**EXPERIMENT NO.1**

**Aim-** To perform Printed Alphabet Recognition on image dataset

**Title**- Printed Alphabet Recognition on MATLAB

**Task**- Classification of all 5 alphabet (A, B, C, D, E)

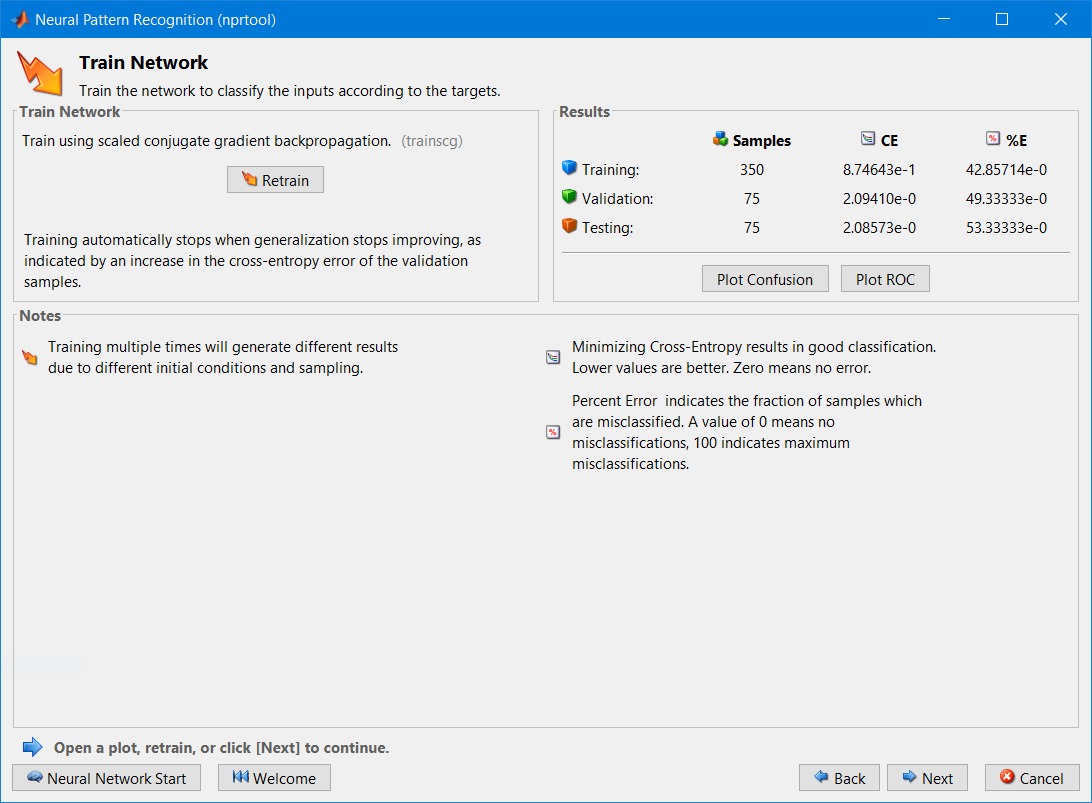
**Dataset used -**

Given below are 2 images for each alphabet from our image dataset.

C:\Users\User\Desktop\Alphabet Database\A\12.png C:\Users\User\Desktop\Alphabet Database\A\10.png C:\Users\User\Desktop\Alphabet Database\B\10.png C:\Users\User\Desktop\Alphabet Database\B\3.png C:\Users\User\Desktop\Alphabet Database\C\5.png C:\Users\User\Desktop\Alphabet Database\C\5.png C:\Users\User\Desktop\Alphabet Database\D\6.png C:\Users\User\Desktop\Alphabet Database\D\14.png C:\Users\User\Desktop\Alphabet Database\E\11.png C:\Users\User\Desktop\Alphabet Database\E\4.png

I have prepared our dataset with 100 image of each alphabet, so a total of 500 image samples for our dataset.

Out of this we have used 15 % for test set, 15% for validation set and 70% for training data. As you can see in our nprtool we have 350 images for Training set, 75 images for validation set and 75 images for test set.



