Kernel Methods for Pattern Analysis
Assignment 2

Jilt SebastianCS13D020
Jom KuriakoseCS13S028
Abil N GeorgeCS13M002

March 21, 2014

Contents

Part I

Classification Tasks

1 Data-sets

1.0.1 Data-set 1: 2-dimensional input data:

(a) Linearly separable classes, (b) Non-linearly separable classes, (c) Overlapping class

1.0.2 Data-set 2 : Image data

Models

2 Experiment No.1: Bayes classifier

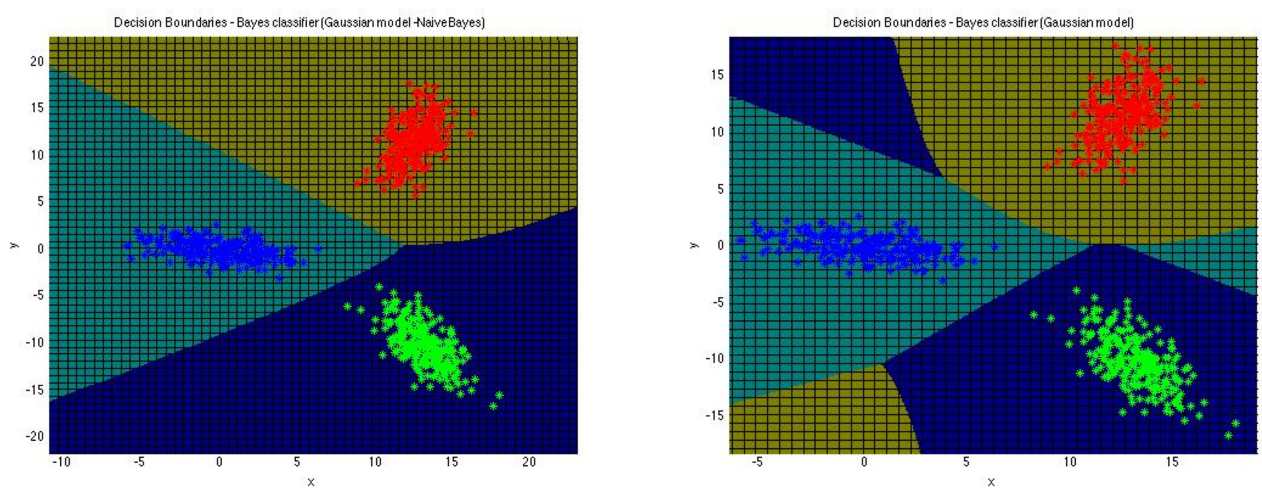
In this experiment,we were suppose to build Bayes classifier (Gaussian Model or Gaussian Mixture Model) for the various types of data-sets given.

2.1 Linearly Separable classes(Data-set 1(a))

Aim was to build classifier for the given data-set of Linearly Separable classes. Data is bi-variate and the classification task was a three class problem. A simple Gaussian Model is built for the given data-set using the training data-set and examples in test data-set are classified using the trained model. Function is written to train a system to classify a given element to 3 classes using BAYES and NAIVE-BAYES Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot of data-sets of different classes .

2.1.1 Decision Region Plots

Below shown are decision region plots for the Naive Bayes Classifier and Bayes Classifier respectively.



(a) Naive-Bayes Classifier

(b) Bayes Classifier

Figure 1: Decision Region Plots for Linearly Separable classes

2.1.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for Naive Bayes classifier

Class Label/Output Label	class1	class2	class3
class1	100	0	0
class2	0	100	0
class3	0	0	100

Classification Accuracy

1	1	1

Average Accuracy=1

Confusion Matrix for Bayes classifier

Class Label/Output Label	class1	class2	class3
class1	100	0	0
class2	0	100	0
class3	0	0	100

Classification Accuracy

1	1	1

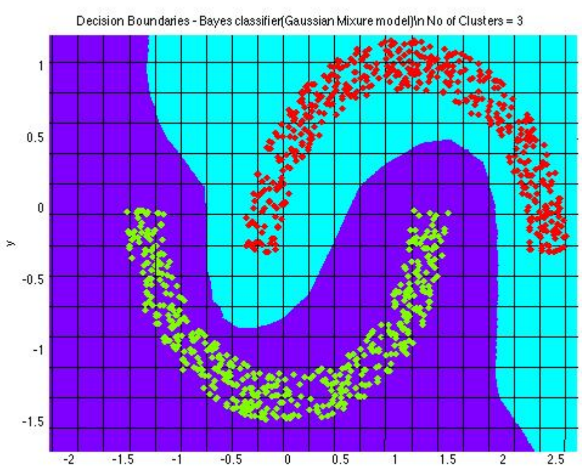
Average Accuracy=1

2.2 Non-linearly Separable classes(Data-set 1(b))

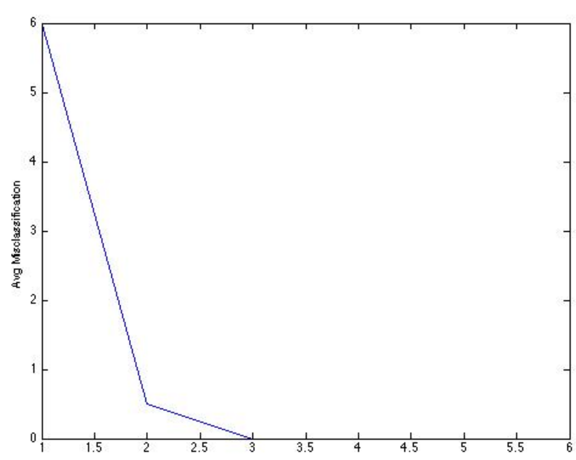
Aim was to build classifier for the given data-set of Non-linearly Separable Classes. Data is bi-variate and the classification task was a two class problem. Model is built to classify a given element to any of the three classes using GMM Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot. Minimum number of mixtures needed per class for best performance is 3.

2.2.1 Decision Region Plots

Below shown are (a) decision region plot for the Bayes Classifier using Gaussian Mixture Model and (b)Average Misclassification vs Number of clusters plot.



(a) Decision Region Plots of Bayes Classifier(GMM). No. of Clusters=3



(b) Average misclassification vs No. of Clusters

Figure 2: Performance plots for Non-linearly Separable classes

2.2.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for Bayes classifier (GMM)

Class Label/Output Label	class1	class2
class1	100	0
class2	0	100

Classification Accuracy

1	1

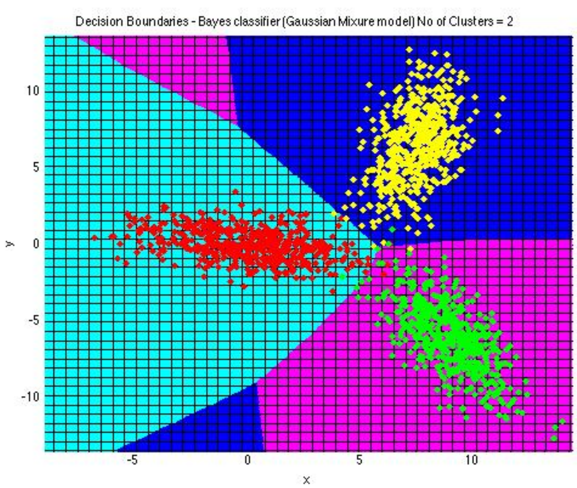
Average Accuracy=1

2.3 Overlapping classes(Data-set 1(c))

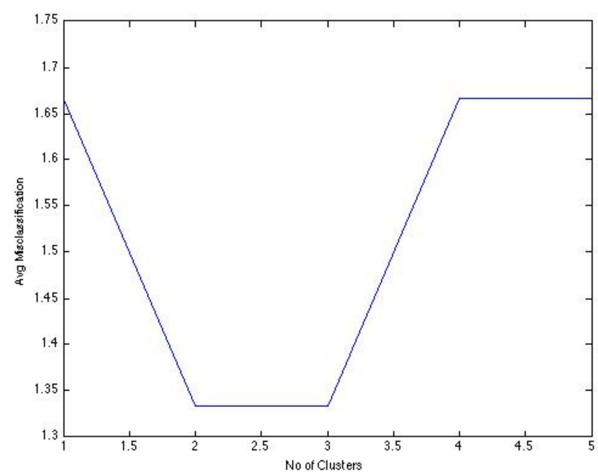
Aim was to build classifier for the given data-set of Overlapping Classes. Data is bi-variate and the classification task was a three class problem. Model is built to classify a given element to any of the three classes using GMM Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot.

2.3.1 Decision Region Plots

Below shown are (a) decision region plot for the Bayes Classifier using Gaussian Mixture Model and (b)Plot of Average Misclassification vs Number of clusters.



(a) Decision Region Plots of Bayes Classifier(GMM). No. of Clusters=2



(b) Average misclassification vs No. of Clusters

Figure 3: Performance plots for Overlapping classes

2.3.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for Bayes classifier (GMM)

Class Label/Output Label	class1	class2	class3
class1	100	0	0
class2	0	100	0
class3	0	1	99

Classification Accuracy

1	1	0.99

Average Accuracy=0.9967

2.4 Image Data(Data-set 2)

Aim was to build classifier for the given data-set of Classes. Data is multivariate with 48 dimensions and the classification task was a five class problem. Image data-set is splitted in ratio 70:10:20 for Train:Validation:Test . Model is built to classify a given element to any of the five classes using GMM Classifier and to find the Classification Accuracy, Confusion Matrix.

