OCR - Optical Character Recognition

Gonçalo Duarte - 2013155376
Tadeu Ferreira - 2013151074

October 13, 2017

Abstract

The aim of the work is to investigate different neural network architecture by developing an Optical Character Recognition. Our object of study was recognition of arabic numbers since nowadays many devices of daily use already have the capability to recognize characters. Two neural network models were developed: associative memory plus the classifier and only the classifier. In addition, 3 activation functions were tested namely hardlim, linear and sigmoidal. The model with more consistent results was Associative Model plus the classifier, nevertheless only the classifier had good results.

1 Introduction

Character recognition, a digitization concept, is an important research area in the field of image processing, and pattern recognition. Optical character recognition is a method of digitizing printed texts so that they can be searched electronically, stored compactly and used in machine processes such as text-to-speech, and machine translation. This work describes the techniques for converting a type or handwritten document into machine readable form using Neural Networks.

2 Methods

2.1 Data Set

The data set is manually created by the user with the help of the program *mpaper.m.* The output of the program is a matrix where each column is a digit defined as a 16x16 matrix and each line is a different digit. The data set is characterized by a 256x800 matrix splitted in a 256x680 for train and 256x120 for validation (ratio 85-15%), and a 256x50 matrix for test. In addition it was built target vectors of size 10x800 and 10x50 respectively. The target vector is the desired output for each of the inputs.

2.2 Neural Network Architecture

In this work it was used two different networks: both associative memory plus classifier and only a classifier. The associative memory work as a filter, providing a "better" output, which becomes the input of the classifier. In both architectures the activation functions hardlim, purelin and logsig were test. In addition, the learning rate was set to 0.5, the number of epochs was 1000 and the goal was 1e-06. All of this architectures were made using Matlab and Neural Networks Toolbox.

3 Results

In order to test the quality of our classifiers, it was created the function *performance.m* that compares the output of simulation of each neural network with the train target and also, the test data with the test target. The results for both architectures are presented below.

AM+Classifier	Hardlim	Purelin	Logsig
Train Data (%)	93.75	97.38	93.00
Test Data (%)	66.00	90.00	80.00

Table 1: Results of performance of AM+Classifier

AM+Classifier	Hardlim	Purelin	Logsig
Train Data (%)	95.50	88.63	81.75
Test Data (%)	76.00	88.00	86.00

Table 2: Results of performance of Only Classifier

Moreover, it was decided to plot the ROC curves for Test data. ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution.

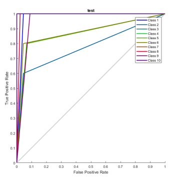

Figure 1: AM + Classifier - Hardlim

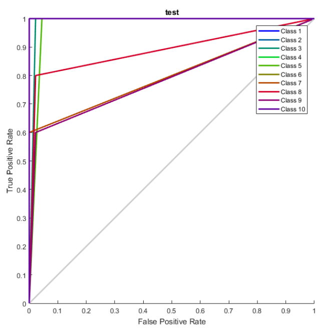

Figure 2: Classifier - Hardlim

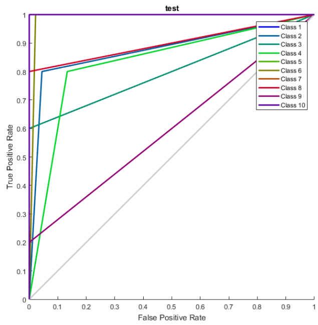

Figure 3: AM + Classifier - Logsig

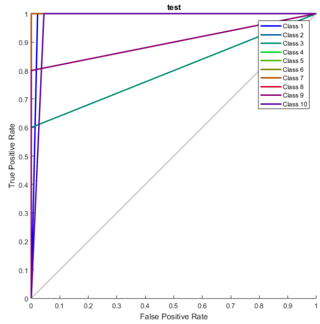

Figure 4: Classifier - Logsig

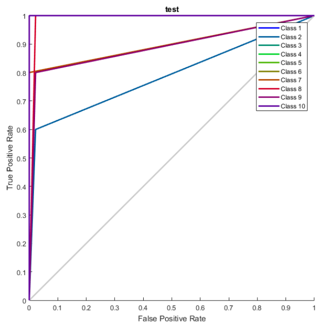

Figure 5: AM + Classifier - Purelin

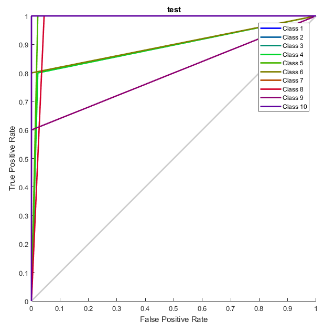

Figure 6: Classifier - Purelin

4 Discussion

4.1 Data set

Having a good data set is major step for having good results. In other words it was created a big data set with size 800 instead of 500 that was advised. Moreover, giving drawing the numbers in different ways gives a the neural network the ability to detected different handwriting. Nevertheless different people drawing numbers would probably present worse results. The solution is having a very big sample of numbers draw by different people.

4.2 Neural network architecture

The architecture with better results was the one with only classifier. Although the architecture with Associative Memory plus the Classifier had better results in Train data, this only transmits if the classifier has overfit reflecting in lower accuracy results in the Test Data. It is believed that the reason that lead to better results in the architecture with only classifier is the difference between the normal human handwriting and "filter" of Arial type letter. That for the architecture with Associative Memory led to worse results. The best activation function was the Purelin. In both architectures the performance was superior regardless of the change of other parameters.

4.3 Results

The architecture of Associative Memory plus the classifier with purelin as activation presented the most consisted results (accuracy of 88%), that is changing the number of epochs the accuracy remained the same. Although AM with classifier and purelin had higher accuracy (90%) we believed that is an outlier since the difference of accuracy to other activation functions is big and that the number of epochs would change considerably the accuracy. The classification system is robust with a good generalization capacity since we reached a percentage of 76% when a colleague draw 50 digits.

5 Conclusion

In conclusion of this study we think that the results were good in general. Although we had good results with Associative Memory plus the Classifier it is important to understand more neural networks architectures. Neural Networks is a big world that needs to be explored. In order to reach for better we describe some important points: - Explore more Neural Architectures and the change of all parameters; - Have a bigger data with a big variability of handwritings. A large and diverse data set lead to better perfomance results; - Increase the type of fonts of the Associative Memory. It is possible that creating a big data base of fonts for the AM we lead to better results comparing with the architecture with only a classifier.