A Combining Method with Many Sets of Weights and Biases in Pattern Recognition Neural Network

Mr. A. Baskaran,
M.Phil. Research Scholar,
P.G. & Research Department of Computer Science,
St.Joseph's College of Arts and Science- Cuddalore,
Tamilnadu.
basmphil15@gmail.com

Mrs. C. Christy
Assistant Professor,
P.G. & Research Department of Computer Science,
St.Joseph's College of Arts and Science- Cuddalore,
Tamilnadu.
canschris@yahoo.com

Mr. P. Arunmani,
M.Phil. Research Scholar,
P.G. & Research Department of Computer Science,
St.Joseph's College of Arts and Science- Cuddalore,
Tamilnadu.
arunmani24mca@gmail.com

Abstract - In Supervised training method, the training data is a pair consisting of an input object (typically a vector) and a desired output value (also called supervisory signal). Pattern recognition systems are in trained from labeled 'training' data, but when no labeled data are available, other algorithms can be used to discover previously unknown patterns. Back propagation requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method, although it is also used in some unsupervised networks such as auto encoders. In this paper, we proposed a training method to classify all the training patterns using designed neural network pattern recognition. For more classification, the neural network designed as training data set has to be separated with a reject output. Neural network will find not only one but many sets of weights and biases with the help of training method being classify all the training patterns , control the recognized rejection and error rate is reduced. The proposed method can reduce the neural network size and to design fast smart sensors for the robots, it can be implemented on a Field Programmable gate array chip.

Keywords*- Neural Networks, Bitmap image, Pattern, Serialization, Layers.*

*****

I.INTRODUCTION

"Pattern analysis using Neural Networks with Back Propagation algorithm" is the complete solution pattern recognition and pattern mapping. This application solves the complicated problems which are unsolvable from artificial intelligence. It helps us to find the clustered patterns. It will give the information for highly matched pattern and low matched patterns. The pattern mapping is done in both color and gray scale images.

The network construction is done by a base class with three layer network. Image recognition is done by a separate class files. For serialization and deserialization is done by the .NET framework predefined class libraries. Virus detection, robot control, intrusion detection systems, pattern (image, fingerprint, noise) recognition also possible in future enhancements.

The Main core of the paper is Artificial Neural Network. In this paper we are storing the images in neurons by constructing the three layers. Three layers are Input layer, Hidden layer and Output layer. In this projects the inputs are must be a BMP(Bit map) file format. Basically the input and output in neural network are not predefined and constant.

The Back Propagation algorithm was used to train the neurons by the patterns. In the back propagation algorithm consist two major training parts. There Forward propagation and Back propagation.

Once the patterns are trained that all stored in a single neural network file in the binary format. Here we are using serialization for converting the patterns from image to binary. Whenever we are testing a pattern (.bmp) with neural network file the deserialization process will be done. Because we can't map the binary format image with an ordinary image (testing pattern). So we must deserialize the network file for recognize the input.

II. RELATED WORK

According to J.K.Basu[1] emphasize a pattern should be finger print image, speech signal. Its recognition is based on the supervised classification and unsupervised classification. In the supervised classification, input pattern is identified as a member of predefined class and the unsupervised classification is nothing but clustering in which the pattern is assigned to unknown class. The pattern recognition is designed as follows 1) data acquisition and preprocessing 2) data representation 3) decision making. Pattern recognition approached in four different ways as follows a) template matching b) statistical classification c) structural matching d) neural networks.

In order to improve the Image preprocessing Tao chen[2] introduced the image binarization using back propagation algorithm. He used a multilayer perceptron with comparison of threshold selecting methods. Multiple trained neural networks between units is invariant to position. Each unit is divided into several groups and different image is incorporated with each neurons.

Applying Back propagation Algorithm for pattern recognition and it can be implemented by using feed forward network has been trained accordingly reveled by Amit Kumar [3]. In training input patterns with associate outputs Artificial Neural Networks are composed of large number of structurally and functionally similar units called neurons. Elements are obtained as vector grid after training. It able to recognize all characters correctly.

Several Surveys have been collected for ANN and Abdullahi[4] illustrated the importance of Artificial Neural Networks[ANN] . It is not possible to train more than two layers having relationship between binary input and output characters. A continuous classification of a input stimuli when a stimulus appear at the network. The input and output functions that is specified for the units. Several supervised and unsupervised network architecture has been supported in neural network toolbox.

