IMPLEMENTATION OF EXTREME LEARNING MACHINE ON MNIST DATASET

A SUMMER INTERNSHIP PROJECT REPORT

Submitted by

JASLEEN KAUR (03801022018)

in partial fulfillment of Summer Internship for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

Under the supervision of

Dr. Nidhi Goel

Associate Professor and HOD ECE

INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

KASHMERE GATE – 110006

CONTENTS

ACKNOWLEDGEMENT

The project title Implementation of ELM on MNIST dataset would not have been possible without the kind support of project mentor Dr. Nidhi Goel. I would like to extend my sincere thanks to her.

I would like to express my sincere gratitude towards Palak Handa for their kind cooperation and encouragement which helped me in completion of our project.

I’m highly indebted to Palak Handa for their guidance and supervisions as well as for providing necessary equipment and information regarding the project and also for their support in completing this project.

I would like to express my gratitude towards my parents and family members for always believing in me.

My thanks and appreciation also goes to Aarushi Goyal (Teammate) in completing the project and people who have willingly helped out with their abilities.

Jasleen Kaur
03801022018
Btech ECE 2nd year
Summer research intern

ABSTRACT

Extreme learning machine is an emerging neural network architecture that offers fast learning and generalization for multiple tasks. Compared with the other conventional network algorithm it has the advantage of over fitting problems and slow training speed. In this work to detect images and to validate the model, MNIST database that is collected from kaggle is used which is widely employed in the neural network architectures and for image classification in discetre machine learning models. Feature extraction using convolutional neural networks is performed on a sample image.

Chapter 1: INTRODUCTION

Around 12 million people are expected to suffer from epilepsy in India. Some basic, clinical, and translational researches have been accounted for epilepsy and one such is the detection of epileptic and healthy data through EEG signals. EEG is cost-effective and used by clinicians to examine abnormal activities in the brain by recording the electrical signals from the scalp and intracranially. People with epilepsy living in low-income countries do not get the treatment they need and hence automatic detection of epilepsy from EEG signals can be of great clinical significance.

This summer research project purely focuses on the classification of different types of signals in the form of images, matrices etc. The classification of EEG signals has been a challenging yet an important area of research. This is due to the abnormal and complex nature of the signals, hence to distinguish the signals, a neural network technique named extreme learning machine is used.

Deep convolutional networks are becoming the default option for difficult tasks on large datasets, such as image and speech recognition. Extreme learning machines are feedforward neural networks that can be extremely easy to implement and offer decent results, considering the speed and simplicity of this algorithm compared to more complex solutions with a very rapid training time (approximately 10 minutes).

The MNIST dataset that was collected from kaggle is used to implement the extreme learning machine algorithm for basic understanding. The dataset consists of a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning with 60,000 training images and 10,000 testing images.

For the basic understanding of feature extraction a cat image is taken and features are extracted then different layers with 1-D convolution neural network were mapped onto the image.

The universal approximation ability of the ELM algorithm with additive or RBF activation function has been proved. The ELM algorithm has been successfully applied to many real-world problems, such as classification and regression problems.

Chapter 2: WORK DONE ON DATASET

Datasets given:

I. Epileptic dataset
1. Hauz Khas
2. Max Saket (Set C and D)
3. University of Bonn (Set C and D)

II. Healthy dataset
1. University of Bonn (Set A and B)
2. Max Saket (Set A and B)

1) Resampling performed on the interictal data from Hauz Khas to normalize all the datasets to a common sampling rate that is 173.6 Hz. Hence the data was downsampled from 200 Hz to 173.6 Hz.
2) After resampling the data, all the healthy datasets were concatenated together and the epileptic datasets together in two ways:
3) In the first merged data, there are 400 EEG signals each with 4097 samples in a healthy datasheet.
4) In epileptic datasheet, there are 450 EEG signals out of which 400 signals have 4097 samples each and later 50 signals have 889 (length of hauz khas) samples which were resampled from 1024 samples.
5) In the second merged data, there are 10 EEG signals with 163880 samples each in a healthy datasheet. In epileptic datasheet, there are 10 EEG signals each with 168325 samples.
6) Class labels added to the merged sheet 2 and the labels are Healthy and Unhealthy to classify both the classes distinctly.
7) Total time to access the data is the same for both the sheets, that is , 2.62 hours for healthy data and 2.69 hours for the unhealthy data.

