Neural network behavior analysis based on transfer functions MLP & RB in face recognition

Conference Paper · October 2018

DOI: 10.1145/3279996.3280011

Tareq Abed Mohammed
Kirkuk University

Abdulbasit Alazzawi
Istanbul Kemerburgaz University

Osman Nuri Ucan
Istanbul Aydin University

Neural Network Behavior Analysis Based on Transfer Functions MLP & RB in Face Recognition

Tareq Abed Mohammed
Altinbas University/ Kirkuk University
Collage of Engineering/ Collage of Science
Istanbul, Turkey/ Kirkuk -Iraq
tareq.mohammed@altinbas.edu.tr

Abdulbasit ALAZZAWI
Altinbas University / Diyala University
Collage of Engineering / Collage of Science
Istanbul, Turkey / Diyala -Iraq
abdulbasit.alazzawi@ogr.altinbas.edu.tr

Oguz BAYAT
Altinbas University
Collage of Engineering
Istanbul, Turkey
oguz.bayat@altinbas.edu.tr

Osman N UÇAN
Altinbas University
Collage of Engineering
Istanbul, Turkey
osman.ucan@ altinbas.edu.tr

ABSTRACT
We performed multi-layer perceptron neural networks MLPNN and Radial Basis neural networks RBNN. In the MLPNN, we applied three layers (input, Hidden, and output) with sigmoid transfer function. Similarly, we used RBNN. Both classifiers are used after preprocessing operations BIOID data set is used in training and testing phases to test the proposed face recognition system combinations. According to the experimental results, the proposed schemes achieved satisfactory results with high accuracy classification

Keywords
Neural Network ; Face Recognition; Artificial intelligence ; Preprocessing Edge Detection ;Machine Learning.

1. INTRODUCTION

Biometrics as a method for identification schemes has been extensively utilized for its strength; one of its used is application that depend on the human face. The face has unmatched features that represent evidence in personal identification because of its features, namely, face print and face geometry. Any camera (with sufficient resolution) possibly used to obtain the digital face image. A critical issue to improve the performance of these systems is enhancing face image. Any scanned picture is accepted tool and can do the same process as well.
The better the image source (i.e. camera or scanner) the further precise results we get. The lighting conditions required are mainly dependent on the quality of the camera used. In unprepared environment, individual features may not be easily discernible [1][2][3]. In face image, human faces are exposed under countless sources of illumination.

Conference'1, juon 19–20, 2018, Bagcilar, Istanbul, Turkey.

DOI: http:// http://iares.net/Conference/

This paper proposed compression approach between two types of neural such as radial basis neural network and multi-layer feedforward regardless of face image quality and illumination conditions. In this study. BIO ID database ("BioID-Face-Database @ www.bioid.com," n.d.) is the dataset used in our empirical experiments. The BIO ID database includes 23 subjects with 1502 face images. The results show that the proposed (Sobel, SLP, ANN) method can enhance the performance of face recognition system and gains high accuracy classification.

In this paper the conclusion are organized in the following form which described the edge detection methods that
That will be used in this paper as Prewitt, Sobel, Roberts, Zero cross, LOG, and Canny filter. Section 3 explains our

Implement algorithm that combines the segmentation process by extracting methods. Therefore Artificial intelligence system in section 5 shows the future work of this research paper.

2. NEURAL NETWORK

Anural network are a main tools and robust classification techniques which can be performed for predicting not only for the known data but also for the unknown data. It works well for linear and nonlinear systems separable data sets. NN has been applied in many areas such as interpreting visual scenes, speech recognition; face recognition Rowley et al used the first power neural approach which registered performance statistics on a huge and difficult dataset. The behavior of NN to robotically learn from samples makes the neural attractive and exciting. Moreover, neural are very robust and adaptive. Therefore, for the applications which undergo many variation issues like biometrics systems (e.g., face, hand, face, fingerprint) neural networks seem to be a good remedy to treats the recognition problem. [4][5]

2.1 Radial Basis Neural Network

Radial basis neural networks techniques are suitable for related biometrics applications such as face detection and recognition, It is not possible to build a robust detector with a great accuracy and response to entirely the probable face image differences since the big inter class change, the variation of ambient light conditions, and the complex structure of the background [10].