2.4.1 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for Bayes classifier (GMM)

Class Label/Output Label	class1	class2	class3	class4	class5
class1	29	9	0	1	3
class2	7	15	2	4	2
class3	6	7	27	3	1
class4	5	10	3	22	1
class5	1	7	0	3	76

Classification Accuracy

0.6579	0.5000	0.6136	0.5366	0.8736

Average Accuracy=0.6363

2.5 Observations

- For linearly Separable classes, Naive Bayes and Bayes Classifier performances are similar.This is because there are not much dependancy between the two features of the feature vector and only diagonals contributes to covariance matrix.
- For the Non-linearly Separable classes, GMM using 3 clusters found to be sufficient to get zero miscalssification. Hence the shape of the decision region is not too complex.
- For overlapping classes, if the number of clusters increases beyond 3, misclassification also increases, this is because when overlapped datapoints tend to form a different cluster in the other class, classifier learns wrongly.
- GMM using 10 clusters found to be suitable for given image data-set.

2.6 Experiment No. 2: Perceptron Model for Data-set 1(a)

The aim was to build a perceptron model for the Linearly Separable Data-set. Data is bi-variate and the classification task was a three class problem. One against one(1V1) method is used for classification of the examples to any of the three classes using Perceptron Model Based Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot.

2.6.1 Decision Region Plots

Below shown are decision region plots for the Perceptron Model based Classifier.

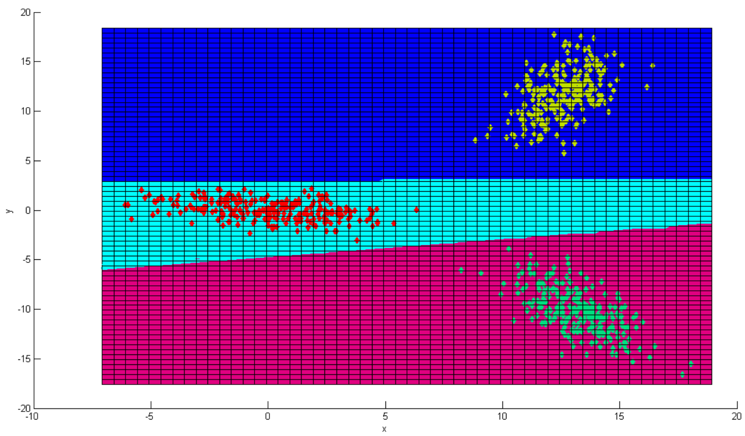


Figure 4: Decision Region Plot for Linearly Separable classes-Perceptron

2.6.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix

Class Label/Output Label	class1	class2	class3
class1	100	0	0
class2	0	100	0
class3	0	0	100

Classification Accuracy

1	1	1

Average Accuracy=1

2.6.3 Observation

- Decision region decided by 1-Vs-1 method is different from that obtained by Bayes Classifier.
- Batch mode is observed to be faster than the pattern mode (online mode) even though the accuracies are same

3 Experiment No.3: MLFFNN Models

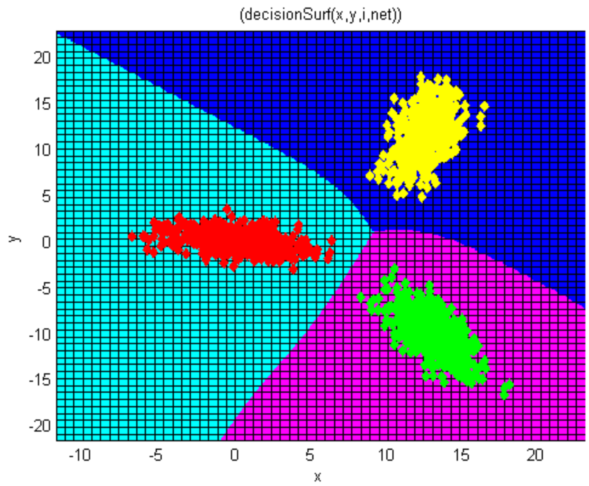
Multi Layer Feed Forward Neural Network Model is built to classify various types of data. Given data is split into Training, Validation and Test data-sets based on their index. Training function, Performance function, Transfer function are chosen appropriately.

3.1 Linearly Separable classes(Data-set 1(a))

Aim was to build MLFFNN Model Classifier for the given data-set of Linearly Separable classes. Data is bi-variate and the classification task was a three class problem. MLFFNN Classifier is built classify the given example and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot.

3.1.1 Decision Region Plots

Below is the decision region plot for the Linearly Separable classes with the trained model.

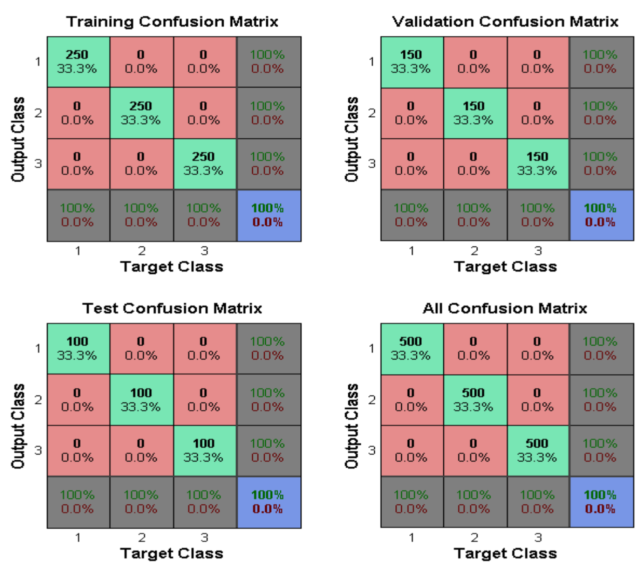


(a) MLFFNN Classifier

Figure 5: Decision Region Plots for Linearly Separable classes

3.1.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for MLFFNN classifier on Linearly Separable Data



(a) MLFFNN Classifier

Figure 6: Confusion Matrices for Linearly Separable classes

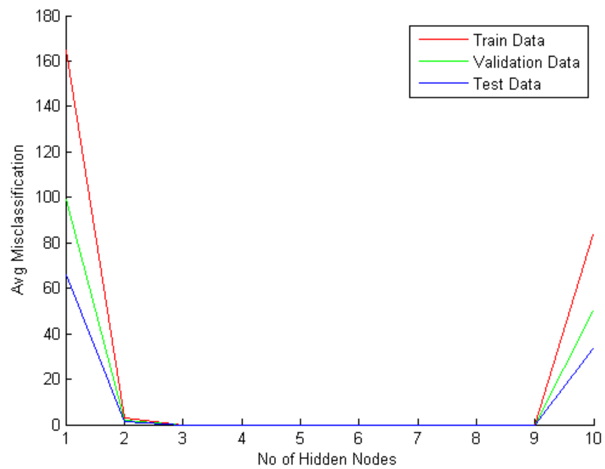
Classification Accuracy

1	1	1

3.1.3 Average misclassification vs No of Hidden Nodes

Below is the plot of average misclassification with the number of nodes in the hidden layer.

Figure 7: Average misclassification vs No. of Hidden Nodes



3.2 Non-linearly Separable classes(Data-set 1(b))

Aim was to build classifier for the given data-set of Non-linearly Separable Classes. Data is bi-variate and the classification task was a two class problem. Model is built to classify a given element to any of the three classes using MLFFNN Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot. Minimum number of mixtures needed per class for best performance was 3.

3.2.1 Decision Region Plots

Below shown is the decision region plot for the MLFFNN Classifier.