Back Propagation Neural Network algorithm (BPNN) is the most popular and the oldest supervised learning multilayer feed – forward network algorithm proposed by Rumelhart, Hinton and Williams. The aim of supervised learning is to update the network weights iteratively to globally minimize the difference between the actual outputs of the network and the desired outputs. Rashmi chaudry[5] surveys on BPNN known for its accuracy and simplicity.

Kuldip yora[6] reveals that it too has pros and cons such as slow convergence rate and problem to get stuck in local minima however, it is known for its accuracy. A multilayer feed forward neural network consists of input layer, hidden layer and output layer of neurons. Every node in a layer is connected to every other node in the neighboring layer. A feed forward neural network consists of one or more layers of usually non-linear processing units. BPLA use the gradient – decent search method to adjust the connection weights. The output of each neuron is the aggregation of the neurons of the previous level multiplied by its corresponding weights. The input values are converted into output functions with the help of active functions.

Sonali Wankhede[7] explains the Multilayer perceptron(MLP) neural network classifier and self organizing maps focuses on the various neural networks techniques. Learning process in MLP has been done by Back Propagation Algorithm. The neural networks are commonly categorized in terms of training algorithms such as fixed weight networks, supervised networks and unsupervised networks. Supervised learning networks have been the mainstream neural model development. The network learn to adapt based on the experiences collected from the previous training patterns.

III. EXISTING SYSTEM

The data set contains examples of input patterns together with the corresponding output results and the network learns to infer the relationship between input patterns and output results through training. The training process requires a training algorithm, which uses the training data set to adjust the network's weights and bias so as to minimize and error function and try to classify all patterns in the training data set. After training, the NN will have a set of weights and biases that will be used to recognize the new patterns. In general, training NN with a larger training data set can reduce the recognizing error rate, but there are some problems that we have to consider is Problems with a large training data set.

Disadvantages

- Data base oriented
- Problem sophisticated
- Implementation cost too high
- Limited input
- Recognizing time too high

IV. PROPOSED SYSTEM

In order to interpret the problems in this paper, we chose the recognition of the handwritten digits patterns as an illustrative example, because the handwritten digits Training NN with a larger training data set, it requires more time to minimize the error function, especially when the data set contains some patterns that are difficult to classify correctly. These patterns keep the error function reducing very slowly. Even though the NN has been trained by many epochs, they are still clustered in a wrong area; hence, they are called the misclassified patterns. The number of the misclassified patterns will increase when the size of the training data set is enlarged. If the NN have to recognize a pattern that approximates in shape to one of the above patterns, the recognition result will be wrong. With a bigger size of the NN, it can classify all patterns in the large training data set. However, if this NN is implemented on FPGA, it is required a larger number of logic cells and RAM for a hardware-based NN design Our idea to solve the above problems is “the training data set should be separated into some parts and the neural network will use many sets of weights and biases to classify all patterns in each part and between parts”. We propose a new structure of recognition NN with an especial output that is called “Reject output”, and build a training method corresponding to this structure. The name “Reject output” that means it is used to separate all difficult recognizing patterns from the training data set. Hence, these patterns are called “Rejected patterns”. Our training method uses the reject output to separate the training data set into some parts, and with a smaller number of patterns in each part, they can be classified by the NN using a distinct set of Weights and biases.

Advantages

- File system
- Fast Recognition
- Well formed algorithm (BP algorithm)
- Great accuracy
- Solution for Complicated task
- Acceptance

V. EXECUTION

5.1 Description

In system flow diagram the user select the image for the learning file then reducing the threshold value and then start the iteration process. If the training image obtains the maximum frequency value it shows the exact output, or else it try to minimum possibility of the related output.

5.2 Use case Diagram

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. An actor is represents a user or another system that will interact with the system modeled.

A use case is an external view of the system that represents some action the user might perform in order to complete a task.

