CSC 369 2.0 Machine Learning I

An Approach for IRIS Plant Classification using Neural Networks

W.B.M.Perera
AS2016467

August 7, 2019

Introduction

Biology is the natural science of studying life and biological organisms, including chemical structure, molecular interactions, physical structure, physical mechanisms, development and evaluation. To extract the knowledge and solved the problems in biology, advanced computational technologies and algorithms such as ANN, Fuzzy Logic, Genetic Algorithms, Support Vector Machines are used.

ANN (Artificial Neural Network) is the one of AI (Artificial Intelligence) techniques commonly in use because it builds a relationship between complex input and outputs. ANN is a biology inspired information paradigm which is approach to build new systems using bio informatics. It is highly interconnected network of information processing elements that mimics the connectivity.

This is a review on how to classify the Iris plant to three groups which are IRIS Setosa, IRIS Versicolour, IRIS Veginica by examine the petal and sepal size of IRIS plant.

Classification goes under the supervised learning which predefined the class of instance before it use as a training set. This training set is used to develop the performance of ANN and it is a method to build a function that maps the input features in to an output space. All methods of classification assume some data knowledge. A test is then performed to determine the category of the input data set.

Since this IRIS data problem is also related to the classification problem we use supervised learning mechanisms to identify the categories of IRIS plants by using multi layer feed-forward neural networks and back propagation algorithms.

Contents

Chapter 1

Methodology

This section carried out how the IRIS process classification was organized with respect to all the steps.

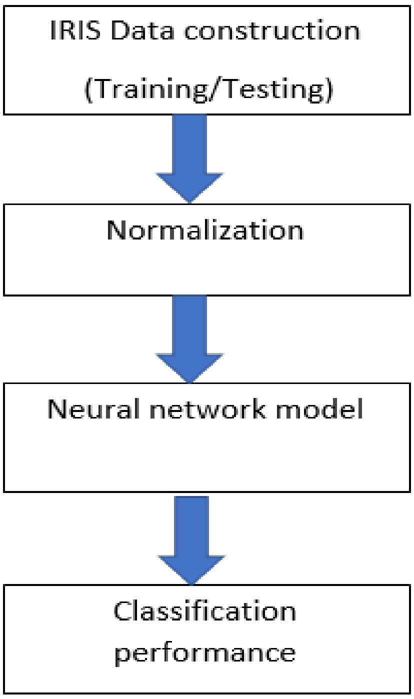

Figure 1.1: proposed model

1.1 Dataset construction

The approach of this research was separate the IRIS plants into to three categories according to the sepal length, petal length, sepal width and petal width. The IRIS data set was taken from the UCI repository. There are 150*4 instances. Out of these 150 75 instances were taken for training Neural Network under the supervised learning. Remain 75 instances were used to test the system performances. So the dataset was divided into major two categories which are training and testing. In each Training and Testing categories, there are 25 instances per IRIS Setosa, IRIS Versicolor and IRIS Virginnica. Under the supervised learning , the target output of the first 25 instances is taken as 0, for the next 25 instances as 0.5 and for the last 25 instances as 1

1.2 Normalization

It is more efficient to use normalized data since we using Neural Networks, which is supported numerical inputs. By using the following formula all the inputs are converted to the appropriate range.

Xij=(Xij –colmin)/(colmax –colmin)

- Xij=(data item) present at ith row jth column
- col min=minimum value of the column
- col max=maximum value of the column

1.3 Network Architecture

There are three layers in this Network which are input layer, hidden layer and output layer that was 4*3*1. Input layers has 4 nodes, hidden layer has 3 nodes and output layer has 1 node.

The weight matrix of input layer is 3*4 and weight matrix of hidden layer is 1*3. These two weight matrices were taken randomly.

1.4 Training Network

Training process was done via the back propagation algorithm that is known as Back propagation neural Network. Functional signal flows in forward direction and error signals propagate backward direction. Log sigmoid activation function is used to compute the outputs hidden layer and output layer. In this task back propagation is applied to the Neural network under the three circumstances which is 500 epochs, 1000 epochs and 5000 epochs. In each process weight is updated to give better outputs that has less errors. After the training process we get more appropriate values for Weight functions.