Since face detection and recognition can be treated as a class of pattern recognition. The benefit of applying a neural network for face recognition is the possibility of training a system to detect the complex class conditional density of face patterns. An Rb neural structure is shown in fig. (), which has an architecture alike to that of a three-layer feed-forward neural network.

3. Architecture

The model of the RBF neural involves three different layers with feed forward architecture. Architecture shown below (Fig1) is a radial basis network with R inputs.

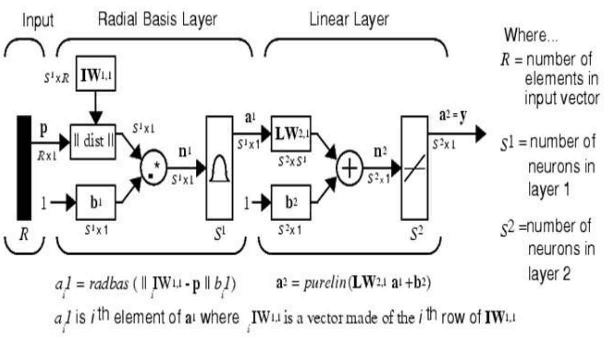

Fig. 1. Radial Basis Neural Network

Rp are centers of RBFs. Centers are set to ci = xi 0 Rp, i=1, N. A very often used form of RBF is the Gaussian function (x)=exp (-x 2 /22), where is a width (parameter).

Functions i, i=1, N form the basis of a linear space and the interpolation function f is their linear combination Interpolation problem is simple to solve, in contrast to approximation problem (there is N given points and n0 functions, where n0 ¡ N.), which is more complicated. Then it is a problem to set centers ci, i=1, n0, also the parameter of each RBF can be not the same for all RBFs.

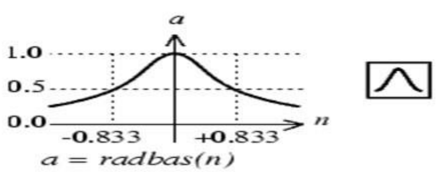

Fig. 2. RAD Function

The transfer function for a radial basis neuron is the radial basis _ Fig. 2.

RAD Function RAD Function has a highest of 1 when its input is 0. As the distance between w and p reduces the output growth. Thus, a radial basis neuron acts as a detector that gives 1 whenever the input p is matching to its weight w [11] [12] [13].

The bias b tolerates the sensitivity of the r neuron to be updated. consisting of input, one hidden, and the output layer. The input layer spreads training data into the network, the hidden layer represent the computational units based on RF. Last layer neurons calculate linear combinations of their inputs. [14]. IW1, 1: Weight metrics to the first layer. //DIST//: Euclidean distance weight function.

b1: basis. LW2, 1: Weight metrics to the second layer. The neural networks that utilized in the proposed algorithm are RBNN which consist of two layers. Radial basis function (RBF) network [15], [16], Given a set of N distinct vectors xi 0 Rp —i = 1, N and di 0 R —i = 1, N, the aim is to catch a function f: Rp R satisfying the condition f(xi)=di, i=1, , N.

RBF method works according to these radial basis functions (RBF) i, where i: Rp R, i=1, N and i = (2x-ci2), where: R+ R, x 0 Rp , 2 2 is a norm on Rp , ci 0 of received , single hidden, and the output layer. RBF network learning consists of more different steps

(a structure of function network learning is explained in [15] Here is a RF neural with R inputs.

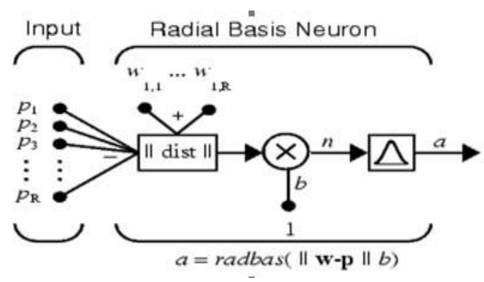

Fig. 3. Radial Basis Neural Network with R Input Vectors

The bias b tolerates the acceptancy of the neuron to be adjusted. For example, if a neuron had a bias of 0.1 it would output 0.5 for any input vector p at vector distance of 8.326 (0.8326/b) from its weight vector w. 2.2 Multi-Layer Perceptron Network Multi-Layer Perceptron (MLP) is the most famous neural network.

