consist of 14 EEG values (means 14 time-series values) representing 14 electrodes.

# 

**Project Design**

# **Algorithm and Techniques –**

Statistical Feature Extraction– As we know to compare two timeseries data we can extract statistical features:

1.Mean

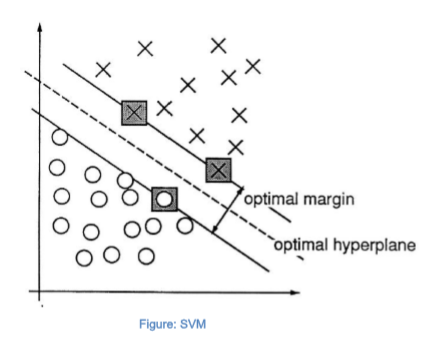
2.Standard Deviation

3.Skewness (Skewness is a measure of symmetry, or more precisely, the lack of symmetry.)

4.Kurtosis (Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution)

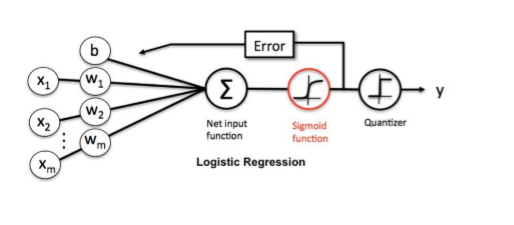
## SVM

SVM maps the input data to high dimensional feature space through kernels, so that it can be separated by a hyperplane. Because of different kernels SVM will provide good accuracy in our problem. In implementation we have used ‘rbf’ kernel which was giving better results than other kernels (poly, linear).



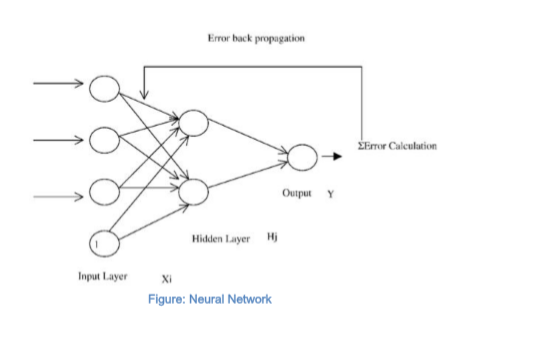
## Logistic Regression

Logistic Regression is another technique for classification, which use sigmoid for two class classification and uses cross-entropy function as a loss function. Justification - Logistic Regression are used as benchmark models because of simple algorithm with non-linear sigmoid function they provide good baseline accuracy.



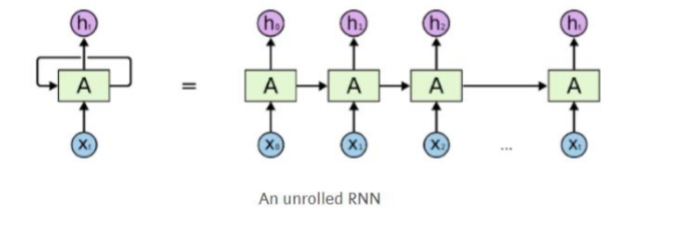
## Neural Networks

Artificial Neural Networks are inspired from our Brain Neural Networks. ANN randomly assigns weights to all layers then use backpropagation for decreasing the error rate and updating the weights. Deep Neural Networks have specialty of function approximation on lot of different type of data. Best part of neural network is that at each layer automatically important features get extract, that’s what make neural network great in field of Machine Learning.