Table 1: Data Explanation before and after merging

Healthy Dataset Explanation					
Dataset		**Columns**	**Sampling frequency**	**Time of each column**	**No. of samples in each column**
Set A	Bonn	100	176.3Hz	23.6s	4097
Set B	Bonn	100	176.3Hz	23.6s	4097
Max Saket	Zanedo	200	176.3Hz	23.6s	4097
Case 1: Healthy	Merged data	400	176.3Hz	23.6*400 = 9440s or 2.62 hours	4097
Case 2: Healthy	Merged data	10 EEGs	176.3Hz	For 1 EEG: 23.6*40 = 944s For 10 EEGs: 9440s or 2.62 hours	163880

Unhealthy Dataset Explanation					
Dataset		Columns	Sampling frequency	Time of each column	No. of samples in each column
Set C	Bonn	100	176.3Hz	23.6s	4097
Set D	Bonn	100	176.3Hz	23.6s	4097
Max Saket	Zanedo	200	176.3Hz	23.6s	4097
Hauz Khas (Resampled)	Interictal	50	Down sampled from 200Hz to 176.3Hz	5.12s	Resampled from 1024 to 889
Case 1: Unhealthy	Merged data	450	176.3Hz	23.6*400 + 5.12*50 = 9696s or 2.69 hours	4097 in first 400 and 889 in next 50 columns
Case 2: Unhealthy	Merged data	10 EEGs	176.3Hz	For 1 EEG: 23.6*40 + 5.12*5 = 969.6s For 10 EEGs: 9696s or 2.69 hours	168325

Chapter 3: METHODOLOGY

1. Data Pre-processing:

   The MNIST dataset contains a series of monochrome images 28x28 of handwritten digits, on each row of the dataset stored as a vector with 784 values, each representing a pixel value, the training data has an additional column containing the label associated with each image.

2. Each row has 785 columns, with the first being the label and the rest of them representing the pixel values (28x28) of the image.
3. Next is to separate the labels from the pixel values.

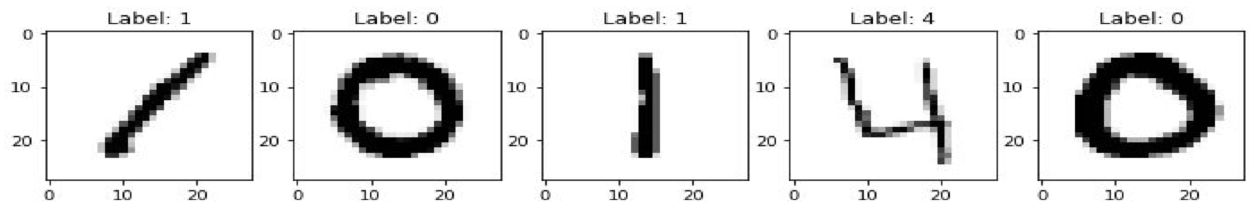

Fig 1: Plot of first 5 images from the dataset to better visualize the data

4. Since this is a multiclass classification problem, so One Hot Encode the labels. This simply means that vectors will be used to represent each class, instead of the label value. Each vector contains the value 1 at the index corresponding to the class it represents, with the rest of the values set to 0.
5. The next step is to split the data into training and testing parts. Training part consists of 90% of data and the testing part consists of 10% of data.
6. The ELM algorithm is similar to other neural networks with 3 key differences:
   - The number of hidden units is usually larger than in other neural networks that are trained using backpropagation.
   - The weights from input to hidden layer are randomly generated, usually using values from a continuous uniform distribution.
   - The output neurons are linear rather than sigmoidal, this means use least square errors regression to solve the output weights.
7. To compute the hidden layer to output weights. This is done in the following way:
   - Compute the dot product between the input and input-to-hidden layer weights,
   - and apply some activation function.
8. To compute output weights, we try to minimize the least square error between the predicted labels and the training labels. The solution to this is:

   $$\widehat{\beta} = (X^T \, . \, X)^{-1} \; X^T \; \mathbf{y}$$

   Where X is the input to the hidden layer matrix computed using the function from the previous step, and y is the training labels.

Feature extraction and mapping with 1-D convolution neural network:

To extract a feature map, the convolutional neural network (CNN) can be used. The convolutional neural network is an architecture of neural networks to extract adequate features from the datasets especially from image datasets. So, along with the basic layers which are input, code and output layers, there are some additional layers to map features.

1. Input layer:- The first layer which are image inputs only.
2. Convolutional layer :- This layer is used for learning features from image data.
3. Padding layer:- This layer will increase depth but will not change dimensionality.
4. Max Pooling layer:- This layer has no parameters(feature parameter) which means it will not learn anything but decrease spatial dimensionality of feature.
5. Code layer:- It is a compressed representation of the image dataset.
6. Conv2dTranspose layer:- Learns feature from input layer's output.
7. Upsampling layer:- This works as an inverse function of max pooling layer. It increases spatial dimensionality.
8. Output layer:- Decodes images from code layer by applying convolutional layer before output layer.

