Distortion Analysis Of Tamil Language Characters Recognition

Gowri.N [1], **R. Bhaskaran[2],**

1. T.B.A.K. College for Women, Kilakarai,

**2. School Of Mathematics, Madurai Kamaraj University,
Madurai.**

ABSTRACT

This research work demonstrates how character recognition can be done with a back propagation network and shows how to implement this using the MATLAB Neural Network toolbox. This is a slightly enhanced version of the character recognition application based on the MATLAB Neural Network toolbox. In this research article we are focusing on the distortion analysis of Tamil Language Characters in order to recognize them effectively using the neural network we have developed. We have used the commonly used representation method for recognizing digits and uppercase English letters as suggested by Guyon et.al [1] to start with and built a method for Tamil language letters over it.

KEYWORDS: *Neural Network, Natural Language Characters recognition , character shape representation, character classification, Pattern Recognition, Tamil Language Characters, Tamil Desktop Publishing, Feed forward network, Distortion*

1. Introduction

The word "recognition" plays an important role in our lives and understanding actually begins with proper recognition. It is a basic property of all human beings; when a person sees an object, he or she first attempts to gather all possible information about the object and compare its properties and behaviors with the existing knowledge stored in the mind. If one finds a proper match, one recognizes it. If no match is found, it is recognized as a new object. However, when pattern recognition has to be employed for identification of fixed and known set of patterns in a commercial or research environment, it is more appropriate to have an automated way. A machine that reads checks in a bank could do the job several times faster than human beings. In effect, machines that can read symbols could provide cost effective alternative to manual methods. This kind of application saves time and money, less error prone, and relieves human labor to work on other interesting jobs rather than doing a mechanical boring job.

A neural network is an information processing system. It consists of massive simple processing units with a high degree of interconnection between each unit. The processing units work cooperatively with each other and achieve massive parallel distributed processing. The design and function of neural networks simulate some functionality of biological brains and neural systems. The advantages of neural networks are their adaptive-learning, self-organization and fault-tolerance capabilities. For these outstanding capabilities, neural networks are used for pattern recognition applications. Some of the best known neural models are back-propagation, high-order nets, time-delay neural networks and recurrent nets.

2. Design of the Neural Network

The twenty-six by thirty five-element input vectors are defined in the MATLAB script file as a matrix of input vectors called alphabet. The target vectors are also defined in this file with variable called targets. Each target vector is a 26-element vector with a 1 in the position of the letter it represents, and 0’s in other positions. For example, the letter “C” is to be represented by a 1 in the third element (as “C” is the third letter of the alphabet), and 0’s in other positions. The network receives the 5×7 real values as a 35-element input vector. It is then required to identify the letter by responding with a 26-element output vector. The 26 elements of the output vector each represent a letter. To operate correctly, the network should respond with a 1 in the position of the letter being presented to the network. All other values in the output vector should be 0. In addition, the network should be able to handle noise. In practice, the network does not receive a perfect letter as input. Specifically, the network should make as few mistakes as possible when classifying vectors with noise of mean 0 and standard deviation of 0.2 or less. We have used the commonly used representation method for recognizing digits and uppercase English letters as suggested by Guyon et.al [1]

3. Network Architecture

Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network produce values outside the range –1 to +1.

The network is a two-layer network. The neural network needs 35 inputs and 26 neurons in its output layer to identify the letters. The ***log-sigmoid*** transfer function at the output layer was picked because its output range (0 to 1) is perfect for learning to output Boolean values. The hidden layer has 10 neurons. This number was picked by through several tests by changing the number of neurons and also the number of epochs and by changing the sum squared error to enhance learning.

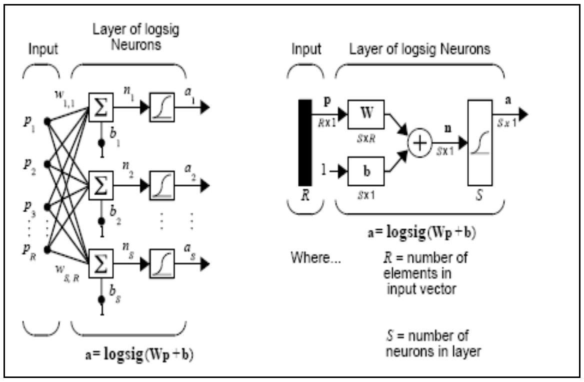

Fig1: Architecture of the network

The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. However, noisy input vectors may result in the network not creating perfect 1's and 0's. After the training, the output from the network is passed through the competitive transfer function ***compet***. This makes sure that the output corresponding to the letter most like the noisy input vector takes on a value of 1, and all others have a value of 0. The result of this post-processing is the output that is actually used.

3.1 Initialization

The two-layer network is created with newff with logsig function and by creating a set of target and input vectors.

