International Journal on Recent Researches In Science, Engineering & Technology
(Division of Computer Science and Engineering)
A Journal Established in early 2000 as National journal and upgraded to International journal in 2013 and is in existence for the last 10 years. It is run by Retired Professors from NIT, Trichy. It is an absolutely free (No processing charges, No publishing charges etc) Journal Indexed in JIR, DIIF and SJIF.
Research Paper
Available online at: www.jrrset.com

ISSN (Print) : 2347-6729
ISSN (Online) : 2348-3105

Volume 5, Issue 7,
July 2017.

JIR IF : 2.54
DIIF IF : 1.46
SJIF IF : 1.329

FACE RECOGNITION USING NEURAL NETWORKS

Dr.JVN. Ramesh
Department of Computer Science and Engineering,Shadan College of Engineering and Technology
HYD,T.S,INDIA

ABSTRACT
In recent times, face recognition plays a significant role in the field of image analysis. A *facial recognition* system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source.The objective of this study is to develop an effective face recognition system that extracts face features to improve the recognition performance. The proposed method derived the face features with principle component analysis (PCA).In this paper we are concentrating on Back propagation for training the neural network, which is the powerful path to machine learning even the data includes complex image input.

Keywords: Neural Networks, Principle Component Analysis, Back Propagation**,**

1. INTRODUCTION

Over the past few years, a lot of research put forward in the area of Face Recognition and Face Detection to make it more efficient, but it makes a revolution in this field when Viola-Jones comes with its Real-Time Face Detector, which is capable of detecting the faces in real-time with high accuracy. Neural networks as shown in Figure 1 have a large collection of nodes or neurons that are interconnected in some pattern to allow communication between the nodes. These nodes are simple processors which operate in parallel.

Every neuron is connected with other neuron through a connection link. Each connection link is associated with a weight that has information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually excites or inhibits the signal that is being communicated. Each neuron has an internal state, which is called an activation signal. Output signals, which are produced after combining the input signals and activation rule, may be sent to other units.

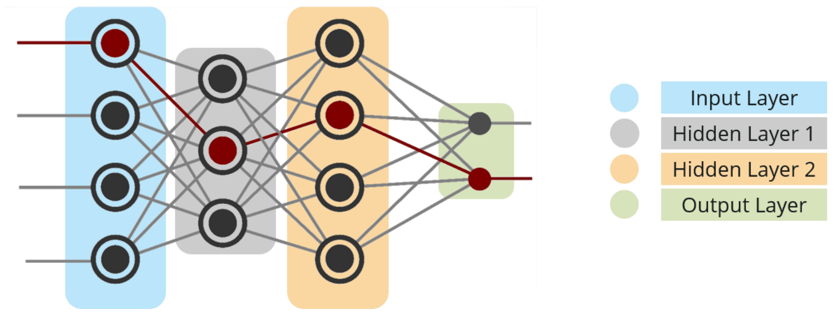

Fig 1: Neural Network Model

- Input Nodes – The Input nodes gets the raw information from the external world to the network and are together referred to as the "Input Layer". No calculation is performed in any of the Input nodes – it just passes the information to the hidden nodes.
- Hidden Nodes – The Hidden nodes have no direct connection with the outside world. They perform computations based on the information from the input nodes and the weights on the connections between the input and the hidden nodes. A collection of hidden nodes forms a "Hidden Layer". A single input layer and a single output layer only present in a network, but it can have zero or multiple Hidden Layers
- Output Nodes – The Output nodes are groupily mentioned as the "Output Layer" and are responsible for computations. It transforms the information from the network to the outside world.

2. FACE RECOGNITION STAGES

Face recognition technology is a merging of various other technologies and their features and characteristics makes face recognition a better performer based on the application. Face recognition as shown in Fig 2 works under three phases- Detection, Extraction and Recognition. The description of each phase of face recognition is given in the following sections.