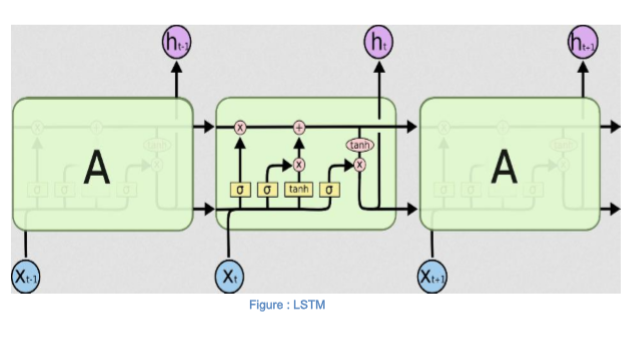
### Recurrent Neural Network

The basic idea of RNN is to make use of sequential information. In traditional neural networks the data was assumed to be independent, but as we know data points in Time Series are not dependent, future values depend on previous values. A RNN has loops in them that allow information to be carried in neurons while reading the inputs but RNN has few drawbacks like vanishing and exploring gradient problem.



## LSTM (Long Short Term Memory)

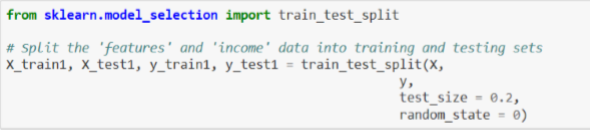
To overcome vanishing gradient problem of Recurrent Neural Network, LSTM was made, which uses a different function to compute the hidden state. LSTMs also have chain structure but instead of having a single neural network layer, it uses 4 – cell, an input gate, an output gate and a forget gate as shown in figure. Justification - LSTM will be very useful in our problem as our data is time – series data, it can capture recurrence relation of the data.



**Implementation details**

Pre-processing the data

a) In this step first we will first separate data (X) and label (y) from raw dataset.

b) Then split the data into train and test in ratio of 80 and 20.

For training in neural network we will normalize train and test data using min – max scalar normalization. This normalization is done so that neural networks can converge faster.





* Feature Extraction – In this step for training on SVM and Logistic Regression we will extract Statistical Features as discussed above.
* Training on Models – In this step we will create SVM, Logistic Regression, Neural Network, LSTM for training.
* Compare the results – After training on training models we will calculate Accuracy, Precision, Recall, F1 Score for different models.

# Implementation

## Neural Network

a) - For creating Neural Network Keras library is used with TensorFlow backend.

b) - To make model we have import Sequential model from Keras.

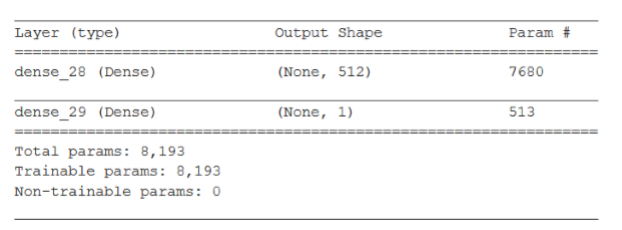
from keras.models import Sequential

c) – Then we can add Dense layers by the following code. model.add(Dense(number\_of\_nuerons, activation\_function, input\_shape))

d) – Activation ‘relu’ function is used in hidden layer for faster convergence and ‘sigmoid’ is used at output layer for classification.

Whole model snippet –



Model Summary – 

Optimizers – ‘Adam’ to overcome local minima.

Loss function – Binary cross entropy as we have 2 classes to classify.

## LSTM

a) First we have to convert our dataset into 3D as LSTM takes 3D array as input.



X\_train.shape[0] gives us total numbers of dataset rows.

14 is used as window size as we want our LSTM to observe all 14 values at a time.

1 indicates that one feature is one observation at a time step.

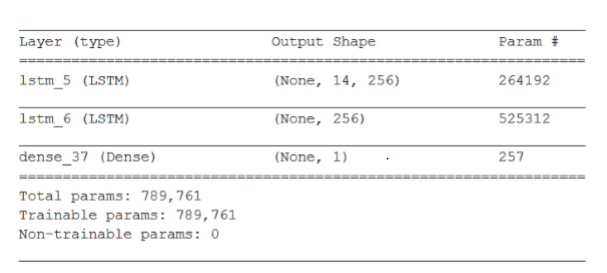
b) Create a Sequential Model in Keras.

c) Then add LSTM layer using following code snippet –

model.add(LSTM(no\_of\_neurons, input\_shape, return\_sequences = True))

return\_sequences - returns the hidden state output for each input time step.

