Image Recognition With the Help of Auto-Associative Neural Network

Moumi Pandit moumi_pandit@yahoo.co.in
Department of Electrical and Electronics Engineering
Sikkim Manipal Institute of Technology
Majhitar,Rangpo,East Sikkim-737132,India

Mousumi Gupta mousmi_gt@yahoo.co.in
Department of computer science Engineering
Sikkim Manipal Institute of Technology
Majhitar,Rangpo,East Sikkim-737132,India

Abstract

This paper proposes a Neural Network model that has been utilized for image recognition. The main issue of Neural Network model here is to train the system for image recognition. In this paper the NN model has been prepared in MATLAB platform. The NN model uses Auto-Associative memory for training. The model reads the image in the form of a matrix, evaluates the weight matrix associated with the image. After training process is done, whenever the image is provided to the system the model recognizes it appropriately. The weight matrix evaluated here is used for image pattern matching. It is noticed that the model developed is accurate enough to recognize the image even if the image is distorted or some portion/ data is missing from the image. This model eliminates the long time consuming process of image recognition.

Keywords: Image Recognition, Artificial Neural Network, Pattern Matching, Associative Memory, Weight Matrix.

1. INTRODUCTION

Neural network as the name suggests is interconnection of nerve cells. The nervous system in human brain is highly interconnected neural cells which makes the nervous system. The Artificial Neural Network technology is built up with a inspiration of functioning of human nervous system. Many of human intelligent behavior which is the direct functioning of human nervous system are implemented artificially by using Artificial Neural Network. Artificial neural network is an information processing devices, which are built from interconnected elementary processing elements called neurons. It is inspired by the way biological nervous systems works.

ANN is composed of a large number of highly interconnected processing elements working in union to solve specific problems. Like human, artificial neural network also learn by example. ANN is configured for specific application, such as pattern recognition or data classification through learning process. Learning involves adjustments to the synaptic connection known as weights that exist between the neurons [8] [9].

In artificial neural network, the information processing elements are known as nodes. There are two or more layers of nodes. The first layers of nodes are known as input layer whereas the last layer is known as output layer. The layers in between which may or may not exist is known as hidden layer(s).The information is transmitted by means of connecting links. These links possess an associated weight, which is multiplied with the incoming signal for any typical neural network. The output is obtained by applying activations to the net [8] [9].

Image recognition is a key component in application areas like biometric identification. Image recognition is also one of the important functions relevant to image processing of brain in addition to image segmentation and associative memory. For this reason, many studies on the image recognition can be found in computer vision and computational intelligence, e.g., see [8], [7]. It is not a surprising to see that actually the human identifying methods by possessions (cards, badges, keys) or by knowledge (userid, password, Personal Identification Number (PIN)), are being replaced by biometrics (fingerprint, face, ear). A human being has the capacity to memorize a pattern and can also recall it.

It is well-known that neural network is effective for classification problems. Some studies that a neural network was applied to Braille recognition have been reported. In [9], Braille documents read by tactile were considered as time series information, and a new SOM architecture with feedback connections was proposed. Wang et al. have introduced the coupled neural oscillator networks to model synchronous oscillatory firings and applied it to image segmentation [3]

The hierarchical neural network with back-propagation method is widely used as network model. However, it requires a lot of time for learning. Furthermore, modifying, adding and deleting memory patterns are not easy. Here Auto Associative Neural Network has been used because the training time is comparatively lower than previous mentioned methods.

2. ARCHITECTURE OF AUTO-ASSOCIATIVE NEURAL NETWORK

An Auto-Associative Neural Network is basically feed forward multilayered neural network which has same number of nodes in the input layer and the output layer. The output layer is actually the computing layer.

The architecture of Auto-Associative Neural Network is given in FIGURE 1. The inputs are given in matrix[x] ,the output is given in matrix [y] and the associated weights are in matrix [w].

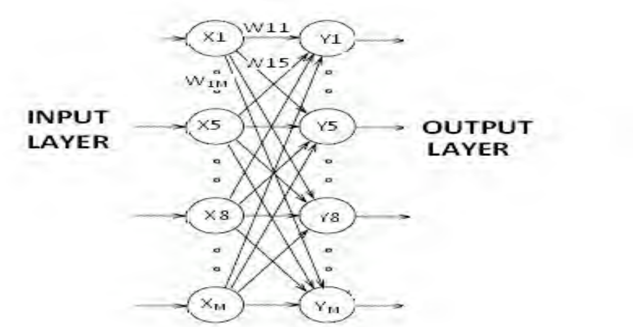

FIGURE1: Architecture of auto-associative network

The output of the matrix[y] is can be represented as:

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \qquad \text{..............(i)}$$

$$\begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix}^t \qquad \text{.............(ii)}$$

When two vectors are processed with outer products the result is a matrix. If one of the two vectors u is having m rows and the other vector v is having n column. Their outer product is defined by a matrix whose dimention is m*n and is defined as u * v^t. The resulting matrix can map the vectors. Applying this rule to the auto associative memory we have:

$$[w(k)] = [x_k] *[y_k^t]$$

The resulting matrix obtained by finding the outer product of vector [Y_k] and[X_k] should be considered associative memory. If in the considered associative memory Neural Network we assume [X_k] = [Y_K] hence the weight matrix becomes [x_k]*[x_k^t].

Let us consider two examples:

Exampe1:

A vector [X] = [1 1 -1 -1] is given as input to the Auto-Associative network. The weight matrix w can then be calculated as :

[w]=[x]*$[y]^t$, where y is the output. As already mentioned that in auto associative net the output and the input is same so [x]=[y],therefore the weight matrix becomes

$$[w]=[x]*[x]^t= [1\ 1\ -1 -1] \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

$$[Y] =[W]*[X]= \begin{bmatrix} 4 \\ 4 \\ -4 \\ -4 \end{bmatrix}$$

The output is passed through activation function. Here the condition is if the input is 1 or above then then the output will be 1.if it is -1 or below then the output will be -1.Passing the output through activation function the output becomes $\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$ which is same as the given vector[x].

Example2:

Now, if the input matrix is changed slightly and it is taken as [0 1 -1 -1].then the output will be :

$$[Y] = [W]*[X]= \begin{bmatrix} 3 \\ 3 \\ -3 \\ -3 \end{bmatrix}$$

The output is passed through activation function. Here the condition is if the input is 1 or above then then the output will be 1.if it is -1 or below then the output will be -

Passing the output through activation function the output becomes $\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$.

The same example can be implemented in real life application as in case of image recognition where the image can be read as a matrix with the help of MATLAB. Even if the image is slightly distorted or some data gets lost then also we can recognize the image provided the distortion do not change the image completely. Firstly the computer is trained with a particular image. The weight matrix is found out. Secondly, the net is tested to check whether it is working properly.

3. DEVELOPMENT OF THE NETWORK

3.1 Training the Network

An Auto-Associative neural network has a very simple architecture. It has an input layer and an output layer. The input layer is linked with the output layer with associated weights. In this process the input and the output is the same that is, the same image is used as input and output. The network is thus trained with an image and associated weight is found out as in (ii) i.e.

When the output matrix is [y], the input matrix is [x] and the weight matrix is [w] then:

$$\begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix}^t \qquad \text{..............(ii)}$$

Where the input matrix is the image read as matrix[x], the output matrix is the same image read as matrix [y] and the weight matrix[w] is calculated by using (ii). As the input and output is the same matrix of the same image we can say [y]=[x],so we can rewrite (ii) as:

$$\begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} = \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix}^t \qquad \text{..............(iii)}$$

3.2 Testing the Network

Case 1: Testing with same image

For testing, first the same image is given to the network as input. The image is read as matrix. The weight matrix is same as calculated during training the network by equation(iii).The output [y] was calculated with (i) i.e.:

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix}$$

Now the output matrix[y] is passed through activation function and then compared with input matrix [x]. As the input and output matrix is same the image is declared to be "**same**".

Case 2: Testing with different image:

A different image is given as input. The weight matrix is same as calculated during training the network by equation (iii).Now the output is calculated as in (i)

$$\begin{bmatrix} y_{k1} \\ y_{k2} \\ \vdots \\ y_{km} \end{bmatrix} = \begin{bmatrix} w_{i1} & w_{i2} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{mm} \end{bmatrix} \begin{bmatrix} x_{k1} \\ x_{k2} \\ \vdots \\ x_{km} \end{bmatrix}$$

Now the output matrix[y] is passed through threshold function and the compared to the input matrix[x]. As the input and output matrix is not same the image is declared to be" **not same**".

Case 3: With distorted form of same image

Though the brightness of the image is changed but the image is same. So the matrix [x] will be same. The weight matrix is same as calculated during training the network by equation(iii). Then again the output matrix [y] is calculated using equation (i) as done in earlier two cases. The output [y] is then passed through activation function and then compared with input matrix [x]. As they are same the image is declared to be "**same**".