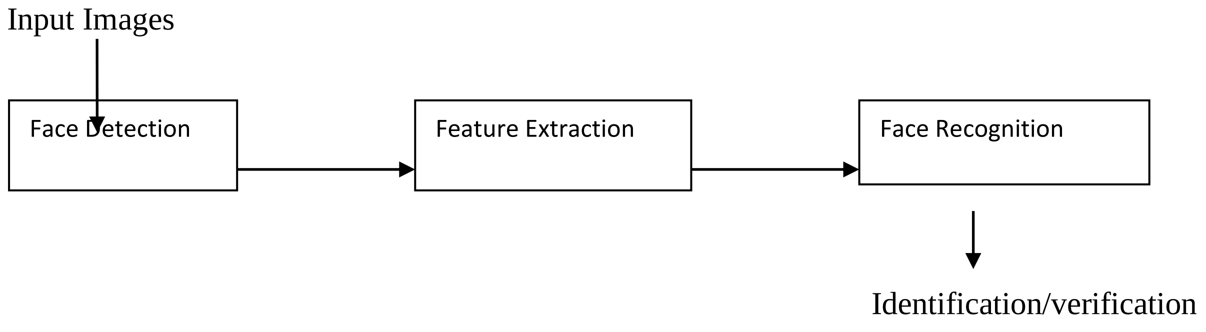

Fig 2: Face recognition stages

2.1 FACE DETECTION

Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. Face detection can consider a substantial part of face recognition operations. According to its strength to focus computational resources on the section of an image holding a face. The method of face detection in pictures is complicated because of variability present across human faces such as pose, expression, position and orientation, skin color, the presence of glasses or facial hair, differences in camera gain, lighting conditions, and image resolution.
Hjemal and Low [8] divides the face detection techniques into two categories named feature based techniques and image based techniques.

2.2 FEATURE EXTRACTION

Face recognition is an evolving area, changing and improving constantly. This section gives the overview of various approaches and techniques along with their advantages and disadvantages. Different approaches of face recognition can be categorized in three main groups such as holistic approach, feature-based approach, and hybrid approach.

2.2.1 METHODS FOR FEATURE EXTRACTION

The analysis of face recognition systems using three different feature extraction techniques-Principal Component Analysis (PCA), Fisher Linear Discriminant analysis (FLD) and Fast Pixel Based Matching (FPBM) is considered.

2.2.1. a.Principal Component Analysis

Principal Component Analysis (PCA), also known as Hotelling Transform or Karhunen-Loeve expansion, is a well-known data representation and feature extraction technique widely used in the areas of pattern recognition, computer vision, etc. [1]-[3]. The purpose of PCA is to reduce the data dimensionality with reveal its essential characteristics i.e. to extract the relevant information from high dimension data set. The Eigen face (PCA) based Method of Turk and Pentland [4] is one of the foremost successful method applied in the literature which is based on the Karhunen-Loeve expansion and their study was motivated by the earlier work of Sirowich and Kirby [3] [5]. It is based on the application of Principal Component Analysis to the human faces. It treats the face images as 2-D data, and classifies the face images by projecting them to the eigenface space which is composed of eigenvectors obtained by the variance of the face images. Eigen face recognition derives its name from the German prefix eigen, meaning own or individual.

The Eigenface method of facial recognition is considered the first working facial recognition technology [6]. When the method was first proposed by Turk and Pentland [4], they worked on the image as a whole. Also, they used Nearest Mean classifier two classify the face images. By using the observation that the projection of a face image and non-face image are quite different, a method of detecting the face in an image is obtained. They applied the method on a database of 2500 face images of 16 subjects, digitized at all combinations of 3 head orientations, 3 head sizes and 3 lighting conditions. They conducted several experiments to test the robustness of their approach to illumination changes, variations in size, head orientation, and the differences between training and test conditions. They reported that the system was fairly robust to illumination changes, but degrades quickly as the scale changes [4]. This can be explained by the correlation between images obtained under different illumination conditions; the correlation between face images at different scales is 35 rather low. The eigenface approach works well as long as the test image is similar to the training images used for obtaining the eigenfaces.

The following steps of Eigenface approach summarize the process:

1. Let a face image X(x, y) be a two dimensional mxn array (8-bit Gray Scale) of intensity values. An image may also be considering the vector of dimension mn. Let the training set of images $\{X_1, X_2, X_3 \ldots X_N\}$. The average face of the set X is defined by

$$\overline{X} = \frac{1}{N} \sum_{i=1}^{N} Xi \qquad (1)$$

2. Calculate the covariance matrix to represent the scatter degree of all feature vectors related to the average vector. The covariance matrix C is defined by

$$C = \frac{1}{N} \sum_{i-1}^{N} (X_i - \overline{X})(X_i - \overline{X})^T \qquad (2)$$

3. The Eigenvectors and corresponding eigenvalues are computed by using

$$CV = \lambda V \qquad (3)$$

Where V is the set of eigenvectors associated with its eigenvalues λ.