Model Summary:



# **Conclusion**

# Results

SVM

Accuracy = 0.6989319092122831

Precision = 0.7059190031152648

Recall = 0.7248880358285349

F1 Score = 0.7152777777777777

Logistic Regression

Accuracy = 0.5240320427236315

Precision = 0.5404607206142942

Recall = 0.5854126679462572

F1 Score = 0.5620393120393121

Neural Network

Accuracy = 0.7557788944723618

Precision = 0.7142857142857143

Recall = 0.8642892521050025

F1 Score = 0.7821604661586732

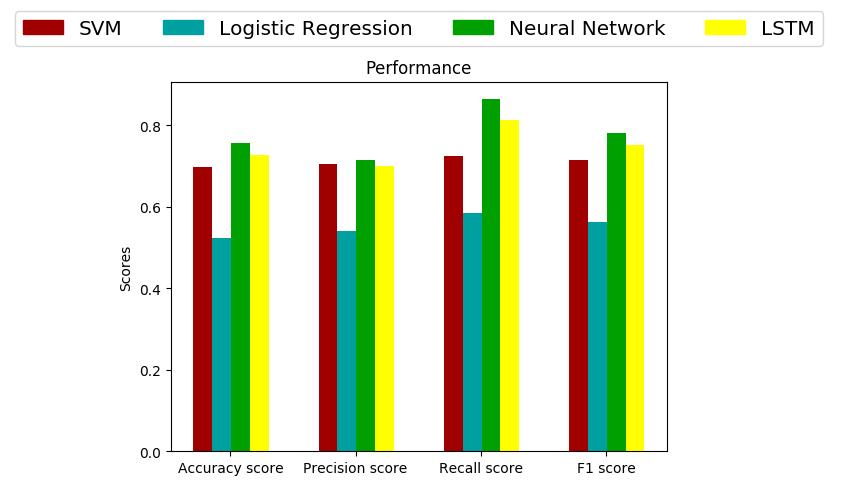
LSTM

Accuracy = 0.728140703517588

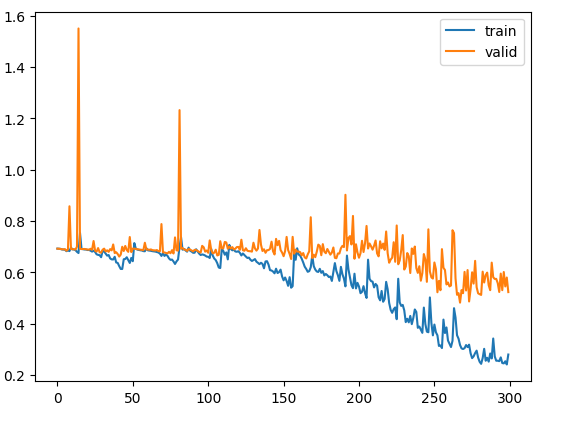
Precision = 0.6999573196756296

Recall = 0.8122833085685983

F1 Score = 0.7519486474094451



In training any neural network very important topic is its convergence of loss function. Convergence helps us to know whether the model is under-fit or over-fit. The below figure shows the learning curve of the training process of LSTM model. We can observe that loss for training and validation set is decreasing as epochs are increasing. From the graph we can say our model is neither over-fitted nor under-fitted.



# Future Enhancements

The main reason for LSTM and Neural Network not able to accuracy greater than 75% is EEG signals are Non – Stationary signals means there frequency keep on changing.

To further improving the model we need to use signal processing techniques and take it to spatial domain (Fourier Transform) or Time – Frequency Domain to better capture those features.

References –

* R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, G. Pfurtscheller. Brain-computer communication: motivation, aim, and impact of exploring a virtual apartment. IEEE Transactions on Neural Systems and Rehabilitation Engineering 15, 473–482, 2007.
* M. Fatourechi, A. Bashashati, R. K. Ward, G. E. Birch. EMG and EOG artifacts in brain computer interface systems: a survey. Clinical Neurophysiology 118, 480–494, 2007.