Optical Character Recognition using Neural Network

Parul Gupta
IIITD, India
Email: parul1370@iiitd.ac.in

Veronica Sharma
IIITD, India
Email: veronical428@iiitd.ac.in

I. Abstract

Optical Character Recognition is a very challenging task in pattern recognition. It is a rapidly evolving subject as it can contribute to manage and organize scanned documents. But at the same time it is very complex and it's complexity is attributed to variations in hand writings of people. In this work, we implement a multilayer neural network based optical character recognition for numerical digits 0-9. We make use of backward propagation algorithm to train our neural network. We found that the classifier was able to predict digits with the accuracy of approx. 80%. We also verified the training of our neural network by monitoring the training error at each updation of weights and identified that the error rate was continuously decreasing in each iteration, hence increasing our confidence in the model. We also found that the most accurately classified digits were 0,1,2,4,7 and 9 with 100% accuracy.

II. Introduction

Optical Character Recognition, usually abbreviated as OCR is the translation of text found in an image into machine- encoded text. For this reason, OCR is an important step towards image processing. It has been widely used in different applications like extracting text from scanned documents. With the use of OCR, we can even manipulate the text present in the scanned documents or we can search for a phrase or a word in the text. Originally, OCR required a text to be in a standard format to be recognized but modern OCR are intelligent enough and we can use neural networks efficiently to categorize the characters. Neural Networks also compensates for the irregularities of printed ink on the paper. It also has an added advantage of recognizing arbitrary font styles.

In this work, we have implemented a multilayer neural network for optical character recognition of numerical digits 0-9. We have used backward propagation algorithm to update weights while training our neural network. Our neural network contains 3 layers: Input layer, Hidden Layer and Output Layer.

III. Data Collection and Method

We have used 7 images each for 10 digits for the training purpose. We have used 3 images each for 10 digits for testing. The training and test data was collected from Google and each was of dimension 16 X 16. We can choose any dimension for the images, we chose 16 X 16 as we found this size to be computationally feasible. Since the images were of good quality, we did not perform any preprocessing techniques except for converting each image into a gray scale image.

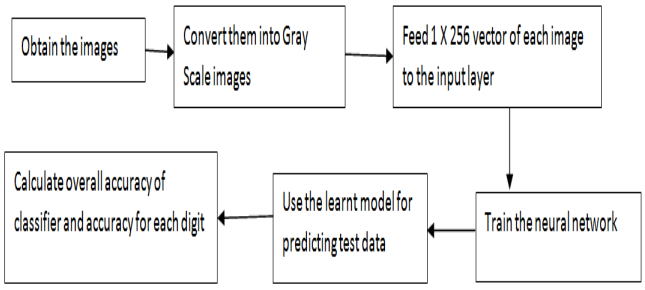


Fig 1. Process Steps

Our Neural Network contained 256(16 X 16) neurons in the Input Layer, 10 neurons were in the

output layer corresponding to each class(i.e, 0-9) and 133((256+10)/2) neurons in the hidden layer. In the training phase, we passed each image as a 1 X 256 vector to the input layer.

A. *Assumptions*

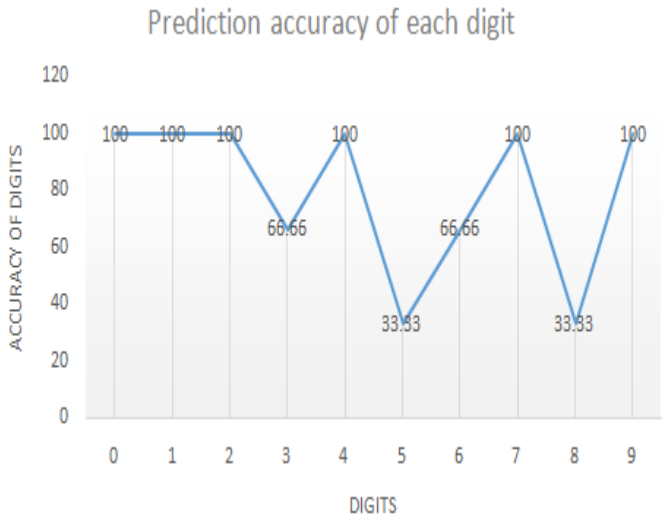
We have assumed acceptable training error to be 0.05. We have used sigmoid function as an activation function.

IV. Results

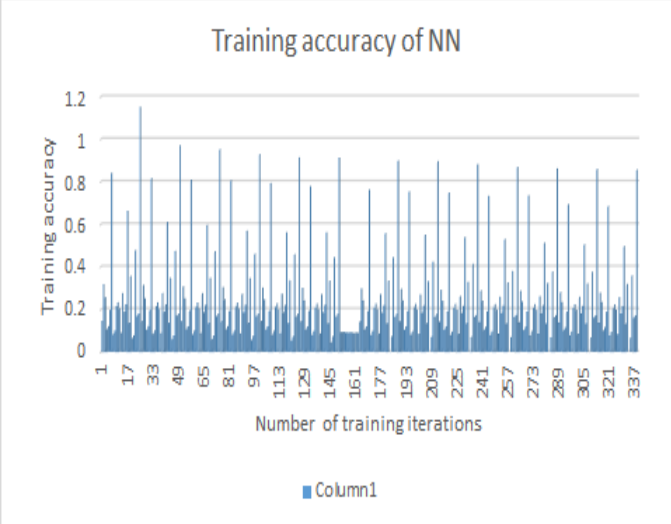
We found that the overall accuracy of the classifier was approx. 80% with digits 0,1,2,4,7 and 9 achieving highest accuracy of 100%. We also found that training error of the neural network in each iteration in the training phase was monotonically decreasing, thus increasing our confidence in the classifier. The following table and graph show the accuracy of classifier for each digit.

Digit	Accuracy Per Digit
0	100
1	100
2	100
3	66.66
4	100
5	33.33
6	66.66
7	100
8	33.33
9	100

Table 1. Accuracy Percentage of Each Digit



The following graph indicates the training error rate is decreasing with each learning iteration of the neural network.



V. Limitations

We have taken refined images and have not performed any preprocessing technique which might be required in other images. We have also not performed feature extraction here, the reason for doing this was our feature set(256 features) was small enough to apply feature extraction and selection. But for images of higher dimensions we need to perform these steps otherwise we will require a huge number of neurons in our neural network and it will lead to slow computation.

VI. Future Work

We can generalize our model for any dimensions of images. We can also explore different neural network topologies and algorithms. We can also perform preprocessing on the data to improve overall accuracy of our classifier.

VII. Conclusion

We were able to train our model successfully and with high confidence. The learnt model predicted the test data with decent accuracy of 83.33%. We also calculated the accuracy rate of every digit and found that digits 0,1,2,4,7 and 9 had 100% accuracy. Although our approach was very naive we were able to produce a neural network with decent

average accuracy. We can improve our results and make it more general approach by overcoming our limitations.

VIII. References

We referred the following papers [2], [1], [3]

References

[1] Eldem, A., and Başçiftçi, F. Dynamic calculation and characterization of threshold value in optical character recognition.

[2] Handley, T. Optical character recognition with neural networks.

[3] Varshney, S., Chaurasiya, R., and Tayal, Y. Optical character recognition using neural network.