4. Sort the eigenvector $V_i \in V$ according to their corresponding eigenvalues $\lambda_i \in \lambda$ from high to low

5. Each of the mean centered image project into Eigen space using

$$W_i = V_i^T (X_i - \overline{X}) \qquad (4)$$

6. In the testing phase each test image should be mean centred, now project the test image into the same Eigen space as defined during the training phase. This projected image is now compared with projected training image in Eigen space. Images are compared with similarity measures. The training image that is close to the test image will be matched and used to identify

2.3 FACE RECOGNITION USINGBACKPROPAGATION NEURAL NETWORKS:

Back propagation is a supervised learning algorithm, for training Multi-layer Perceptrons (Artificial Neural Networks).

2.3.1 Implementing Back propagation

Now that we've developed the math and intuition behind backpropagation, let's try to implement it. We'll divide our implementation into three distinct steps:

1. **Feed-forward**. In this step, we take the inputs and forward them through the network, layer by layer, to generate the output activations (as well as all of the activations in the hidden layers). When we are actually using our network (rather than training it), this is the only step we'll need to perform.

2. **Back propagation**. While designing a Neural Network, in the beginning, we initialize weights with some random values or any variable for that fact.So, it's not necessary that whatever weight values we have selected will be correct, or it fits our model the best. we have selected some weight values in the beginning, but our model output is way different than our actual output i.e. the error value is huge.Basically, what we need to do, we need to somehow explain the model to change the parameters (weights), such that error becomes minimum.Let's put it in an another way, we need to train our model.

One way to train our model is called as Back propagation. Consider the Figure 3 below:

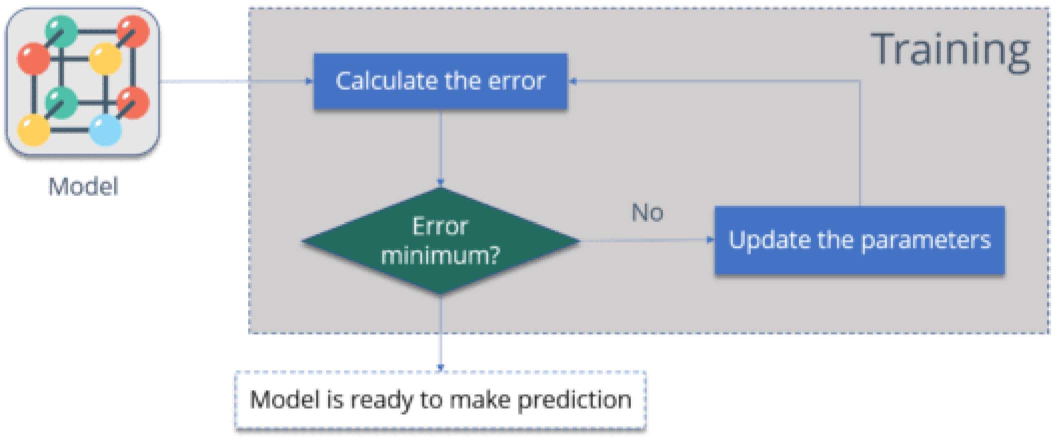

Fig 3.Predictive Model based on Back Propagation

Here, we'll take our error function and compute the weight gradients for each layer. We'll use the algorithm just described to compute the derivative of the cost function w.r.t. each of the weights in our network, which will in turn allow us to complete step 3.

3. Weight update. Finally, we'll use the gradients computed in step 2 to update our weights. We can use any of the update rules discussed previously during this step (gradient descent, momentum, and so on).

The Backpropagation Algorithm

For any given supervised machine learning problem, we (aim to) select weights that provide the optimal estimation of a function that models our training data. In other words, we want to find a set of weights **W** that minimizes on the output of **J(W)**. The gradient descent algorithm—one where we update each weight by some negative, scalar reduction of the error derivative with respect to that weight. If we do choose to use gradient descent (or almost any other convex optimization algorithm), we need to find said derivatives in numerical form.

For other machine learning algorithms like logistic regression or linear regression, computing the derivatives is an elementary application of differentiation. This is because the outputs of these models are just the inputs multiplied by some chosen weights, and at most fed through a single activation function (the sigmoid function in logistic regression). The same, however, cannot be said for neural networks. To demonstrate this, here is a diagram of a double-layered neural network:

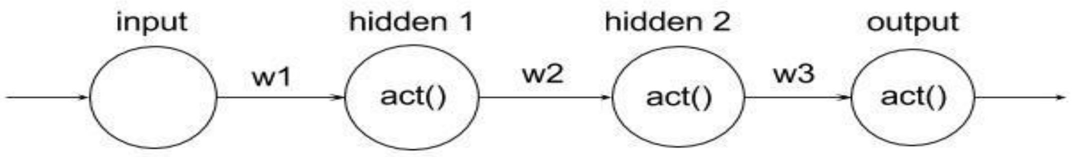

Fig 4.Double-layered Neural Network Model

As you can see, each neu**r**on is a function of the previous one connected to it. In other words, if one were to change the value of **w1**, both "hidden 1" *and* "hidden 2" (and ultimately the output) neurons would change. Because of this notion of functional dependencies, we can mathematically formulate the output as an extensive composite function:

$$output = act(w3 * hidden2)$$
$$hidden2 = act(w2 * hidden1)$$
$$hidden1 = act(w1 * input)$$