**Neural network configuration** -

We have 24 input nodes for 6 x 4 =24 pixels = 24 features

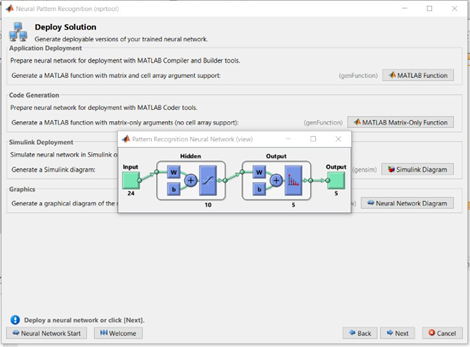
We have 5 output nodes as we have 5 different values representing 5 different alphabets

We have 10 nodes in each of the hidden layers (2)

Each layer (input, hidden, output) is denoted by weights.

As this is an example of supervised learning, we know the correct output is known, and the output is result our Deep learning NN classification.

The ‘correct output’ – ‘output’ = error, this error is passed back to the hidden layer to adjust weights of the nodes



**Code Explanation-**

Firstly, all the file locations were put in the code. Then a function cellmat() was used which takes 4 integer values and 1 scaler value. Here the scalar value is 1 and integer values have been included/given. Then for each alphabets the size of the pictures was resized to (32,42). This was done so as to get 24 features for each alphabets. The rest of the code is for the grey scale image conversion. In the end the mean of all the results in accordance to the training data was taken. In addition to that a validation set was created which was a completely unseen data and the model was implemented on that data.

The accuracy chart for training, validation and test is given below:

Below is the curve of the AUC-ROC which is a performance measurement for classification problem at various threshold. It tells us how much he model is capable of distinguishing the classes. It is a graph between False Positive rate (the predicted value is true/positive but the actual value is negative) and True Positive Rate (The predicted and the actual value is true).

**Code to run on MATLAB Software (Classification using Neural Network Back Propagation) :**

clc;

clear all;

close all;

A\_folder = dir('E:\Notes\SEM 7\PRML\Alphabet Database\A\\*.png');

B\_folder = dir('E:\Notes\SEM 7\PRML\Alphabet Database\B\\*.png');

C\_folder = dir('E:\Notes\SEM 7\PRML\Alphabet Database\C\\*.png');

D\_folder = dir('E:\Notes\SEM 7\PRML\Alphabet Database\D\\*.png');

E\_folder = dir('E:\Notes\SEM 7\PRML\Alphabet Database\E\\*.png');

A\_result=cellmat(1,length(A\_folder),32,42);

B\_result=cellmat(1,length(B\_folder),32,42);

C\_result=cellmat(1,length(C\_folder),32,42);

D\_result=cellmat(1,length(D\_folder),32,42);

E\_result=cellmat(1,length(E\_folder),32,42);

m=100;

for i=1:m

A\_result{1,i}=imresize(imread(strcat('E:\Notes\SEM 7\PRML\Alphabet Database\A\',A\_folder(i).name)),[32 42]);

A\_DB=A\_result{1,i};

for j=1:32

for k=1:42

if A\_DB(j,k)>=157

A\_DB(j,k)=1;

else

A\_DB(j,k)=0;

end

end

end

A\_result{1,i}=A\_DB;

A\_result{1,i}=reshape(A\_result{1,i},[],24);

B\_result{1,i}=imresize(imread(strcat('E:\Notes\SEM 7\PRML\Alphabet Database\B\',B\_folder(i).name)),[32 42]);

B\_DB=B\_result{1,i};

for j=1:32

for k=1:42

if B\_DB(j,k)>=157

B\_DB(j,k)=1;

else

B\_DB(j,k)=0;

end

end

end

B\_result{1,i}=B\_DB;

B\_result{1,i}=reshape(B\_result{1,i},[],24);

C\_result{1,i}=imresize(imread(strcat('E:\Notes\SEM 7\PRML\Alphabet Database\C\',C\_folder(i).name)),[32 42]);