Case 4: With same image but with some data missing:

Now in this case the same image is given but with slight change ie some data is missing. This case is same given in example 2. Here the input image is same as the trained one but with some data missing. The matrix is calculated as in equation (i). Though the output is slightly different from the input ,but after passing through the activation it becomes the same. So ,the image is again declared **"same"**.

The main advantages of this process**:**

i) The process is very simple

ii) Do not require any specific complicated software or hardware.

iii) Computational time is very less.

iv) Image can be recognized even if it is distorted.

v) Cost effective.

4. ALGORITHM

The algorithm for training the network and testing it is given below.

4.1 Training the Net

Step1: The image [y] is read in the form of square matrix [x] [mXm].

Step2: The image is changed to grayscale if it is in RGB format.

Step3: The matrix may be reduced to suitable size for quicker result.

Step4: The Gray scale image has been converted to binary 0,1 image (B1) by using a user defined threshold(t) parameter. The Gray value >= t is converted to 1 in original image. The gray value <t converted to 0.

Step5: The weight matrix is calculated as $[w] = [x] X [x]^t$ ………. (A)

4.2 Testing the Output

Step1: Various images are taken as input in form of matrix [x] and changed it to grayscale. The matrix (B2) is then changed in terms of 0 and 1 using the previous threshold function as stated above. The matrix may be reduced to the size as given in training process i,e m X m matrix for quicker result.

Step2: Weight matrix [w] as evaluated by (A) is provided to the network.

Step3: Output is calculated as:

[Y]=[W]*[X] where [Y] is the output matrix.

Step4: The output is passed through the activation function by using equ (B) and output image is converted to 0 or 1.

Step5: For i=1 to m

For j=1 to m

if matrix B1== matrix B2

Then display the Y is matched

Otherwise Y not matched.

5. RESULT AND ANALYSIS

In the proposed method, auto-associative network is used for image recognition. The entire work is done in two parts: Firstly the network is trained with a particular image. The weight matrix is calculated which is acting as the mapping function. Once the network is trained, it can be used for matching the matrices for image recognition. In our work we have used gray scale images which have later been reconstructed as a square matrix. The whole process was done in MATLAB platform.

Here the network is trained with a gray scale image of a graph which is in JPEG. This validity of the process is tested with four different kind of images and has proved to be successful. The method is successful in recognizing the image even if the brightness of the image is reduced by 80%. Moreover the method can successfully recognize the image even if 10% data of the trained image is missing.

The network can be trained with various kinds of images whether it is in grayscale or it is a color image. The performance of the system is invariant to the size and brightness of the image. By using this method the network tests the identity of an image in 0.25 seconds in a Pentium machine at 500 MHz, and gives an average recognition rate of 99%. This technique is computationally inexpensive and involves only three parameters that are input, output and a weight matrix.

The training of the network is done by the following image;

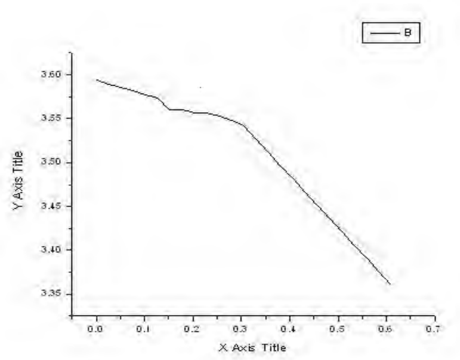

FIGURE2: Image with which the network is trained

The image was read as 5*5 matrix in term of 0 and 1. The weight matrix of this particular image was found to be:

$$\begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

This weight matrix will be used as mapping parameter for image recognition. The output is tested based on four kinds of data as discussed in the following four cases:

Data 1: With the same image

Data 2: With different image

Data 3: With same image but distorted

Data 4: With the same image but some data missing

5.1 Case1- With the Same Image

The same image with which the network was trained is given as input as given in FIGURE 3. The weight matrix which was derived after training the network is also given. It was seen that with the same image the output given was "**same**".

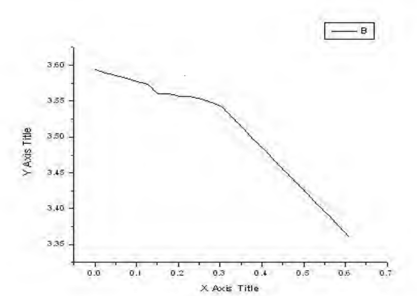

FIGURE3: The same image with which the training was done

5.2 Case 2-With Different Image

Now a different image image as given in FIGURE4 trained is given to the network as input. The weight matrix which was derived after training the network is also given. It was seen that with the different image the output given was "**not same**"

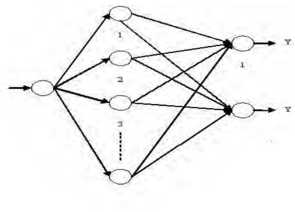

FIGURE4. A different image

5.3 Case 3- With Same Image Dut Distorted

The same image with which the network was trained is given as input with lot of distortion i.e the brightness of the has been changed completely as shown in FIGURE 5. The weight matrix which was derived after training the network is also given. It was seen that with the distorted form of the same image the output given was “match”.

