

# Artificial Neural Network Design Flow for Classification Problem Using MATLAB

Dr. E. Jebamalar Leavline

Department of Electronics and Communication Engineering, Bharathidasan Institute of Technology,  
Anna University, Tiruchirappalli – 620 024, India

**Abstract**—Artificial neural network (ANN) is an important soft computing technique that is employed in a variety of application areas in the field of engineering and technology. Classification is a typical application where ANN is inevitable to perform supervised learning. This paper presents the ANN design flow for classification problem on biological data in MATLAB environment.

**Index Terms**— Artificial neural network, classification, machine learning, MATLAB

## I. INTRODUCTION

Artificial neural network is a mathematical model to solve engineering problems. It incorporates a group of highly connected neurons to realize compositions of non linear functions. In computational point of view, ANN is a method of representing functions wherein a network is formed using elements to carry out simple arithmetic operations. In biological viewpoint, it is a mathematical model for the operation of the brain where each computational element corresponds to a neuron. These neurons do information processing in the brain and interconnection of neurons is called as neural network [1].

The inspiration for the neural network concept is derived from Neurobiology. A neuron in the brain has many-inputs and one-output unit and the output can be excited or not excited. The incoming signals from other neurons determine if the neuron shall excite ("fire") or not and the output is subject to attenuation in the synapses, which are junction parts of the neuron. The structure of a biological neuron or nerve cell is shown in Figure 1 [2, 3].

The rest of the paper is organized as follows. Section II describes artificial neuron model and neural network structures. Section III describes the process of learning and classification. Design of ANN for classification using MATLAB is presented in Section IV and the details of the data set are summarized in Section V. The step by step Procedure for classification is presented in Section VI and Section VII concludes the paper.

## II. ARTIFICIAL NEURON MODEL AND NEURAL NETWORK STRUCTURES

Inspired by the biological neuron, the artificial neuron is modeled as shown in Figure 2.  $I_j$  are the inputs,  $W_j$  are the weights and  $T$  is the activation threshold. If the weighted sum

of inputs  $x$  is greater than  $T$ , the output is activated. This activation operation is performed with non linear functions such as step, sign and sigmoid functions.

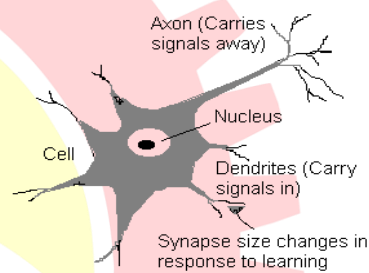


Figure 1. Structure of a biological neuron or nerve cell

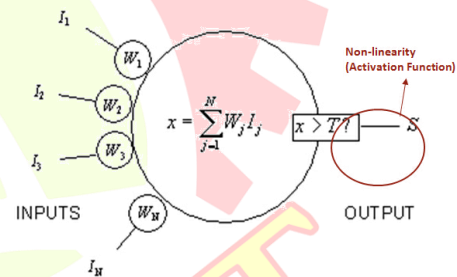


Figure 2. Artificial neuron model

This artificial neuron can be considered as a node and interconnection of nodes is called as network. Each node is connected by a link with numerical weights and these weights are stored in the neural network and updated through the process called learning. An ANN essentially has three layers of neurons namely, input layer, output layer and hidden layer as shown in Figure 3. Number of nodes in a network is determined based on the application requirement.

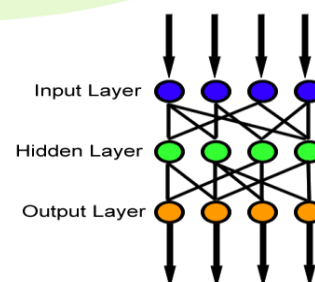


Figure 3. A feed-forward network with input, hidden and output layer

The neural networks can be classified based on the network structure as single layer and multi layered networks. According to input flow it is classified as feed forward neural networks and recurrent neural networks. Feed forward networks have unidirectional links and the computations proceed uniformly from input to output. In recurrent networks, there is possibility of loops since the activation is fed-back to the units that caused it. The internal state conditions are stored in the network itself. The recurrent networks are unstable, oscillatory and exhibit chaotic behavior and the learning is slow compared to feed forward networks.

Hence optimal network structure and size has to be chosen for a particular application. If the network is too small, it may be incapable of representing desired function. On the other hand, if it is too big, it cannot memorize all training examples and may not able to generalize the function [1, 2].

### III. LEARNING AND CLASSIFICATION

#### A. Learning

The procedure that consists in estimating the parameters of neurons (setting up the weights) so that the whole network can perform a specific task is called learning. There are types of learning namely supervised learning and unsupervised learning.