1.5 Back propagation Algorithm

Steps

Initialize all weights in network by randomly generate matrices;

Oj=Ij *//output of the input layer is actual input value;*

For each hidden or output layer

Ij=Σ I WijOi *//the net input of unit j*
Oj=1/(1+e-Ij) *// the output of each unit j*

Back propagation errors

For each unit j in the output layer

Sj=Oj(1-Oj)(Tj-Oj)
For each unit j in the hidden layer, from the last to the first hidden layer
Sj=Oj(1-Oj)Σ SkWij *//error with respect to the next higher* level

For each weight Wij in network
Δ Wij=(η)SjOi *//weight increment*
Wij= Wij+Δ Wij *//weight update*

1.6 Testing Process

After training process under the each three conditions, then the testing process is carried out. In this process system performance was tested by using the mean square error and number of classified instances. Against the each number of epochs

- Mean square Error
- Classification plot
- Classification table

Was taken to measure the performance of IRIS plant separation activity.

Chapter 2

Simulation Results

2.1 for 500 epochs

2.1.1 Mean Square Error Graph for 500 epochs

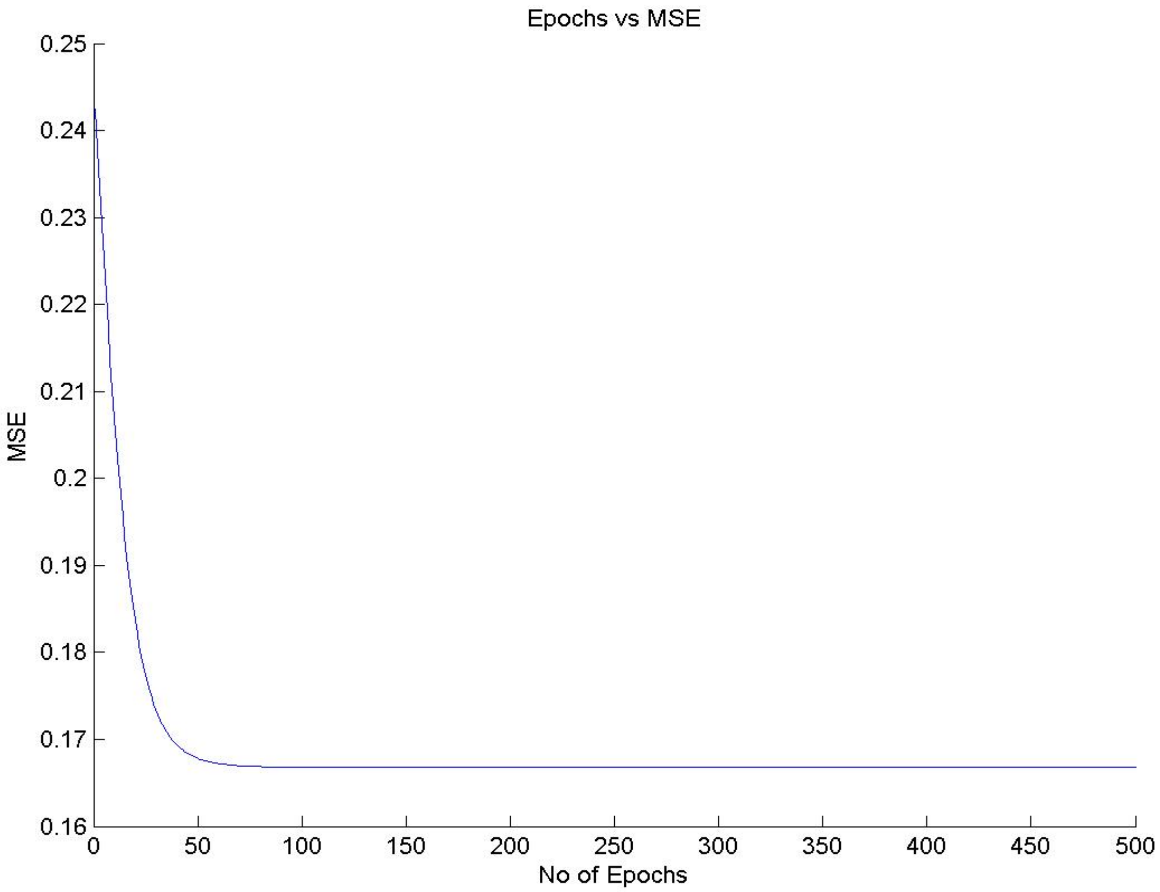

Figure 2.1: Mean Square Error Graph for 500 epochs at η=0.7

2.1.2 Result of classification for 500 epochs

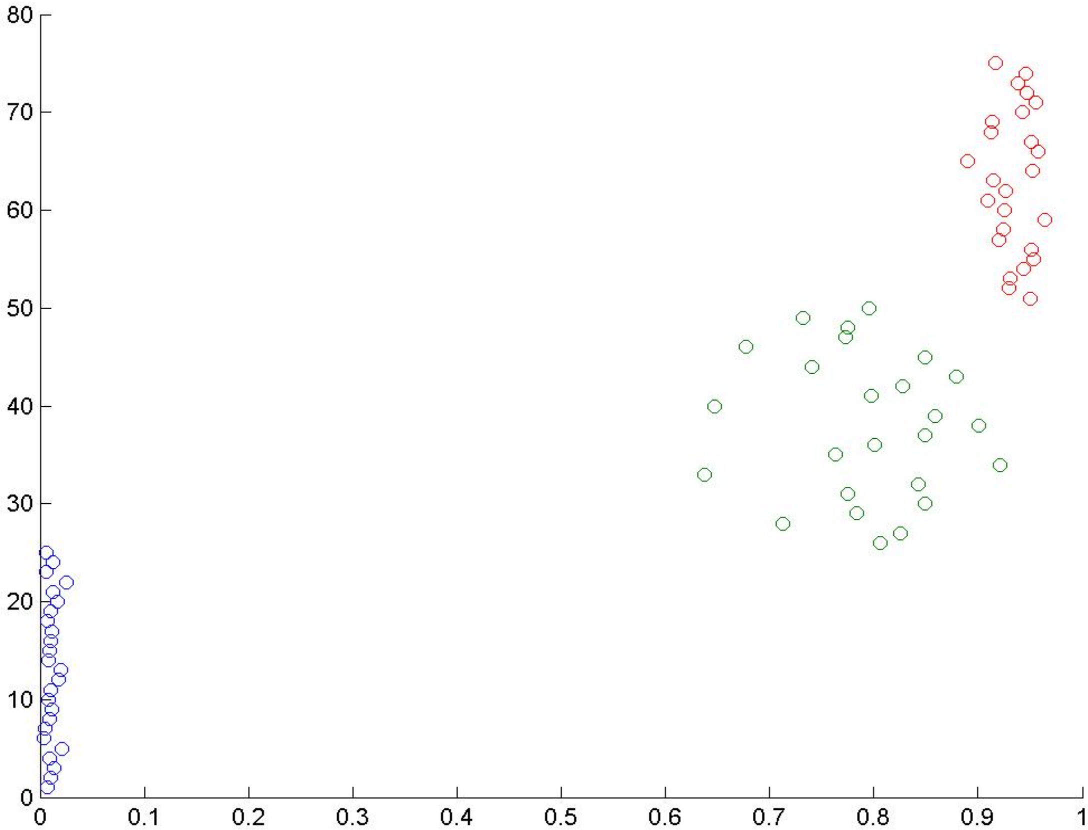

Figure 2.2: Result of classification for 500 epochs

2.1.3 Accuracy of Neural Network for 500 Epochs

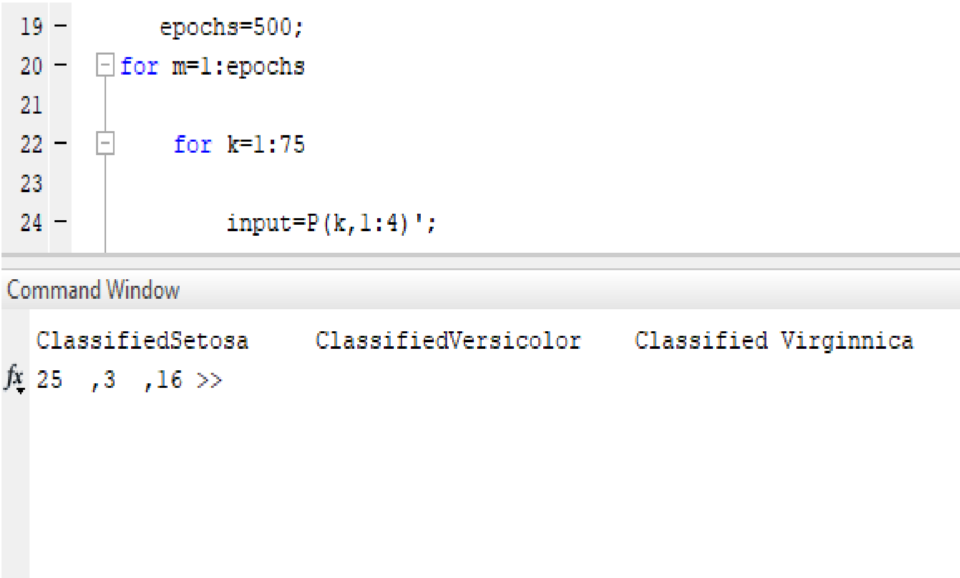

Figure 2.3: mat lab command result for number of classifications in 500 epochs

According to the figure 3 the following table is created.

Table 2.1: Test data classification for 500 epochs

IRISPlant	Total	Classified	Not Classified
Setosa	25	25	0
Versicolor	25	3	22
Virginnica	25	16	9

In 500 epochs, out of 25 instances of Setosa class all 25 instances are classified, Out of 25 instances of Versicolor class only 3 instances are classified and out of 25 instances of Virginnica class 16 instances are classified. As a total out of 75 instances 44 instances are classified correctly. Hence accuracy is 58.66%.

