Face Recognition by Artificial Neural Network using MATLAB Toolbox

Aman Arora
Systems Engineer
Infosys Limited, Pune
Amanarora9909102169ece@gmail.com

Dishant Chawla
Software Engineering Analyst
Accenture Services Pvt. Ltd., Mumbai
Dishantchawla19@gmail.com

Kinjal Thakkar
Systems Engineer
Infosys Limited, Pune
thakkar.kinjal14@gmail.com

Bhavika Bhanushali
Software Engineering Analyst
Accenture Services Pvt. Ltd, Mumbai
bhanushalibhavika@gmail.com

Sheetal Thakkar
Computer Technology Department
Shah & Anchor Kutchhi Polytechnic, Mumbai
skthakkar2@gmail.com

Abstract— Technology has always aimed at making human life easier and Artificial Neural Network has played an integral part in achieving this. Neural networks include simple elements operating in parallel which are inspired by biological nervous systems. These networks can be trained to perform specific task which is remedy for the problems faced by conventional computers or human beings. This electronic document mainly focuses on implementation of face recognition software which uses Neural Network tool box of MATLAB with back propagation algorithm. This software recognizes faces from standard set of images and also calculates error probability.

Keywords- *Artificial neural networks; neurons; back propagation algorithm; MATLAB toolbox; feedforward networks; face recognition algorithm.*

*****

I. Introduction

A. Biological inspiration:

Human beings have ability to react adaptively and accept the changes in their external and internal environment. Humans use their nervous system to perform this behavior. A similar model to our nervous system can be used to perform similar response and behavior in artificial system. Human nervous system communication includes synapses, dendrites and axon. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This model can be easily compared with artificial neuron model. It basically consists of input ‘p’ (like synapses), which is multiplied by weights ‘w’ (strength of signals), and then passed through a mathematical function resulting in output ‘a’ which determines the activation of the neurons.

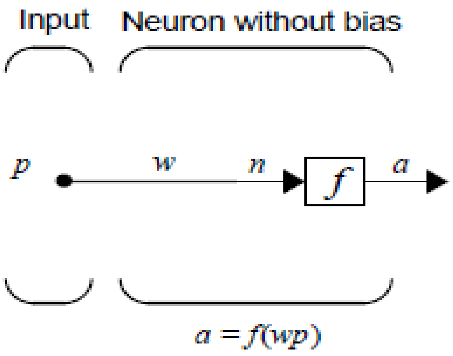


Figure 1. A neuron with a single scalar input and no bias[1]

B. Artificial Neural Network (ANN) architecture:

Many neurons can be combined in a layer and similarly a particular network could have one or more such layers. In this network, each element of the input vector 1 to X^m is connected to each neuron input through the weight matrix 1 to W^m. The neuron has a bias *b*, which is summed with the weighted inputs. This sum is passed through the transfer function f to get the neuron’s output. Multiple weights and bias can be used to train our neural network to get the desired output.

$$\text{Output} = \int (\sum (X^m W^m) + \sum b^m)$$

II. Characteristics of ANN

- Massive parallelism: Artificial Neural networks includes simple elements operating in parallel which makes its operating fast and this characteristics can offer data compression, because information can be stored in the weights of the connections.
- Distributed representation and computation: In ANN systems a given activation pattern is assigned a semantic interpretation; such assignments are referred to as “distributed representation”.[4]
- Learning ability: Neural network has ability to get trained to perform a particular function by adjusting the values of the connections (weights) between elements.
- Generalization ability: It is ability to handle unseen data. Generalization ability is determined by system complexity and training of network. Poor generalization is occurred when the network is over trained or system complexity is relatively more than the training data.
- Adaptability: Artificial neural networks can be designed directly or trained to find minimum error solution. In addition, linear Networks can be trained adaptively allowing the network to track changes in its environment.

- Inherent contextual information processing: ANN can be designed to have an innate characteristics to process the information depending upon the context.
- Fault tolerance: Small destruction of a network leads to a corresponding degradation of performance. However, some network capabilities may be retained even after major damage due to this feature.
- Low energy consumption: To optimize energy consumption behaviors, various techniques to monitor energy consumption may be adopted.

