

Artificial Neural Network-Based Face Recognition

Adjoudj Réda
*Evolutionary Engineering and Distributed
Information Systems Laboratory, EEDIS,
Computer Science Department
University of Sidi Bel-Abbès, Algeria
phone/Fax: (213)-48-57 77 50 ,
E-mail : AdjReda@yahoo.fr*

Dr Boukelif Aoued
*Digital Signal processing laboratory,
Electronic Department
University of Sidi Bel-Abbès, Algeria
phone:(213)-48-57 82 81 ,
E-mail : aboukelif@yahoo.fr*

Abstract

This document demonstrates how a face recognition system can be designed with artificial neural network. Note that the training process did not consist of a single call to a training function. Instead, the network was trained several times on various input ideal and noisy images, the images which contents faces. In this case training a network on different sets of noisy images forced the network to learn how to deal with noise, a common problem in the real world.

1. Introduction

It is often useful to have a machine perform pattern recognition. In particular, machines which can read face images are very cost effective. A machine that reads passenger passports can process many more passports than a human being in the same time[1].

This kind of application saves time and money, and eliminates the requirement that a human perform such a repetitive task. This document demonstrates how face recognition can be done with a backpropagation artificial neural network.

2. PROBLEM STATEMENT

A artificial neural network is to be designed and trained to recognize the persons of the database. An imaging system that converts each face image centered in the system's field of vision is available. The result is that each face image is represented as a 32 by 30 matrix of reals values. (Image size ~ 32 x 30).

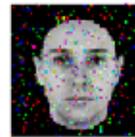
For example, here is the faces from database[2], [3] :

Exemple image, face
from data base



However, the imaging system is not perfect and the faces may suffer from noise [4], [5]:

"salt and pepper" noise is added
to the exemple image



"poisson" noise is added
to the exemple image



"speckle" noise is added
to the exemple image



"gaussian" noise is added
to the exemple image



Perfect classification of ideal input images is required, and reasonably accurate classification of noisy images.

The ninety-four 960-pixel input images are defined as a matrix of input vectors (Image size ~ 32 x 30) [5]. The target vectors are also defined with a variable called targets[5],[6]. Each target vector is a 94-element vector with a 1 in the position of the face it represents, and 0's everywhere else[5]. For example, the face number one is to be represented by a 1 in the first element (as this example is the first face of the database), and 0's in elements two through ninety-four[5].

3. Neural network

The network will receive the 960 real values as a 960-pixel input image (Image size $\sim 32 \times 30$). It will then be required to identify the face by responding with a 94-element output vector[5].

The 94 elements of the output vector each represent a face. To operate correctly the network should respond with a 1 in the position of the face being presented to the network All other values in the output vector should be 0 [5].

In addition, the network should be able to handle noise. In practice the network will not receive a perfect image of face which represented by vector as input. Specifically, the network should make as few mistakes as possible when classifying images with noise of mean 0 and standard deviation of 0.2 or less.

4. Architecture of neural network

The neural network needs 960 inputs and 94 neurons in its output layer to identify the faces. The network is a two-layer log-sigmoid/log-sigmoid network [6], [7], [8]. The log-sigmoid transfer function was picked because its output range (0 to 1) is perfect for learning to output Boolean values (see figure1) [5].

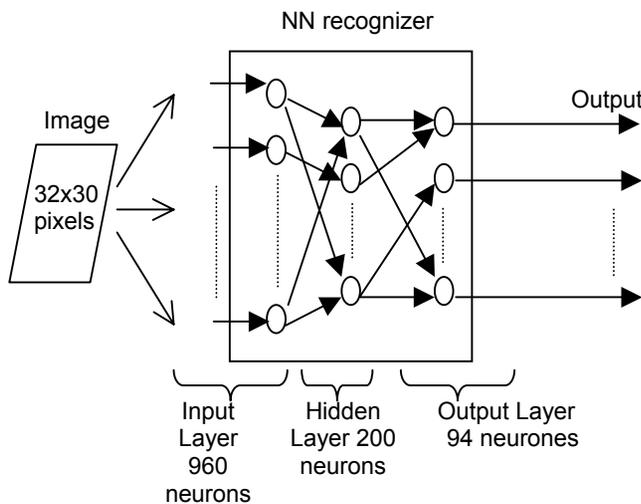


Figure 1. Architecture of neural network

The hidden layer has 200 neurons [5]. This number was picked by guesswork and experience[5]. If the network has trouble learning, then neurons can be added to this layer [5], [9].

