ISSN:0975-9646
Malothu Nagu et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1685-1692

A novel method for Handwritten Digit Recognition with Neural Networks

MALOTHU NAGU[*1], N VIJAY SHANKAR[#]2,K.ANNAPURNA[3]**
[*1]*Department of ECE, V.K.R &V.N.B.Engg College,Gudivada. Krishna (Dist), A. P, S INDIA.*
[#2]*Department of EIE ,S R T I S T, Ramananda Nagar, Nalgonda (Dist).A.P, S.INDIA.*
[**3]*School of Electronics, Vignan University, Guntur (Dist).A.P, S.INDIA.*

***Abstract* :Character recognition plays an important role in the modern world. It can solve more complex problems and makes humans' job easier. An example is handwritten character recognition. This is a system widely used in the world to recognize zip code or postal code for mail sorting. There are different techniques that can be used to recognize handwritten characters. Two techniques researched in this paper are Pattern Recognition and Artificial Neural Network (ANN). 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, Chinese 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, neural network, handwritten characters**

1. INTRODUCTION AND HISTORY

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 [4] which is widely used in the world.

This system is developed for zipcode 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 [4] has continually increased. 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 [4] 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, both Pattern Recognition and Neural Networks [2] will be defined. Examples of types of Pattern Recognition and Neural Networks will be discussed. The advantages of using Neural Networks][2]to recognize handwritten characters will be listed. Finally, Artificial Neural Networks, using back-Propagation method will be used to train and identify handwritten digits.

2. PATTERN RECOGNITION

2.1. What is Pattern Recognition?

Pattern recognition system consists of two-stage process. The first stage is feature extraction and the second stage is classification. Feature extraction is the measurement on a population of entities that will be classified. This assists the classification stage by looking for features that allows fairly easy to distinguish between the different classes. Several different features have to be used for classification. The set of features that are used makes up a feature vector, which represents each member of the population. Then, Pattern recognition system classifies each member of the population on the basis of information contained in the feature vector. The following is an example of feature vectors that have been plotted on a graph.

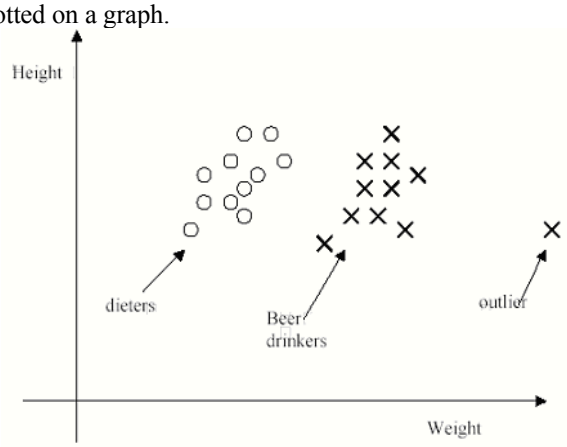


Figure 2.1 Two classes fully separated

From Figure2.1, it shows two clusters of points. Each of them corresponds to one of the classes. These classes are fully separated and can be easily distinguished.

2.2. Pattern recognition methods.

2.2.1. Bayesian decision theory.

The Bayesian decision theory is a system that minimizes the classification error. This theory plays a role of a prioi. This is when there is priority information about something that we would like to classify. For example, suppose we do not know much about the fruits in the conveyer belt. The only information we know is that 80% of the fruit in the conveyer

belt are apples, and the rest of them are oranges. If this is the only information we have, then we can classify that a random fruit from the Conveyer belt is apple. In this case, the prior information is the probability of either an apple or an orange is in the conveyer belt. If we only have so little information, then we would have the following rule:

Decide"apple'if P (apple) > P (orange), otherwise decide"orange'

Here, P (apple) is the probability of being an apple in the conveyer belt. This means that P(apple) = 0.8 (80%). This is probably strange, because if the above rule is used, then we are classifying a random fruit as an apple. But if we use this rule, we will be right 80% of the time.