III. APPLICATIONS OF ANN

Artificial Neural Networks (ANNs) can be used to solve a variety of problems such as pattern recognition, prediction, associative memory, optimization and control. Conventional approaches could solve these problems but they are not flexible enough to perform well outside their domain. ANNs provide exciting alternatives and many applications could benefit by using them.

ANN is widely utilized in the world of aerospace and automobiles. It can be to track unusual credit card transactions, which would help to detect frauds. It is used to read and check documents in the banking and other financial fields like insurance policies. It has huge application in army and defense field. It used for Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times in the medical field. The combination of robotics and ANN seems to be popularizing in the technical world. Neural networks are being trained to predict the output gasses of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past. ANN can be used for exploration of oil and gas. Besides, it can be used for voice recognition, gesture recognition, text to speech conversion and etc. Use of ANN in the entertainment industry is also increasing. It is also applicable in the manufacturing, marketing and other departments of industrial sector. [1]

IV. BACKPROPAGATION ALGORITHM

Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. The Backpropagation algorithm gives a solution to the learning mind with minimum of the error function in weight space using the method of gradient descent. Properly trained backpropagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs. There are generally four steps in the training process:

1 Assemble the training data
2 Create the network object
3 Train the network
4 Simulate the network response to new inputs[5]

V. MATLAB TOOLBOX

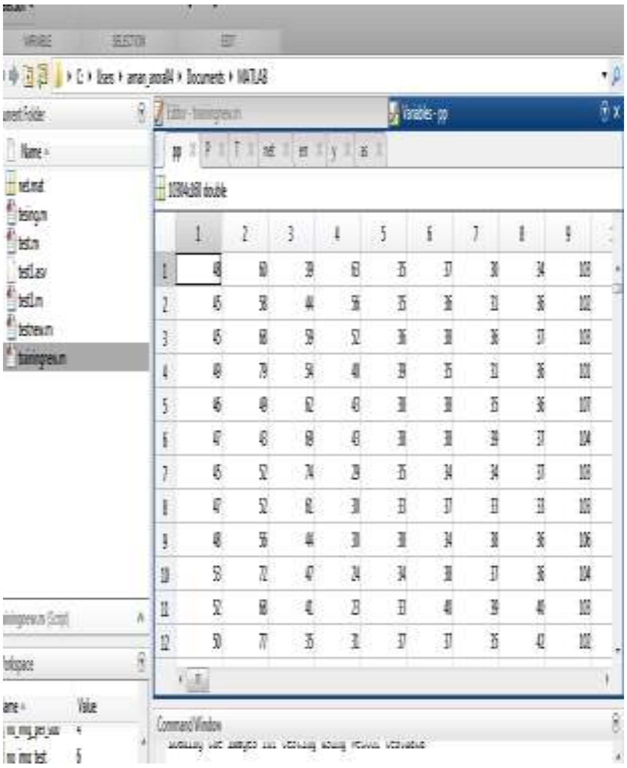
Neural Network provides many inbuilt tools for designing, implementing, visualizing, and simulating neural networks. Neural networks are used for applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control. Neural Network Toolbox supports feedforward networks, radial basis networks, dynamic networks, self-organizing maps, and other proven network paradigms.

VI. IMPLEMENTATION

A. *Face Recognition Algorithm:*

Step 1.Creating input matrix (PP)

We have standard set of images provided by AT&T. This set includes 10 images of 40 persons each. For the training purpose we take 4 images of each person. Each image is of size 112*92 pixels resulting into 10304 pixels. So the input matrix is of size 10304*160.



10304x160 double

	1	2	3	4	5	6	7	8	9
1	48	60	39	63	35	37	38	34	103
2	45	58	44	56	35	36	31	36	102
3	45	60	59	52	36	38	36	37	103
4	49	79	54	41	39	35	31	36	101
5	46	49	62	43	38	38	35	36	107
6	47	63	69	43	38	38	39	37	104
7	45	52	74	39	35	34	34	37	103
8	47	52	62	38	33	37	33	33	103
9	48	55	44	38	38	34	38	36	106
10	53	72	47	34	34	38	37	36	104
11	52	60	41	33	33	41	39	41	103
12	50	77	35	32	37	37	35	42	102

Figure 2. Input Matrix

Step 2. Normalizing input matrix (P)

The input matrix is then normalized which scales the data in matrix in range [-1 1].