The mullite layer neural network basically have three layers which are input layer, hidden layer, and an output layer. Each layer is connected to the next layer by means of weights which are going to be defined based on a learning algorithm. Each layer contains various numbers of neurons which are represented by a transform function.

The number of input neurons is fixed and usually, there is no mapping in this layer. It means that the data directly propagates from the input units to the hidden unit. The number of hidden neurons is unknown, and it is considered as the meta-parameter of the MLP.

However, the hidden neurons have the mapping function which can be the sigmoid function: Y = 1/(1+eBX) (1) Where are the coefficient and x is the input of the sigmoid function and Y is the results of the sigmoid function. The last layer is also fixed. However, the output layer depends on the representation of desired output.

As it spreads through the MLNN layers, the difference between calculated and desired output is calculated by the root to mean square method: p E = 1/pX(Øi -Di)2 (2) i=1 Where E is the function that calculate error and O is yield of last layer and D is the target (desired output), and p represents the number of samples. The weights of the neural are defined based on gradient descent rule.

3.1.Multi-Layer Perceptron Network

Multi-Layer Perceptron (MLP) is the most famous neural network. The MLP neural network can have three main layers which are input layer, hidden layer, and an output layer. Each layer is connected to the next layer by means of weights which are going to be defined based on a learning algorithm. Each layer contains various numbers of neurons which are represented by a transform function. The number of input neurons is fixed and usually, there is no mapping in this layer. It means that the data directly propagates from the input units to the hidden unit. The number of hidden neurons is unknown, and it is considered as the meta-parameter of the MLP. However, the hidden neurons have the mapping function which can be the sigmoid function:

$$Y = 1/(1+e^{B}X) \quad (1)$$

Where the coefficient and x are is the input of the sigmoid function and Y is the output of the sigmoid function. The output layer is also fixed. However, the number of the output layer depends on the representation of desired output. The figure ( ) represents a typical MLP neural

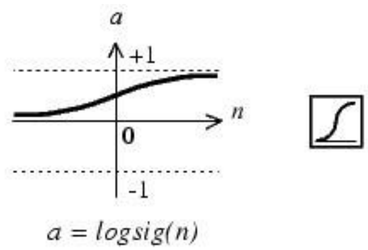

Fig. 4. Main Figure

network. There are two series of unknown weights in the MLP neural network. [15][16]The back propagation learning is applied to calculate the unknown weights. The learning process starts with the forward propagation of data in the network. At the beginning, the weight of hidden of the hidden and output layer is selected by a random value. As it propagates through the neural network, the difference between calculated and desired output is calculated by the root to mean square method:

$E = 1/p^{X}(Ø_i - D_i)^2$ (2)

$i=1$

Where E is the error function and O is the output of the neural network and D is the desired output, and p is the number of samples. The weight of the neural network is defined based on gradient descent rule. For further studies please refer to.

3.2.Backpropagation

Back propagation is the generalization of the Widrow-Hoff learning rule to multilayer networks (4) and nonlinear differentiable transfer functions. The input vectors and the corresponding target vectors are used to train a network until you approximate a function by approximating the corresponding output vectors to specific output vectors or by classifying the input vectors as defined by you. Biased networks, a sigmoid layer and a linear output layer can approach any work. [18] [17]

3.3.Architecture

A basic neuron with R inputs is shown below. Each input is weighed with an appropriate w. The sum of the weighted entries and the prejudices constitutes the input to the transfer function f. The neurons can use any separable transfer function to produce their output. Multilayer networks often use log-sigmoid transfer function.