5.3 System Architecture

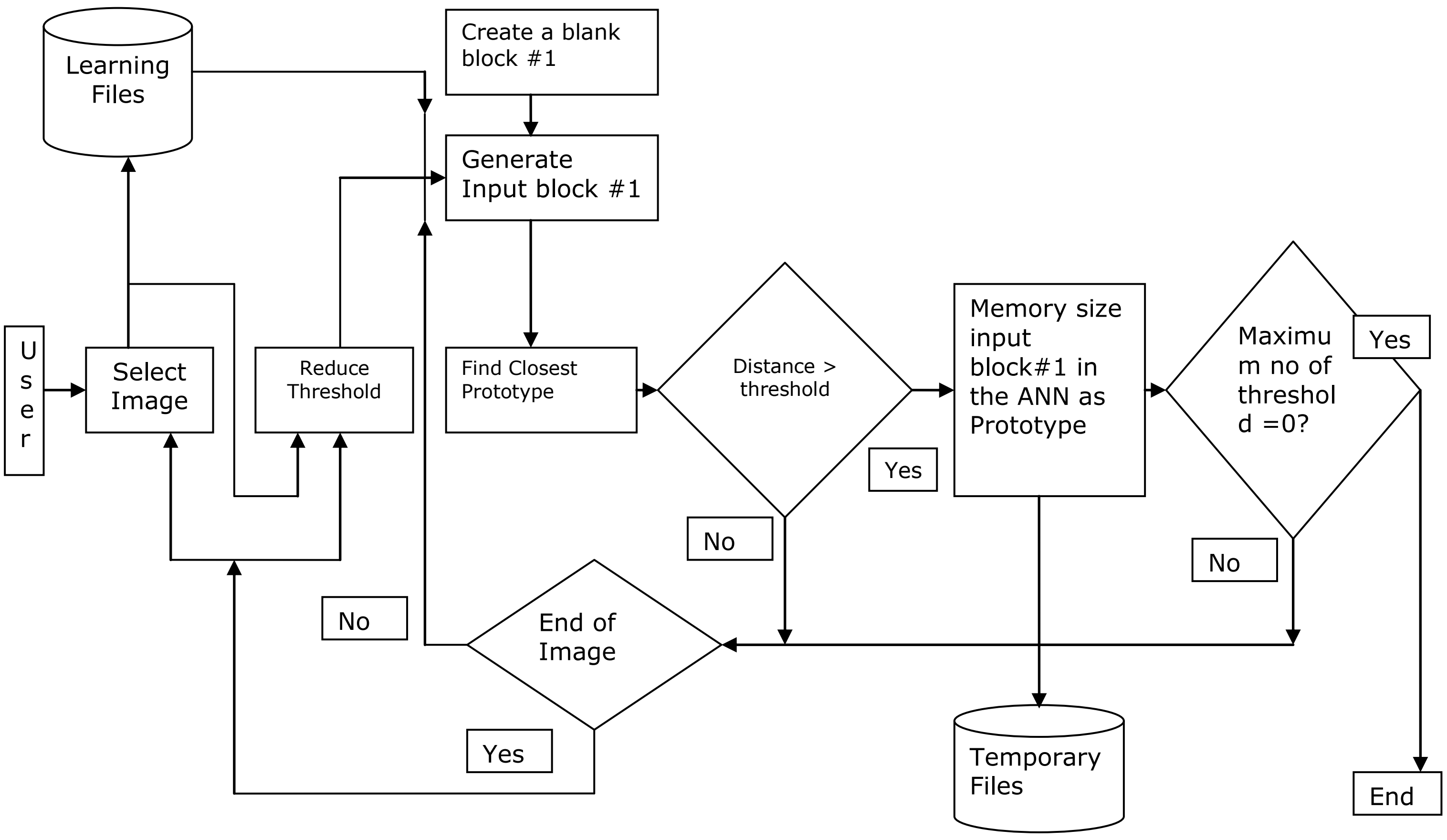

Figure :1

Activity diagrams(Fig:2) are typically used for business process modeling, for modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. Although UML activity diagrams could potentially model the internal logic of a complex operation it would be far better to simply rewrite the operation so that it is simple enough that you don't require an activity diagram. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and dataflow diagram (DFD), from structured development.

Class diagrams are the mainstay of object-oriented analysis and design. Class diagrams show the classes of the system, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. Class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling.