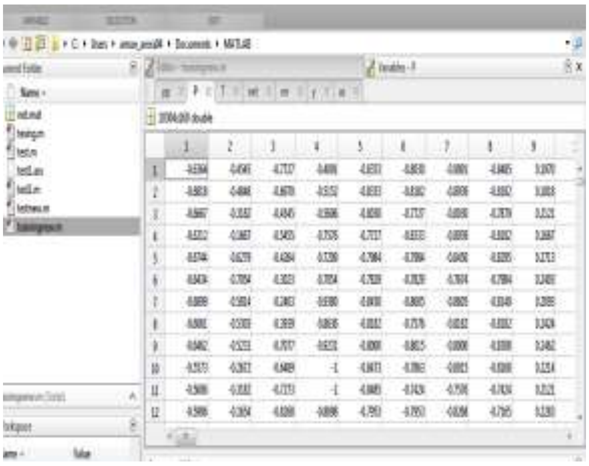


Figure 3. Normalized Input Matrix

Step 3.Creating target matrix (T)

Target matrix is of size 40*160. Data in this matrix is as follows, first 4 images corresponds to first person, so first 4 entries in first row of matrix will be 1, similarly other 4 entries of other row will be 1.

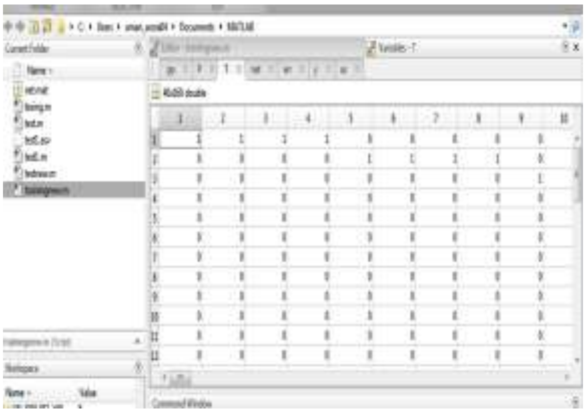
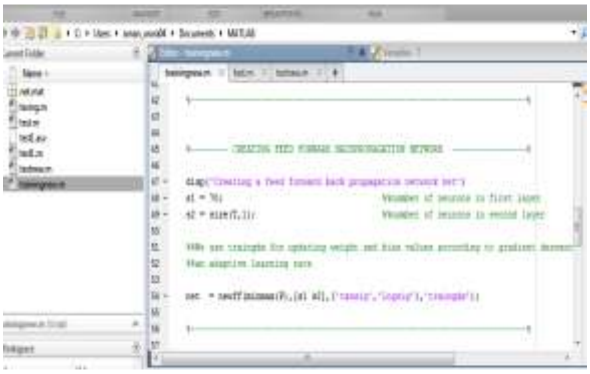


Figure 4. Target Matrix

Step 4.Creating feed forward back propagation matrix (NET)

A feedforward network is set up with the help of inbuilt MATLAB functions. This function creates a structure of neurons used. We also specify transfer function we utilized in code.

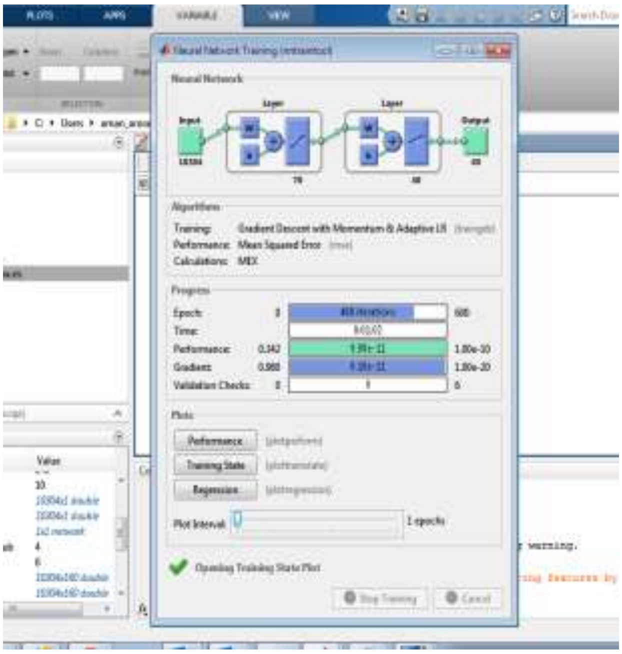


```
%------ CREATING FEED FORWARD BACKPROPAGATION NETWORK ------

disp('Creating a feed forward back propagation network net')
s1 = 70;                              %Number of neurons in first layer
s2 = size(T,1);                       %Number of neurons in second layer

%We use traingdx for updating weight and bias values according to gradient descent
%wit adaptive learning rate

net = newff(minmax(P),[s1 s2],{'tansig','logsig'},'traingdx');
```