The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. However, noisy input images may result in the network not creating perfect 1's and 0's. After the network has been trained the output will

be passed through the competitive transfer function. This function makes sure that the output corresponding to the face most like the noisy input image takes on a value of 1 and all others have a value of 0. The result of this post-processing is the output that is actually used [9].

5. Training

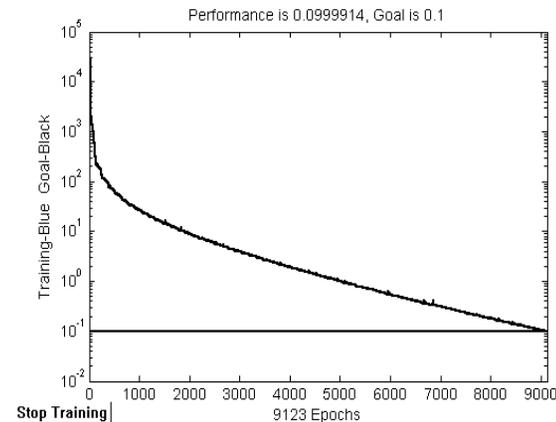
To create a neural network that can handle noisy input images it is best to train the network on both ideal and noisy images. To do this the network will first be trained on ideal images until it has a low sum-squared error. Then the network will be trained on 10 sets of ideal and noisy images. The network is trained on two copies of the noise-free database at the same time as it is trained on noisy images. The two copies of the noise-free database are used to maintain the network's ability to classify ideal input images. Unfortunately, after the training described above the network may have learned to classify some difficult noisy images at the expense of properly classifying a noise free image.

Therefore, the network will again be trained on just ideal images. This ensures that the network will respond perfectly when presented with an ideal face.

All training is done using backpropagation with both adaptive learning rate and momentum.

5.1. Training without noise (network1 "net")

The network is initially trained without noise for a maximum of 10 000 epochs or until the network sum-squared error falls below 0.1 (see this figure).



5.2. Training with noise(network 2 "netN")

To obtain a network not sensitive to noise, we trained with two ideal copies and two noisy copies of the images in database. The noisy images have noise of mean 0.1 ("salt & pepper" noise) and 0.2 ("poisson")

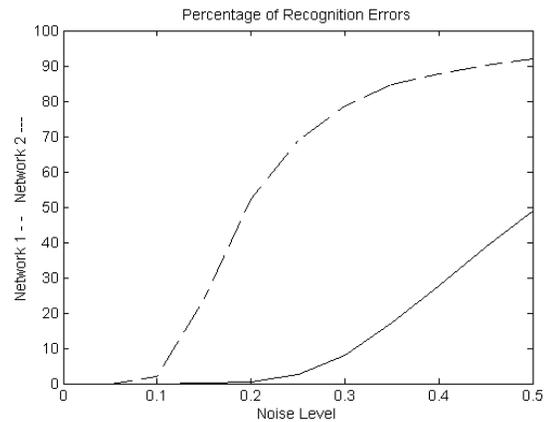
noise) added to them. This forces the neurons of network to learn how to properly identify noisy faces, while requiring that it can still respond well to ideal images. To train with noise the maximum number of epochs is reduced to 300 and the error goal is increased to 0.6, reflecting that higher error is expected due to more images, including some with noise, are being presented.

5.3. Training without noise again

Once the network has been trained with noise it makes sense to train it without noise once more to ensure that ideal input images are always classified correctly.

6. System performance

The reliability of the neural network pattern recognition system is measured by testing the network with hundreds of input images with varying quantities of noise. We test the network at various noise levels and then graphs the percentage of network errors versus noise. Noise with mean of 0 and standard deviation from 0 to 0.5 are added to input images. At each noise level 100 presentations of different noisy versions of each face are made and the network's output is calculated. The output is then passed through the competitive transfer function so that only one of the 94 outputs, representing the faces of the database, has a value of 1. The number of erroneous classifications are then added and percentages are obtained (see this figure) :



The solid line (black line) on the graph shows the reliability for the network trained with and without noise. The reliability of the same network when it had only been trained without noise is shown with a dashed line. Thus, training the network on noisy input images of face greatly reduced its errors when it had to classify or to recognize noisy images.