Supervised learning present the network a number of inputs and their corresponding outputs (Training) and see how closely the actual outputs match the desired ones. Further, the network parameters are updated to better approximate the desired outputs. This is done in several passes over the data. In this case, the real outputs of the model for the given inputs are known in advance. So, the task of the network is to approximate those outputs. A “Supervisor” provides examples and teaches the neural network how to fulfill a certain task.

Unsupervised learning groups typical input data according to some function. The network itself finds the correlations between the data and groups the data. This needs no supervisor.

There are several learning methods proposed in the literature such as error correction learning, memory based learning, Hebbian learning, Competitive and Boltzman learning. Link network structure, the learning method also needs to be chosen in accordance with the application [1].

#### B. Classification

In machine learning, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership (label) is known. The classification process is considered as an instance of supervised learning with a training set of correctly identified observations. A computer based algorithm that implements classification is known as a classifier. The classification algorithm uses a mathematical

function to map the data to a category.

There are several types of classifiers such as statistical classifier, Bayesian classifiers, linear classifier, nearest neighbor classifier, support vector machine, and neural network based classifier [4].

#### C. ANN Based Classifiers

ANN based classifiers are widely used for classification in pattern recognition applications. ANN will be trained to act as a classifier for a given problem. This includes the linear perceptron classifier, feed-forward networks, radial basis networks, recurrent networks and learning vector quantization (LVQ) networks structures.

ANN based classification works in two phases namely training phase and testing phase. During training, the labeled data are used to train (learning) the network and to build the classifier model. During testing, the trained network (model) is given with the unlabeled data and it is classified based on the model built during training phase.

### IV. DESIGN OF ANN FOR CLASSIFICATION USING MATLAB

MATLAB is a multi-paradigm numerical computing environment [5]. It allows the use of a set of predefined functions called Toolboxes and the user defined functions and programs. MATLAB Neural Network Toolbox™ provides tools for designing, implementing, visualizing, and simulating neural networks for various applications. Neural Network Toolbox™ supports various ANN architectures as shown in Table 1.

TABLE 1. SUPPORTED NETWORK ARCHITECTURES IN MATLAB

Supervised Networks	Unsupervised Networks
Feed-forward networks	Competitive layers
Radial basis networks	Self-organizing maps
Recurrent networks	
Learning vector quantization (LVQ)	

#### A. Neural Network Design Flow for Classification

MATLAB facilitates the user to develop applications using graphical user interface (GUI) or command line interface (CLI). The general ANN design flow for classification is as follows.

- Step 1:** Collect data
- Step 2:** Create the network
- Step 3:** Configure the network
- Step 4:** Initialize the weights and biases
- Step 5:** Train the network
- Step 6:** Validate the network
- Step 7:** Use the network

The prior knowledge of the data set and the type of network to be used is required to build up the ANN for a specific classification problem. Generally the number of input and output layer nodes is the same as that of the number of input features that describes the data instance and the number of target (output) classes respectively. The number of hidden

layer neurons is generally greater than or equal to the number of output classes.

### V. DATA SET DESCRIPTION

In order to explain the ANN based classification problem, the dataset namely 'Iris Data Set' created by R.A. Fisher [6, 7] is collected from UCI machine learning repository [8] and the details of this data set are described in Table 2.

Data Set Characteristics	Multivariate
Attribute Characteristics	Real
Number of Instances	150
Number of Attributes	4
Number of Target Classes	3

This data set consists of 150 samples from each of three species of Iris flower family namely Iris setosa, Iris virginica and Iris versicolor as shown in Figure 4. Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other. This is a typical benchmark data set for the classification problem.



Figure 4. three species of Iris flower family (Iris setosa, Iris virginica and Iris versicolor)

### VI. STEP BY STEP PROCEDURE FOR ANN BASED CLASSIFICATION USING MATLAB

#### A. Classification Using GUI

**Step 1:** Open the pattern recognition/classification tool using 'nprtool'. This will generate a predefined network structure with a two layer feed forward network with sigmoid hidden and output neurons which uses a scale conjugate gradient back propagation algorithm for training the network.

**Step 2:** Load the Iris flower data set.

**Step 3:** Randomly divide the samples for training, validation and testing.

**Step 4:** Set the number of hidden layer neurons. Then the required network structure will be created as shown in Figure 5.

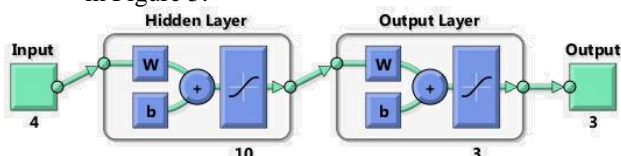


Figure 5. Network structure for classification of Iris data set with 10 hidden layer neurons

**Step 5:** Train the network. As a result, the training and testing results will be displayed as shown in Figure 6. MSE is the mean square error Percentage error (%E) denotes the fraction of misclassified samples.