Figure 5. Creating Network

Step 5. Training the created network

In this we set the values of several variables which are part of training purpose. For example learning rate, Epochs between showing progress, Performance goal.



Neural Network Training (nntraintool)

Algorithms
Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress
Epoch: 0 | 4[illegible] iterations | 600
Time: 0:00:02
Performance: 0.342 | 9.39e-11 | 1.00e-10
Gradient: 0.980 | 9.18e-11 | 1.00e-20
Validation Checks: 0 | 0 | 6

Opening Training State Plot

Figure 6. NN Training

Step 6.Creating testing matrix (PT)

The remaining 6 images of each person is used to test our software. The size of input matrix is 10304*240.

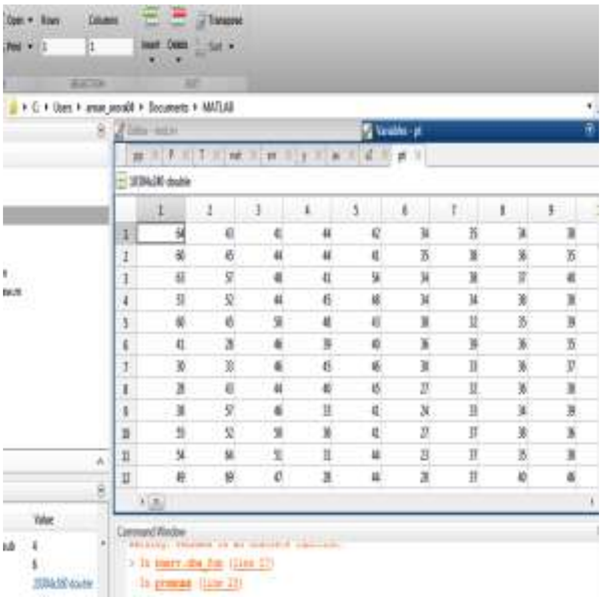


Figure 7. Testing Matrix

Step 7.Normalising testing matrix (TD)

The input matrix is then normalized. Normalization scales the data in matrix in range [-1 1].

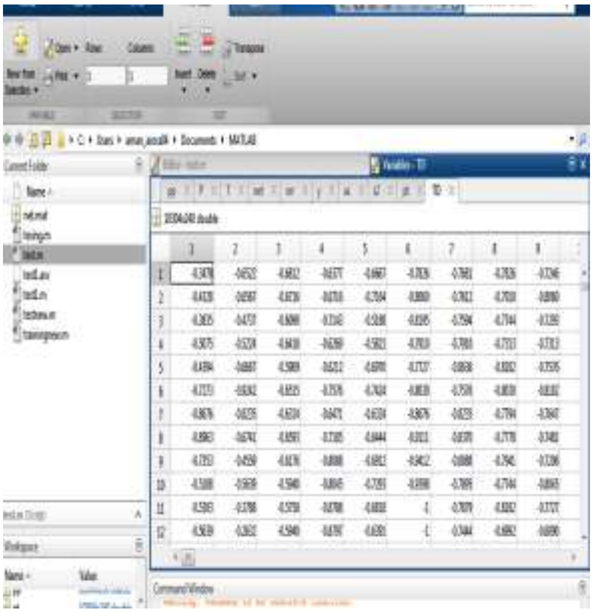


Figure 8. Normalized Testing matrix

Step 8. Testing the network

The Testing matrix (TD) and net is simulated and final output is obtained which determines the percentage error of our software.