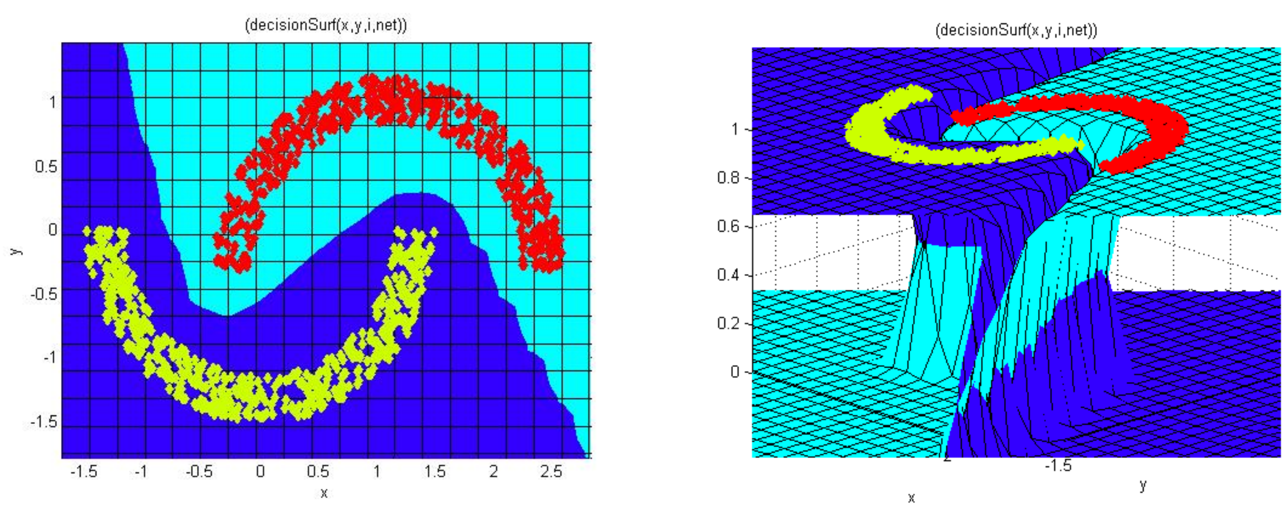


Figure 8: Decision Region Plot for Non- linearly Separable classes

3.2.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for classifier

(a) MLFFNN Classifier

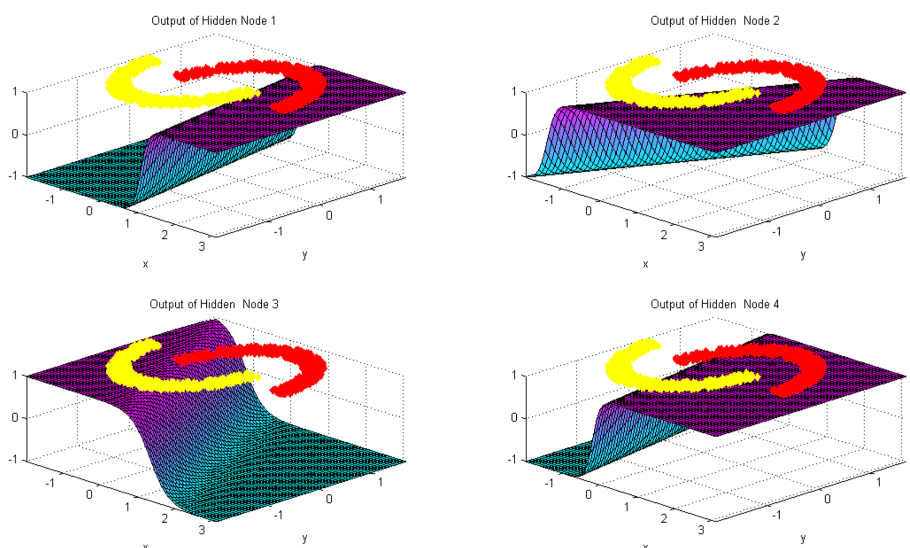
Figure 9: Confusion Matrices for Linearly Separable classes

Classification Accuracy

1	1

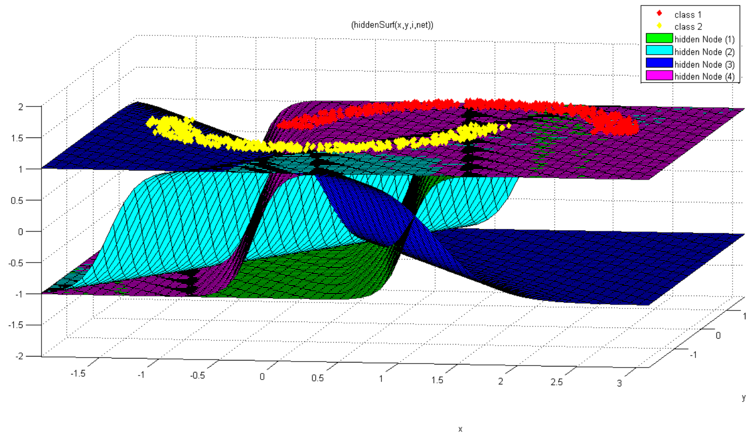
Average Accuracy=1

3.2.3 Plots of outputs each of the hidden nodes and output nodes in MLFFNN after the model is trained



(a) MLFFNN Classifier

Figure 10: Output of individual nodes in Hidden Layer

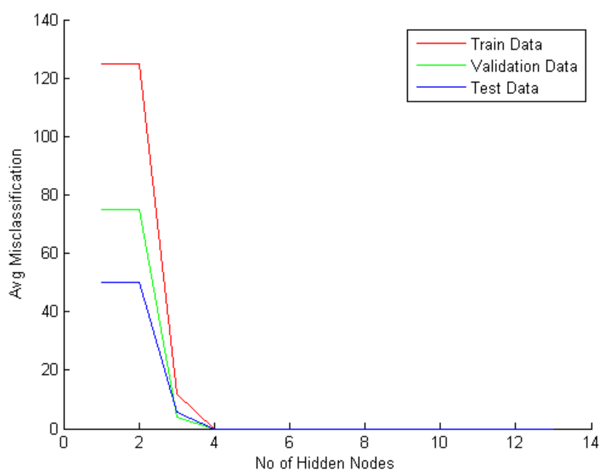


(a) MLFFNN Classifier

Figure 11: Output of Hidden Layer for Non-linearly Separable Classes

3.2.4 Average misclassification vs No of Hidden Nodes

Figure 12: Average misclassification vs No. of Hidden Nodes

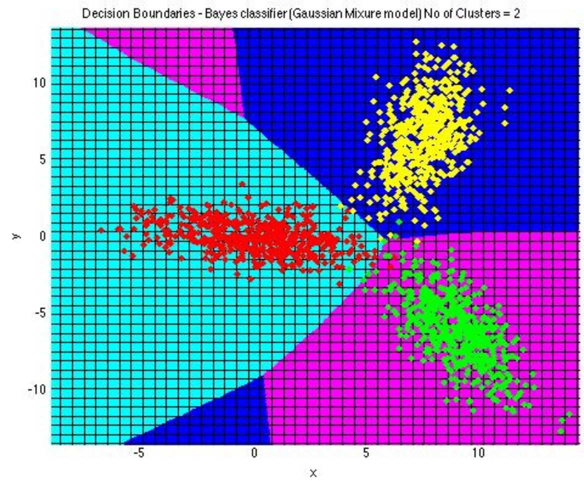


3.3 Overlapping classes(Data-set 1(c))

Aim was to build MLFFNN classifier for the given data-set of Overlapping Classes. Data is bi-variate and the classification task was a three class problem. Model is built to classify a given element to any of the three classes using the Classifier and to find the Classification Accuracy, Confusion Matrix and Decision Region Plot.

3.3.1 Decision Region Plots

Below shown is the decision region plot for the MLFFNN Classifier.



(a) MLFFNN classifier

Figure 13: Decision Region Plot for Overlapping classes

3.3.2 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for classifier

(a) MLFFNN Classifier

Figure 14: Confusion Matrices for Overlapping classes

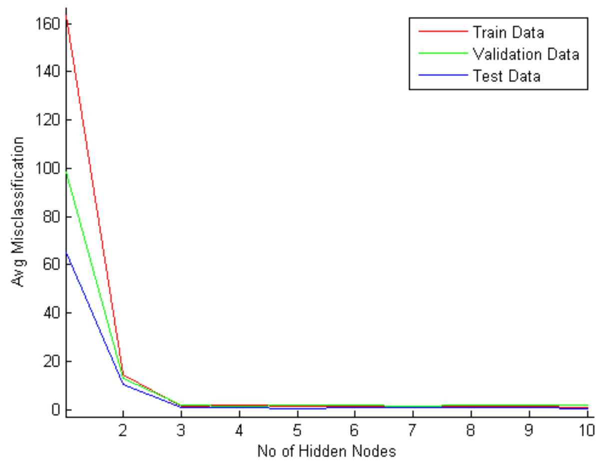
Classification Accuracy

1	1	0.97

Average Accuracy=0.99

3.3.3 Average misclassification vs No of Hidden Nodes

Figure 15: Average misclassification vs No. of Hidden Nodes

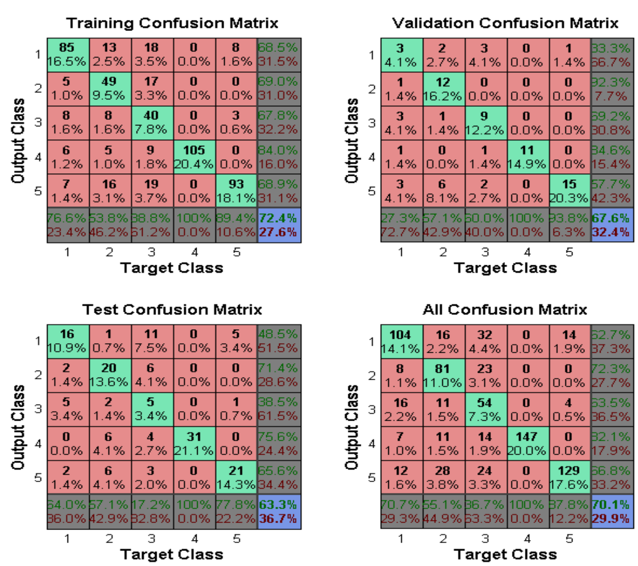


3.4 Image Data(Data-set 2)

Aim was to build classifier for the given data-set of Classes. Data is multivariate with 48 dimensions and the classification task was a five class problem. Image data-set is splitted in ratio 70:10:20 for Train:Validation:Test . Model is built to classify a given element to any of the five classes using MLFFNN Classifier and to find the Classification Accuracy, Confusion Matrix.