2.2 for 1000 epochs

2.2.1 Mean Square Error Graph for 1000 epochs

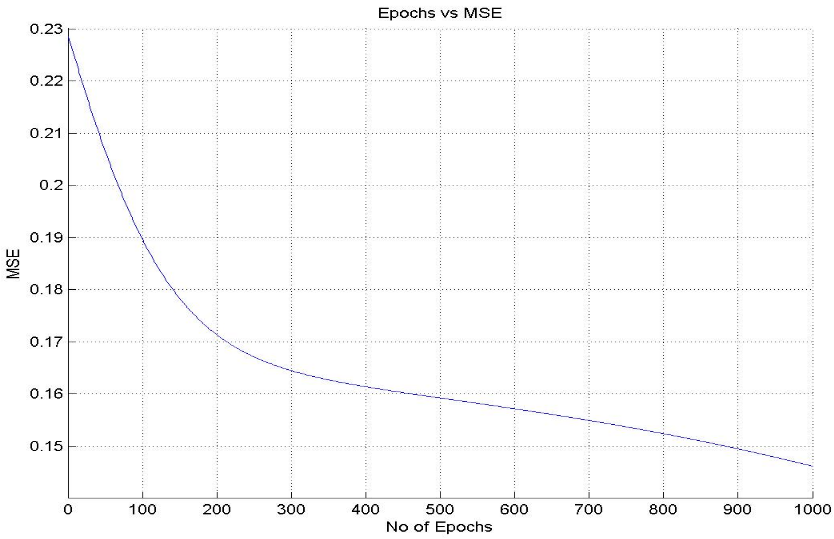

Figure 2.4: Mean Square Error Graph for 1000 epochs at η=0.7

2.2.2 Result of classification for 1000 epochs

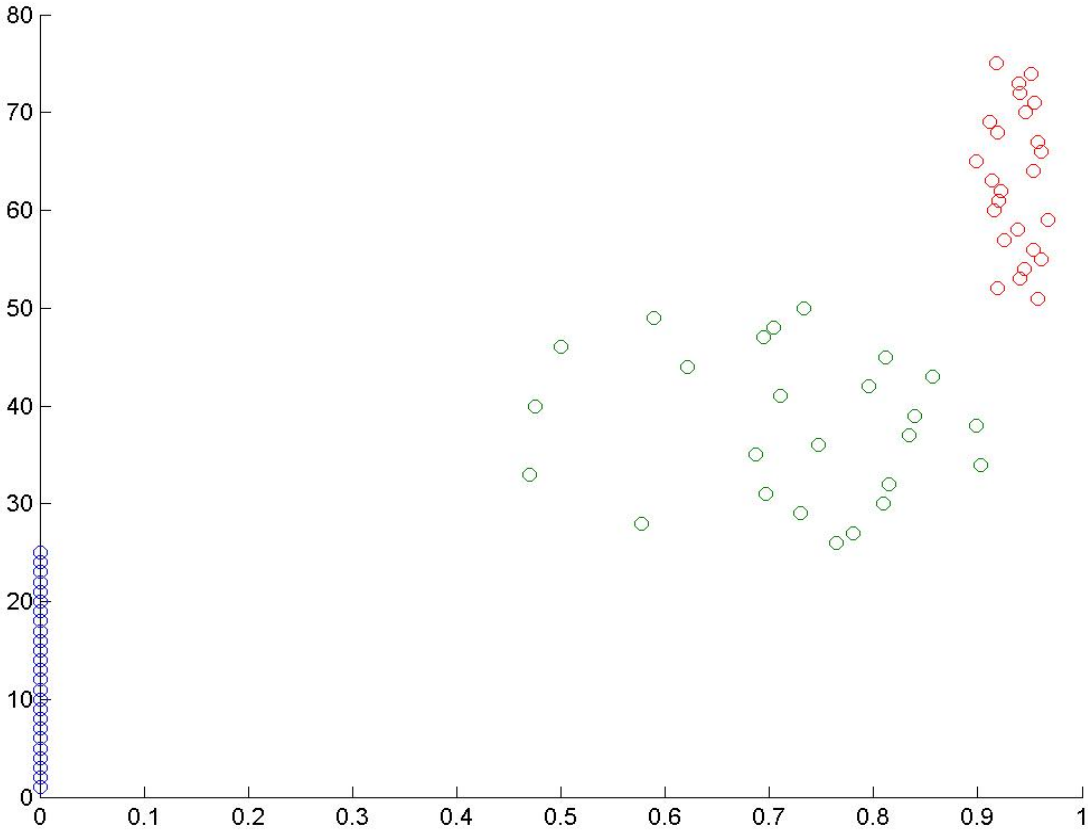