The network did not make any errors for faces with noise of mean 0.00 or 0.05 . When noise of mean 0.10 was added to the images both networks began to make errors. If a higher accuracy is needed the network could be trained for a longer time or retrained with more neurons in its hidden layer. Also, the resolution of the input images of face could be increased to say, a 640 by 480 matrix.

Finally, the network could be trained on input images with greater amounts of noise if greater reliability were needed for higher levels of noise.

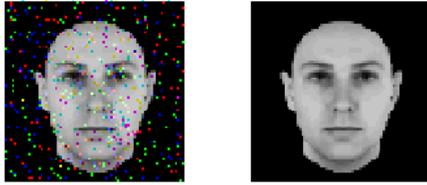
7. Test

To test the system, a face with noise can be created and presented to the network [5], [10]. (see table 1 for more example of faces with different kind of noise).

Table 1. Recognition results on the face image database (Image size ~ 32 x 30) on a PC with 1.7 GHz CPU, RR: Recognition Rate.

Architecture Of Neural Network	Training & Test database	Time of Training (sec)	Recognition Rate (RR)				
			Database Without Noise	Database With Noise			
				Salt & Pepper noise	Poisson noise	Speckle noise	Gaussian noise
Artificial Neural Network1 "Net"	CMU faces, MIT faces & Web faces	1 hour 51 min 06 sec	100 %	81.91 %	100 %	93.62 %	94.68 %
Artificial Neural Network2 "Netn"	CMU faces, MIT faces & Web faces.	2 hour 16 min 42 sec	100 %	100 %	100 %	96.81 %	96.81 %

Here is the noisy face and the face which the network picked (correctly):



8. Conclusion

Face recognition is challenging problems and there is still a lot of work that needs to be done in this area. Over the past ten years, face recognition has received substantial attention from researchers in biometrics, pattern recognition, computer vision, and cognitive psychology communities[1]. This common interest in facial recognition technology among researchers working in diverse fields is motivated both by the remarkable ability to recognize people and by the increased attention being devoted to security applications. Applications of face recognition can be found in security, tracking, multimedia, and entertainment domains.

We have demonstrated how a face recognition system can be designed by artificial neural network. Note that the training process did not consist of a single call to a training function. Instead, the network was trained several times on various input ideal and noisy images of faces.

In this case training a network on different sets of noisy images forced the network to learn how to deal with noise, a common problem in the real world.

9. References

- [1] Rein-Lien Hsu, "Face Detection and Modeling for Recognition," PhD thesis, Department of Computer Science & Engineering, Michigan State University, USA, 2002.
- [2] Tom Mitchell, "neural net & Face images", Home work 3, CMU, Carnegie Mellon University, Pittsburgh, USA, October 1997.
- [3] David J.Beymer, "Pose-Invariant face Recognition Using Real & Virtual Views", PhD thesis, MIT, USA, 1996.
- [4] Encyclopaedic, ©1993-2002 Microsoft Corporation, Collection Microsoft® Encarta® 2003.
- [5] Réda Adjoudj, "Détection & Reconnaissance des Visages En utilisant les Réseaux de Neurones Artificiels", Thèse de MAGISTER, Spécialité Informatique, Option Intelligence artificielle, Université de Djillali Liabès, département d'informatique, SBA, Algeria, October 2002.

[6] Henry A.Rowley, "Neural Network-based face detection" PhD thesis, Carnegie Mellon University, Pittsburgh, USA, May 1999.

[7] Jeffrey S. Norris, "Face Detection and Recognition in Office Environments", Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, M.I.T, USA, May 21, 1999.

[8] Ying-Li Tian & Takeo Kanade, "Recognizing Action Units For Facial Expression Analysis", Carnegie Mellon University, Pittsburgh, USA, 1999.

[9] Howard Demuth, Mark Beale, "Neural Network Toolbox User's Guide For Use with MATLAB", by The MathWorks, Inc. 1998.

[10] John Daugman, "Face and Gesture Recognition: Overview" IEEE PAMI, vol.19, no.7, July 1997.