The network layers on the figure below(Fig:3) are implemented as arrays of struts [1], [3]. The nodes of the layers are implemented as

1. PREINPUT layer,
2. INPUT layer,
3. HIDDEN layer and
4. OUTPUT layer.

These layers are implemented in BASE class. The property of the Base class is serializable. Here the serializable keyword used to serialize the data (converting image into binary).

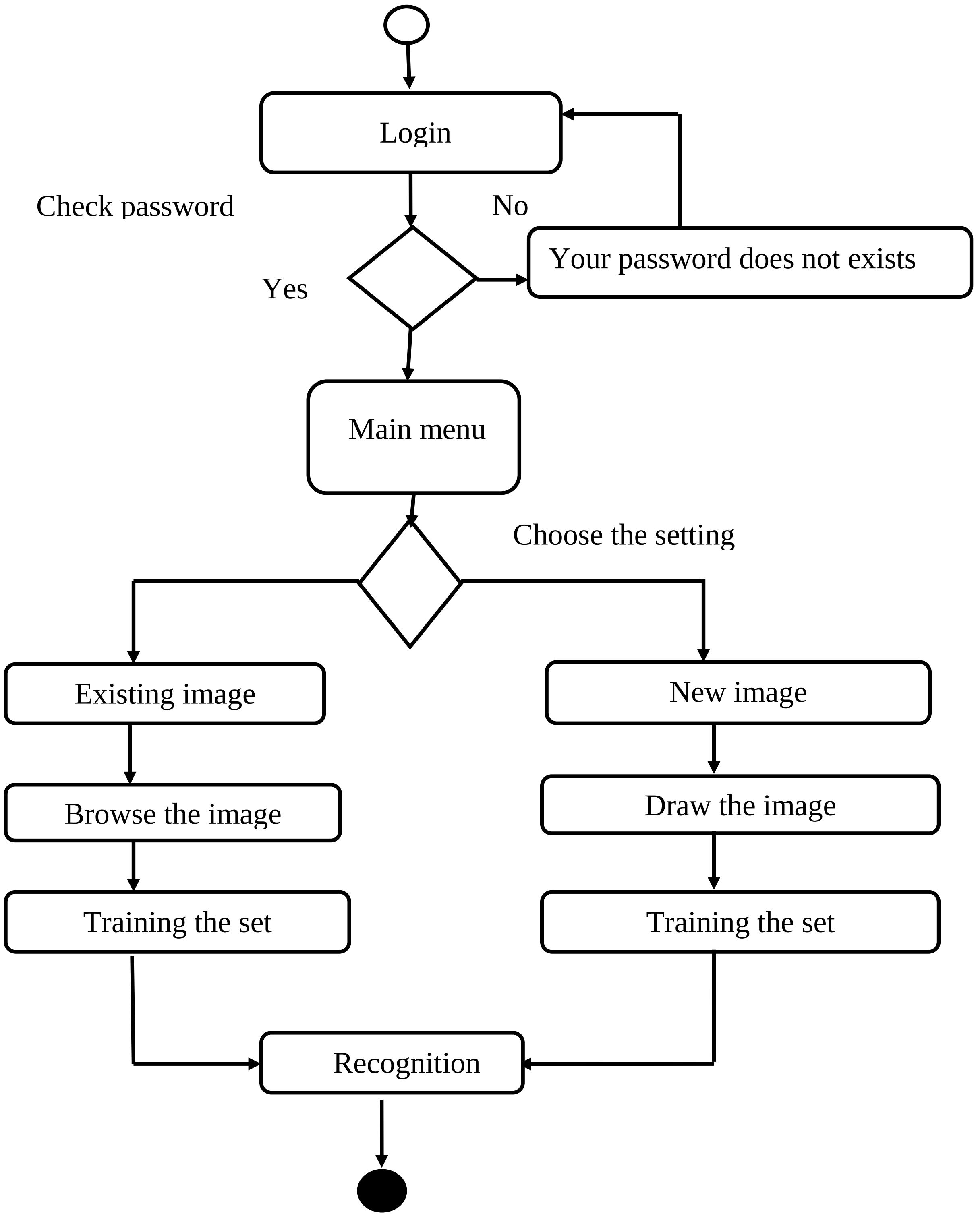

Figure 2: Activity Diagram

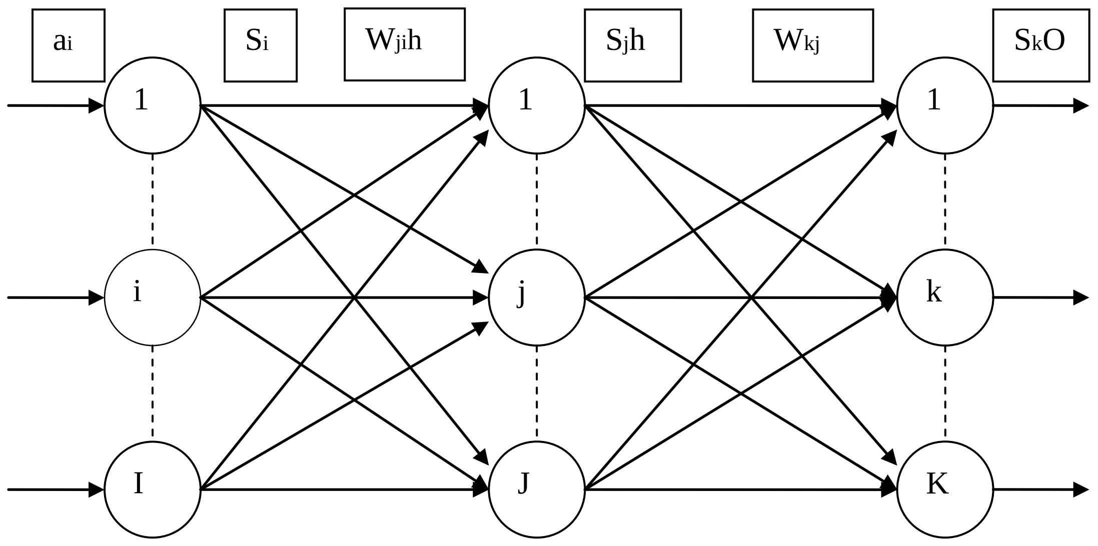

Figure 3 : Three Layer Feed Forward Neural Network

VI. TOOLS

MAIN FORM

Form Name	:	NeuralDemo
Code Behind	:	C#

6.1 Calculate the Input Unit

The total input unit can be calculated by the number of images containing the input's folder. This is done by the setting tab. From this setting tab we change the number of input layer, number of output layer and error[2]. Whenever we want to map the folder we can save the change settings and after that it can be automatically map for input units folder.

The implementation of the neural network can be done through C# class files. The main form contains the three layer network construction base class named as "BPBase.cs". The three layers can be classified into three class files. The Input layer is named as "BP1Layer.cs", Hidden layer is named as "BP2Layer.cs" and output layer is named as "BP3Layer.cs". The interface class is named as "IBackPropagation.cs".

6.2 Drawing Panel

The drawing can be initialized when the Initialize component setting intialized. The drawing panel is usercontrol and it act separately. It will displays the images when we giving the training sets and the testing samples.

Here the drawn image and filled image both must be a bitmap file format. The both type images can be processed in the buffer. Here the painting and drawing method is invoked by namespace "**System.Drawing;".**

