Character recognition with artificial neural networks

George Malandrakis

Abstract

During the last years, there has been an increasing need to deal with problems that cannot be solved with the use of mere conventional programming and are to be solved with artificial intelligence instead. Artificial neural networks are the most eminent tools for processing data without conventional algorithms and code, and the research indicates that technically all problems involving pattern recognition can be and will be solved with their use.

A typical problem that can easily be solved with the use of a neural network is character recognition from images. While it has been partly solved with conventional algorithms, those algorithms often take thousands of lines, while a neural network can be trained to do the same within minutes, or even seconds. And unlike neural networks, conventional algorithms are most often unable to successfully recognize handwritten text, while even with computer-written text they rarely exceed 95% accuracy.

This project is demonstrating a matlab-based application that allows a researcher to set and train neural networks of various specifications –including architecture, number of neurons, and training method- for character recognition. Unlike other neural network based applications easily found on the internet, it can be used for to recognize not only isolated characters, but complete text as well. Its' purpose is to provide the base for a practical text recognition application. With the proper editing, it could also provide the base for a general purpose neural network application, allowing a user to set a neural network and train it for any purpose.

Along with developing and demonstrating the application, an attempt to find the most efficient neural network training algorithms in terms of speed and accuracy has been made, with its' experimental results included in the paper.

1. Introduction

The paper begins with a detailed theoretical part concerning neural networks, their history, the problems they are supposed to solve, their fundamental principles, their main architectures, the most used training methods suitable for them, and proceeds afterwards to the description of the application's structure and algorithms. The application is based upon Matlab r2015a, including neural network toolbox 8.3.

2. Overview

2.1. The main algorithms of the application

The application uses two distinct algorithms for discerning the characters in the images and sending them to the neural network for training or recognition.

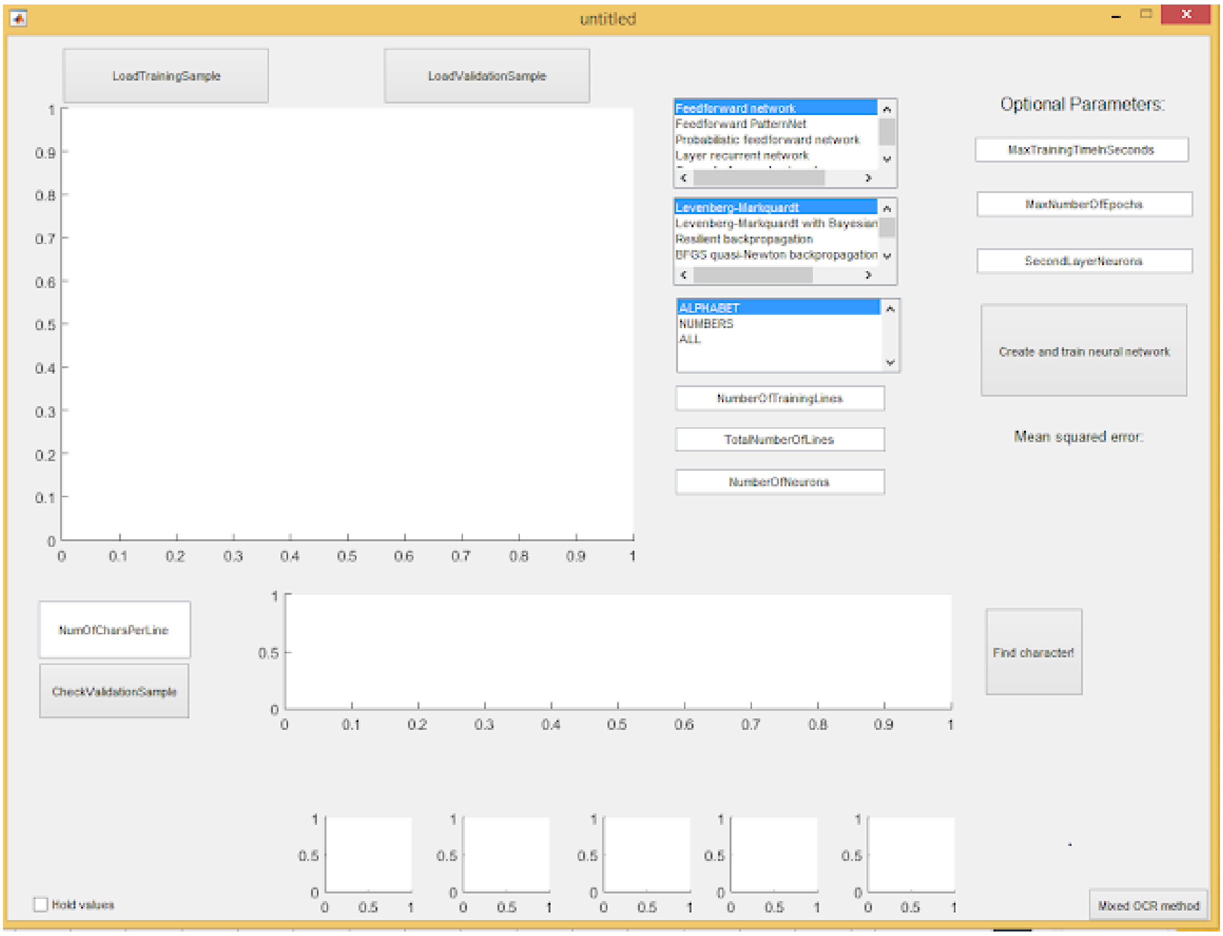
The first one, used in the main interface of the application with the purpose of training the network, uses matlab libraries that treat characters as objects in the image, yet cannot properly sort them from left to right as when we read, nor recognize the space character, which makes the method unsuitable for practical text recognition. On the other hand, and unlike the second method to be described , it has no trouble accurately recognizing the exact position and number of characters.