This is a simple example and can be used to understand the basic idea of pattern recognition. In real life, there will be a lot more information given about things that we are trying to classify. For example, we know that the color of the apples is red. Therefore if we can observe a red fruit, we should be able to classify it as an apple. We can have the probability distribution for the color of apples and oranges.

Let w_{app} represent the state of nature where the fruit is an apple, let w_{ora} represent the state of nature where the fruit is an orange and let x be a continuous random variable that represents the color of a fruit. Then we can have the expression $p(x|w_{app})$ representing the density function for x given that the state of nature is an apple.

In a typical problem, we would be able to calculate the conditional densities $p(x|w_j)$ for j so it will be either an apple or an orange. We would also know the prior probabilities $P(w_{app})$ and $P(w_{ora})$. These represent the total number of apples versus oranges in the conveyer belt. Here we are looking for a formula that will tell us about the probability of a fruit being an apple or an orange just by observing a certain color x. If we have the probability, then for the given color that we observed, we can classify the fruit by comparing it to the probability that an orange had such a color versus the probability that an apple had such a color. If we were more certain that an apple had such a color, then the fruit would be classified as an apple.

So, we can use Baye's formula, which states the following:

$$\mathbf{P(w_j|x) = p(x|w_j)\,P(w_j)/p(x)}$$

What the formula means is that using a priori information, we can calculate the a posteriori probability of the state of nature being in state w hat we have given that the feature value x has been measured. So, if we observe a certain x for a random fruit in the conveyer belt, then by calculating $P(w_{app}/x)$ and $P(w_{org}/x)$. we would decide that the fruit is apple if the first value is greater than the second one and if $P(w_{org}/x)$ is org greater, then we would decide that the fruit is orange. So, the Bayesian decision rule can be stated as:

Decide w_{org} if $P(w_{org}|x) > P(w_{app}|x)$, otherwise, decide w_{app} ,

Since p(x) occurs on both sides of the comparison, the rule can also be equivalent to the following rule:

Decide w_{org} if $p(x|w_{org})P(w_{org}) > p(x|w_{app})P(w_{app})$, otherwise decide w_{app}

The following graph shows the a posteriori probabilities for the two-class decision problem. For every x, the posteriors has to sum to 1. The red region on the x axes represents the values for x for which would decide as "apple'. The orange region represents values for x for which would decide as "orange'.

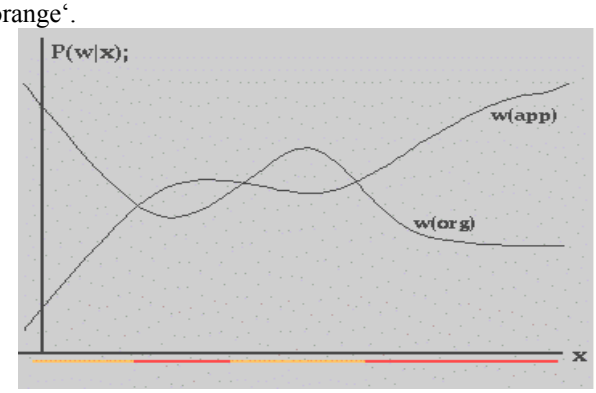


Figure 2.2 A posteriori probabilities for two class decision problem.

The probability that we would probably make an error is any minimum of the 2 curves at any point, because it represents the smaller probability that we did not pick. The formula for the error is the following:

$P(error|x) = \min[p(w_{app}|x), p(w_{org}|x)]$.

2.2.2 Nearest Neighbor rule.

The Nearest Neighbor (NN) [10]rule is used to classify handwritten characters. The distance measured between the two character images is needed in order to use this rule. Without a priori assumptions about the distributions from which the training examples are drawn, the NN [2]rule achieves very high performance. The rule involves a training set of both positive and negative cases. A new sample is classified by calculating the distance to the nearest training case. The sign of the point determines the classification of the sample. The following figure shows an example of NN rule

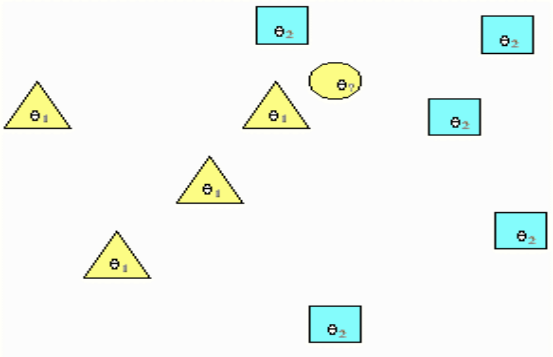


Figure 2.3 The Neural Network rule.