3.2 Network Training

In order to create a network that can handle noisy input vectors it is imperative to train the network on both ideal and noisy vectors. To do this, the network is first trained on ideal vectors until it has a low sum-squared error. Then, the network is trained on 10 sets of ideal and noisy vectors each. The network is trained on two copies of the noise-free alphabet at the same time as it is trained on each noisy vector. The two copies of the noise-free alphabet are used to maintain the network's ability to classify ideal input vectors.

However after, the training described above the network may have learned to classify some difficult noisy vectors at the expense of properly classifying a noise-free vector. Therefore, the network is again trained on just ideal vectors. This ensures that the network responds perfectly when presented with an ideal letter. All training is done using back propagation with both adaptive learning rate and momentum with the function ***traingdx.***

3.3 Training without Noise

The network is initially trained without noise for a maximum of 5000 epochs or until the network sum-squared error falls beneath 0.1.

3.4 Training with Noise

To obtain a network not sensitive to noise, we trained with two ideal copies and two noisy copies of the vectors in alphabet. The target vectors consist of four copies of the vectors in target. The noisy vectors have noise of Standard Deviation 0.1 and 0.2 added to them. This forces the neuron to learn how to properly identify noisy letters, while requiring that it can still respond well to ideal vectors. To train with noise, the maximum number of epochs is reduced to 300 and the error goal is increased to 0.6, reflecting that higher error is expected because more vectors (including some with noise), are being presented.

3.5 Training Again Without Noise

Once the network is trained with noise, it makes for good judgment to train it without noise once more to ensure that the system goes back to its original state and ideal input vectors are always classified correctly. Therefore, the network is again trained with code identical to the Training without Noise section.

3.6 Estimating the System Performance

The reliability of the neural network pattern recognition system is measured by testing the network with hundreds of input vectors with varying quantities of noise. The network is tested with a noisy version of the Tamil Language Characters. The MATLAB script file tests the network at various noise levels, and then graphs the percentage of network errors versus noise. Noise with a mean of 0 and a standard deviation from 0 to 0.5 is added to input vectors. At each noise level, 100 presentations of different noisy versions of each letter are made and the network's output is calculated. The output is then passed through the competitive transfer function so that only one of the 26 outputs (representing the letters of the alphabet), has a value of 1. The number of erroneous classifications is then added and percentages are obtained. The solid line on the graph of Fig 6 and Fig 7 shows the reliability for the network trained with and without noise. The reliability of the same network when it had only been trained without noise is shown with a dashed line. Thus, training the network on noisy input vectors greatly reduces its errors when it has to classify noisy vectors. The network did not make any errors for vectors with noise of standard deviation 0.0 or 0.05. When noise of standard deviation 0.2 was added to the vectors both networks began making errors.

For example we create a noisy version (Standard Deviation 0.2) of the Tamil Language Characters. The Network is trained with distortions and a testing of networks with noise level of 0.00 to 0.50 in steps of 0.05 is done for all the input characters of the input character given. The Fig 2 and Fig 4 are the result of distorted characters of Tamil language with a maximum sum squared error output of 0.5. Fig 3 and Fig 5 is the result with a maximum distortion of 0.5 given as a test input and the character recognized by the network.

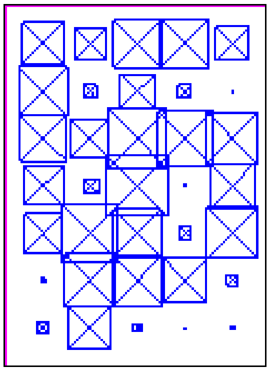

Fig 2: Tamil letter 'tha' distorted

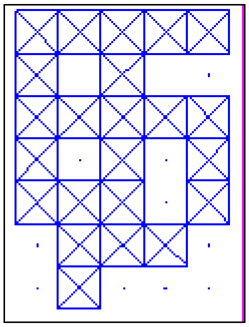

Fig 3: Tamil letter 'tha' recognized

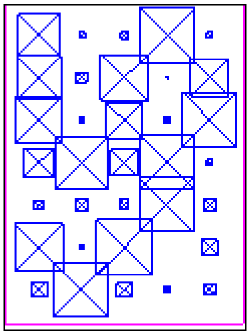

Fig 4.Tamil letter 'zha' distorted

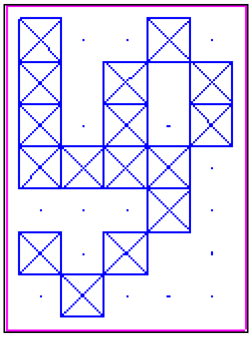

Fig 5.Tamil letter ‘zha’recognized

The graphs in Fig 7 and Fig 8 show the comparison of percentage of recognition errors of training the network with and without distortion respectively and for varying values of number of epochs.