The second method, used in the "MixedOCR" window, utilizes the conventional matlab OCR with the purpose of extracting the exact position of the characters in the image. It has the advantage of recognizing space characters, and can they can be cropped, even though the conventional OCR could be inaccurate – which makes it unsuitable to use during the training.

After being found, the characters are individually cropped and converted to 5x7 matlab arrays that can be processed by a neural network. The values of the array positions depend on the pixels of the cropped image.

2.2. The matlab application

The main interface of the application is as in the image below

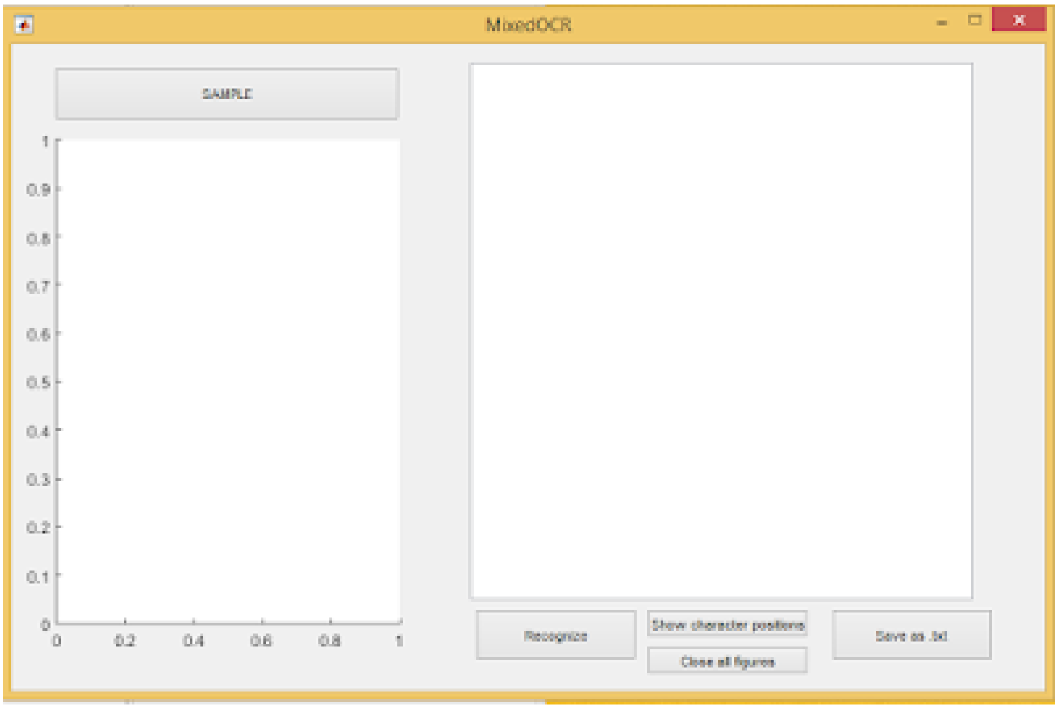


with detailed instructions and descriptions of the components provided in the paper.

The main interface is used to set the technical parameters of the neural network (i.e. number of neurons, training method, etc) and train it for a particular style of writing –depending on the chosen image.

The user can then crop individual characters from an image and use the network for their recognition, while viewing their distinct levels of editing before they are recognized by the neural network.

The MixedOCR interface, which can be used for practical purposes, is as follows:



The user chooses a sample image the text of which can be recognized, and the recognized text appears in the textbox beside the image. One can save it as .txt and view the positions of the characters of the image, as detected by the conventional OCR.