In this example there are two classes θ_1, which are the yellow triangles and θ_2 which are blue squares. The yellow circle represents the unknown sample X.We can see that the unknown samples nearest neighbor is from class θ_1 therefore it is labeled as class θ_1.When the amount of pre-classified points is large, it is good to use the majority vote of the nearest k neighbors instead of the single nearest neighbor. This method is called the k nearest neighbor (k-NN) rule.

The k-NN[2] rule extends the idea by taking and assigning the k nearest points the sign of the majority. It is common to choose k small (with respect to the number of samples) so that the points are closer to x to give accurate estimate of the true

class of x.If the k values are large, it will help reduce the effects of noisy points within training data set. Also, large k values will minimize the probability of misclassifying x. The following figure shows an example of k-NN rule with k value equal 3:

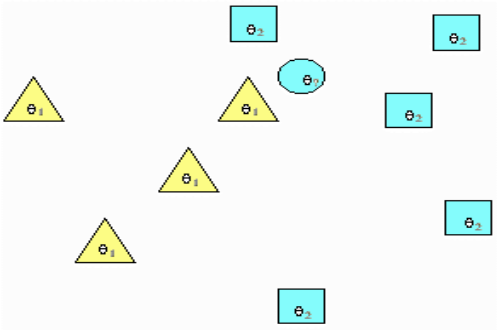


Figure 2.4 The Nearest k-Neural Network rule with k=3.

As before, there are two classes: θ_1 which are the yellow triangles, and θ_2 which are the blue squares. The blue circle represents the unknown sample x. We see that two of its nearest neighbors are from class θ_2 , so it is labeled as class θ_2.

2.2.3 Linear Classification Discrimination.

The goal of Linear Classification is to assign observations into the classes. This can be used to establish a classifier rule so that it can assign a new observation into a class. In another words, the rule deals with assigning a new point in a vector space to a class separated by a boundary. Linear classification provides a mathematical formula to predict a binary result. This result is a true or false (positive or negative) result or it can be any other pair of characters. In general, we will assume that our Results are Boolean variable. To do this prediction, we use a linear formula over the given input data. We refer this as inputs. The linear form is computed over the inputs and the result is compared against a basis constant. It is depending on the result of the comparison that we would be able to predict true or false. The following is the equation that can be stated as the discriminator:

$$\mathbf{a_1 x_1 + a_2 x_2 + \ldots + a_n x_n > x_0}$$

Here, a_1, a_2 are the variables that correspond to one observation. and x_1,x_2 together with x_0 are the solution vector plus the basis constant. Corresponding to each of the input vector a, there is a variable b. This variable b is the dependent Boolean variable. We refer this as the output. We normally have many m data points (a row vector of inputs a and the corresponding output,

b). For convenience, we place all the data in matrices and vectors. A denotes the matrix of the inputs and it is in the dimension of m by n. The m Boolean outputs will be stored in a vector B. Now, the

problem of linear classification can be stated in terms of the matrices and vectors. For example: we want to find x and x_0 such that:

$$\mathbf{(Ax > x_0)\ \ equiv\ \ B}$$

This equivalence means that every row of the left-hand-side is a relation. The relation for the given data will be true or false. This has to match the corresponding B value. When we have more data points than columns, this is m > n, the problem does not have a solution. We can not find a vector x that will satisfy all the true values. Therefore we try to find a solution that can match vector B as good as possible. The best matching means that there is a minimum number of errors or there is a weighted minimum of errors. There will be a weight that is assigned to the true errors and another weight that is assigned to the false errors.

