Network and Complex Systems www.iiste.org
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol 2, No.4, 2012

Comparison of Different Neural Networks for Iris Recognition: A Review

Shivani Godara*, Dr. Rajeev Gupta

UCE, Rajasthan Technical University, Kota 324010, India

* E-mail of the corresponding author: shivanigodara48@gmail.com

Abstract

Biometrics is the science of verifying the identity of an individual through physiological measurements or behavioral traits. Since biometric identifiers are associated permanently with the user they are more reliable than token or knowledge based authentication methods. Among all the biometric modalities, iris has emerged as a popular choice due to its variability, stability and security. In this paper, we are presenting the various iris recognition techniques and its learning algorithm with neural network. Implementation of various techniques can be standardized on dedicated architectures and learning algorithm. It has been observed that SOM has stronger adaptive capacity and robustness. HSOM, which is based on hamming distance, has improved accuracy over LSOM. SANN model is more suitable in measuring the shape similarity, while cascaded FFBPNN are more reliable and efficient method for iris recognition.

Key words: Biometrics, Iris recognition, Artificial Neural Networks.

1. Introduction

Iris Recognition is usually known as eye iris network pattern recognition technology. The technique uses human's iris network features map information. This is used as a special and auto-recognizable identity card inputting the computer using Computer Science Technology and Imaging Technique. A typical iris recognition system includes four proceedings iris collection, pretreatment, feature extraction and pattern classification respectively. During last decade, numerous attempts have been made to apply Artificial Neural Networks for iris recognition. The Self-adaptive neural networks, SOM-NN & ICA [2], LSOM, HSOM ,Feed forward back propagation, Cascade forward back propagation, feed-forward multi-layer perceptron artificial neural network[9], feed-forward multi-layer perceptron artificial neural network with feature extraction through Hough transform[6] are some of the techniques used for feature extraction.

2. Self Organizing Maps (SOM)

The principal goal of the SOM algorithm is to transform high-dimensional input patterns into a one or two-dimensional discrete map and to perform this transformation adaptively in a topological ordered fashion. In pattern recognition, the Self-Organizing Map (SOM) also called as Kohonen network [1], [2] performs a high quality classification. Assigning the similar input vectors to the same neuron or to neighbor neurons. Thus, this network transforms the relation of similarity between input vectors into a relation of neighborhood of the neurons. The map uses the competition principle, by evaluating the distances between the input vector and the weight vectors corresponding to each neuron, instead of using the classical Euclidean distance.

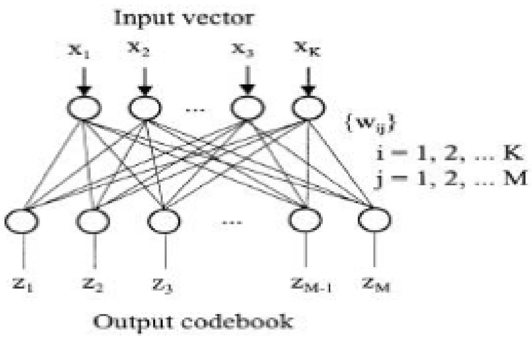

Fig 1- Architecture of the SOM

3. Levensthein Self-Organizing Map (LSOM)

The LSOM is a SOM based on the Levensthein metrics. LSOM uses a symbolic representation for both the input and also for the weight rows. Here the inputs and the weights of the LSOM are represented into a symbolic form where as SOM uses numerical form. For the LSOM, eliminate the condition that the two representations used in the competition phase to have the same length. Instead of evaluating the Euclidean distance between two real vectors, belonging to the same space (for the case of conventional SOM), evaluate the weighted Levensthein distance between two rows of symbols with different lengths, for the LSOM. The LSOM uses the competition principle (like SOM). One computes the Levensthein distances between the input row of symbols and all the rows of weights corresponding to the network neurons. The winner is the neuron that minimizes the above distances.

The training algorithm for LSOM is the following:

i. Initialize the weights of the LSOM.

ii. Apply one by one the words (expressions) belonging to the training set. For each of them, compute the winner neuron, by minimizing the Levenshein distance between the input word and all the weight rows corresponding to the LSOM neurons. Make identical the weight row of the winner neuron with that of the input word (corresponding row). Refine the weight rows of the neurons belonging to the neighborhood of the winner by performing the elementary operations of substitution, insertion and deletion. This is used in order to reduce their Levensthein distances to the input word but not to make them zero. The reduction of the Levensthein distance is a function of the neuron position regarding the winner. This reduction increases when the Euclidean distance (in the map co-ordinates) between the corresponding neuron and the winner neuron decreases.

iii. Compute the classification error as a sum of all the minimum Levenshein distances of the words (expressions) belonging to the training lot. Such a distance is the minimum of the distances between the corresponding input word and the weight rows of the LSOM neurons.

iv. Test the stop condition (if the classification error is-zero).

