International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Use of Auto associative Network for signature recognition

Ashutosh Mukherjee, Khushi Priya, Moumi Pandit* and Dibyadeep Bhattacharya

EE Dept, SMU, Sikkim, India

Accepted 05 March 2017, Available online 09 March 2017, **Vol.7, No.2 (April 2017)**

Abstract

A Neural Network model has been designed and proposed for signature recognition using Auto-Associative memory for training. The model reads the images of the signatures individually in the form of matrices and evaluates the weight matrices associated with each image. After training process is complete, whenever any of the signatures is provided to the system, the model recognizes it appropriately and displays it with the help of arduino. This model eliminates the long time consuming process of signature recognition.

Keywords: *Signature Recognition, Associative Memory, Weight Matrix*

1. Introduction

With the growing number of reported forgery cases, proper personal identification and authentication of signatures has become an issue of concern and an interesting topic of research. Various methods of authentication include retina, face, voice, fingerprint and various other biometrics. There are many distinct advantages of using biometrics as an authentication techniques compared to the traditional methods which includes PIN numbers, passwords, etc. Biometrics are unique to a person and is not lost or forgotten and moreover biometric cannot be easily replicated [Malik.V, Arora. A, 2015]. Signature is one such biometrics. Literature survey reveals various techniques that are already implemented in authentication process of signature. In paper [Kiani, V, 2009], the authors have used cluster technique for offline signature verification. They have implemented feature extraction from small blocks of signature based on vertical and horizontal segmentation. In another paper [Biswas, S, *et al*, 2010], the authors have created separate clusters of sample signatures for each person and then implemented K-Nearest Neighbours' (KNN) technique to test the authenticity of the signature. An algorithm based on some statistical methods has been proposed by the authors in paper [Bhattacharyya D, *et al*,] where the testing data samples are compared with the trained data based on the deviation.

In this proposed work, a neural network model is developed based on auto-associative memory. The network is trained with a set of images of signatures by assigning individual weight. Then the net weight matrix is calculated and stored along with the images. When the testing sample is presented, the output is calculated and after passing it through an activation function it is matched with the matrix of the stored images.

This paper is organized in five sections. Section 2 discusses about the architecture of auto associative network, the methodology of the proposed work is presented in section 3 while section 4 includes the results and discussion followed by a brief conclusion in section 5

2. Architecture of the Auto-associative 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 (Kramer, M. A. 1992 and Pandit, M., *et al* 2011). The output layer is actually the computing layer.

The architecture of Auto-Associative Neural Network is given in **Fig.3**. The inputs are given in matrix [X] ,the output is given in matrix [Y] and the associated weights are in matrix [W].

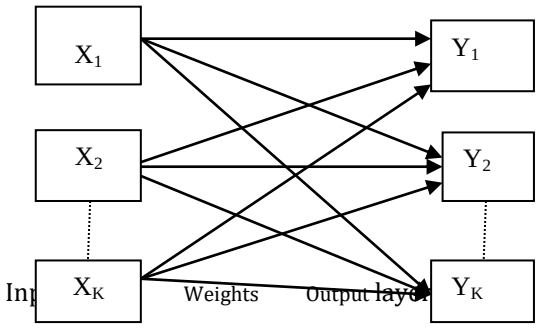

Fig.3 Auto associative Network

The output of matrix [Y] can be represented by:

*Corresponding author: **Moumi Pandit**

Outer product of two vectors is a matrix. If one of the two vectors **p** is having m rows and the other vector **t** is having n column. Their outer product is defined by a matrix whose dimension is m*n and is defined as m * t^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 [Yk] and[Xk] should be considered associative memory. If in the considered associative memory Neural Network we assume [Xk] = [YK] hence the weight matrix becomes $[X_k]*[X_k^t]$.

3. Methodology

3.1 Initial process

Initially, each individual signature is scanned and stored in the system in soft copy. However, scanned signatures again need to be resized to make data easy to store & accordingly consume less memory in system.