3.4.1 Confusion Matrix and Average Classification Accuracy

Confusion Matrix for classifier



(a) MLFFNN Classifier

Figure 16: Confusion Matrices for Classifier- Image Data-set

Classification Accuracy

0.64	0.571	0.172	0.100	0.778

Average Accuracy=0.633

3.5 Observations

- For linearly Separable classes, decision surface is almost similar to what we got from Naive Bayes Classifier. Number of hidden nodes of the is 3 for classifier (MLFFNN) which gives 100% accuracy.
- Average miscalssification is decreasing w.r.to the number of hidden layers until a certain number of nodes and then increases.
- For Non-linearly Separable classes, MLFFNN with 4 hidden nodes gave 100% accuracy.
- Decision region is a weighted sum of hidden layer outputs. We can see from the hidden layer outputs plot that, how each node contribute to decision region.
- For overlapping classes, highest accuracy is obtained when number of hidden nodes is 4.
- For image data-set single hidden layer did not give good accuracy (~20%) . But, two hidden layers with 100 and 30 nodes respectively gave highest accuracy (63%) which is same as that given by Bayes Classifier.

Part II
Regression Tasks

4 Data-sets

4.0.1 Data-set 1: 1-dimensional (Uni-variate) input data

4.0.2 Data-set 2: 2-dimensional (Bi-variate) input data

4.0.3 Data-set 3: Multivariate input data

Models

5 Experiment No.1: MLFFNN model

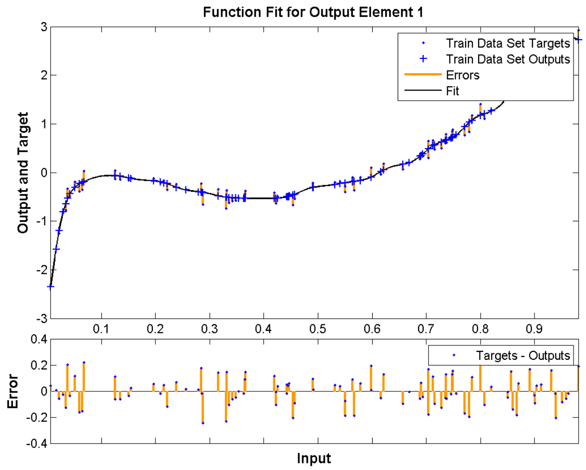
Multi Layer Feed Forward Neural Network Model is built to approximate various types of functions. Given data is split into Training,Validation and Test data-sets based on their index. Training function, Performance function, Transfer function are chosen appropriately.

5.1 Uni-variate Data (Data-set 1)

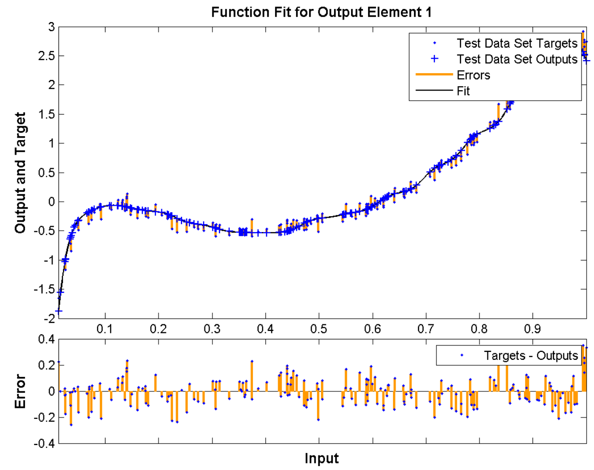
Aim was to build MLFFNN regression model for the given data-set. Data is uni-variate and the regression task was to approximate the underlying function for the test data-set using the trained model

5.1.1 Plots of model output and target output

Below are the plots of model output and target output for training data, validation data and test data using MLFFNN Model.



(b) Train data



(d) Test data

Figure 17: Realization of model output and target output

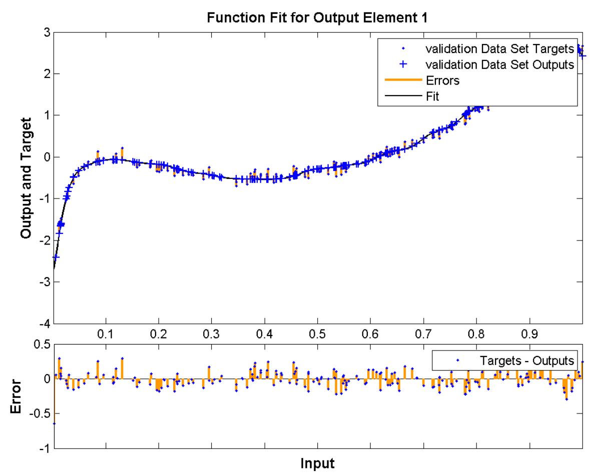
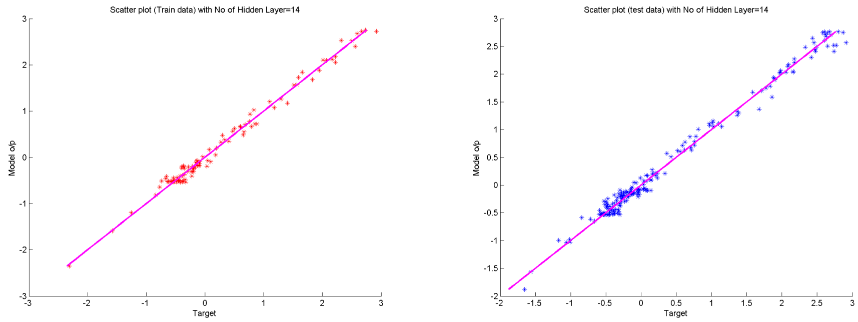


Figure 18: Validation data-Realization of model and target output

5.1.2 Scatter Plot of Target versus Model Output



(b) Target vs Model o/p for train data (d) Target vs Model o/p for test data

Figure 19: Scatter Plot of Target vs Model output

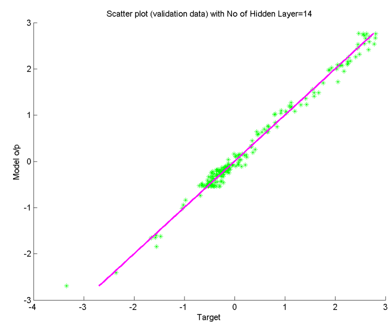
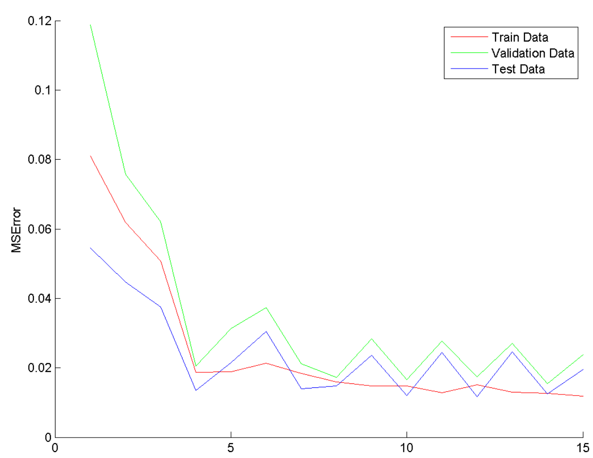


Figure 20: Scatter Plot of Target vs Model o/p for validation data

5.1.3 Plot of Number of Hidden Layers versus Mean Squared Error



(a) Number of Hidden Layers versus Mean Squared Error

Figure 21: MLFFNN Regression model for uni-variate data

5.1.4 Results

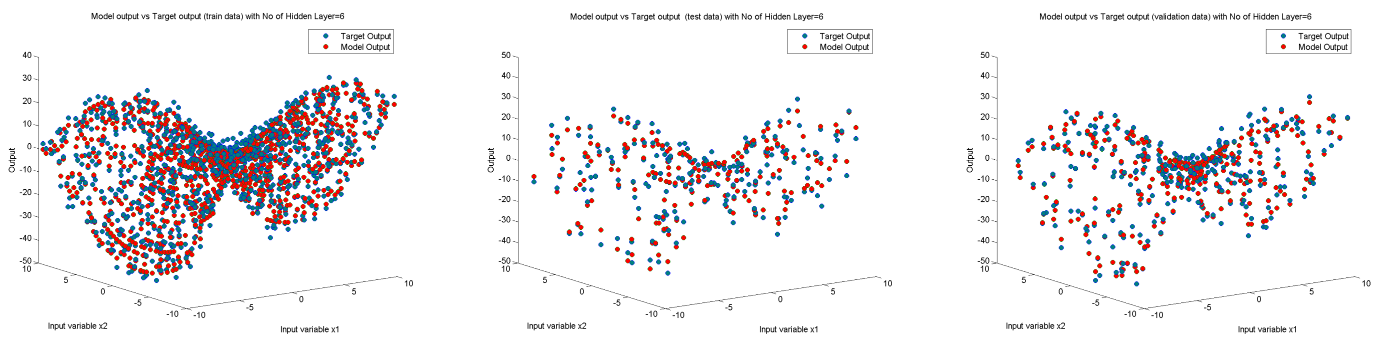
No. of Hidden Nodes	10
MSE on Validation Data	0.0165
MSE on Test Data	0.0119

