

## Handwritten libretto recognition using multilayer and cluster neural network

ANURAG LAL

CSE Department, CIT, Rajanadgaon (India)

(Received: April 12, 2010; Accepted: June 04, 2010)

### ABSTRACT

There are different techniques that can be used to recognize handwritten digits and characters. Two techniques discussed in this paper are: Pattern Recognition and Artificial Neural Network. Both techniques are defined and different methods for each technique is also discussed. Bayesian Decision theory, Nearest Neighbor rule, and Linear Classification or Discrimination is types of methods for Pattern Recognition. Shape recognition, Character and Handwritten Digit recognition uses Neural Network to recognize them. Neural Network is used to train and identify written digits. After training and testing, the accuracy rate reached 99%. This accuracy rate is very high.

**Keywords:** Pattern recognition, multilayer , cluster neural network.

### INTRODUCTION

Character recognition is becoming more and more important in the modern world. It helps humans ease their jobs and solve more complex problems. An example is handwritten character recognition. Handwritten digit recognition is a system widely used in the United States. This system is developed for zip code or postal code recognition that can be employed in mail sorting. This can help humans to sort mails with postal codes that are difficult to identify. For more than thirty years, researchers have been working on handwriting recognition. Over the past few years, the number of companies involved in research on handwriting recognition has continually. The advance of handwriting processing results from a combination of various elements, for example:

Improvements in the recognition rates, the use of complex systems to integrate various kinds of information, and new technologies such as high quality high speed scanners and cheaper and more powerful CPUs. Some handwriting recognition system allows us to input our handwriting into the system. This can be done either by controlling a

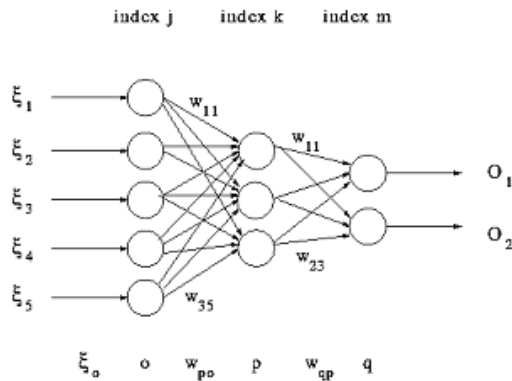
mouse or using a third-party drawing tablet. The input can be converted into typed text or can be left as an "ink object" in our own handwriting. We can also enter the text we would like the system to recognize into any Microsoft Office program file by typing. We can do this by typing 1s and 0s. This works as a Boolean variable. Handwriting recognition is not a new technology, but it has not gained public attention until recently. The ultimate goal of designing a handwriting recognition system with an accuracy rate of 100% is quite illusionary, because even human beings are not able to recognize every handwritten text without any doubt. For example, most people can not even read their own notes. Therefore there is an obligation for a writer to write clearly. In this paper, Neural Networks will be defined. The advantages of using Neural Networks to recognize handwritten characters will be listed. Finally, Artificial Neural Networks, using back-propagation method, will be used to train and identify handwritten digits. Some of the advantages of using Neural Networks for recognition are-More like a real nervous system, can solve problems with multiple constraints, it is insensible to noise, often good for solving complex problems.

**Artificial Neural Network**

Artificial Neural Network is also a good pattern recognition engines and robust classifiers. They have the ability to generalize by making decisions about imprecise input data. They also offer solutions to a variety of classification problems such as speech, character and signal recognition.

Artificial Neural Network (ANN) is a collection of very simple and massively interconnected cells. The cells are arranged in a way that each cell derives its input from one or more other cells. It is linked through weighted connections to one or more other cells. This way, input to the ANN is distributed throughout the network so that an output is in the form of one or more activated cells.

The information in an ANN is always stored in a number of parameters. These parameters can be pre-set by the operator or trained by presenting the ANN with examples of input and also possibly together with the desired output. The following figure is an example of a simple of ANN:



**Fig. 2.1 Multilayer Artificial Neural Network**

**Examples of types of Neural Networks**

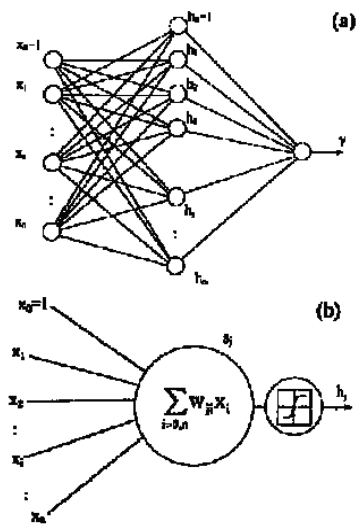
**Multi-Layer Feed-forward Neural Networks**

Multi-Layer Feed-forward neural networks (FFNN) have high performances in input and output function approximation. In a three-layer FFNN, the first layer connects the input variables. This layer is called the input layer. The last layer connects the output variables. This layer is called output layer. Layers between the input and output layers are called hidden layers. In a system, there can be

more than one hidden layer. The processing unit elements are called nodes. Each of these nodes is connected to the nodes of neighboring layers.