C\_DB=C\_result{1,i};

for j=1:32

for k=1:42

if C\_DB(j,k)>=157

C\_DB(j,k)=1;

else

C\_DB(j,k)=0;

end

end

end

C\_result{1,i}=C\_DB;

C\_result{1,i}=reshape(C\_result{1,i},[],24);

D\_result{1,i}=imresize(imread(strcat('E:\Notes\SEM 7\PRML\Alphabet Database\D\',D\_folder(i).name)),[32 42]);

D\_DB=D\_result{1,i};

for j=1:32

for k=1:42

if D\_DB(j,k)>=157

D\_DB(j,k)=1;

else

D\_DB(j,k)=0;

end

end

end

D\_result{1,i}=D\_DB;

D\_result{1,i}=reshape(D\_result{1,i},[],24);

E\_result{1,i}=imresize(imread(strcat('E:\Notes\SEM 7\PRML\Alphabet Database\E\',E\_folder(i).name)),[32 42]);

E\_DB=E\_result{1,i};

for j=1:32

for k=1:42

if E\_DB(j,k)>=157

E\_DB(j,k)=1;

else

E\_DB(j,k)=0;

end

end

end

E\_result{1,i}=E\_DB;

E\_result{1,i}=reshape(E\_result{1,i},[],24);

A(i,:)=mean(A\_result{1,i});

B(i,:)=mean(B\_result{1,i});

C(i,:)=mean(C\_result{1,i});

D(i,:)=mean(D\_result{1,i});

E(i,:)=mean(E\_result{1,i});

end

X{1,1}=A;

X{1,2}=B;

X{1,3}=C;

X{1,4}=D;

X{1,5}=E;

X=cell2mat(reshape(X,[5 1]));

Y=zeros(500,5);

Y(1:100,1)=1;

Y(101:200,2)=1;

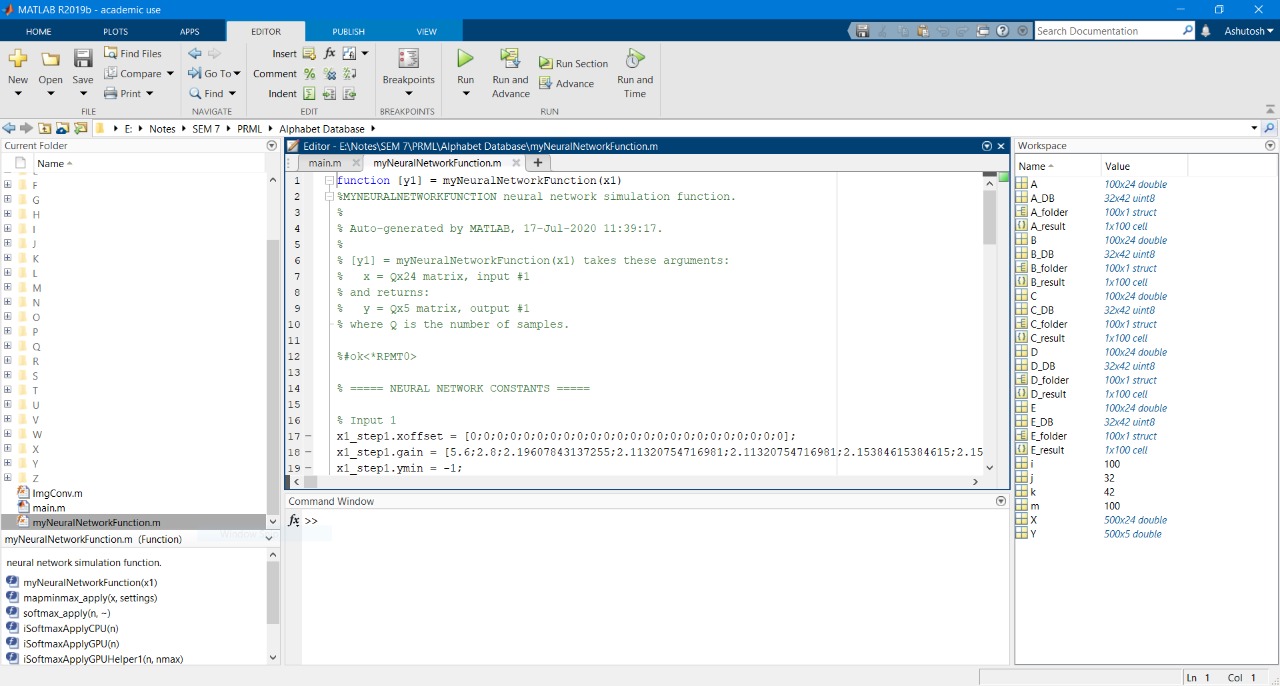
Y(201:300,3)=1;