	Samples	MSE	%E
Training:	104	1.83545e-4	0
Validation:	23	7.46799e-9	0
Testing:	23	5.74616e-2	8.69565e-0

Figure 6. Training results

**Step 6:** Plot confusion matrix and receiver operating characteristics (ROC) to analyze the classification performance.



Figure 7. Performance evaluation of classification using confusion matrix

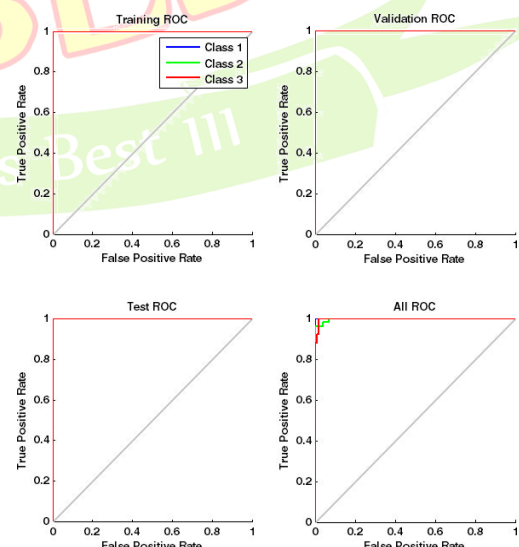


Figure 8. Performance evaluation of classification using receiver operating characteristics (ROC)

### B. Classification Using CLI

MATLAB allows the comments / functions to be entered in the command window. The sequence of comments can also be typed and saved in a script file and executed. The step by step procedure of the later is given below. The major advantage of this procedure is that the user has a variety of choices to select the network structures, learning algorithms and training algorithms.

**Step 1:** Open MATLAB.

**Step 2:** Open a new MATLAB script by selecting File → New → Script

**Step 3:** Type code in the MATLAB editor

**Step 4:** Save the script file with .m extension in the desired directory

**Step 5:** Press run button or F5 to execute the program

Similar to GUI, the performance evaluation shown in Figure 7 and Figure 8 can also be done. Figure 9 shows a sample MATLAB program for Iris data set classification using LVQ network and a sample program for the same with a probabilistic neural network is shown in Figure 10.

```
clc; clear all; close all;
[x,t] = iris_dataset; %Load data
net = lvqnet(10) % Define network (Learning Vector Quantization)
net = train(net, x, t); %Train the Network
view(net) %View network architecture
y = net(x); %Simulate the network
perf = perform(net, y, t) % Evaluate the performance
classes = vec2ind(y)
```

Figure 9. Iris data set classification using LVQ network

```
clc; clear all; close all;
[x,t] = iris_dataset; %Load data
net = newpnn(x,t); % Define network (Two layer network (radial basis, competitive))
view(net); %View network architecture
Y = sim(net, x); % Simulate the network
err=t-Y; % Simulate the network
perf=mse(err); % Evaluate the performance
plotconfusion(t,Y); % Plot confusion matrix
```

Figure 10. Iris data set classification using probabilistic neural network

### VII. CONCLUSION

This paper presented the step by step procedure for solving classification problem with ANN. The classification process can be done in MATLAB environment using GUI and CLI. This may serve as a guide for those who work with ANN based classification.

### REFERENCES

- [1] Simon Haykin, "Neural Networks: A Comprehensive Foundation", 2ed., Addison Wesley Longman (Singapore) Private Limited, Delhi, 2001.
- [2] Satish Kumar, "Neural Networks: A Classroom Approach", Tata McGraw-Hill Publishing Company Limited, New Delhi, 2004.
- [3] <http://ulcar.uml.edu/~iag/CS/Intro-to-ANN.html>
- [4] Singh, D. A. A. G., Balamurugan, S., & Leavline, E. L. (2012, August). Towards higher accuracy in supervised learning and dimensionality reduction by attribute subset selection-A pragmatic analysis. In IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp. 125-130.
- [5] [www.mathworks.com](http://www.mathworks.com)
- [6] R. A. Fisher, "The use of multiple measurements in taxonomic problems", Annals of Eugenics, 7 (2), pp. 179-188, 1936.
- [7] Edgar Anderson (1936). "The species problem in Iris". Annals of the Missouri Botanical Garden 23 (3): 457-509. JSTOR 2394164.
- [8] <https://archive.ics.uci.edu/ml/datasets/Iris>