The drawing panel includes two type of properties:

1. Drawing properties
2. Fill image properties

1. Drawing properties:

   It includes the mouse properties to draw the image by using the left mouse button. Here the muose handling function can be used draw the characters.

2. Fill image properties:

   The fill image properties can be accessed by filling the image on choosing the by browse. Here the chose image can be fill in the drawing area.

6.3. Training the network

a. **Calculate the error**: Which is defined as the square of the difference between the actual and the desired activities. Example: **error < 1%**(1.1) all the neurons was trained successfully. Modify the weights for all neurons using the error

b. **Weights adjustment:** The error between the desired output and the actual output is reduced.

c. **Reduce the error:** Change the weight of each connection

d. **Iteration:** Repeat this training process for many different images of each different images of each kind of digit until the network classifies every image correctly.

6.4 The Back-Propagation Algorithm

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW).** The back propagation algorithm is the most widely used method for determining the **EW**.

The back-propagation algorithm is easiest to understand if all the units in the network are linear[4]. The algorithm computes each **EW** by first computing the **EA**, the rate at which the error changes as the activity level of a unit is changed. For output units, the **EA** is simply the difference between the actual and the desired output. To compute the **EA** for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, we can compute in like fashion the **EA**s for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the **EA** has been computed for a unit, it is straight forward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection.