Figure 2.5: Result of classification for 1000 epochs

2.2.3 Accuracy of Neural Network for 1000 Epochs

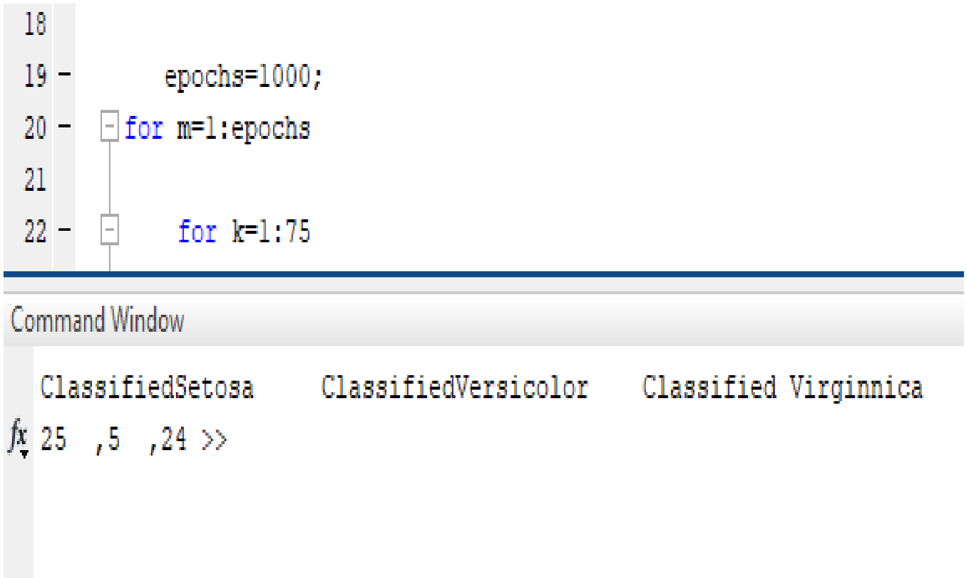

Figure 2.6: mat lab command result for number of classifications in 1000 epochs

According to the figure 2.6 the following table is created.

