**Paper Title:** Handwritten Character Recognition Using Neural Network

**Author:** Chirag Patel, Palak Patel , Ripal Patel

**Year:** 2011

**Journal Name**: International Journal of Scientific & Engineering Research

**Problem(s) Focused:** Studying behaviors of different Models of Neural Network used in Optical Character Recognition (OCR).

**Algorithms/Approach Used:**

* Multilayer Feed Forward Network
* Back Propagation with biases, a sigmoid layer, and a linear output layer.
* For Preprocessing some basic algorithms for segmentation of characters, normalizing of characters and De-skewing are used.

For a typical OCR, there are two parts to the process: Feature Extractor/Character Extracter and Character Recognition.

PreProcessing (Done for converting a RGB file to binary file):

For performing preprocessing , TraverseList and EdgeDetection Algorithm (graphics-8-way connected Algo) are used. The TraverseList maintains the record whether a pixel has been traversed or not whereas EdgeDetection Algorithm converets the encountered pixel to 0 or 1 depending upon the threshold value.

Normalizing: The obtained matrix size of character after preprocessing is mapped to a general considered matrix size.

Skew Detection: To remove skewness from the character obtained, “Line Fitting” method i.e Linear Regression is used to calculate angle ϴ (by which the character is rotated) to counter rotate the character.

Neural Network Design: Two documents are presented to the model (one being training input and one as testing input). As image size of 10x10 is fed as inupt to the model, hence there are 100 neurons in the input layer.

Due to different models being considered, observations are recorded on the factors such as training time, accuracy etc. to find the effectiveness of the Neural Network.

**Evaluation Parameter:** Number of Hidden Layer, Size of Hidden Layer and Epochs

**Dataset Used/Domain:**

Two types of documents are presented to the input layer of the models (one for training and one for testing).

The dataset consisted of A-Z typed characters of different size and type. Thus the input layer consisted of 100 neurons, and the output layer 26 neurons (one for each character).

**Size of Dataset:**

300,600 and 1000 epochs are run for each of the training models with 1,2,3 and 4 hidden layers (each differeing in configuration in the hidden layer).

Totally 6 models have been used and the accuracy percentage is calculated respectively.

**Performance Evaluation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Epochs** | **Number Of Hidden Layers** | **Configuration**  **(No. Of nodes in Hidden Layer)** | **Accuracy (%)** |
| 1000 | 4 | 78-26-52-104 | 98 |

As it can be seen from the above data (for model 5), an accuracy of 98% was achieved using the meintioned algorithms and approach.

The following observations were made from the above experimentation:

* A small number of nodes in the hidden layer results in lower accuracy.
* A large number of neurons in the hidden layer help in increasing the accuracy; however there is probably some upper limit to this which is dependent on the data being used. Additionally, high neuron counts in the hidden layers increase training time significantly.
* As number of hidden layer increases the accuracy increases initially and then saturates at certain rate probably due to the data used in training.
* Mostly Accuracy is increased by increasing the number of cycles.
* Accuracy could also be increased by increasing the training set.

**Advantages:**

* Useful when there is a need to convert hard copies into soft copies.
* Such models reduce upto 80% of the conversion work.
* Translate images of typewritten text (usually captured by a scanner) into machine-editable text
* To translate pictures of characters into a standard encoding scheme representing them in ASCII or Unicode.

**Drawbacks:**

The model cannot handle major variations in translation, rotation, or scaling of the characters. While a few pre-processing steps can be implemented in order to account for these variances. In general they are difficult to solve completely.

The parameters like skewing, slanting,thickening, cursive handwriting, joint characters affect the accuracy of OCR. If all these parameters are taken care in the preprocessing phase then overall accuracy of the Neural Network would increase.