The idea of obtaining good classification is to be able to use it to predict the output for new data points. Here, the discriminator is a prediction formula, A and B are the training data, and x and x are the parameters of the model fitted to the training 0 data. The following figure shows the main components of linear classification:

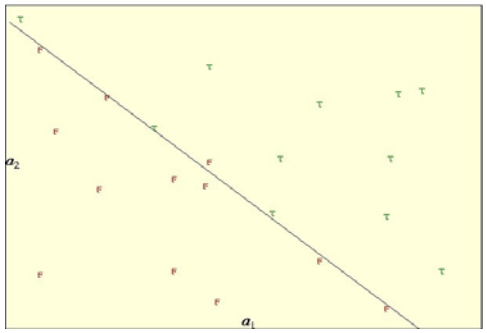


Figure 2.5 Linear Classification in two dimensions.

Here, true are marked with a green T and false are marked with a red F. The inputs are two independent variables, a_1 and a_2 . Here we use them to position the points in the plane. There are eleven Ts (positives) and twelve Fs (negatives). This linear classification in two dimensions is a straight line. The straight line drawn is a very good classification because the line separates most of the points correctly. We can see that all the Ts are on one side of the graph and all the Fs, but one, are on the other side of the graph. Therefore this classification has one "false positive" and no "false negatives". A false positive is a data point that is classified as positive but it is negative and vice versa for false negatives.

3. HANDWRITTEN CHARACTER RECOGNITION IN PATTERN RECOGNITION.

Linear Classification [9] is a useful method to recognize handwritten characters.[4] The background basis of Artificial Neural Network (ANN)[2] can be implemented as a classification function. Linear Classification works very similar to Artificial Neural Network because the mapping of the ANN cell or one layer of the ANN cell is equivalent to the linear discrimination function. Therefore, if the ANN is a two-layer network, which is consisting of an input and an output layer, it can act as a linear classifier.

3.1. What is Neural Network?

A Neural Network (NN) [2]is a function with adjustable or tunable parameters. Let the input to a neural network be denoted by x. This is a real-valued or row vector of length and is typically referred to as input or input vector or regressor or sometimes pattern vector. The length of the vector x is the number of inputs to the network. So let the network output be denoted b Y. This is an approximation of the desired output y, which is also a real-valued vector having one or more components and the number of outputs from the network. The data sets often contain many input and output

pairs. The x and y denote matrices with one input and one output vector on each row.

A neural network[2] is a structure involving weighted interconnections between neurons or units. They are often non-linear scalar transformations but can also be linear scalar transformation. The following figure shows an example of a one-hidden-layer neural network with three inputs, $x = \{x_1, x_2, x_3\}$.

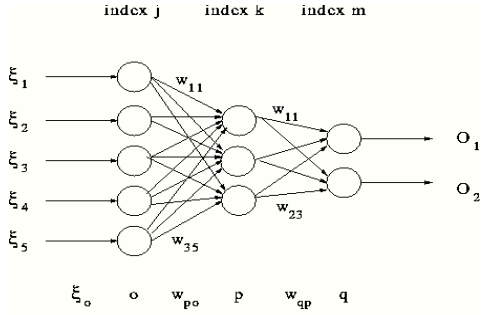


Figure 3.1 Feed-forward Neural Network with 3 inputs, two hidden neurons and one output neuron.

The three inputs, along with a unity bias input, are fed each of the two neurons into the hidden layer. The two outputs from this layer and from a unity bias are then fed into the single output layer neuron. This produces the scalar output Y.The layer of neurons is called hidden layer because the outputs are not directly seen in the data.

Each arrow in the Figure 3.1 corresponds to a real-valued parameter, or a weight, of the network. The values of these parameters are tuned in the training network.

A neuron is structured to process multiple inputs. This includes the unity bias in a non-linear way. Then, this produces a single output. All inputs to the neuron are first augmented by multiplicative weights. These weighted inputs are summed and then transformed via a non-linear activation function and as indicated from the above Figure 3.1, the neurons in the first layer of the network are non-linear. The single output neuron is linear because no activation function is used. 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 example.