Back Propagation ANNs contain one or more layers each of which are linked to the next layer. The first layer is called "input layer" which meets the initial input (e.g. pixels from a letter) and so do the last one "output layer" which usually holds input's identifier (e.g. name of the input letter).

The layers between input and output layers are called "hidden layer(s)" which only propagates previous layer's outputs to the next layer and [back] propagates the following layer's error to the previous layer. Actually, these are the main operations of training a Back Propagation ANN which follows a few steps.

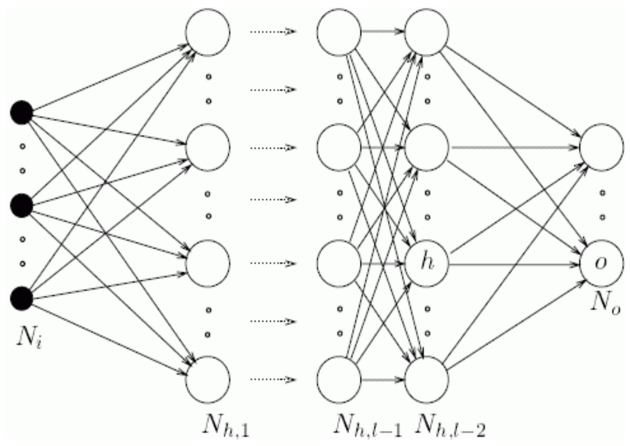

Figure 4 : Back-Propagation Algorithm

A typical Back Propagation ANN is as depicted below. The black nodes (on the leftist) are the initial inputs. Training such a network involves two phases. In the first phase, the inputs are propagated forward to compute the outputs for each output node. Then, each of these outputs is subtracted from its desired output, causing an error [an error for each output node]. In the second phase, each of these output errors is passed backward and the weights are fixed. These two phases is continued until sum of [square of output errors] reaches to an acceptable value.

Units are connected to one another. Connections correspond to the edges of the underlying directed graph. There is a real number associated with each connection, which is called the weight of the connection. We denote by Wij the weight of the connection from unit ui to unit uj. It is then convenient to represent the pattern of connectivity in the network by a weight matrix W whose elements are the weights Wij. Two types of connection are usually distinguished: excitatory and inhibitory. A positive weight represents an excitatory connection whereas a negative weight represents an inhibitory connection. The pattern of connectivity characterizes the architecture of the network.

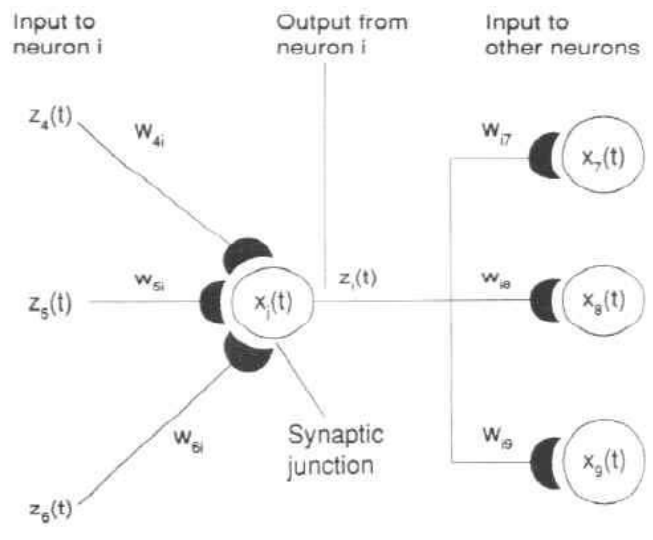

Figure 5 : Output Layer Determination

Activity by following a two step procedure.

First, it computes the total weighted input xj, using the formula:

$$X_j = \sum_i y_i W_{ij}$$

where yi is the activity level of the jth unit in the previous layer and Wij is the weight of the connection between the ith and the jth unit. Next, the unit calculates the activity yj using some function of the total weighted input. Typically we use the sigmoid function:

$$y_j = \frac{1}{1+e^{-x_j}}$$

Once the activities of all output units have been determined, the network computes the error E, which is defined by the expression:

$$E = \frac{1}{2}\sum_i (y_i - d_i)^2$$

where yj is the activity level of the jth unit in the top layer and dj is the desired output of the jth unit.