5.2 Bi-variate Data (Data-set 2)

Aim was to build MLFFNN regression model for the given data-set. Data is bi-variate and the regression task was to approximate the underlying function for the test data-set using the trained model

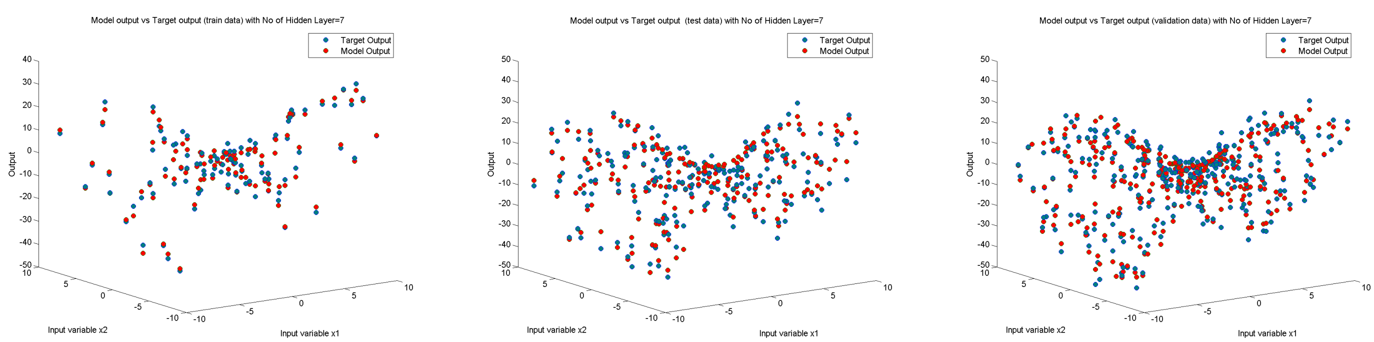
5.2.1 Plots of model output and target output

Below are the plots of model output and target output for training data, validation data and test data using MLFFNN Model.



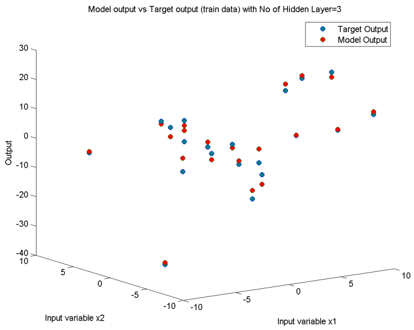
(b) Train data (d) Test data (f) Validation data

Figure 22: Realization model output and target output,N=1000

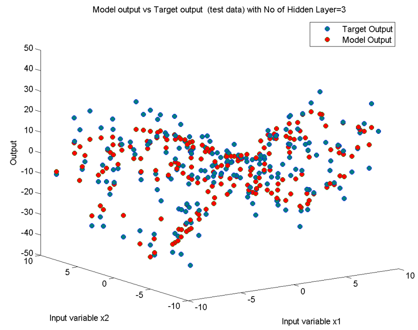


(b) Train data (d) Test data (f) Validation data

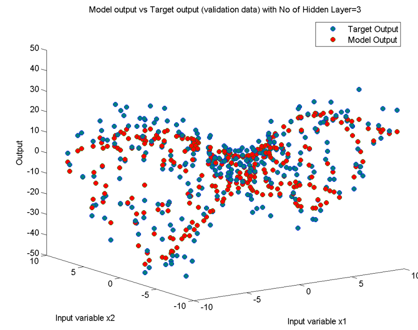
Figure 23: Realization model output and target output,N=100



(b) Train data



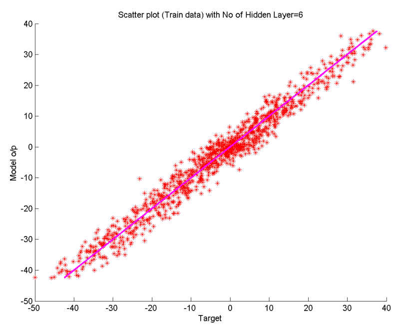
(d) Test data



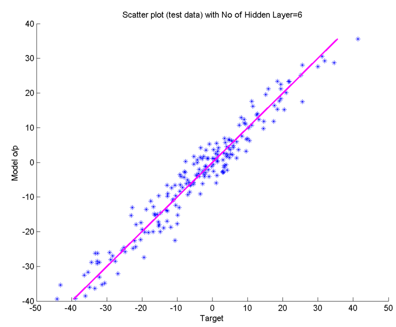
(f) Validation data

Figure 24: Realization model output and target output,N=20

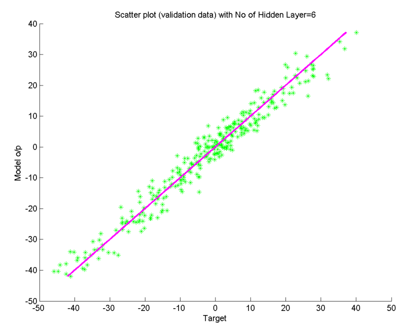
5.2.2 Scatter Plot of Target versus Model Output.



(b) Target vs Model o/p for train data

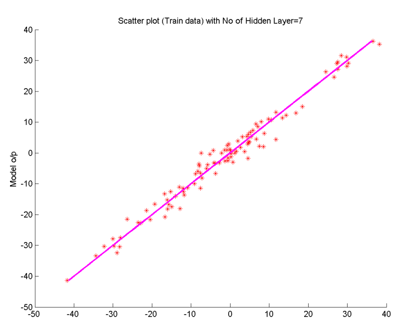


(d) Target vs Model o/p for test data



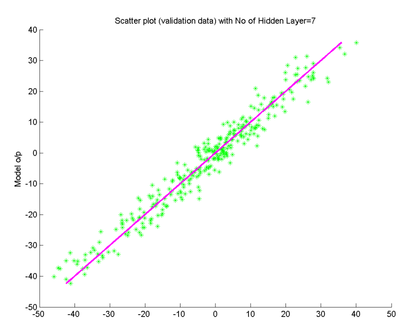
(f) Target vs Model o/p for validation data

Figure 25: Target vs Model output,N=1000



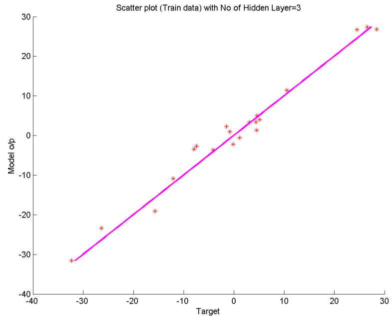
(b) Target vs Model o/p for train data

(d) Target vs Model o/p for test data

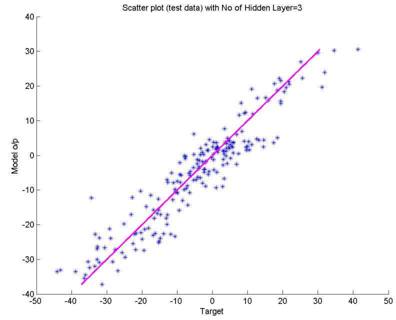


(f) Target vs Model o/p for validation data

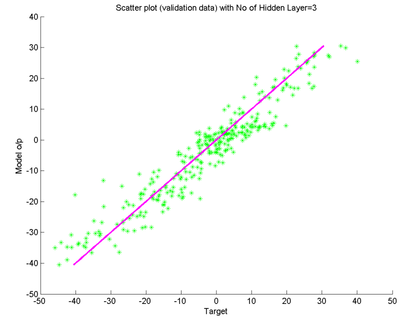
Figure 26: Target vs Model output, N=100



(b) Target vs Model o/p for train data



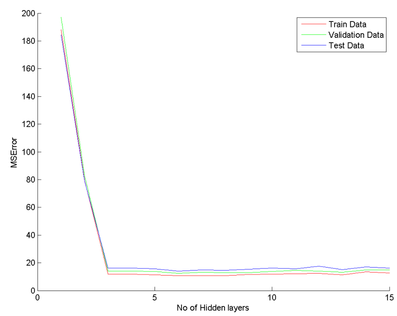
(d) Target vs Model o/p for test data



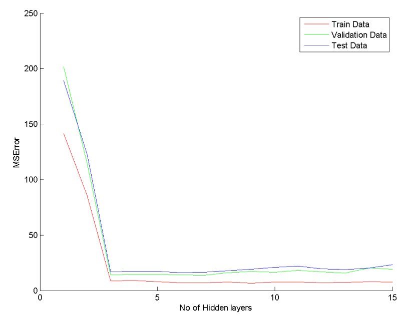
(f) Target vs Model o/p for validation data