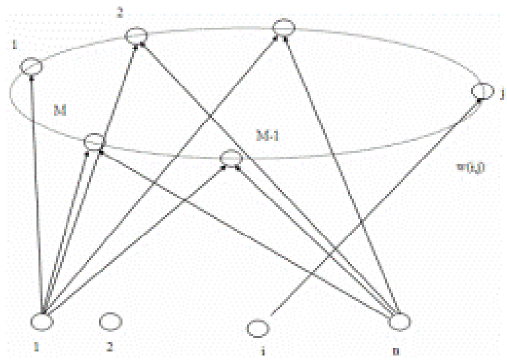

Fig 2: The circular architecture of the LSOM

4. Hamming Self-Organizing Map (HSOM)

SOM variant based on Hamming metric, called Hamming Self-Organizing Map (HSOM). It can be considered as a particular case of the previous LSOM. Both the input vector elements and the weights of the HSOM are represented as binary integers “0” or “1”. The HSOM is based on hamming distance. Assuming two binary vectors, $A=(a_1, a_2, \ldots, a_n)^t$, $B=(b_1, b_2, \ldots, b_n)^t$; where $a_i, b_i \in \{0, 1\}$, the Hamming distance between A and B is $DH(A, B) = \Sigma(a_i - b_i)$. Consequently, the Hamming distance between the input vector X and the weight vector W_j of the jth neuron in the competitive layer is calculated by the equation $DH(X, W_j) = \text{bit count}\{ x_i \text{ XOR } w_{ji} \}$. $DH(X, W_j) = \text{bit count } \{(x_i \wedge w_{ji}) \vee (x_i \wedge w_{ji})$. here, $i = \{1,\ldots, n\}$, $j= \{1, \ldots, M\}$; M = number of output neurons. The HSOM uses the competition principle (like SOM). One computes the Hamming distances between the binary input vector and all the binary weight vectors. The winner is the neuron that minimizes the above distances $c = j \operatorname{argmin} \{DH(X, W_j)\}$. To update the binary weight vectors of HSOM, firstly compute exclusive-OR (XOR) of each element of X and W_j. If XOR $(x_i, w_{ji}) = 1$, then $\mathbf{w_{ji}}$ is a candidate for inversion. The number of inverted bits (belonging to the weight vector W_j) is defined as a learning rate; it gradually decreases as learning progresses.

-

5. Self Adaptive Neural Network (SANN)

SANN uses the combination of the information related to both the shape and magnitude of the data. This implements new similarity matching criteria and error accumulation strategies for network growth. The SANN model is randomly initialized with four neurons. Such an initial structure allows the network to grow in any direction solely depending on the input data. Once the network has been initialized, each input sample xi is sequentially presented and each presentation involves the following two basic operations:

i. Finding the winning neuron for each input sample; and

ii. Updating the weights associated with both the winning neuron and its topological neighbors. Determining a winning neuron for each input data sample is the fundamental process for SANN models. Euclidean distance-based approaches are perhaps the most widely used matching criterion in the development of SANN models. Initialization: let's start the network with four neurons on a 2D grid Initialize each neuron with random values.

iii. Repeat growing for each learning cycle.

iv. Select a sample xi, from the input dataset and Compute the distance between the input sample, xi, and each neuron, wj.

v. Find the winning neuron using the minimum-distance criterion and then update the weights of the winning neuron and its neighbors.

vi. Increase the error value of the winner and find a neuron with the highest cumulative deviation and initiate the growth of new neurons.

vii. Until stopping criterion is satisfied. The learning process is normally stopped when computational bounds, such as the number of learning epochs exceed or when the quantization error of neurons in the network fall below a given threshold.

viii. Repeat smoothing for each learning cycle, Present each input sample and determine the winning neuron.

ix. Update the weights of the winner and its immediate neighbors, until the error values of the neurons become very small or computational bounds are exceeded.