3.2 Artificial Neural Network.

Artificial Neural Network (ANN) has been around since the late 1950's. But it Was not until the mid-1980 that they became sophisticated enough for applications. Today, ANN[2] is applied to a lot of real- world problems. These problems are considered complex problems. ANN's are 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. es of input and also possibly together with the desired output. The following figure3.2 is an example of a simple of ANN:

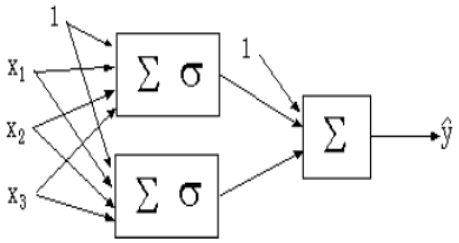


Figure 3.2 Multi-Layer Artificial Neural Networks.

3.3. Examples of types of Neural Networks.

3.3.1. Multi-Layer Feed-forward Neural Networks.

Multi-Layer Feed-forward neural networks (FFNN)[9] 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 is a weight. The effective incoming signal to node j is the weighted sum of all incoming signals. The following figures (Figure 3.3 (a)) are an example of a usual FFNN and nodes (Figure 3.3 (b)):

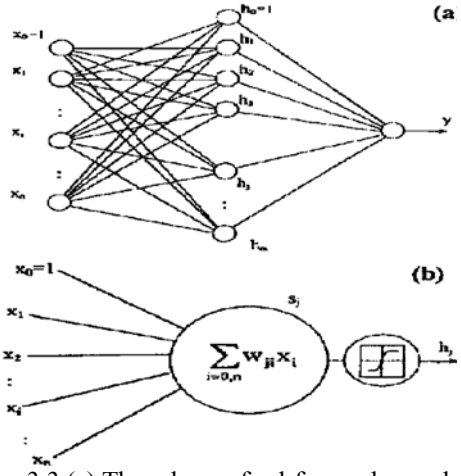


Figure 3.3 (a) Three layers feed-forward neural net,
(b) Processing unit element.

3.3.2 Back-propagation algorithm.

Back-propagation algorithm[11] 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 is the phase where then the observed actual value and the requested nominal value in the output layer are propagated backwards so it can modify the weights and bias values.

The following figure 3.4 is an example of the forward propagation and backward propagation.

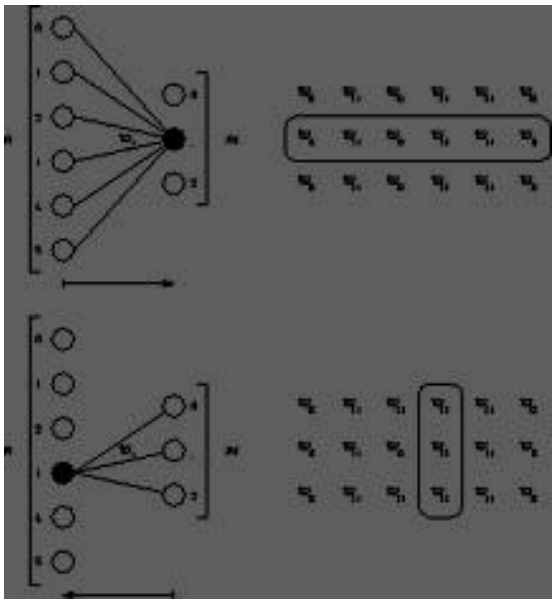


Figure 3.4 Forward and Backward propagation.

In forward propagation, the weights of the needed receptive connections of neuron j are in one row of the weight matrix. In backward propagation, the neuron j in the output layer calculates the error between the expected nominal targets. 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. n the actual output values. This output value is known from both the forward propagation and backward propagation.

3.3.3. 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[1], handwritten character recognition[4], such as handwritten Chinese character recognition[3] and handwritten digit recognition.

Shape Recognition[1].

Shape describes a spatial region. Most shapes are a 2-D space. Shape recognition[1] 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.