Figure 27: Target vs Model output

5.2.3 Plot of Number of Hidden Layers versus Mean Squared Error.



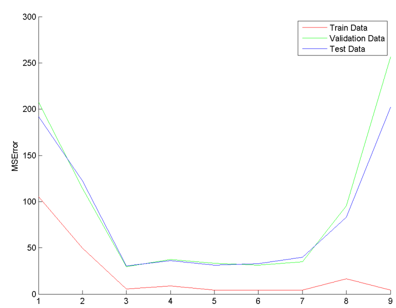
(a) N=1000



(b) N=100

Figure 28: Number of Hidden Layers versus Mean Squared Error

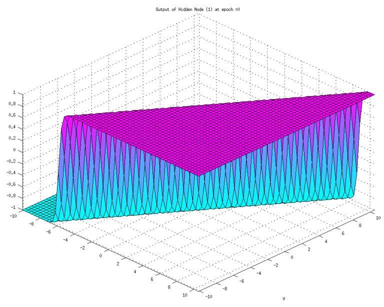
Figure 29: N=20



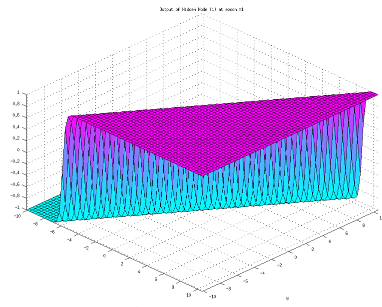
5.2.4 Plots of outputs each of the hidden nodes in MLFFNN after different number of epochs during training.

Below are the plots of hidden layer outputs for different number of epochs. For this we considered Hidden nodes 1 and 4 outputs.

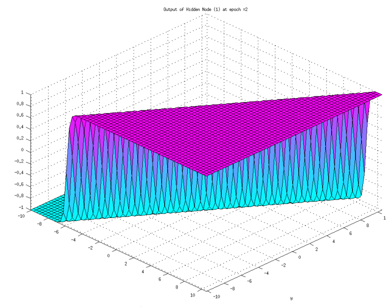
Figure 30: Outputs of Hidden Node (1)



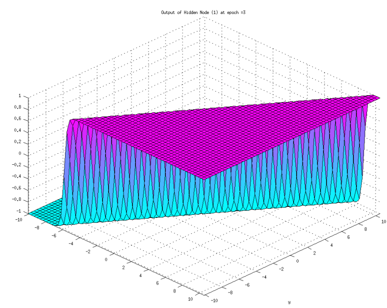
(b) Epoch0



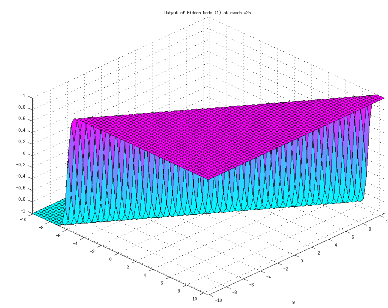
(d) Epoch 1



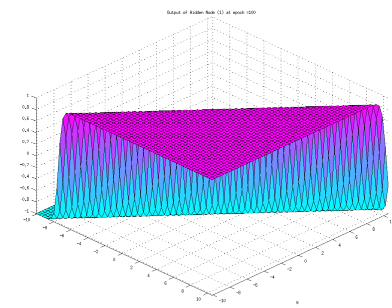
(f) Epoch 2



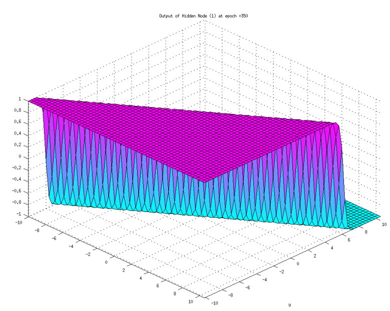
(h) Epoch 3



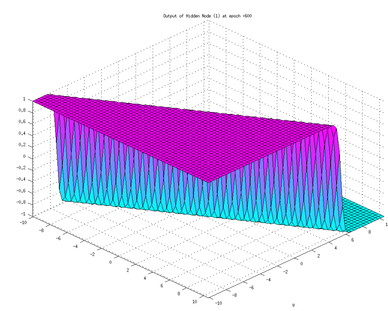
(j) Epoch 25



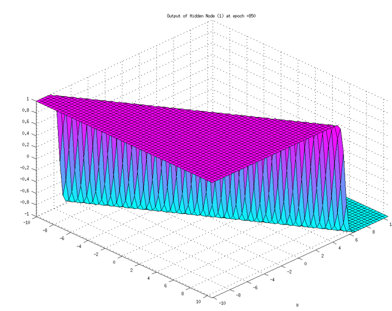
(l) Epoch 100



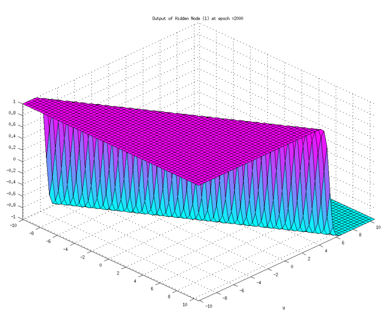
(n) Epoch 350



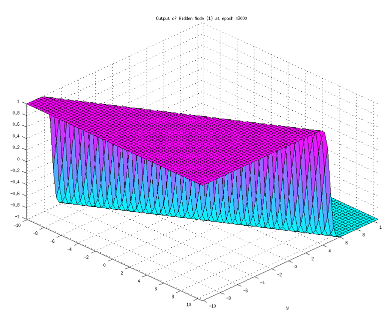
(p) Epoch 600



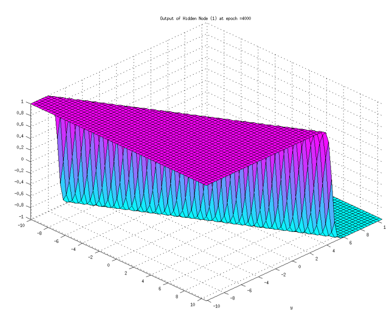
(r) Epoch 850



(t) Epoch 2000

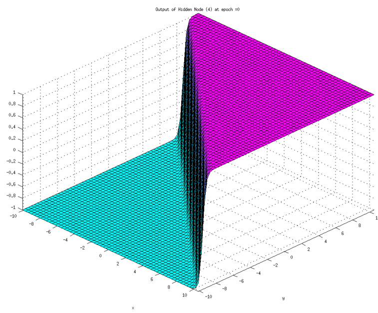


(v) Epoch 3000

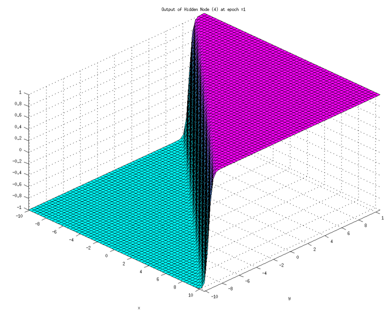


(x) Epoch 4000

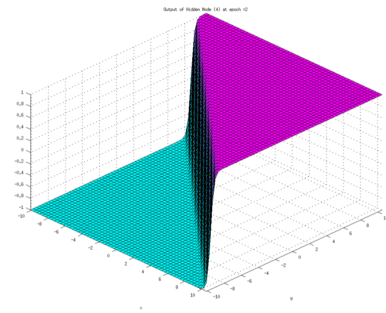
Figure 31: Outputs of Hidden Node (4)