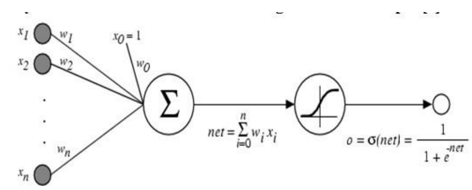

Fig. 5. Feed Forward Neural Network

The function logs generate outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity. Alternatively, multilayer networks can use the tansigmoid transfer function tang Occasionally, the linear trans-

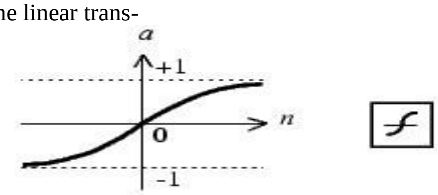

Fig. 6. sigmoid function

for function purely is used in backpropagation networks. Feedforward networks often have one or more hidden

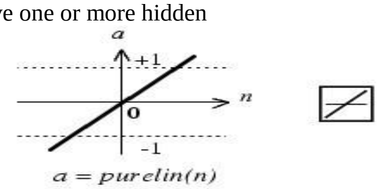

Fig. 7. pure line function

layers of sigmoid neurons, followed by an output layer of linear neurons. Multi-neuron layers with non-linear transfer functions allow to learn non-linear and linear relationships between input and output vectors of a network. The linear output layer allows the network to generate values outside the range of -1 to +1. For multi-layered networks, the number of layers determines the super-definitions on the weight matrices. The appropriate display is used in the next two layer tansig / purelin network

4. PROBLEM DEFINITION

This paper study two kinds of neural networks, RB and MLP neural network. The aim is to study the effects of a transfer function of these neural networks on face image analysis problem. We have studied the RAD from a radial basis neural network and sigmoid function from MLP and explain the behavior of these functions during neural network training.

5. METHODOLOGY

The aim of the proposed system is a comparison between two types of neural network in the face recognition, in the other words, the ability of neural (back propagation and radial basis) to detect face pattern in the image and compare the results.

The proposed face recognition image system use preprocessing stage because it is too difficult to use an image (in the neural network to training) without preprocessing operations, the preprocessing operation makes an image without noise or decrease the noise from the original colored input face image, so it is necessary to use the further preprocessing. The preprocessing stage has operations used to find the object of interest (in the proposed system) to be prepared for the next levels.

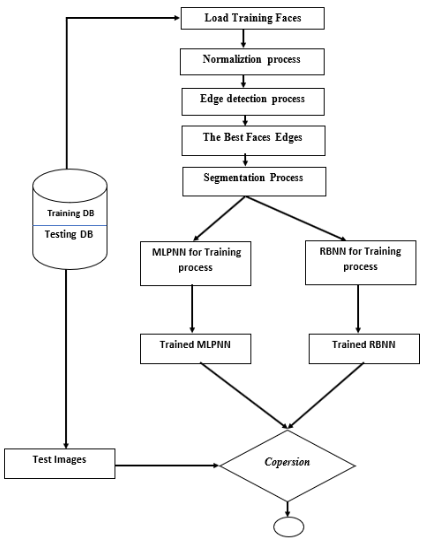

Fig. 8. Main Figure

5.1 Image Reading

The first function of the proposed system is reading the image, the user can read an image of file format JPEG, from any location in PC. A = imread(filename) reads the face image from dataset. If filename is a multi-image file, then imread reads the first image in the file.

The function of the base layer is to captures the main structural information, meanwhile the rest layer covers the residual smaller details [21].One of the fundamentals goals of preprocessing is to minimize data as much as possible. In the proposed system, we minimize date by convert gray image to binary image. To obtains, data from the image, convert the image to grayscale(this operation will convert 24-bit/pixel images to 256 grayscale images.

The color image is represented as red, green and blue or RGB image. Using the 8-bit image format as standards model, c color image would have 24bits/pixel, 8 bits for each of the color bands (red green and blue). after that, we read image.

5.1.1.Transformation Gray level to Binary

In this operation, the gray image of a face that entered to the proposed system will transform to binary level. each gray input image must convert to binary scale image, in other words (this operation will convert 24-bit/pixel images to 256 grayscale images.