Y(301:400,4)=1;

Y(401:500,5)=1;

**Next step after execution of the code** - Run 'nnstart' in command window

We observe this **MATLAB generated function** with back-propagation, which is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch. is an important mathematical tool for improving the accuracy.



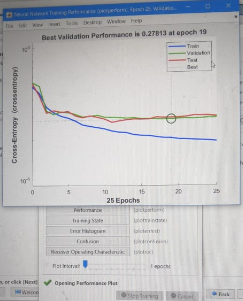
**Output observed** –

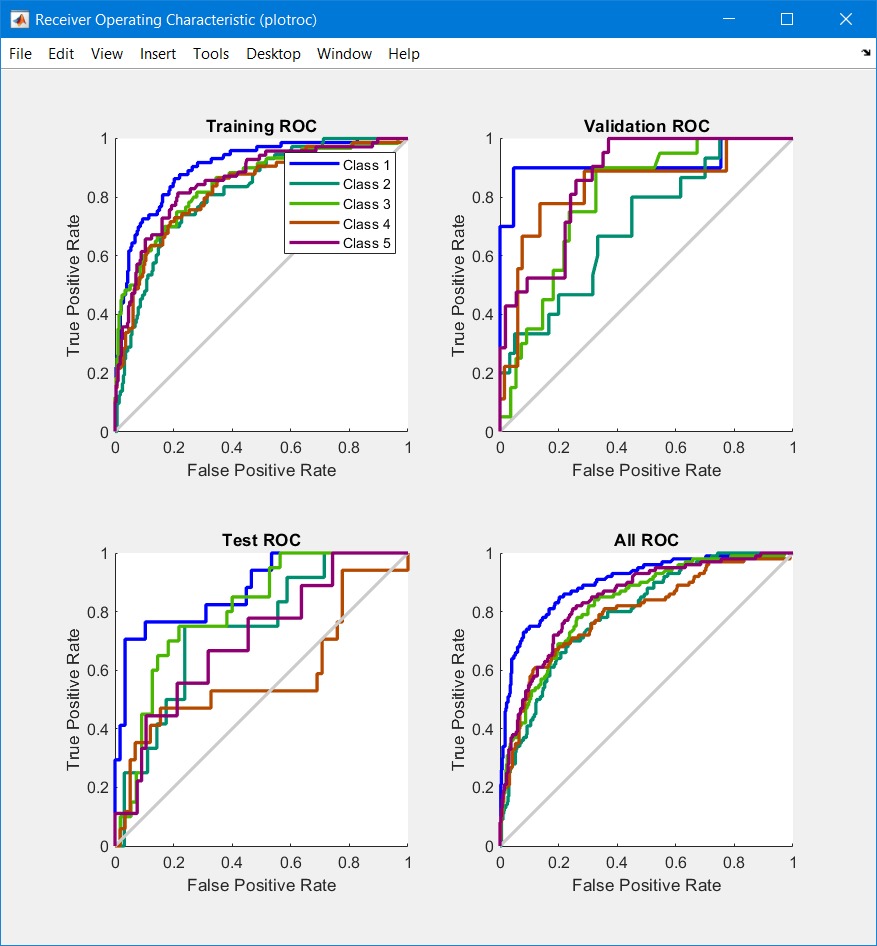
We obtained the graph plotted between Cross- Entropy and the 25 Epochs.