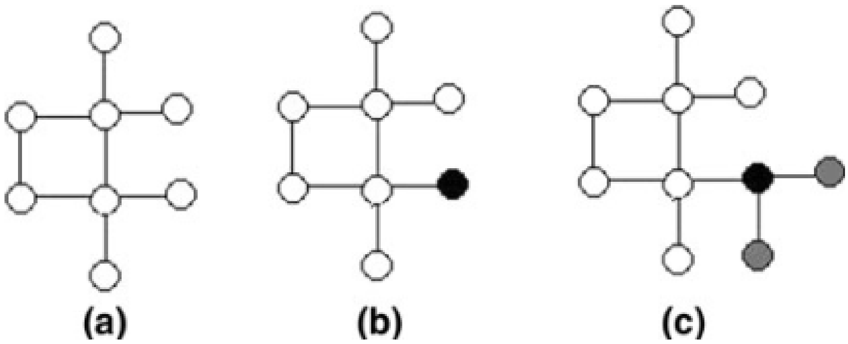

Fig 3: New neurons generation process for the proposed SANN model. (a) The topology before generation, (b) the accumulation of errors during learning process, the neuron marked with a filled circle has the highest cumulative error after a learning epoch, (c) neuron growth on all free neighboring positions. The neurons marked with shaded circle are newly generated.

6. Feed & Cascade – Forward Back Propagation

Feed – Forward Back propagation neural network (FFBPNN) and Cascade Forward Back propagation neural network (CFBPNN) shown in Figs. [4]. A FFBPNN and CFBPNN have three layers: an input layer, hidden layer and an output layer. The neurons in the input layer only act as buffer for distributing the input signals to neuron in hidden layer. Each neuron in hidden layer sums up its input signal after weighting them and computes it outputs. Training a network consists of adjusting its weights using learning algorithms.

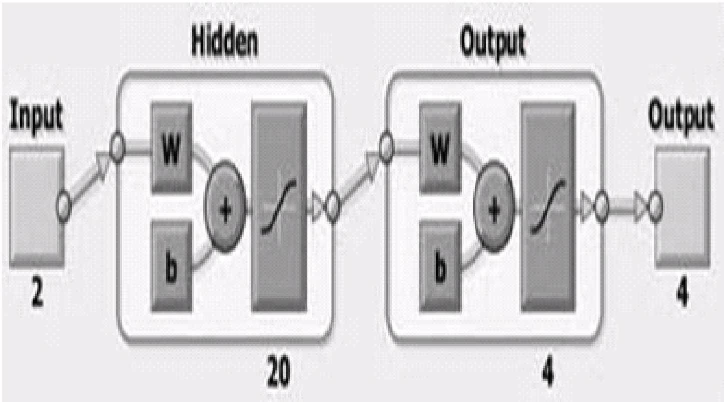

Fig 4: Feed & Cascade – Forward Back propagation

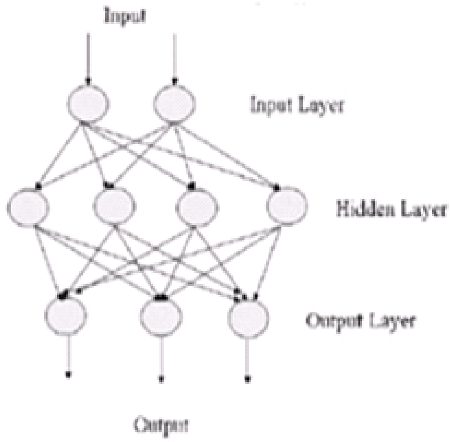

Fig 5: Architecture of feed forward back propagation artificial neural network

7. Feed Forward Multilayer Perceptron Artificial Neural Network

A feed forward multilayer perceptron (MLP) artificial neural network is used to form the match decision. The error back-propagation training algorithm used to adjust the internal neural network weights. Since a typical multilayer perceptron neural network produces a class membership decision as an output, the maximally responding output node represents the class membership of the input pattern. To achieve a more descriptive comparison to the previously described distance metrics, it is desirable that the neural network produce a distance value as an output. For iris template matching, this distance value would represent a similarity score between two iris templates. One of the output nodes was removed, leaving one remaining output node which represents a similarity measurement. The sigmoid function on the remaining output node was also removed. The purpose of the output sigmoid function was to force and limit the output values to one or negative one. Removing the sigmoid function allowed the neural network to output numerical values that represent the degree to which it determined two iris templates match. This numerical value can be considered a similarity score between two iris templates and will be called the neural network distance. As with the other distance metrics, a smaller number from the neural network metric denotes a greater similarity between two templates. Accuracy results, identical to those presented in this paper, could be achieved by simply training and executing the unmodified neural network. A neural network with a large number of hidden nodes has the ability to memorize input data points that are statistical outliers. Another issue is computational efficiency. A neural network with a large number of hidden nodes can be computationally expensive to execute over a large database.