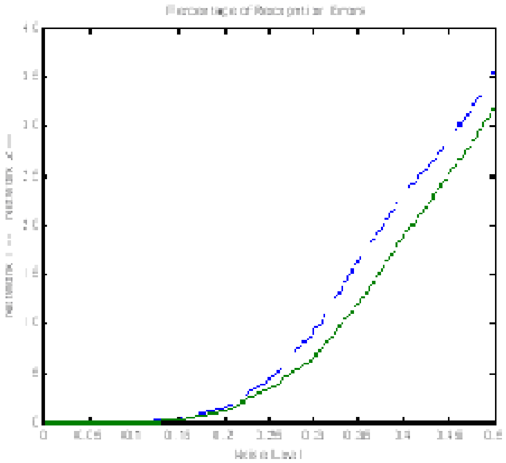

Fig 6: Percentage of recognition errors

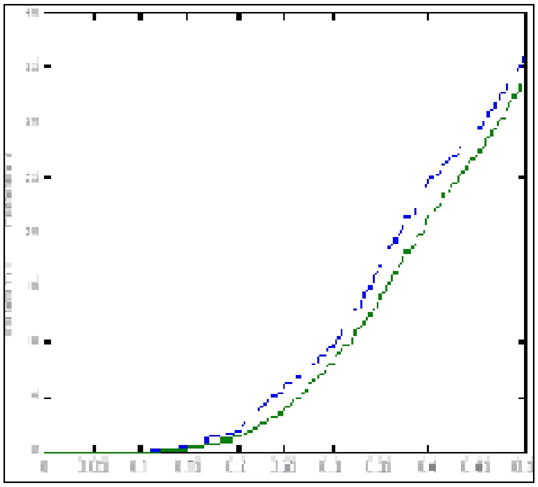

Fig 7: Percentage of recognition errors

If a higher accuracy is needed, the network can be trained for a longer time or retrained with more neurons in its hidden layer. Also, the resolution of the input vectors can be increased to a 10-by-14 grid. Finally, the network could be trained on input vectors with greater amounts of noise if greater reliability were needed for higher levels of noise.

The diagrams and the graphs demonstrate how a simple pattern recognition system could be designed. Note that the training process did not consist of a single call to a training function. Instead, the network was trained several times on various input vectors. In this case, training a network on different sets of noisy vectors forced the network to learn how to deal with noise, a common problem in the real world.

Normally, only feed-forward networks are used for pattern recognition. Feed-forward means that there is no feedback to the input. Similar to the way that human beings learn from mistakes, neural networks also could learn from their mistakes by giving feedback to the input patterns. This kind of feedback would be used to refine and reconstruct the input patterns and make them free from error; thus increasing the performance of the neural networks. Of course, it is very complex to construct such types of neural networks. These kinds of networks are called as auto associative neural networks. As the name implies, they use back-propagation algorithms.

4. Conclusion

Pattern recognition could be achieved using both normal computing techniques and neural networks. In normal computing conventional arithmetic algorithms are used to detect whether the given pattern matches a known one. It is a straightforward method. It will say either yes or no. It does not tolerate noisy patterns. On the other hand, neural networks can tolerate noise and, if trained properly, will respond correctly for noisy patterns. Neural networks may not perform miracles, but if constructed with the proper architecture and trained correctly with good data, they give amazing results in pattern recognition.

References :

[1] I.Guyon.P.Albrecht,Y.Le.Cn, J. Denker, and W.Hubbard. *Design of neural network character recognition for a touch terminal, Pattern Recognition, 1991.*

[2] Gerald Wheatley- *Applied Numerical Analysis*

[3] A. Mitiche and J. Aggarwal. *Pattern category assignment by neural networks and the nearest neighbors rule.* International Journal on Pattern Recognition and Artificial Intelligence,10 :393–408, 1996.

[4] R. Plamondon. *A model-based segmentation framework for computer processing of handwriting*. Proceedings of the Eleventh IAPR International Conference on Pattern Recognition, pages 303–307, 1992

[5] K. Bhattacharyya and K. K. Sarma, *"Innovative Segmentation of Handwritten Text in Assamese Using Neural Network"*, The 2009 International Conference on Genetic and Evolutionary methods (GEM'09), Las Vegas, July 2009.

[6] Z. Chen, *"Handwritten Digits Recognition"*, The 2009 International Conference on Genetic and Evolutionary Methods (GEM'09), Las Vegas, July 2009.

[7] Tamil Standard Code for Information Interchange, http://www.tamil.net/tscii

[8] S. Mori, H. Nishida, H. Yamada, *"Optical Character Recognition"*, John Wiley & Sons, 1999.

[9] R. Gonzalez, R. Woods, and S. Eddins, "*Digital Image Processing UsingMatlab"*, Prentice Hall, 2004.

Gowri N. completed M.Sc. in Physics and obtained M.Phil. in Physics from Alagappa University, Karaikudi for her thesis on "Simulation Studies in Crystal Growth". She is currently employed in TBAK College for Women in Kilakarai as Associate Professor in Computer Science Dept. and doing her Ph.D. in Pattern Recognition in School of Mathematics, Madurai Kamaraj University, Madurai.

Dr. R. Bhaskaran is Chariperson and Head of Department, School of Mathematics, Madurai Kamaraj University, Madurai.