Epochs basically indicates the number of passes of the entire training dataset the machine learning algorithm has completed. An epoch refers to one cycle through the full training dataset, we used 25 epochs as too many of them can lead to overfitting.

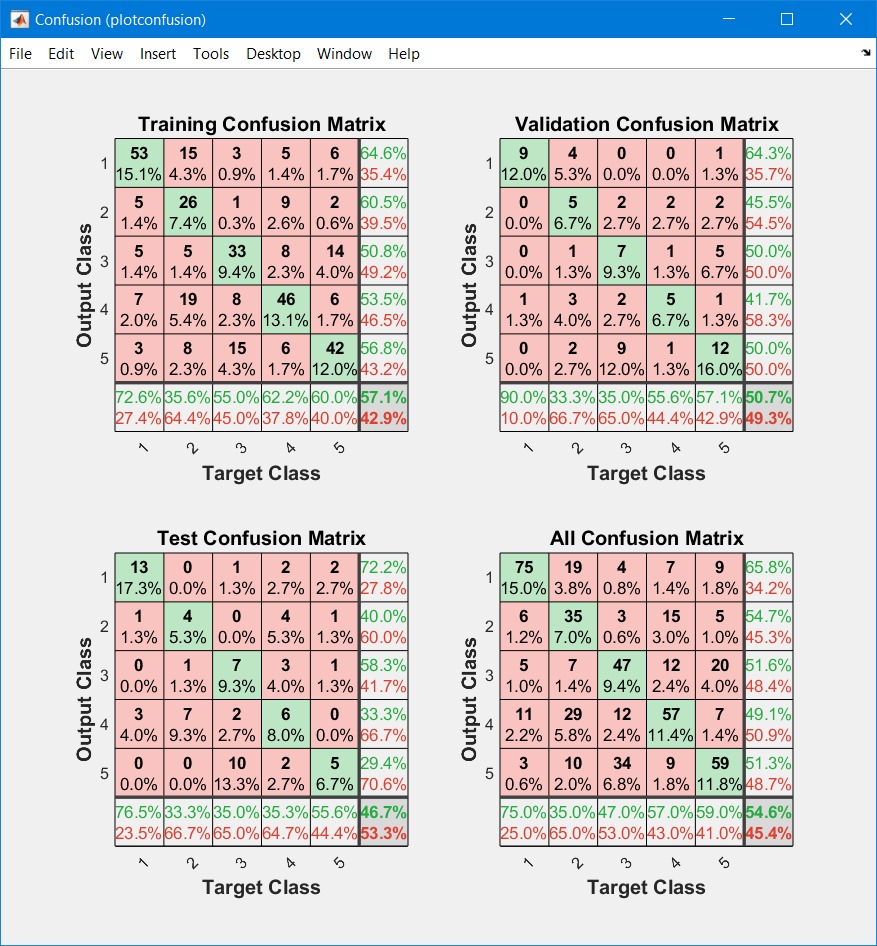
Out of 25 cycles the best performance is obtained for the dataset is observed at epoch 19.

We obtained the plot for performance of training set, validation set and test set for Class 1 to Class 5 of alphabets.





Given below is the plot for the 24 feature matrix i.e. 24 boxes considering the Output class and Target class.



**Conclusion** –

I was able to successfully run the code for Printed Alphabet Recognition on MATLAB with Classification of all 5 alphabet based on the concept of Neural Network method back propagation. First we collected the image dataset of 500 images. We used 15 % for test set, 15% for validation set and 70% for training data Resized the images into 24 pixels Black-white on Matlab with command imresize (imread) , we observed that the shaded region with alphabet was considered as 0 rest as 1 that is how binary function for 0 and 1. We tested the deep learning neural network with selecting input nodes, output nodes, weight matrices, who data are passed into the network and how to calculate. Overall we got a good amount of knowledge on what machine learning really is and how beneficial it is for Pattern recognition.