The parameters associated with node connections are called weights. All connections are feed forward, therefore they allow information transfer from previous layer to the next consecutive layers only. For example, the node  $j$  receives incoming signals from node  $i$  in the previous layer. Each incoming signal,  $x_i$ , is a weight  $w_{ji}$ . The effective incoming signal  $S_j$  to node  $j$  is the weighted sum of all incoming signals.

The following figures (Figure 2.1 (a)) are an example of an usual FFNN and nodes (Figure 2.1 (b)):

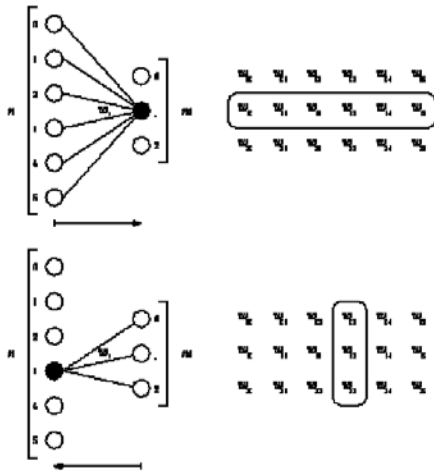


**Fig. 2.1 (a) Three layers feed-forward neural net (b) Processing unit element**

**Back-propagation algorithm**

Back-propagation algorithm consists of two phases. First phase is the forward phase. This is the phase where the activations propagate from the input layer to the output layer. The second phase is the backward phase. This the phase where the observed actual value And the requested nominal values in the output layer are propagated backwards so it can modify the weights and bias value.

The following figure is an example of the forward propagation and backward propagation.



**Fig. 2.1 (b) Forward and Backward propagation**

The forward propagation, the weights of the needed receptive connection of neuron  $j$  are in one row of the weight matrix. The backward propagation, the neuron  $j$  in the output layers calculates the error between the actual output values. This output value is known from both the forward propagation and the expected nominal target value. The error is propagated backwards to the previous hidden layer and the neuron  $i$  in the hidden layer calculates this error that is propagated backwards to its previous layer. This is why the column of the weight matrix is used

**Handwritten Character Recognition**

There are many different types of recognitions in the modern time, which can really solve complex problems in the real world today. Examples of recognitions are: face recognition, shape recognition, handwritten character recognition and handwritten digit recognition.

**Examples of types of Recognitions**

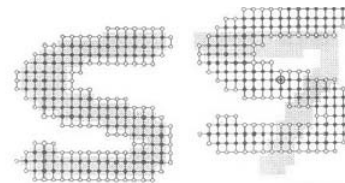
**Shape Recognition**

Shape describes a spatial region. Most shapes are a 2-D space. Shape recognition works on the similarity measure so that it can determine that two shapes correspond to each other. The recognition needs to respect the properties of imperfect perception, for example: noise, rotation, shearing, etc.

One of the techniques used in shape recognition is elastic matching distance. Here we use a binary-valued image  $X$  on the square lattice  $S$  as an example. The value of  $X$  at pixel belonging to  $S$  is denoted  $X(s)$ . The images we are interested in this example are the images of the handwritten numerals. Pixel with value 1 stands for "black" or "numeral" and pixel with value 0 stands for "white" or "background". There are ten numeral classes numbered 0 to 9. These ten numeral classes come in different shapes. The goal is to provide a space of images on  $S$  with an alternative metric  $(X, X')$  that can reduce this intra-class spread as much as possible.

Matching problems are not easy tasks. Satisfactory matches can sometimes be obtained reliably and rapidly under two general conditions:

1. Objects to be matched should be topologically structured.
2. Initial conditions should provide a rough guess of the map to be constructed.