A convolutional layer with 1 filter and 3x3 kernel is added.
Then the second convolutional layer with 15x15 dimensions and 1 filter layer is added.
Max pooling with reduced dimensions and pooling size of (5,5) filter layer is added with one more convolutional layer along with the activation layer and 3rd filter layer.
After the code layer, a convolutional layer with max- pooling with pool size 3x3 and activation function is added.
One more convolution layer along with activation function and 1 filter layer is added.
PIPELINE:

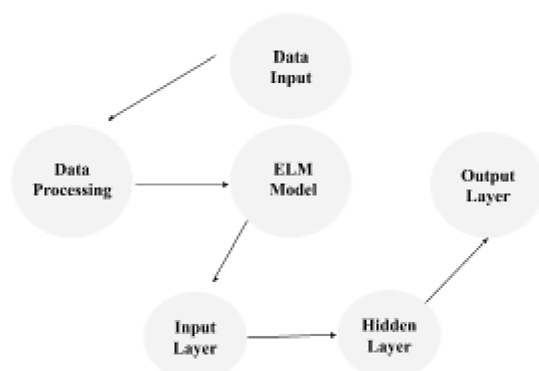

Fig 2: Pipeline Design

Chapter 4: RESULTS AND DISCUSSIONS

Resampling:

1. Hauz khas data down sampled to 173.6Hz.
2. Conclusion after studying about sampling : Down sampling may violate the Nyquist rule, as the new sample rate may be less than twice the signal's bandwidth, producing aliasing, so the data should go through a low-pass filter first.
   a. Downsampling or decimation helps in reducing data size, compression or image reduction.
   b. Upsampling or interpolation increases resolution, improves anti-aliasing filter performance and reduces noise.
3. No effect on the information after sampling as the clinical range of EEG is from 0-30 Hz.

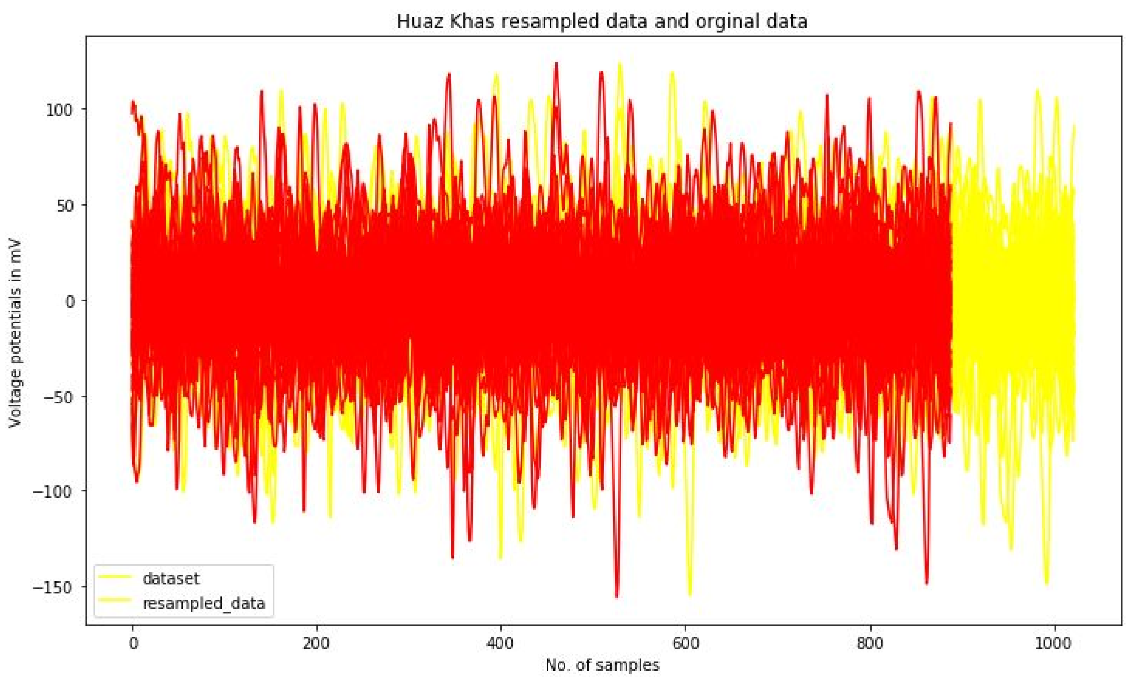

Fig 3: Hauz Khas resampling data and original data

Extreme Learning Machine:

Any activation function can be used, ReLU and sigmoid are used in this project.