FIGURE 5: The distorted form of same image with which the training was done

5.4 Case 4 - With the Same Image But Some Data Missing

The same image with which the network was trained is given as input with some data missing as shown in FIGURE 6. The weight matrix which was derived after training the network is also given. It was seen that with the distorted form of the same image the output given was “match”.

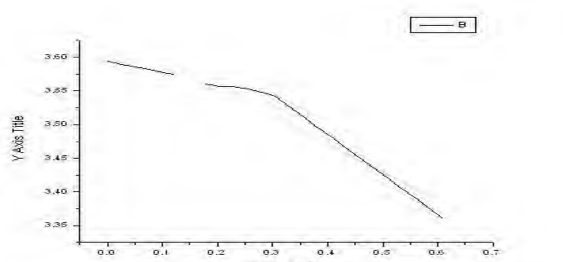

FIGURE 6: The same image with which the training was done but with some data missing

The whole process was done in MATLAB platform. This process is tested with numerous images and has proved to be successful. The proposed method takes less than 0.5 secs for recognizing. From experimental results we have to

5.5 Comparison With Others Algorithm

The authors [5] [9] uses different parameters to get optimum recognition rate, the parameters they have used were distance measurement technique to find distance between two points for face recognition. The calculation for different parameters increases the computational time and its complexity.

The strength in our proposed methods is that we are using very few parameters to train the system. Therefore we can conclude that the proposed method is able to train the image and gives output within very short memory space requirement, and the network can be trained with even one image.

6. CONCLUSION AND FUTURE WORK

In this paper an image recognition model for pattern matching has been proposed. The weight matrix is the correlation matrix and the advantage of correlation matrix is the ease of storing new association or deleting odd ones. The image is trained with the proposed auto associative memory architecture. The advantage of this model is training time is very less and this model recognize the image even if the original image contains less information.

The limitation of this model is that it can store only m-1 images for recognition where m is the number of elements in the input matrix. To improve the performance, input patterns can be selected orthogonal among themselves [8] [9]. Storing of more than one image can be done by simply updating the current weight matrix by adding the weight matrix of the new image [9]. The future scope of this project is to overcome this limitation.

7. REFERENCES

1. **FOR JOURNALS:** M. Burl, T. Leung, and P. Persona. "*Recognition of Planar Object Classes*". In Proc. IEEE Comput. Soc. Con! Computer. Vision and Pattern Recognition., 223-230 ,1996.

2. **FOR JOURNALS** : H. Waukee, H. Harada, *et al.* "*An Architecture of Self-Organizing Map for Temporal Processing and Its Application to a Braille Recognition Task,*" *IEICE Trans. on Information and Systems*, J87(3), 884–892, 2004.

3. **FOR JOURNALS** : Wang, D. L., & Terman, D. "*Locally excitatory globally inhibitory oscillator networks.*" IEEE Trans. Neural Networks, 6, 283-286 1995.

4. **FOR JOURNALS** : José R. Dorronsoro, Vicente López, Carlos Santa Cruz, and Juan A. Sigüenza, "*Autoassociative Neural Networks and Noise Filtering"* IEEE transactions on signal processing, 51(5) , 2003

5. **FOR JOURNALS***: S. Palanivel, B.S. Venkatesh and B. Yegnanarayana,* **R**eal time face recognition system using autoassociative Neural Network models, ICASSP 2003.

6. **FOR JOURNALS:** S. Amari "*Neural Theory of association and concept formation*" Biological cybernetics, 26, 175-185,1977.

7. **FOR CONFERENCES** : Csurka, G., Dance, c., Bray, c., and Fan, L., "*Visual categorization with bags of key points,*" In Proceedings Workshop on Statistical Learning in Computer Vision, I -22, 2004.

8. **FOR BOOKS**: Simon Haykin, "*Neural Networks A Comprehensive Foundation*", Pearson Educartion (Singapore) Pvt. Ltd. pp. 1-49 (2004).

9. **FOR BOOKS** : S.N. Sivanandam, S. Sumathi, S.N. Deepa, "*Introduction to Neural Networks using Matlab 6.0*", Tata McGraw-Hill pp. 10-29, pp 109-165(2006).