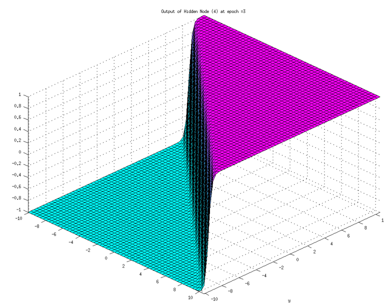
(b) Epoch0



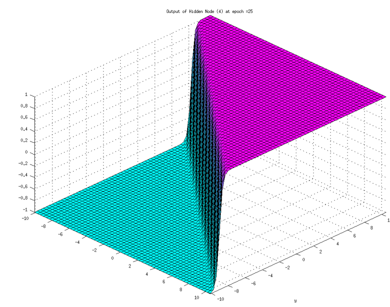
(d) Epoch 1



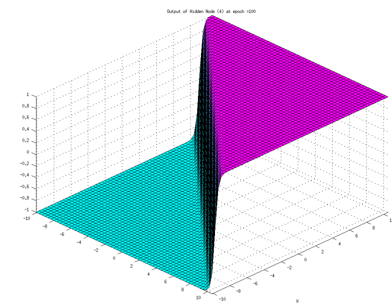
(f) Epoch 2



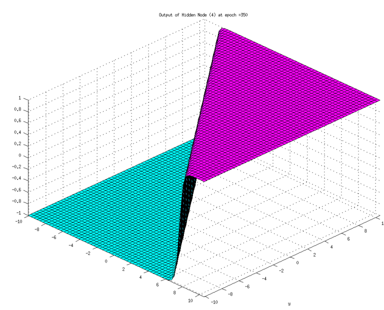
(h) Epoch 3



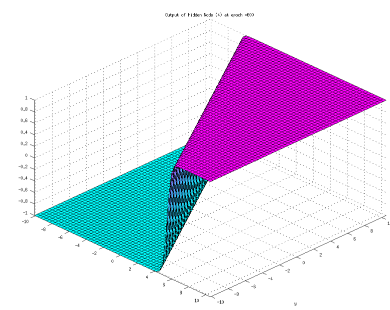
(j) Epoch 25



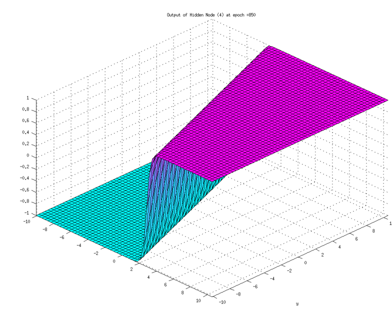
(l) Epoch 100



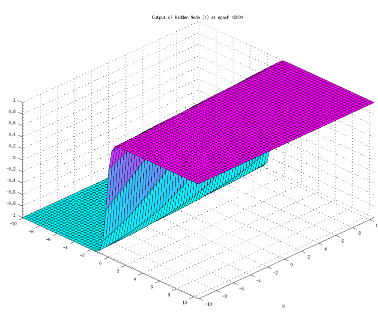
(n) Epoch 350



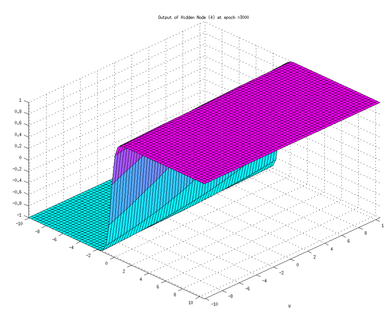
(p) Epoch 600



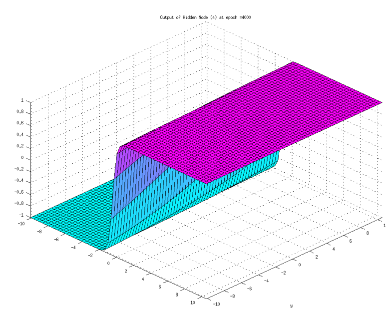
(r) Epoch 850



(t) Epoch 2000



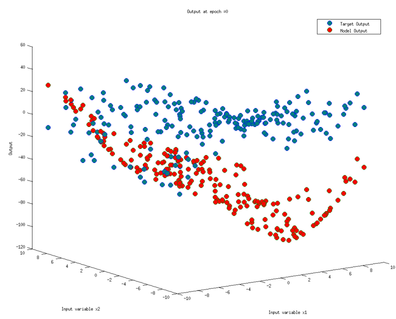
(v) Epoch 3000



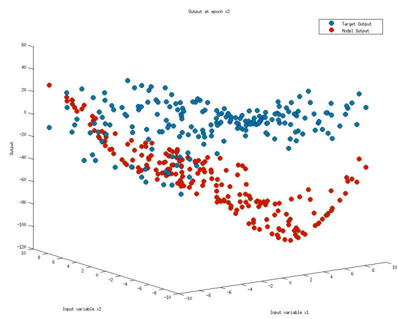
(x) Epoch 4000

5.2.5 Plots of outputs of the output nodes in MLFFNN after different number of epochs during training.

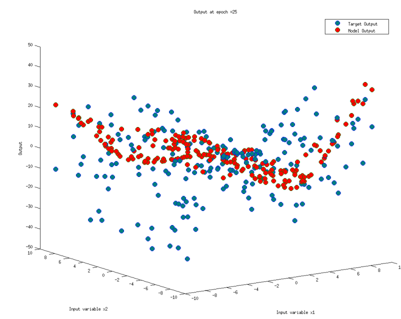
Figure 32: Outputs of Output Layer



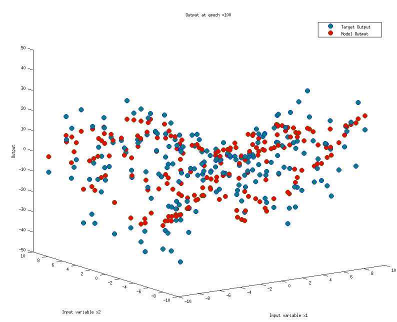
(b) Epoch0



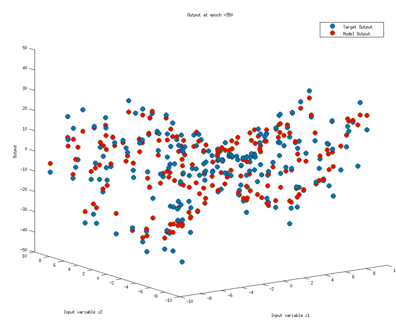
(d) Epoch 2



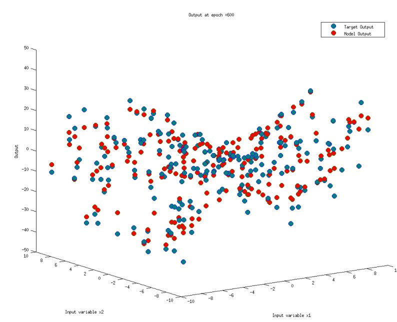
(f) Epoch 25



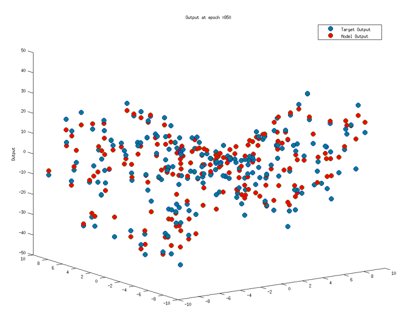
(h) Epoch 100



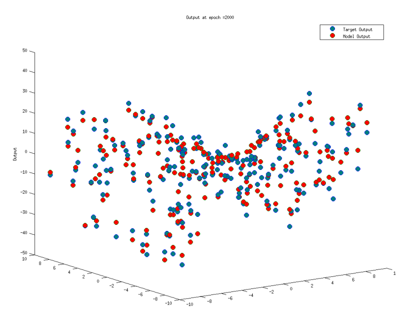
(j) Epoch 350



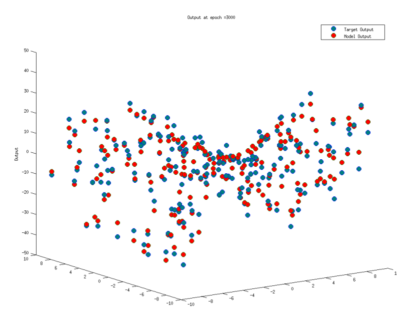
(l) Epoch 600



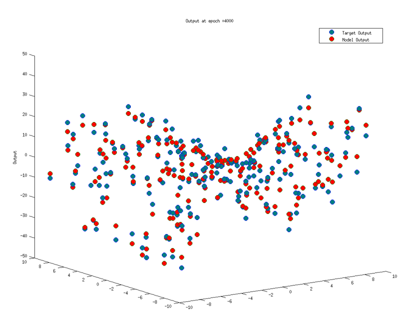
(n) Epoch 850



(p) Epoch 2000



(r) Epoch 3000



(t) Epoch 4000

5.2.6 Plots of Mean Squared Error versus Number of Epochs.

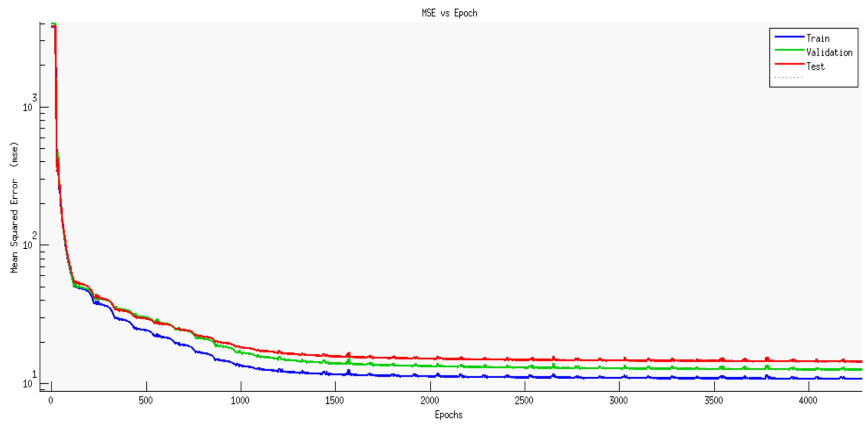


Figure 33: Mean Squared Error versus Number of Epochs.