3.2 Training :

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.

Therefore, the images of the signatures can be treated as the training data. The stored images can be read in the matlab platform as seen in **Fig.2** and converted to matrices in form of binary.

Fig.2 Signature sample 1

Weight matrix of each image is calculated individually using the formulae as given below (Haykin S. ,2004 and Sivanandam 2006):

$[w_1] = [x_1]^t [x_1]$

Where w_1 is the weight associated with image 1 which is stored in form of matrix x_1

Now the data is stored in the system for further testing and recognition of signature in the form of final weight which is to be calculated by summing of all individual weights assigned to the inputs. This calculation of weight is the simple algebraic summation used in auto associative network. Similarly, weight of each image is calculated individually and the then all the weight matrix is added together.

$W = w_{1+} w_{2+}.....w_n$

3.3 Testing

Step 1: To test the validity of the network, any random image of signature is to be presented to the system in form of matrix [Z]. The output [Y] is computed with the following formulae (Haykin S. ,2004 and Sivanandam 2006):

[Y] = [W] [Z]

Step 2: The next step involves passing the Y matrix through an activation function.

Step 3: After passing the matrix through a linear activation function, matrix [Y] is matched with the existing database. If the matrix matches, the network will display "known" along with the name of the person and if it does not match with the data stored in the database it will display "unknown".

4. Results and discussion

In this proposed work, two signatures are taken as samples. BW be the input matrix for image of signature 1 and BW1 be the input for signature 2 in reduced matrix form binary as seen in **Fig.3.**

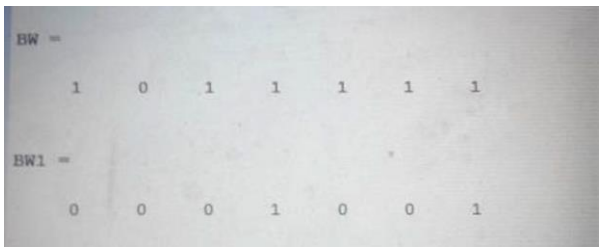

Fig.3 Input in reduced matrix form for two signatures

Fig.4 and Fig.5 shows the weight matrices associated with the two signatures.

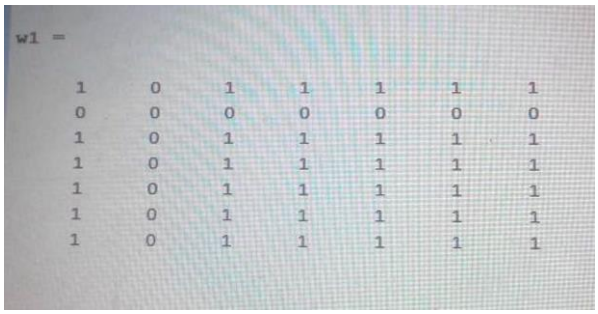

Fig.4 Weight matrix assigned to signature1

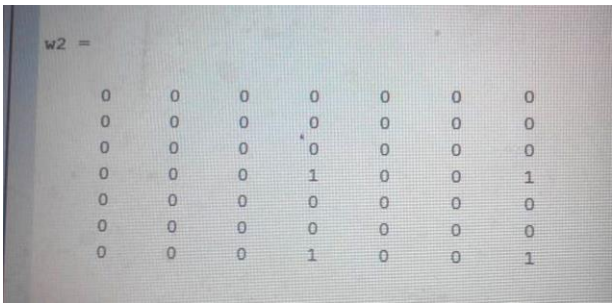

Fig. 5 Weight matrix assigned to signature 2

Then the total weight matrix is calculated by summing the individual weights. For testing, when any random image is given, it is first resized and then output is calculated by multiplying the matrix of the resized given image with the total weight matrix and passed through a linear activation function. If the output after passing through the activation function matches with any matrix of image stored in database then the network will display the person's name otherwise it will display unknown. **Fig.6 and Fig.7** shows the output of the network which displays the person's name.