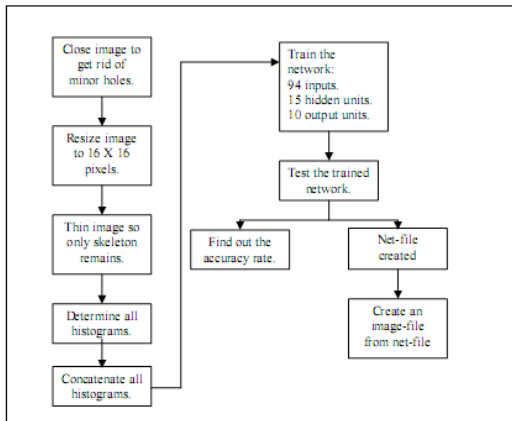
**Fig. 3.1 Numeral -5" is mapping to numeral -7"**

**Neural Network based Handwritten Digit Recognition**

Artificial Neural Network system is used to recognize ten different handwritten digits. These are digits from zero to nine. Here, back-propagation neural network is used to train all the data. The major problem is the digits are handwritten; therefore it is subject to enormous variability. Digits were written by different people, using a great variety of sizes, styles, and instruments.

Back-propagation can be applied to real image recognition problems without a complex pre-processing stage, which requires a detailed engineering. The learning network is fed directly with images rather than feature vectors.

Before inputting the data into the network, the image has to be closed first so there would have no minor holes. Then the image is resized to 16 X 16 pixels. Afterwards, the image is thinned so only the skeleton remains. When the skeleton image is obtained, the horizontal, vertical, right diagonal, and left diagonal histogram of the image is determined. Then the histograms are concatenated into one large integer sequence. The integer sequence is the digit representation. This is fed into the neural network. A three-layered neural network is used. This is 94 input units, 15 hidden units, and 10 output units.



**Fig. 3.2 Flow chart of the Neural Network System**

An image, which contains 100 samples of number, is fed into the system to train and test. They are 100 samples of the same number with different writing styles. Then a net-file is created and can be used to create an image-file. This image-file shows the recognized number.

**Handwritten Digit Recognition with Neural Networks**

Handwritten digit recognition is a created system that is used to recognize handwritten digits. The handwritten digit images get transformed into histograms and these histograms are fed into a neural network. This neural network outputs scores for matching the input digit against the ten possible digits (0-9). The data is trained and tested and it outputs the accuracy rate. The results can show us which numeral needs more training to reach high accuracies and which numeral the system had a difficulty to identify.

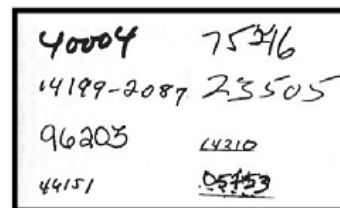
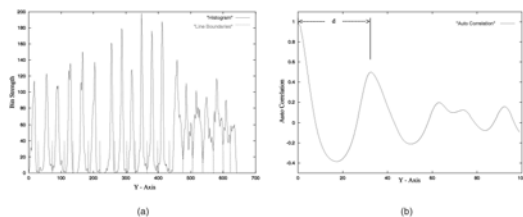
**Neural Network Digit Recognition**

In order to have a learning task that is reasonably workable, a great amount of pre-processing of the digits is carried out using conventional Artificial Intelligence techniques. This is done before the digits are fed to the Artificial Neural Network.

The difficult task is there are some handwritten digits that often run together or not fully connected. Numeral 5 is an example. But once these tasks have been carried out, the digits are available as individual items. But the digits are still in different sizes. Therefore a normalization step has to be performed so we can have to have digits in equal sizes. After the digits are normalized, they are fed into the ANN. This is a feed-forward network with three hidden layers. The input is a 16 x 16 array that corresponds to the size of a normalized pixel image.

The first hidden layer contains 12 groups of units with 64 units per group. Each unit in the group is connected to a 5 x 5 square in the input array and all 64 units in the group have the same 25 weight values.

The second hidden layer consists of 12 groups of 16 units. This layer operates very similar to the first hidden layer, but now it seeks features in the first hidden layer. The third hidden layer consists of 30 units that are fully connected to the units in the previous layer. The output units are in turn fully connected to the third hidden layer.