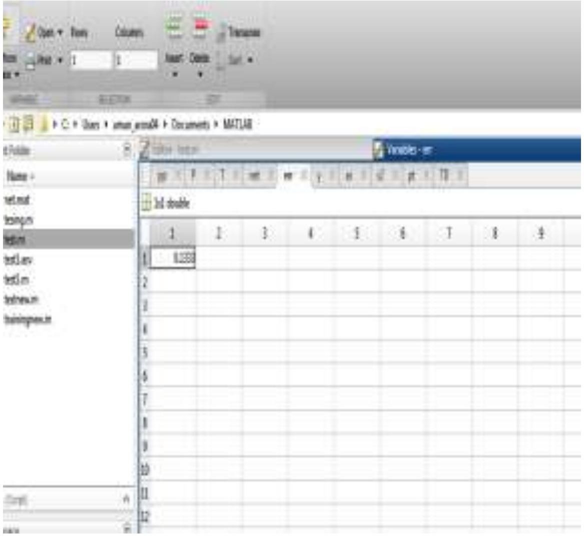


Figure 9. Error Probability

B. *Commands used:*

- Imread: Reads a grayscale or color image from the file specified by the string filename.
- Premnmx: It is used to scale the matrix in range [-1 1].
- Newff: The function newff creates a feedforward network. It requires four inputs and returns the network object. The first input is an R by 2matrix of minimum and maximum values for each of the R elements of the input vector. The second input is an array containing the sizes of each layer. The third input is a cell array containing the names of the transfer functions to be used in each layer. The final input contains the name of the training function to be used.
- Traingdx: The batch steepest descent training function is traingdx. The weights and biases are updated in the direction of the negative gradient of the performance function. If you want to train a network using batch steepest descent, you should set the network trainFcn to traingdx, and then call the function train. There is only one training function associated with a given network.

VII. RECENT ADVANCEMENT AND FUTURE APPLICATIONS OF ANN

Various avenues of ANN were studied and researched during the above case study. As ANN being such a wide area to explore, its future scope is very vast and soon to be the technology establishing its supremacy in all domains. The future advancements and applications of ANN are as follows:

- Research on design of interfaces between humans and computers (Human computer interfaces).
- Smart rooms can be designed for finest utilization of power with the help of ANN.
- Neuro-fuzzy Systems- This is the term used for integration of fuzzy logic and ANN systems which combines the human-like reasoning style of fuzzy system.
- Simulation for biological neural networks which communicate with the pulse timings has stimulated research on pulsed neural networks.
- Systems for monitoring and surveillance for scrutinizing, storing and retrieving health information of patients can be integrated with ANN
- Intelligent robots, the cutting-edge research in the field of ANN could be implemented for complex and integrated day-to-day tasks.
- Biomedical image analysis is the next avenue with special emphasis on efforts related to the applications of computer vision, virtual reality and robotics to biomedical imaging problems

VIII. CONCLUSION

In this paper we have discussed relation of biological neurons with the artificial neurons, artificial neural network architecture, major characteristics of ANN, applications of ANN, advantageous and disadvantageous of ANN, backpropagation algorithm and MATLAB toolbox used for implementation of software. Along with this we have also cited our software implementation of ANN on MATLAB with the steps in sequence. This software is able to recognize a face from a standard set of images using backpropogation algorithm and calculates error probability of our software.

REFERENCES

[1] Howard Demuth, Mark Beale, Martin Hagan “Neural Network Toolbox 6 – User’s Guide”

[2] Martin T.Hagan “Neural network design”

[3] Simon Haykin “Neural network – A Comprehensive foundation”

[4] “Mathematical Methods for Neural network Analysis and Design” by Richard M. Golden

[5] K. V. Kale, Ramesh Raybhan Manza, “Advances in Computer Vision and Information Technology”

[6] http://www.journals.elsevier.com/medical-image-analysis/

[7] N. Revathy, T. Guhan, “Face Recognition System Using Back Propagation Artificial Neural Networks” IJAET Volume-3, Issue-1.

[8] “Artificial Neural Network”, Neha Gupta, IISTE, Volume-3

[9] “Study of Artificial Neural Network”, by Kiran Sharma, Ankit Naik, Purushottam Patel, IJARTET, Volume-2, Issue-4.