The back-propagation algorithm consists of four steps:

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$\mathrm{EA}_j = \frac{\partial E}{\partial y_j} = y_j - d_j$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$\mathrm{EI}_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{dy_j}{dx_j} = \mathrm{EA}_j y_j (1-y_j)$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$\mathrm{EW}_{ij} = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial W_{ij}} = \mathrm{EI}_j y_i$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to multilayer networks. When the activity of a unit in the previous layer changes, it affects the activites of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these seperate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit.

$$\mathrm{EA}_i = \frac{\partial E}{\partial y_i} = \sum_j \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j \mathrm{EI}_j W_{ij}$$

By using steps 2 and 4, we can convert the EAs of one layer of units into EAs for the previous layer. This procedure can be repeated to get the EAs for as many previous layers as desired. Once we know the EA of a unit, we can use steps 2 and 3 to compute the EWs on its incoming connections.

6.5 Image Processing

The image processing is done through a separate class files. The file was named as "ImageProcessing.cs". The image processing include two type of functions.

1. Converting the color image into Gray Scale type
2. Converting the Gray scale image into matrix format.

Converting the color image into Gray Scale type:

The image can be mapped into the matrix by getting row and column values. Here the bitmap image can be converted by the RGB values.

Converting the Gray scale image into matrix format:

The bmp image can be mapped into the matrix then it will be converted by changing the RGB values and the result value can stored in the temporary matrix.

6.6 Image Recognition & Displaying Output

Image recognition process is done by the serialization and deserialization process. First the training sets (input patterns) can be stored in the file with the extension of **.ann [5].** The serialization and deserialization process can be done by the temporary memory (buffer).

Binary Conversion: [Serialization]

The image can be converted into the binary format is called as serialization. This is done by using the stream class in .NET framework.

ImageConversion: [DeSerialization]

The **.net** file contains the images in the arrays. Converting the binary images from file to **.bmp** image format is used to mapping the pattern in the matrix. The process of converting binary image into object (bitmap image) is called as deserialization.

The output can be obtained after the pattern mapping. Two output units will be shown after the recognition process completed. The highly matched pattern and low matched pattern.

BROWSING THE PATTERN FOR GIVING THE NPUT TO TRAINING SETS

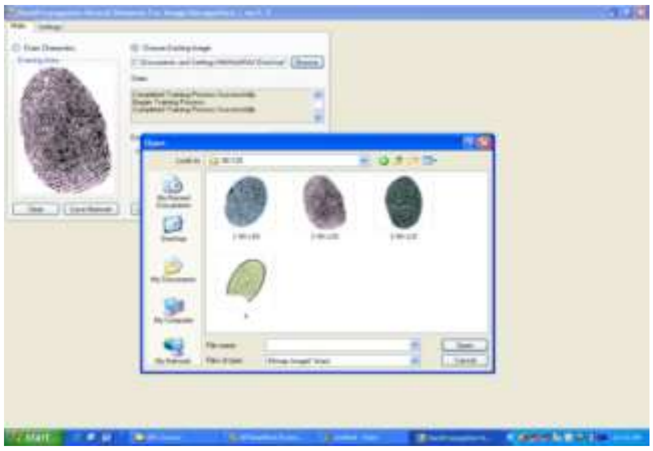

Figure 6: Browsing the pattern for giving the input to training sets

Here the user is browsing the necessary folders to select the pattern to be given as the input for the neurons after selecting the Error rate, neuron layers and saving the settings in the settings form for recognition the exact output.

LOAD THE TRAINED NETWORK TO RECOGNIZE THE IMAGE

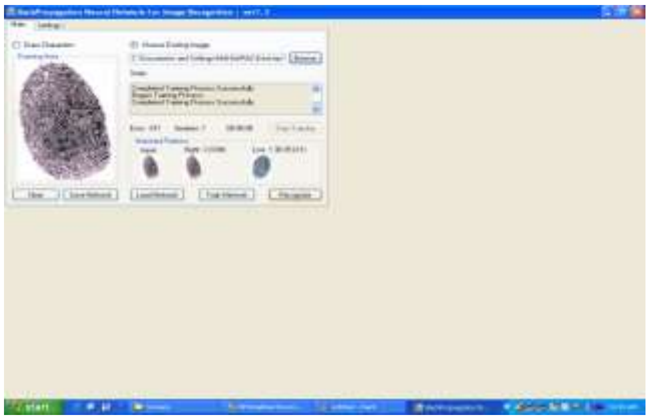

Figure 7 :Load the trained network to recognize the image

With this form the user is loading the trained network to recognize the image. This form shows us how the user trained the input pattern with the saved network for recognition the input pattern and displays the exact output.