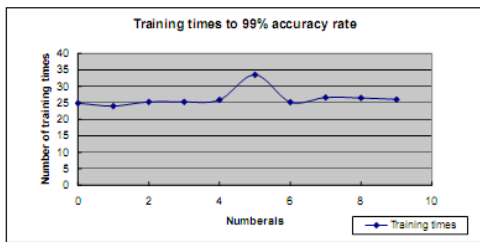


**Fig. 4.1 Example of Handwritten Digits**

**Imitation**

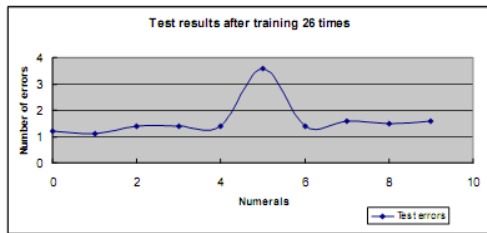
An image is fed into the network to train. Back-propagation neural network is used for training the network. Although it is only one image, it contains 100 samples of the same number. For every 10 epochs, the information is saved into the network. After training, the network is tested and the accuracy rate reached to 99%. This is a very high accuracy rate. The network was not stable because the training results changes everyday. If we take numeral —2" as an example, today we might have to train 20 times in order to reach 99% accuracy, but tomorrow we maybe have to train 25 times in order to reach 99% accuracy. The following graph is the average number of times that a data needs to be trained in order to reach 99% accuracy:

- Fig. 5. Interline distance estimation.
- (a) The y-axis projection of the document in Fig. 1, and the text-line boundaries.
  - (b) Autocorrelation of the projection used to estimate the value of d.



**Fig. 5.1 Results of trained data**

Here is the test results after training 26 times



**Fig. 5.2 Results of tested data**

After training 26 times, the network is tested. We can see that there were more errors for numeral -5". This means that numeral -5" still need more network training to reach an accuracy of 99%.

The following is the image-file produced using the net-file:



**Fig. 5.3 Image -file of numeral "5"**

**Discussion and Future Improvements**

From the Imitation, the training and testing results gives an accuracy rate of 99%. This is a high accuracy rate. From the results, we also realized that the system has trouble identifying numeral —5". This is probably caused by the —head" of the numeral is not fully connected or maybe it looks like a numeral —6".Every day the system gives a different result for the training of each numeral. Therefore an average of training times was taken to produce the results. This should be improved by having a close look at the program and the system. The image-file produced does not show a clear numeral that was trained. The Unix's Editor Paint was used to open the image-file. Maybe this was not suitable software to open the image-file. Maybe we should try to find another way to open the file to improve the image.

**CONCLUSION**

Using Neural Network system, back-propagation learning, to recognize handwritten digits was very successful. An image, which contained 100 samples of each number, was trained and tested. The accuracy rate of recognizing the number was 99%. This accuracy rate is very high. From the training and testing results, it was concluded that the system had more trouble identifying numeral —5". This maybe caused by the fact that the digit is running together or maybe it is not fully connected. The system was not stable. It gave different training and testing results every day for each numeral. It will need to take a close look at the system and should look for improvements for the future. From the net-file, the system was able to produce an

image-file. The image-file produced showed the recognized number. By looking at figure 5.2, it is concluded that the image-file produced does not show the numeral "5" clear enough. This part will also need more improvements.

Apart from the above problems and parts that need improvements, the overall recognition system was successful.

## REFERENCES

1. Bernard Haasdonk , Hans Burkhardt, Invariant kernel functions for pattern analysis and machine learning, *Machine Learning*, v.68 n.1, p.35-61, July 2007
2. Kussul, E., Baidyk, T., Kasatkina, L., Lukovich, V "Rosenblatt Perceptrons for Handwritten Digit Recognition". Proceedings of International Joint Conference on Neural Networks IJCNN, 2001, V.2, 2001, pp. 1516 –1520.
3. Bock, H.H., Diday, E. (eds. 2000): Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, Series: Studies in Classification, Data Analysis, and Knowledge Organisation, Vol. 15, Springer-Verlag, Berlin.
4. Esposito, F., Malerba, D., Tamma, V. (2000): Dissimilarity Measures for Symbolic Objects. Chapter 8.3 in H.-H. Bock and E. Diday (Eds.), Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data, Series: Studies in Classification, Data Analysis, and Knowledge Organization, vol. 15, pp. 165-185, Springer-Verlag.
5. Hao Y., Shi Y., Zhang D., Zhu X. 2001, "An effective result-feedback neural algorithm for handwritten character recognition" International Journal of Neural Parallel & Science Computations, Vol. 9, No. 2, pp.139~150
6. Keyzers D., NeyH., Paredes R., Vidal E. 2002, "Combination of Tangent Vectors and Local Representations for Handwritten Digit Recognition", <http://wasserstoff.informatik.rwth-aachen.de/~keyzers/Pubs/SPR2002/comb.html>
7. Neural Networks. 2003, <http://documents.wolfram.com/applications/neuralnetworks/index2.html>
8. Neural Networks: Advanced tutorial. 2002, <http://www.pmsi.fr/home-gb.htm>
9. Schofield, E. 2001,"Concepts in Pattern Recognition", <http://userver.ftw.at/~schofiel/patterns.html>