Table 2.2: Test data classification for 1000 epochs

IRISPlant	Total	Classified	Not Classified
Setosa	25	25	0
Versicolor	25	5	20
Virginnica	25	24	1

In 1000 epochs, out of 25 instances of Setosa class all 25 instances are classified, Out of 25 instances of Versicolor class only 5 instances are classified and out of 25 instances of Virginnica class 24 instances are classified. As a total out of 75 instances 54 instances are classified correctly. Hence accuracy is 72% .Performance is better than 500 epochs.

2.3 for 5000 epochs

2.3.1 Mean Square Error Graph for 5000 epochs

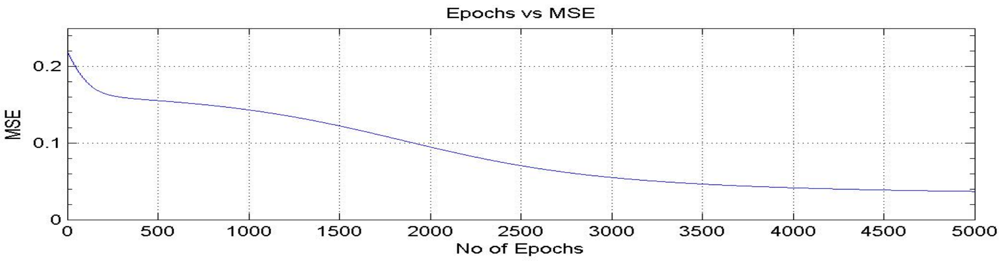

Figure 2.7: Mean Square Error Graph for 1000 epochs at η=0.7

2.3.2 Result of classification for 5000 epochs

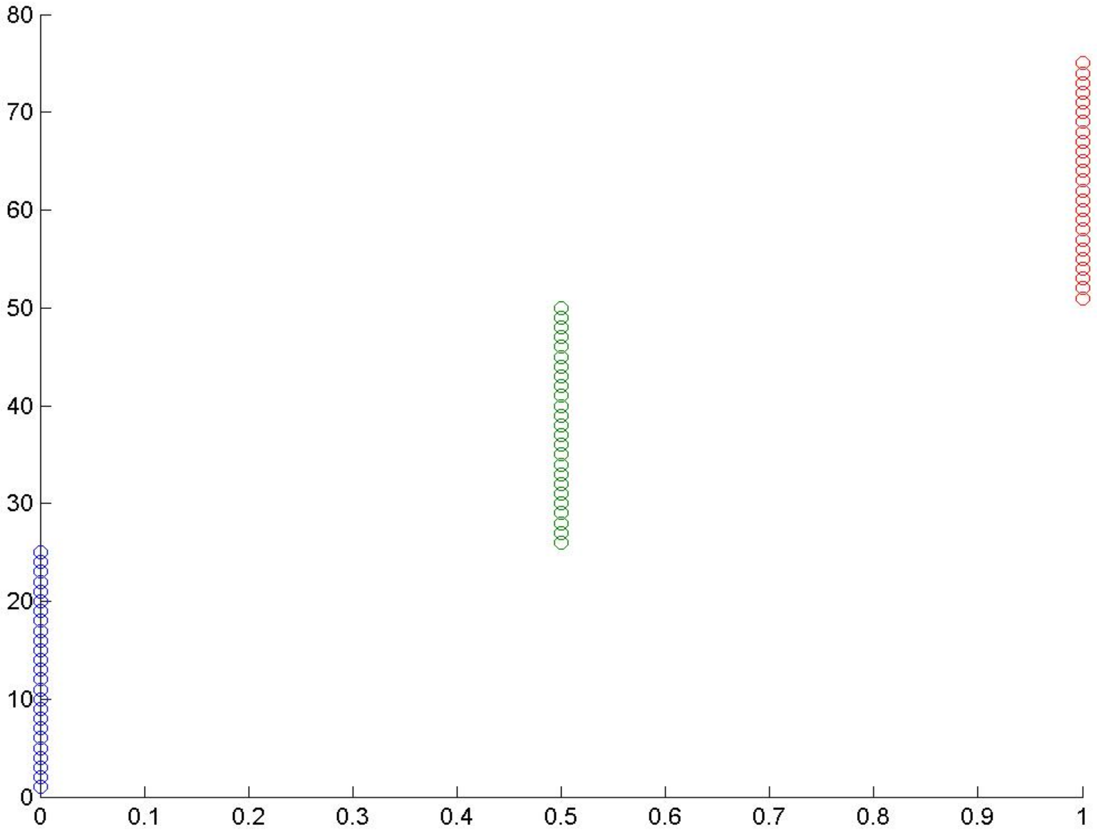

Figure 2.8: Result of classification for 5000 epochs

2.3.3 Accuracy of Neural Network for 5000 Epochs

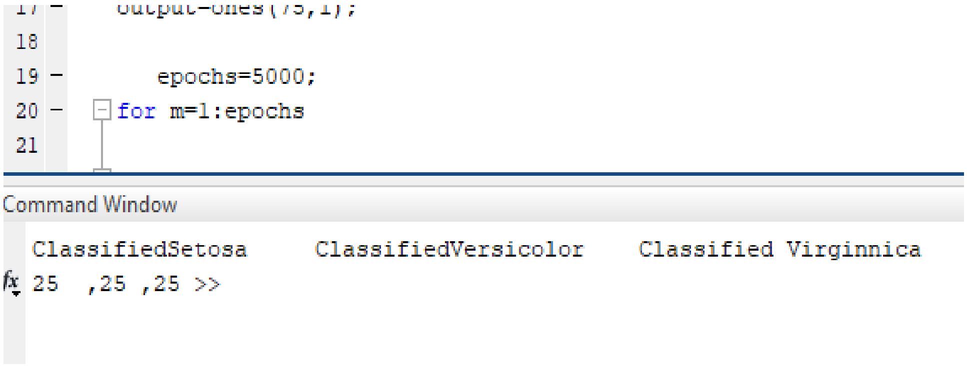

Figure 2.9: mat lab command result for number of classifications in 5000 epochs

According to the figure 3 the following table is created.

Table 2.3: Test data classification for 5000 epochs

IRISPlant	Total	Classified	Not Classified
Setosa	25	25	0
Versicolor	25	25	0
Virginnica	25	25	0

Out of 75 instances all instances are classified correctly. Hence accuracy is 100%. Performance is very high in the 5000 epochs.

Chapter 3

Discussion

- According to the results in above session increasing the number of epochs leads the data set for higher performances.
- But there is limitation of epochs since over training the Neural Network it is harmful to predict outputs for new data.
- Over training is a serious problem where NN reduces extent that is simply memorizes the data that used in training. Then it will be not possible to categorize the new data set and generalization is not possible, which is not required behaviour of the NN.
- To prevent over training validation of data set is important. As the training procedures the training error will decreases .
- Normalizing data before using training is also useful to scale the data.

Chapter 4

Conclusion

The multilayer Feed Forward Neural Network gives a satisfactory result, since it able to classify the three different categories of IRIS plant with minimal errors. Since we increase the number of epochs or the number of iterations the performance of Neural Network is increasing. At the beginning, when number of epochs are 500 accuracy level is 58.66%. And for 5000 epochs the accuracy level is approximately 100%. So, the number of epochs required to train the neural network range from 500 to 5000 and the accuracy ranges from 58.66% to 100%. From the above results, graphs and discussion, it is concluded that Multi-Layer Feed Forward Neural Network is faster and higher accuracy classification.