Fig. 9. Convert image to Gray Image and Get Edge of the face

5.1.2. Pseudo Code

The conversion algorithm is as follow: Begin Step 1: load image file from memory Step 2:
For i=0 to image width
For j=0 to image width
Split the components (red, green and blue) of pixel[i,j] as follows :
C= pixel [i,j];
Red [i, j] = (C and (logical and) HFF),
C= (C Red [i, j])/256;
Green [i, j]=(C and (logical and) HFF);
C= (C green[i,j])/256 ;
Blue [i,j]=(C and (logical and) HFF); r= Red[i,j]/(Red[i,j]+ Green [i,j]+Blue[i,j]); g= Green[i,j]/(Red[i,j]+ Green [i,j]+Blue[i,j]); b= Blue[i,j]/(Red[i,j]+ Green[i,j]+Blue [i,j]); gray[i,j]= (Red[i,j]*r+ Green[i,j]*g +Blue [i,j]*b);
Draw the gray [i,j];
Step 3: Save the data of image; End
Where:
H: hexadecimal. F: value in hexadecimal=1111. r : red value. g : green value. b: blue value .

5.2 Normalization Processing

One of the preprocessing stages is normalization. The process of normalization is nothing except preparing the date of pixels that more than some thresholds. This operation is very important to reduce data because most of the images don't appear in the center of images. The data of the database of the image is very huge and we have to do some process to save time and efforts

5.2.1. Normalization Pseudo Code

Begin nr=p1(1);nc=p1(2);or=64;oc=64; fr=mod(or,nr); sr=fix(or/fr); fc=mod(oc,nc); sc=fix(oc/fc); j=0;k=0; for i=1:or; if i¡=fr*sr; if mod(i,sr) =0; j=j+1; im1(j,:)=im(i,:); end; else; j=j+1; im1(j,:)=im(i,:); end j=0;k=0; for i=1:oc; if i¡=fc*sc; if mod(i,sc) =0 ; j=j+1; nim(:,j)=im1(:,i); end else j=j+1; nim(:,j)=im1(:,i); end end str=r/nr; stc=c/nc; if str¿1; om1=nim(1:str:end,:); om2=nim(2:str:end,:); om3=(double(om1)+double(om2))/2; om=om3; end; if stc¿1; om4=om3(:,1:stc:end); om5=om3(:,2:stc:end); om6=(om4+om5)/2; om=om3; end; oim=uint8(om); end

6. DECISION NEURAL NETWORK

DECISION NEURAL NETWORK The following operation applied before (to) previous both types of neural networks .to get the input vectors ,we should divide each an image in the training image set to 2 dimensional blokes with the same size .this operation called segmentation operation .we convert each block to one dimensional vector as follows : We assume each block in an image has u*v: size of block g(p,q): gray level value for each coordinate (point)in the block at (p,q) position, now we represent each block as vector as follows: x =(x1 x2.....xi,...... xn) where xi : g(p,q) n=u.v
i: vector coordinate x : input vector to convert from position coordinate (p,q) to vector coordinate (i) do the following operation i =(s-l)u+p....(1) where f= 1,2.....u s=1,2......v to all training an image set we do segmentation operation .the segmentation operation create a TS(training set) array with two dimensions(n*l),where GM= [x1 x2 x3xk ..xl] where xt = x1 x2....... xi,.....
xn ,column vector no of block in each image (f)=(N*N)/n N*N: an image dimensions n=u.v l=no of training images *f each column in TS array represents input vector to both neural nets .in each iteration we apply one column in GM array .the select of vector (optionally) randomly or sequentially from first to the last vector in TS array .

7. RESULTS

In the proposal system, we focus on the transformation function of multilayer perceptron and radial basis neural networks for comparison.
Whoever, some of the results have explained there are another factor also effects in results. the study is just to show the differences in use between two types of neural networks and to explain the difference in effects on a decision. We applied edge detection filters on the database to get the edges of faces. group of the filters used with two neural network .
According to near to 2000 experiments that included these edge detection filters ,we copmared the results of both neural outputs in (50%-85%)training data set
The following sections summarize the results of the detection of both neural network in table 1 that shows the ability of both neural (backpropagation and radial basis) to accommodate complex decision region.