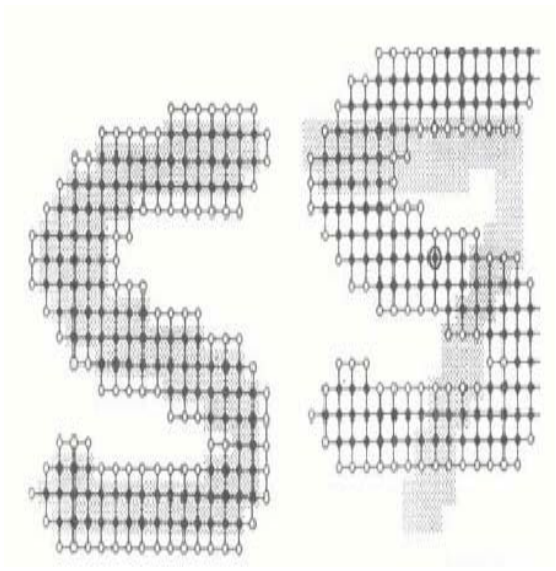


Figure 4.1 Numeral —5“ is mapping to numeral —7“.

3.4.3 Chinese Character Recognition.

Recognition of similar characters (or confusion characters) is a difficult task. An example of similar characters is Chinese characters[3]. Chinese characters have a wide range of complexity. The characters may consist of one to thirty or more distinct strokes. The differences between characters can be quite small. One of the techniques that can be used to recognize handwritten Chinese characters is using Optical Character Recognition (OCR). Here, OCR uses probabilistic neural network to recognize Chinese characters[3]. The training of the classifier starts with using the distortion-modeled characters from four fonts. Statistical measures are taken the set of features computed from the distorted character. Based on these measures, the space of feature vectors is transformed to the optimal discriminant space for nearest neighbor classifier. In the discriminant space, a probabilistic neural network classifier is trained. For classification, the modifications are presented to the standard approach by implying the probabilistic neural network structure. The approach is compared using discriminant analysis and Geva and Sitte's

Decision Surface Mapping classifiers. All methods are tested using 39,644 characters in three different fonts.

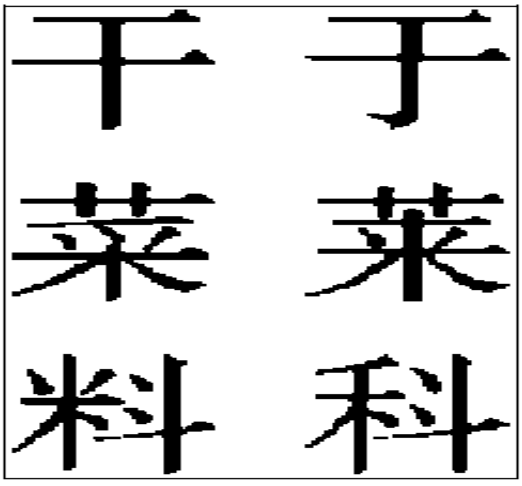


Figure 4.2 Examples of visually similar characters.

4. NEURAL NETWORK BASED HANDWRITTEN DIGIT RECOGNITION.

Artificial Neural Network[2] system is used to recognize ten different handwritten digits. These are digits from zero to nine. Here, back-propagation neural network[11] 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 [11]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 [6]. This is fed into the neural network. A three-layered neural network issued. This is 94 input units, 15 hidden units, and 10 output units (Appendix A for picture format).

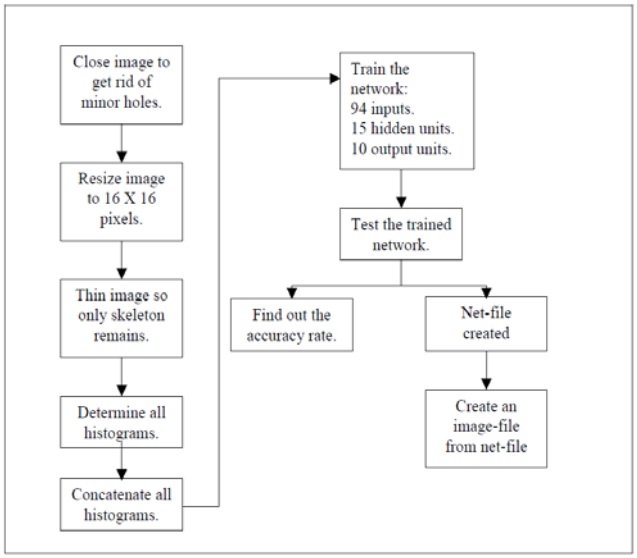


Fig4.3. Flowchart of Neural Network System.