And thus: $$output = act(w3 * act(w2 * act(w1 * input)))$$

Here, the output is a composite function of the weights, inputs, and activation function(s). It is important to realize that the hidden units/nodes are simply intermediary computations that, in actuality, can be reduced down to computations of the input layer.
If we were to then take the derivative of said function with respect to some arbitrary weight (for example **w1**), we would iteratively apply the chain rule. The result would look similar to the following:

$$\frac{\partial}{\partial w1} output = \frac{\partial}{\partial hidden2} output * \frac{\partial}{\partial hidden1} hidden2 * \frac{\partial}{\partial w1} hidden1$$

If you fail to get an intuition of this, try researching about the chain rule.Now, let's attach a black box to the tail of our neural network. This black box will compute and return the error—using the cost function—from our output:

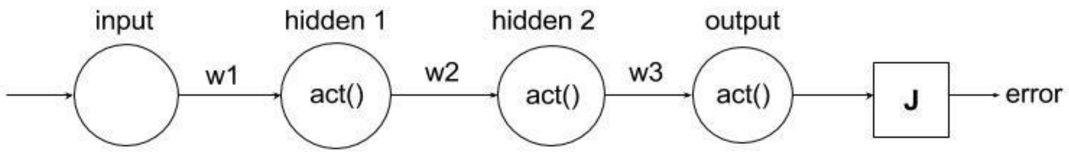

Fig 5.Double-layered Neural Network Model with Error function

All we've done is add another functional dependency; our error is now a function of the output and hence a function of the input, weights, and activation function. If we were to compute the derivative of the error with any arbitrary weight (again, we'll choose **w1**), the result would be:

$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \frac{\partial output}{\partial hidden2} * \frac{\partial hidden2}{\partial hidden1} * \frac{\partial hidden1}{\partial w1}$$

Each of these derivatives can be simplified once we choose an activation and error function, such that the entire result would represent a numerical value. At that point, any abstraction has been removed, and the error derivative can be used in gradient descent (as discussed earlier) to iteratively improve upon the weight. We compute the error derivatives w.r.t. every other weight in the network and apply gradient descent in the same way. *This* is backpropagation—simply the computation of derivatives that are fed to a convex optimization algorithm. We call it "backpropagation" because it almost seems as if we are traversing from the output error to the weights, taking iterative steps using chain the rule until we "reach" our weight.

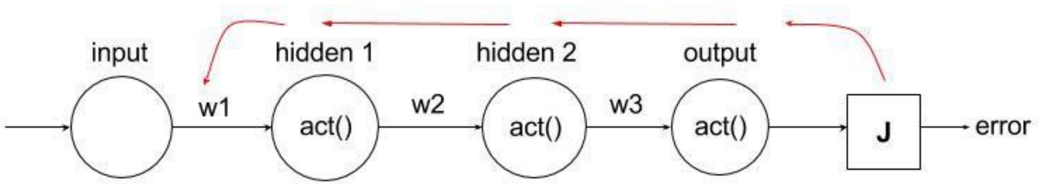

Fig 6.Back propagation Neural Network Model

4. CONCLUSION

Face recognition technologies have been associated generally with very costly top secure applications. Today the core technologies have evolved and the cost of equipments is going down dramatically due to the integration and the increasing processing power. Certain application of face recognition technology are now cost effective, reliable and highly accurate. As a result there are no technological or financial barriers for stepping from the pilot project to widespread deployment.

It is used to detect faces in real time for surveillance and tracking of person or objects. It is widely used in cameras to identify multiple appearances in the frame Ex- Mobile cameras and DSLR's. Facebook is also using face detection algorithm to detect faces in the images and recognise them.

The face recognition system can be implemented using a MATLAB software package. We can use the neural networks tool box in matlab. and can find the transformation for different inputs and compared with unknown face that the given face is in database or not.

REFERENCE

1.W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: a literature survey" ACM Computing Surveys, Vol. 35, No. 4, pp. 399–458, December 2003.

2. A. S. Tolba, A.H. El-Baz, and A.A. El-Harby, "Face Recognition: A Literature Review", International Journal Of Signal Processing Volume 2 Number 2 ISSN 1304-4494, 2005.