8. Feed Forward Neural Network (FFNN) using Hough Transform

In FFNN information flows in forward direction. Signal flows from the input nodes to output nodes through the hidden nodes. For training of the network, training data is fed into the input layers to output layer though hidden layers which are adjusted to fit the data points to the curve. This is known as forward back propagation algorithm. Input data received by hidden layers which are multiplexed with appropriate weights and summed. It's a hidden layer output which is non linear transformation of the resulting sum. At the output layer same operation is performed finally output values are compared with target value. The error between two is propagated back towards the hidden layer. This is the backward pass of the back propagation algorithm. The procedure is repeated to get the desired accuracy. Hough Transform being the most efficient techniques is used to identify positions of arbitrary shapes most commonly circles and ellipses. The purpose of this technique is to find imperfect instances of objects in a parameter space within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. The local maxima obtained from the accumulator array are used in training of back propagation neural network.

9. Conclusion

In this paper, we have discussed various up-to-date neural network architectures for iris recognition. These include direct development of neural learning algorithms for iris recognition. Theoretically, all the existing iris recognition algorithms can be possibly implemented by modified neural networks such as SOM with ICA, Levensthein Self-organizing Map (LSOM), Hamming Self-Organizing Map (HSOM)[3], Feed forward back propagation neural network, cascade forward back propagation and Self Adaptive neural network (SANN). One of the advantages of doing so is that implementation of Redundancy of feature space. It is also used for iris classification and recognition. Various techniques can be standardized on dedicated architectures and learning algorithm. Extensive evaluation and assessment for a wide range of different techniques and algorithms can be conveniently carried out on generalized neural networks. An Iris Recognition Algorithm Based on ICA and SOM Neural Network almost remove the redundancy of feature space. It is also used for iris classification and recognition. ICA and SOM neural network has stronger adaptive capacity and robustness but as the neural network has many inherent defect such as convergence slow rate, easy to relapse into local minimum, large quantity of parametric and parameter determination need to be from experience and so on, and original neural network need be trained again after addition sample into iris database, this method need to improve further. Due to this two variant [LSOM & HSOM] of SOM comes into picture. Levensthein Self-organizing Map (LSOM) uses the weighted Levensthein distance between two rows of symbols with different lengths while SOM variant, Hamming Self-Organizing Map (HSOM) uses the specific Hamming distance in the competition phase. Trained SOM is not able to accurately represent the input space. So to overcome this limitation SANN models are used, where each training sample is presented multiple times and the desired performance can only be achieved after gradual adaptation of weights associated with each neuron that may influence the selection of learning models. This may affect the performance of the model in large-scale applications. SANN model would be more suitable when there is a need to include the magnitude information in measuring the shape similarity. Some other networks like Feed forward back propagation neural network, cascade forward back propagation network (CFFBPN)) are also used for the iris recognition. In these algorithms, error signal is calculated between output layer and target value, and this error signal will move-back to hidden layers to improve the performance. This evaluation and comparison among them indicate that the cascaded FFBPNN are reliable and efficient method for iris recognition. Various techniques can be standardized on dedicated architectures and learning algorithm. Extensive evaluation and assessment for a wide range of different techniques and algorithms can be conveniently carried out on generalized neural networks. An Iris Recognition Algorithm Based on ICA and SOM Neural Network almost remove the

References

M. Gopikrishnan, T.Santhanam (2011), 'Effect of different neural network on the accuracy in iris patterns recognition', International Journal of Reviews in Computing, Vol. 7.

Bo Lu1, Jing-jing Wu, Yu Wang (2010), 'An iris recognition algorithm based on ICA and SOM neural network', CISP, pp 2246-2448.

Victor-emil Neagoe (2007), 'New self-organizing maps with non-conventional metrics and their applications for iris recognition and automatic translation', proce 11th WSEAS International Conference on computers, Greece, pp 145-151.

Huiru Zheng, Haiying Wang (2011), 'Improving pattern discovery and visualisation with self-adaptive neural networks through data transformations', Springer.

Venkata Rama Prasad Vaddella, Kurupati Rama (2009-2010), 'artificial neural networks for compression of digital images: a review', IJRIC& LLS, pp 75-82.

Shylaja S.S., K.N.Balasubramanya, Murthy, S. Nataranjan, Nisecheth, Muthuraj R.,Ajay S(2011), 'Feed forward neural network based eye localization and recognition using hough transform', IJACSA Vol. 2,No. 3.

Mr.Amir M.Usman Wagdarikar, Mr.Patil B.G, Mrs. Dr. Shaila subbaraman(2010), 'Performance Evalution of IRIS Recognition Algorithms using Neural Network Classifier', IEEE,pp 146-149.

Thomas Heseltine, Nick Pears, Jim Austin, Zezhi Chen (2003), 'Face Recognition: A Comparison of Appearance-Based Approaches', Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), Sydney, pp 50-69.

M. Gopikrishnan, T.Santhanam (2010-2011), 'Improved biometric recognition and indetification of human iris patterns using neural networks', JATIT & LLS, Vol. 31.