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.

4.1 Handwritten Digit Recognition With Neural Networks

4.1.1. Introduction.

Handwritten digit recognition[4] [9] is a created system that is used to recognize handwritten digits.[4] The handwritten digit images get transformed into histograms and these histograms are fed into a neural network. This neural network[2] 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.

4.1.2 Neural Network Digit Recognition System

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 (AI) techniques. This is done before the digits are fed to the ANN.[2] 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.

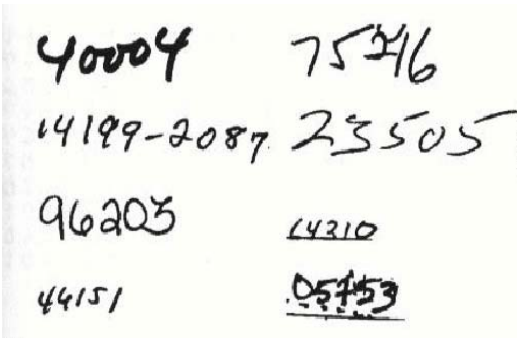


Figure 4.4: Example of handwritten digits.

5. SIMULATION. AND RESULTS.

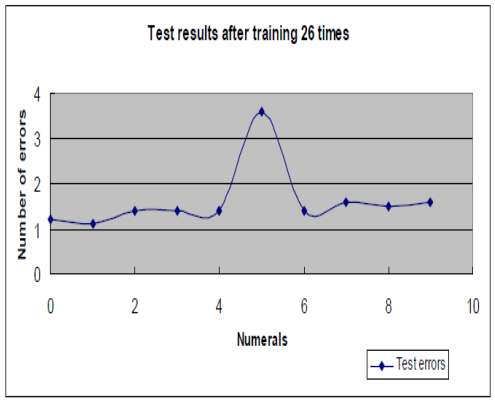
An image is fed into the network to train. Back-propagation neural network is used for training the network. Although it is only one image, it contains 100 samples of the same number. For every 10 epochs, the information is saved into the network. After training, the network is tested and the accuracy 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: Training times.

Graph 5.1 Results of trained data.

From the training results (See Appendix B for table results) and from the above graph, we can see that the system has more trouble identifying numeral —5“. Here is the test result after training 26 times (See Appendix B for table results).
Test results after training 26 times



Graph 5.2 Results of tested data.

Test errors
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:

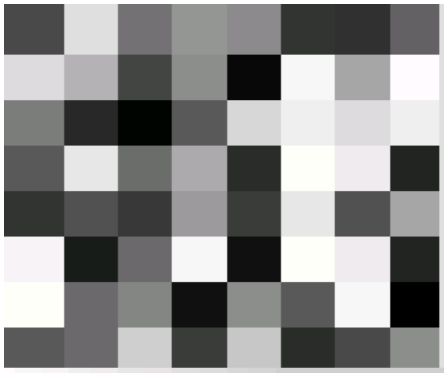


Figure 5.3 Image-file of numeral —5“.

6. DISCUSSION AND FUTURE IMPROVEMENTS.

From the simulation, 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“.

From Appendix B, we realize that the system is not very stable. 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. This was not suitable software to open the image-file. We May try to find another way to open the file to improve the image.

7. 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 numberal —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.