3. M. Kirby and L. Sirovich, "Application of the KL procedure for the characterization of human faces," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103-108, Jan. 1990.

4. M. Turk and A. Pentland, "Eigenfaces for recognition", J. of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

5. Sirovich, L. & Kirby, M. "A low-dimensional procedure for the characterization of human face", Journal of the Optical Society of America A, Vol. 4, No. 3, pp.519-524 1987.

6. B. Kepenekci, "Face Recognition Using Gabor Wavelet Transform", MSc. Thesis, METU, September 2001.

7. A. Pentland, B. Moghaddam, T. Starner, "View-based and modular eigenspaces for face recognition", Proceedings of IEEE, CVPR, 1994.

8. B. Moghaddam and A. Pentland, "Face Recognition using View-Based and Modular Eigenspace", Automatic Systems for the Identification and Inspection of Humans, 2277, 1994.
9. Kim, K.I., Jung, K., Kim, H.J. "Face recognition using kernel principal component analysis", IEEE Signal Processing Letters, 9 (2), pp. 40-42, 2002.
10. Ming - Hsuan Yang, "Kernel eigenfaces vs. Kernel Fisherfaces: Face recognition using kernel methods", Proceedings of the fifth international conference on automatic face and gesture recognition 2002.ss
11. The ORL Database of Faces, Available: http://www.uk.research.att.com/facedatabase.html
12. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711– 720, July 1997.Fgfgg
13. .PriyankaDhoke, M.P. Parsai"A MATLAB based Face Recognition using PCA with Back Propagation Neural network"2014
14. Chen, T., Hsu, Y.J., Liu, X., Zhang, W. "Principle component analysis and its variants for biometrics" IEEE International Conference on Image Processing, 1, pp. I/61-I/64 2002.
16. Gottumukkal, Rajkiran. Asari, V.K. "An improved face recognition technique based on modular PCA approach" Pattern Recognition Letters, 25 (4), pp. 429-436, 2004.
17. Oravec, M., Pavlovic ová, J. "Face recognition methods based on principal component analysis and feedforward neural networks" IEEE International Conference on Neural Networks - Conference Proceedings, 1, pp. 437-441, 2004.
18.Yang, J., Zhang, D., Frangi, A.F., Yang, J.-Y. "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition" EEE Transactions on Pattern Analysis and Machine Intelligence, 26 (1), pp. 131-137, 2004.
19. Liu, C. "Gabor-based kernel PCA with fractional power polynomial models for face recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (5), pp. 572-581, 2004.
20. Zhang, D Daoqiang., Zhou, Z.-H. "(2D)2 PCA: Two-directional twodimensional PCA for efficient face representation and recognition" Neurocomputing, 69 (1-3), pp. 224-231, 2005.
21. Kong, H., Wang, L., Teoh, E.K., Li, X., Wang, J.-G., Venkateswarlu, R. "Generalized 2D principal component analysis for face image representation and recognition" Neural Networks, 18 (5-6), pp. 585-594, 2005.
22. Sanguansat, P., Asdornwised, W., Jitapunkul, S., Marukatat, S. "Classspecific subspace-based two-dimensional principal component analysis for face recognition" Proceedings - International Conference on Pattern Recognition, 2, art. no. 1699435, pp. 1246-1249, 2006.
23. Dagher, I., Nachar, R. "Face recognition using IPCA-ICA algorithm" IEEE Transactions on Pattern Analysis and Machine Intelligence, 28 (6), pp. 996-1000, 2006.
24. Wang, H., Yang, S., Liao, W. "An improved PCA face recognition algorithm based on the discrete wavelet transform and the support vector machines" Proceedings - International Conference on Computational Intelligence and Security Workshops, pp. 308-311, 2007.
25. P. Philips, P. Flynn, T. Scruggs, and K. Bowyer. Overview of the face recognition grand challenge. CVPR, 2005. 36. Mandal, T., Wu, Q.M.J. "Face recognition using curvelet based PCA" 19th International Conference on Pattern Recognition, ICPR 2008.
26. http://cswww.essex.ac.uk/mv/allfaces/grimace.zip
27. Li, C., Diao, Y., Ma, H., Li, Y. "A Statistical PCA Method for face recognition" Proceedings - 2nd International Symposium on Intelligent Information Technology Application, 2008, 3, pp. 376-380 40. CVL FACE DATABASE: http://www.lrv.fri.unilj.si/facedb.html.
28. Lee, Y.-C., Chen, C.-H. "Face recognition based on gabor features and two-dimensional PCA" Proceedings - 4th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2008, pp. 572-576.