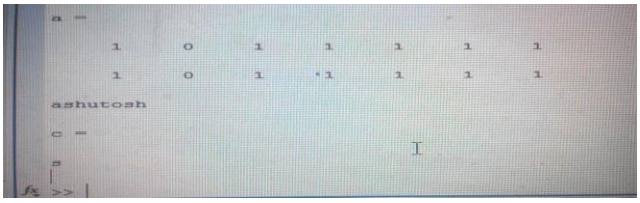

Fig.6 Matched signature showing the name of the name of the person

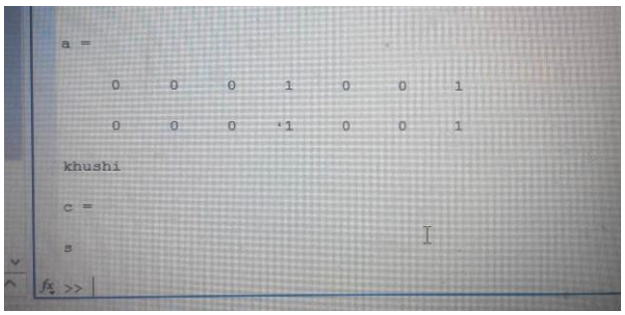

Fig.7 Matched signature showing the name of the name of the person

However if the signature is not there in the database it will display as unknown.

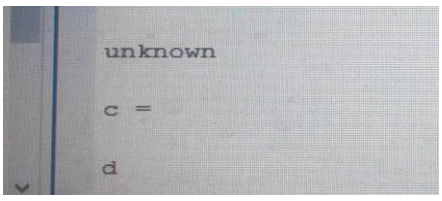

Fig.8 Different signature testing showing unknown as an output

5. Interfacing with arduino

Arduino Juang, (H. S., & Lurrr, K. Y. 2013) is a device which can interface the software and hardware in a single platform. In this proposed work, arduino is interfaced with matlab to display the signature verification.

Arduino will display the output given by matlab by glowing the led as seen in **Fig.8** which is inbuilt within the device if the signature is found to be within the database. But the led will not glow in case the signature is unknown.

Fig.8 Display of Arduino showing verification of signature

Conclusions

Signature recognition is a very essential process for security in the banking and the financial sector. This process of using neural network for signature recognition is much simpler than the traditional processes in terms of complexity, time and cost.

References

Malik.V, Arora. A, (2015) A Review Paper on Signature Recognition, International Journal for Research in Applied Science & Engineering Technology, 3(6),670-674

Kiani, V., Pourreza, R., & Pourreza, H. R. (2009). Offline signature verification using local radon transform and support vector machines. *International Journal of Image Processing*, *3*(5), 184-194.

Biswas, S., Bhattacharyya, D., Kim, T. H., & Bandyopadhyay, S. K. (2010) Extraction of Features from Signature Image and Signature Verification Using Clustering Techniques. In Security-Enriched Urban Computing and Smart Grid (pp. 493-503). Springer

Bhattacharyya D, Bandyopadhyay S.K, Das P, (2008) Statistical Approach for Off-Li ne Handwritten Signature Verification," Journal of Computer Science, 4, 181-1 85.

Kramer, M. A. (1992). Autoassociative neural networks. *Computers & chemical engineering*, *16*(4), 313-328.

Pandit, M., & Gupta, M. (2011). Image Recognition With the Help of Auto-Associative Neural Network. International Journal of Computer Science and Security (IJCSS), 5(1), 54.

Haykin S. (2004). Neural networks. A comprehensive foundation. Singapore: Pearson Education, Pte. Lt

Sivanandam, S. N., Sumathi, S., Deepa, S. N. (2006). Introduction to neural networks using matlab 6.0. New Delhi: Tata McGraw-Hill.

Juang, H. S., & Lurrr, K. Y. (2013, June). Design and control of a two-wheel self-balancing robot using the arduino microcontroller board. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on* (pp. 634-639). IEEE.