Chapter 5

Appendices

By changing the variable epochs 500, 1000, 5000 above results are carried out.Training and the testing process also contained in the same script.

```
%part 1-training data
%read normalized data from dataset excel sheet for training

clc;
data1=xlsread('dataset.xlsx','Training','B2:J76');
P1=data1(1:end,2);
P2=data1(1:end,4);
P3=data1(1:end,6);
P4=data1(1:end,8);
t=data1(1:end,9);
P=[P1,P2,P3,P4];


%initialize weight functions W1=ones(3,4);
W2=rand(1,3);

%epochs can be 500,1000,5000

epochs=500;
%initialize variables

niu=0.7;
MSE=ones(epochs,1);
error=ones(75,1);
err=ones(75,1);
output=ones(75,1);
```

```
for m=1:epochs

for k=1:75

input=P(k,1:4)';
n1=W1*input;
o1=logsig(n1);

%output of the input layer concerned as input to the hidden layer
n2=W2*o1;
output(k,1)=logsig(n2);

error(k,1)=t(k,1)-output(k,1);

err(k,1)=error(k,1)∧2;

S2=output(k,1)*(1-output(k,1))'*error(k,1);

a=[1;1;1];
j=a-o1;
S1=o1'*j*S2*W2;

dweight1=input*S1*niu;
dweight2=niu*S2*o1';

W1=W1+dweight1';
W2=W2+dweight2;

end

MSE(m,1)=mean(err);

end

%plot MSE against epochs

figure(1);

line(1:epochs,MSE);
title('Epochs vs MSE')
xlabel('No of Epochs')
ylabel('MSE')

% end of training
```

```
%part 2- test data

%read data from dataset excel sheet for testing


data2=xlsread('dataset.xlsx','Test','A2:I76');
T1=data2(1:end,2);
T2=data2(1:end,4);
T3=data2(1:end,6);
T4=data2(1:end,8);
y=data2(1:end,9);
T=[T1,T2,T3,T4];


%initialize variables


classifiedSetosa=0;
classifiedVersicolor=0;
classifiedVirginnica=0;
OT=ones(75,1);
e=ones(75,1);


%test Iris Setosa classification performance


for a=1:25

input=T(a,1:4)';
n1=W1*input;
o1=logsig(n1);
%output of the input layer concerned as input to the hidden layer


n2=W2*o1;
OT(a,1)=logsig(n2);


error(a,1)=y(a,1)-OT(a,1);
e(a,1)=abs(error(a,1));
if e(a,1)¡0.01
classifiedSetosa=classifiedSetosa+1;


end


end
```

```
%test Iris Versicolor classification performance

for a=26:50

input=T(a,1:4)';
n1=W1*input;
o1=logsig(n1);

%output of the input layer concerned as input to the hidden layer

n2=W2*o1;
OT(a,1)=logsig(n2);
error(a,1)=y(a,1)-OT(a,1);
e(a,1)=abs(error(a,1));
if e(a,1)¡0.01
classifiedVersicolor=classifiedVersicolor+1;

end

end

%Test Iris virginnnica classification performance

for a=51:75

input=T(a,1:4)';
n1=W1*input;
o1=logsig(n1);

%output of the input layer concerned as input to the hidden layer

n2=W2*o1;
OT(a,1)=logsig(n2);

error(a,1)=y(a,1)-OT(a,1);
e(a,1)=abs(error(a,1));
if e(a,1)¡0.01
classifiedVirginnica=classifiedVirginnica+1;

end

end
```

% scatter plot of classification results

```
figure(2);
scatter(OT(1:25,1),1:25);
hold on;
scatter(OT(26:50,1),26:50);
scatter(OT(51:75,1),51:75);
hold off;
```

%print the classification result

```
fprintf('ClassifiedSetosa ClassifiedVersicolor Classified Virginnica');
fprintf('%d,%d,%d',[classifiedSetosa,classifiedVersicolor,classifiedVirginnica]);
```

Bibliography

[1] AN APPROACH FOR IRIS PLANT CLASSIFICATION USING NEURAL NETWORK *Madhusmita Swain1, Sanjit Kumar Dash2, Sweta Dash3 and Ayeskanta Mohapatra4*

1, 2 Department of Information Technology, College of Engineering and Technology, Bhubaneswar, Odisha, India 3Department of Computer Science and Engineering, Synergy Institute of Engineering and Technology, Dhenkanal, Odisha, India 4Department of Computer Science and Engineering, Hi-tech Institute of Technology, Bhubaneswar, Odisha, India