8.REFERENCE

1. Bienenstock E., Doursat R. 1993, ”A shape recognition model using dynamical links‘ Neural: Computation in Neural Systems, Vol. 5, No. 2, pp. 241~258
2. Buffa F., Porceddu I. 1997 ”Temperature forecast and dome seeing minimization. A case study using neural network model‘, http://www.pd.astro.it/TNG/TechRep/rep67/rep67.html
3. Chang H., Fu H., Xu Y. 1990, ”Recognition of handwritten similar Chinese characters by self-growing probabilistic decision-based neural network‘ International Journal of Neural Parallel & Science Computations, Vol. 9, No. 6, pp.545~561
4. Claus, D. ”Handwritten Digit Recognition‘, http://www.robots.ox.ac.uk/~dclaus
5. Fassnacht C., Zippelius A. 1990, ”Recognition and categorization in a structured neural network with attractor dynamics‘ Neural: Computation in Neural Systems, Vol. 2, pp.63~84
6. 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. 9z No. 2, pp.139~150
7. Hubbard J. 2000, Schaum‘s outline of theory and problems of programming with C++, 2 nd ed., McGraw Hill, New York.
8 Jones R. 1991, The C programmer‘s Companion: ANSI C library functions, Silicon Press, New Jersey.
9. Keysers D., NeyH., Paredes R., Vidal E. 2002, ”Combination of Tangent Vectors and Local Representations for Handwritten Digit Recognition‘,
10. Middelberg U., Thiesing F., Vornberger O. 1995 ”Back-Propagation algorithm‘,http://www.informatik.uni-osnabrueck.de /papers_ html3/ wtc_95_frank/node7.html
11. Nearest Neighbor Rule: A Short Tutorial. 1995, http://cgm.cs.mcgill.ca/~soss/cs644/projects/simard/nn_theory.html

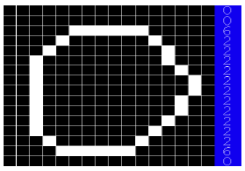
Appendix A – Methodology.

1) Close the image to get rid of minor holes.

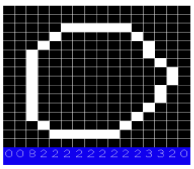
2) Resize the image to 16 X 16.

3) Thin the image so only the skeleton remains.

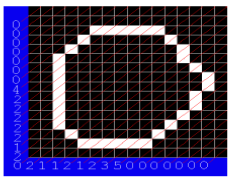
4) Determine all the histograms.



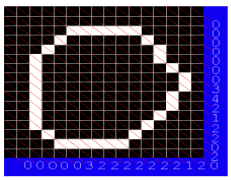
Determine horizontal histograms.



Determine vertical histograms.



Determine right diagonal histograms.



Determine left diagonal histogram

5) Concatenate all the histograms.

(0062222222222260
0082222222223320
00000004222221202112123500000000
00000322222221202022124300000000)

Appendix B – Results.

1. Training results.

Numerals	Training 1	Training 2	Training 3	Training 4	Training 5	Training 6	Training 7	Training 8	Training 9	Training 10	Average
0	29	24	24	24	26	25	23	28	26	20	24.9
1	20	26	25	25	28	23	22	21	24	26	24
2	22	27	27	25	25	23	31	26	23	24	25.3
3	21	25	23	29	24	24	25	24	28	30	25.3
4	27	23	23	28	26	24	26	25	26	31	25.9
5	35	29	32	34	32	30	38	39	30	36	33.5
6	22	21	23	24	31	28	29	23	25	26	25.2
7	28	23	24	24	25	25	35	31	26	25	26.6
8	24	25	26	26	29	26	21	32	28	28	26.5
9	27	26	28	30	26	24	24	25	21	29	26

2. Test results.

Numeral	Test 1 Errors	Test 2 Errors	Test 3 Errors	Test 4 Errors	Test 5 Errors	Test 6 Errors	Test 7 Errors	Test 8 Errors	Test 9 Errors	Test 10 Errors	Average
0	2	1	1	1	1	1	1	2	1	1	1
1	1	1	1	1	2	1	1	1	1	1	1
2	1	2	2	1	1	1	3	1	1	1	1
3	1	1	1	2	1	1	1	1	2	3	1
4	2	1	1	2	1	1	1	1	1	3	1
5	4	2	3	4	3	3	5	5	3	4	4
6	1	1	1	1	3	2	2	1	1	1	1
7	2	1	1	1	1	1	4	3	1	1	2
8	1	1	1	1	2	1	1	3	2	2	2
9	3	1	2	3	1	1	1	1	1	2	2

3. Hidden Layers.

INPUT UNITS	94 units
HIDDEN UNITS	15 units
OUTPUT UNITS	10 units