VII. CONCLUSION

The network has been trained and tested for a number of widely used font type in the Latin alphabet. Since the implementation of the software is open and the program code is scalable, the inclusion of more number of fonts from any typed language alphabet is straight forward.

The necessary steps are preparing the sequence of input symbol images in a single image file (*.bmp [bitmap] extension), typing the corresponding characters in a text file (*.cts [character trainer set] extension) and saving the two in the same folder (both must have the same file name except for their extensions). The application will provide a file opener dialog for the user to locate the *.cts text file and will load the corresponding image file by itself.

Although the results listed in the subsequent tables are from a training/testing process of symbol images created with a 72pt. font size the use of any other size is also straight forward by preparing the input/desired output set as explained. The application can be operated with symbol images as small as 20pt font size.

Increasing the number of iterations has generally a positive proportionality relation to the performance of the network. However in certain cases further increasing the number of epochs has an adverse effect of introducing more number of wrong recognitions.

The design of neural network structure and selection of training are very important factors affecting the recognition results. Using AE signals produced by as unitary as possible mechanism to train the network, the pattern recognition results will be better. Designed BP neural network can successfully distinguish the nature of field PV AE sources with the characteristic parameters of AE signals.

VIII. FUTURE ENHANCEMENT

Medical Diagnosis - Assisting doctors with their diagnosis by analyzing the reported symptoms.

Input data: patient's personal information, patterns of symptoms, heart rate, blood pressure, temperature, laboratory results, etc. The major problem in medical field is to diagnose disease. Human being always make mistake and because of their limitation, diagnosis would give the major issue of human expertise. One of the most important problems of medical diagnosis, in general, is the subjectivity of the specialist. It can be noted, in particular in pattern recognition activities, that the experience of the professional is closely related to the final diagnosis.

Example: **Diagnose Breast Cancer**

Bibliography :

[1] J. K. Basu, D. Bhattacharyya, and T. Kim, “Use of Artificial Neural Network in Pattern Recognition,” vol. 4, no. 2, pp. 23–34, 2010.

[2] T. Chen, “Image Binarization By Back Propagation Algorithm,” pp. 345–349.

[3] A. K. Gupta and Y. P. Singh, “Analysis of Back Propagation of Neural Network Method in the String Recognition,” vol. 3, no. 5, pp. 5–8, 2011.

[4] A. U. Muhammad, A. G. Musa, and K. I. Yarima, “Survey on Training Neural Networks,” vol. 5, no. 3, pp. 169–173, 2015.

[5] M. E. Scholar, “A SURVEY ON BACKPROPAGATION ALGORITHM FOR NEURAL,” vol. 2, no. 7, pp. 729–733, 2015.

[6] K. Vora and S. Yagnik, “A Survey on Backpropagation Algorithms for Feedforward Neural Networks,” pp. 193–197.

[7] S. B. Wankhede, “Analytical Study of Neural Network Techniques : SOM , MLP and Classifier-A Survey,” vol. 16, no. 3, pp. 86–92, 2014.

[8] Bishop, C.M. "Neural Networks for Pattern Recognition", Oxford University., 1995

[9] Gloger.J.M, Kaltenmeier.A, Mandler.E, Andrews.L , " Reject Management in Handwriting Recognition System",ICDAR'97

[10] Le Dung, Makoto Mizukawa, "Automatic handwritten postcode reading system using image processing and neural network technology", System Integration Division Conference ,pp 930-931,2006

[11] Martin Hegan, Howard Demuth, Mark Beale "Neural Network Design",Thomson Learning, 2006

[12] Patrice Simard.Y, Dave Steinkraus, John Platt, "Best Practices for Convolution Neural Networks Applied to Visual Document Analysis", ICDAR, IEEE Computer Society,Los Alamitos, pp.958-962