Table (1) explains the results of both neural networks (MLPNN and RBNN) acording to different training rate (50%-85%) form data set

CLASSIFIER	TRAINING DB.%	Training Classification rate	Testing Classification rate
MLPNN	0.85	97.8330	95.653
RBNN	0.85	95.1260	93.3355
MLPNN	0.8	98.8887	94.6508
RBNN	0.8	97.3129	91.0635
MLPNN	0.7	88.3392	83.7692
RBNN	0.7	98.8987	94.5522
MLPNN	0.6	85.1795	80.2371
RBNN	0.6	97.4359	94.5671
MLPNN	0.5	75.8659	70.9355
RBNN	0.5	86.3902	83.9255

8.CONCLUSION

In the proposed system Database (Bio-Id) includes more than 1000 stand ad images. we applied Multi layer perceptron and Radial basis neural networks.

After applied training on the data sets, we registered some another factors addition to transformation functions can effect on results of decision place. The experiments appear that the error rate is a little bit higher in the back propagation as compared with radial basis. The training time in the radial basis is less than back propagation. The performance of radial basis is higher than back propagation. Radial basis networks can require more neurons than standard feed forward back propagation networks, but often they can be designed in a fraction of the time it takes to train standard feed forward networks. They work best when many training vectors are available.

REFERENCES

1. M. Agarwal, H. Agrawal, N. Jain, and M. Kumar, Face Recognition Using Principal Component Analysis, Eigen face and Neural Network, in Int Conf on Signal Acquisition and Processing, 2010 - ICSAP 10, 2010, pp. 310314.
2. M. Agarwal, N. Jain, M. Kumar, and H. Agrawal, Face Recognition Using Eigen Faces and Artificial Neural Network, Int. J. Comput. Theory Eng., vol. 2, no. 4.
3. M. J. Aitkenhead and A. J. S. McDonald, A neural network face recognition system, Eng. Appl. Artif. Intell., vol. 16, no. 3, pp. 167176, 2003.
4. M. J. Aitkenhead and a. J. S. McDonald, A neural network face recognition system, Eng. Appl. Artif. Intell., vol. 16, pp. 167176, 2003. Fig. 13. Threshold of Correct Classification Rate.
5. N. Amani, A. Shahbahrami, and M. Nahvi, A new approach for Face Image Enhancement and Recognition, vol. 52, pp. 110, 2013.
6. S. Anila and N. Devarajan, Simple and Fast FaceDetection System Based on.
7. K. a. a. Aziz, R. a. Ramlee, S. S. Abdullah, and a. N. Jahari, Face Detection Using Radial Basis Function Neural Networks with Variance Spread Value, 2009 Int. Conf. Soft Comput. Pattern Recognit., no. 1, 2009.
8. C. M. Bishop, Neural networks for pattern recognition,J. Am. Stat. Assoc., vol. 92, p. 482, 1995.
9. N. Cao, T. Hai, D. Kim, and H. Park, A heuristic approach for Shadow and Light Regions Fast Detection in Face Images, pp. 610614, 2012.
10. S. Chen, C. F. N. Cowan, and P. M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, Neural Networks, IEEE.
11. Dharmarajan and T. Velmurugan, Applications ofpartition based clustering algorithms: A survey, 2013 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE.
12. S. Duffner, Face Image Analysis With ConvolutionalNeural Networks, Thesis.
13. H. M. El-Bakry and Q. Zhao, Fast Neural Implementation of PCA for Face Detection, Neural Networks, 2006. IJCNN 06. Int. Jt. Conf., pp. 806811, 2006.
14. Y. S. Gao and M. K. H. Leung, Face recognition usingline edge map, IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 6, pp. 764779, 2002.
15. P. Goel and S. Agarwal, An Illumination InvariantRobust and Fast Face Detection, Feature Extraction Based Face Recognition System, Comput. Commun. Technol. (ICCCT), 2012 Third Int. Conf., pp. 110115, 2012.
16. M. M. Kasar, D. Bhattacharyya, and T. Kim, FaceRecognition Using Neural Network: A Review, vol. 10, no. 3, pp. 81100, 2016.
17. M. M. Kasar, D. Bhattacharyya, and T. Kim, FaceRecognition Using Neural Network: A Review, vol. 10, no. 3, pp. 81100, 2016.
18. T. P. Singh, Face Recognition by using Feed ForwardBack Propagation Neural Network, Int. J. Innov. Res. Technol. Sci., vol. 1, pp. 1319, 2013.