The detailed description of the code and the code itself are included in the paper.

3. Experimental results

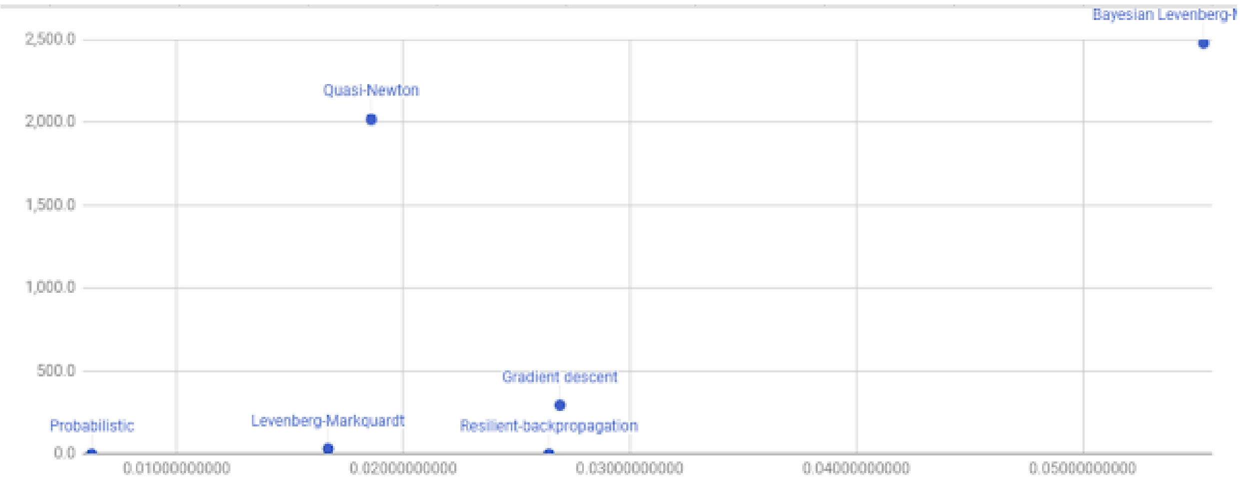
The purpose of the experimental part was to find out the most efficient type of neural network in terms of training speed and error rate. In order to reach a conclusion, we used a particular training set to train neural networks of various specifications –including different architectures, number of neurons, and training methods.

What was taken in to account was the training time in seconds, the mean squared error, the number of training iterations, and the number of mistakes in a given alphabet. The two latter are merely indicative, and largely depend on parameters such as the system's specifications (i.e. speed of the CPU) or the initial weights of the neurons in the network, which are randomly chosen. Still, the results were based upon the training time and mean squared error, since they both can be reliable indicators of the network performance.

There ought to be mentioned that no two networks can have exactly the same training time and mean squared error, since they depend on the CPU and the initial weights of the network respectively, with the latter being, as mentioned above, random. The degree to which the results would be the same for two distinct networks was checked with the retraining of networks of the same specifications, in order to compare the results. One can see that the differences were existent yet negligible, and that the measurements are, therefore, fairly reliable and allow a researcher to reach safe conclusions. This is further elaborated in the paper.

Four predictions were made before the measurements:

The measurements were taken for feed-forward and probabilistic networks using all available training methods. The performance was calculated using equations given and explained in the paper. The performance of an ideal neural network – instantaneously trained and with no errors-, is equal to 1. A network's suitability is judged by how close its' performance is to the ideal one. The general results appear in the following chart:



The X axis represents the mean squared error rates, while the Y axis the training time in seconds. The ideal network would appear on the (0,0) position, and as one may see the probabilistic neural network is the closest to the ideal one. The exact performances are as follows:

Ideal: 1
Probabilistic: 0.966764302
Feed-forward with Levenberg-Markquardt training: 0.0942357778
Feed-forward with resilient-backpropagation training: 0.927209553
Feed-forward with gradient-descent training: 0.0111970987
Feed-forward with quasi-newton training: 0.00164904112
Feed-forward with Bayesian-regularized Levenberg-Markquardt training: 0.00134347435

4. References

[1] Simon Haykin, 2008, Neural networks - A comprehensive foundation

[2] Ι. Βλαχαβάς, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κοκκορας, Η. Σακελλαρίου, 2006, Τεχνητή Νοημοσύνη

[3] Fausett Laurene, 1993, Fundamentals of Neural Networks: Architectures, Algorithms And Applications

[4] B. Krose, P. van der Smagt, 1996, An introduction to Neural Networks

[5] The industrial electronics handbook-Intelligent systems - B.M. Wilamowski, J.D. Irwin

[6] https://en.wikipedia.org

[7] https://www.mathworks.com/