The accuracy of the model came out to be 93.4%.

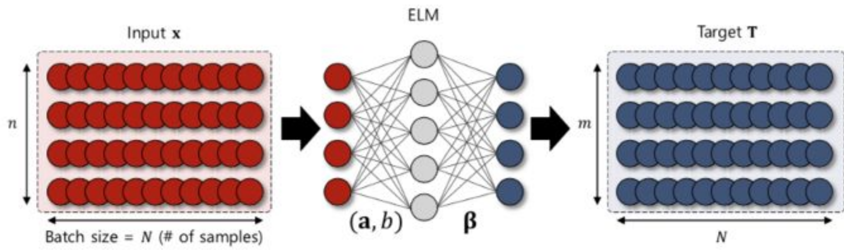

Fig 4: Architecture of extreme learning machine

Feature mapping:

Extracted feature and mapped different layers with 1-D convolution neural network.

1. For data, an image of a cat in png format is used.
2. After loading the image through OpenCV, different layers were mapped.

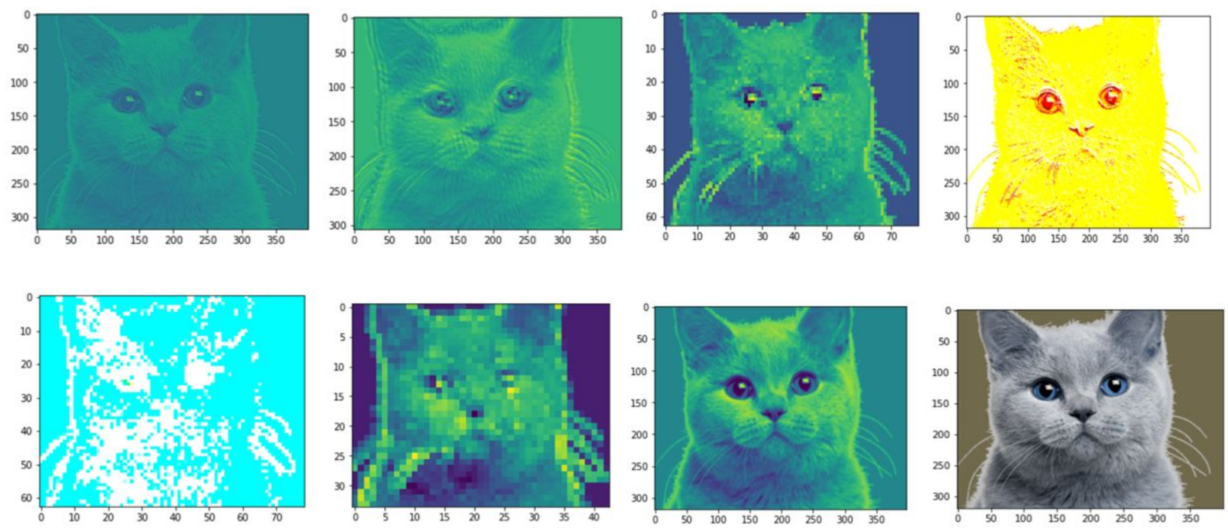

Fig 5: Feature extraction after applying different layers

Chapter 5: CONCLUSION AND FUTURE PROSPECTIVE

- At the settling this summer research work, I learnt how to make a machine learning model from the raw data to the final result.
- I learnt how to manage different datasets all together by performing resampling of the Hauz Khas dataset to concatenate it with other datasets.
- I discovered the complexity of the EEG signals and many techniques to simplify them. Grasped a few nonlinear operators and techniques which helps to reduce complexity of different types of signals.
- Worked on MNIST image dataset which introduced me to the concept of OpenCV.
- Worked on an Iris dataset which cleared my concepts of cross validation and splitting of data.
- Get in touch with neural networks by performing ELM on MNIST dataset.
- Learn how to find suitable research papers and how to do research work.

With the fullness of time, the ELM can also be implemented to EEG signals by converting signal sets to image databases with proper research work.

BIBLIOGRAPHY

1. https://www.hindawi.com/journals/cin/2016/3049632/
2. https://sci-hub.tw/https://ieeexplore.ieee.org/abstract/document/7966279
3. https://www.diva-portal.org/smash/get/diva2:1130092/FULLTEXT01.pdf
4. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0134254
5. https://pubmed.ncbi.nlm.nih.gov/29747439/