5.2.7 Table showing weight change with epochs for a hidden layer.

Epochs	Weights					
0	2.8044	-3.4066	-2.1670	2.2787	1.2365	-2.2289
	-1.9737	0.3936	2.6578	2.5627	-3.1986	2.6062
25	2.8817	3.4041	-2.1120	2.2084	1.2412	-2.2079
	-2.0407	0.3963	2.6635	2.6304	-3.2110	2.6153
100	2.4499	3.4202	-2.1097	2.2789	1.1886	-1.9120
	-2.1671	0.0131	2.5813	2.4944	-3.2437	2.5169
350	2.0055	3.0144	-2.0077	2.2936	0.4445	-2.1658
	-2.9059	-0.4539	2.2872	1.8508	-3.1856	2.4392
600	1.8382	2.6318	-1.8042	2.2572	0.4026	-2.2869
	-2.9779	-0.4113	2.3297	1.3705	-3.0470	2.3783
850	1.7862	2.3098	-1.6038	2.2539	0.5530	-2.3477
	-2.8663	-0.2354	2.3937	0.8063	-2.9250	2.3296
2000	1.6301	2.0357	-1.3836	2.2555	0.8578	-2.3491
	-2.6738	0.0448	2.4059	0.1267	-2.7609	2.2902
3000	1.5274	1.9581	-1.2900	2.2097	0.9560	-2.3071
	-2.6309	0.0871	2.4142	0.0800	-2.7035	2.2171
4000	1.4573	1.9023	-1.2155	2.1569	1.0260	-2.2450
	-2.6122	0.1139	2.4177	0.0686	-2.6583	2.1499
5000	1.4073	1.8575	-1.1576	2.1037	1.0771	-2.1771
	-2.6041	0.1315	2.4124	0.0645	-2.6250	2.1030
6000	1.3721	1.8191	-1.1136	2.0535	1.1122	-2.1127
	-2.5996	0.1423	2.4000	0.0626	-2.6011	2.0773
7000	1.3506	1.7893	-1.0840	2.0136	1.1331	-2.0620
	-2.5962	0.1478	2.3856	0.0617	-2.5860	2.0681
8000	1.3327	1.7570	-1.0562	1.9709	1.1505	-2.0080
	-2.5915	0.1513	2.3660	0.0610	-2.5729	2.0680
9000	1.3219	1.7310	-1.0366	1.9374	1.1618	-1.9655
	-2.5866	0.1526	2.3475	0.0603	-2.5645	2.0743
10000	1.3144	1.7070	-1.0204	1.9076	1.1711	-1.9271
	-2.5813	0.1526	2.3286	0.0595	-2.5581	2.0843

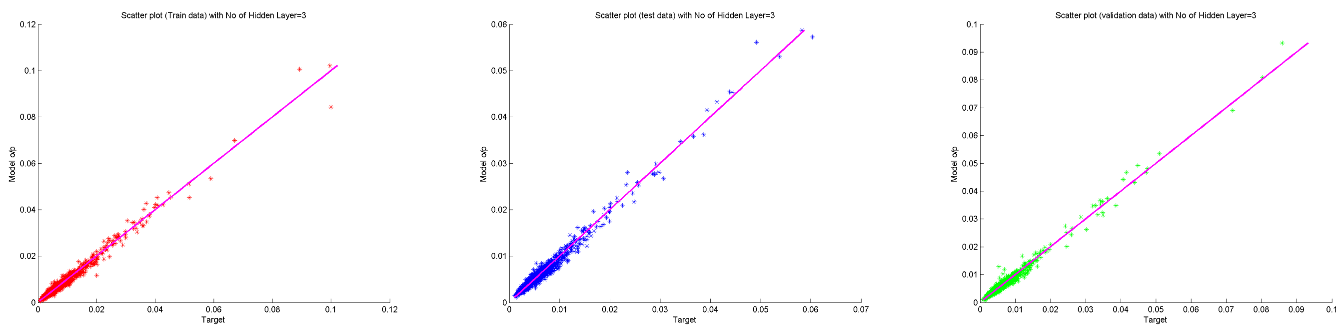
5.2.8 Results

No. of Hidden Nodes	6
MSE on Validation Data	1.23844e+01
MSE on Test Data	1.4142e+01

5.3 Multivariate Data (Data-set 3)

Aim was to build MLFFNN regression model for the given data-set. Data is multivariate with 18 dimensions and the regression task was to approximate the underlying function for the test data-set using the trained model

5.3.1 Scatter Plot of Target versus Model Output



(b) Target vs Model o/p for train data

(d) Target vs Model o/p for test data

(f) Target vs Model o/p for validation data

Figure 34: Target vs Model output,N=1000

5.3.2 Results

No. of Hidden Nodes	4
MSE on Validation Data	6.3983e-07
MSE on Test Data	6.7703e-07

5.4 Observations

- Mean Squared Error first decreases w.r.to increase in number of hidden nodes, then increases. (we can see this clearly for N=20, bi-variate case)
- Weights of hidden layer changes very slowly during higher epochs compared to that of lower epochs.
- The model output tries to match with the target faster at lower epochs and fine tunes it at higher epochs.
- For the multivariate data, the approximation seems to be better as compared with other regression methods.
- Best possible value of number of nodes in hidden layer are 10 for uni-variate data, 6 for Bi-variate data and 4 for Multivariate data.

6 Experiment No.2: Generalized RBF model

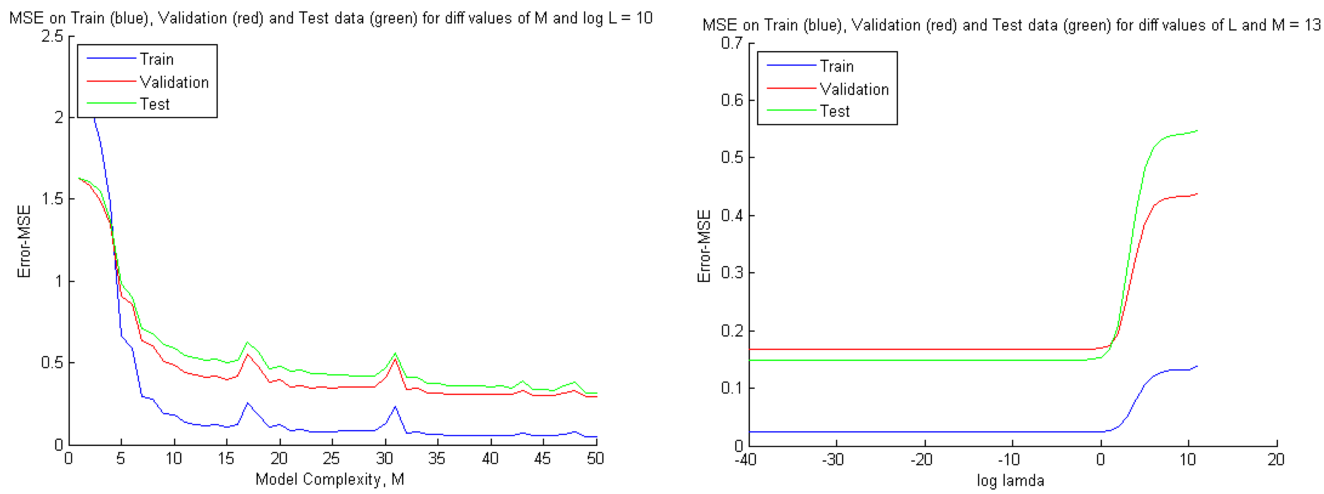
Multi Layer Feed Forward Neural Network Model is built to classify various types of data. Given data is split into Training, Validation and Test data-sets based on their index. Training function, Performance function, Transfer function are chosen appropriately.

6.1 Uni-variate Data (Data-set 1)

Aim was to build Generalized RBF Model for the given data-set. Data is uni-variate and the regression task was to approximate the underlying function for the test data-set using the trained model.

6.1.1 Plots of MSE Error on Training data, Validation data and Test data.

Following are the plots of Mean Squared Error for (a) different model complexities and for (b) different regularization parameter values.



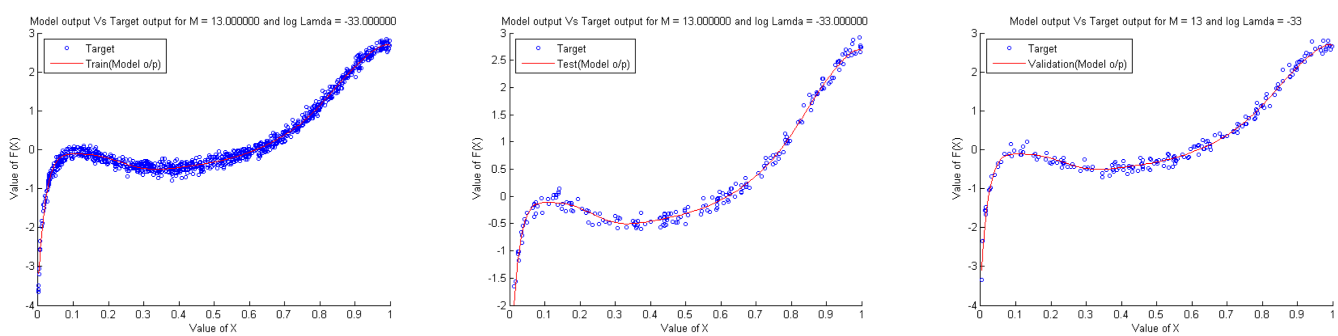
(a) MSE vs Model complexity

(b) MSE vs Regularization Parameter

Figure 35: Generalized RBF Model

6.1.2 Plots of model output and target output

Below are the plots of model output and target output for training data, validation data and test data using Generalized RBF Model.



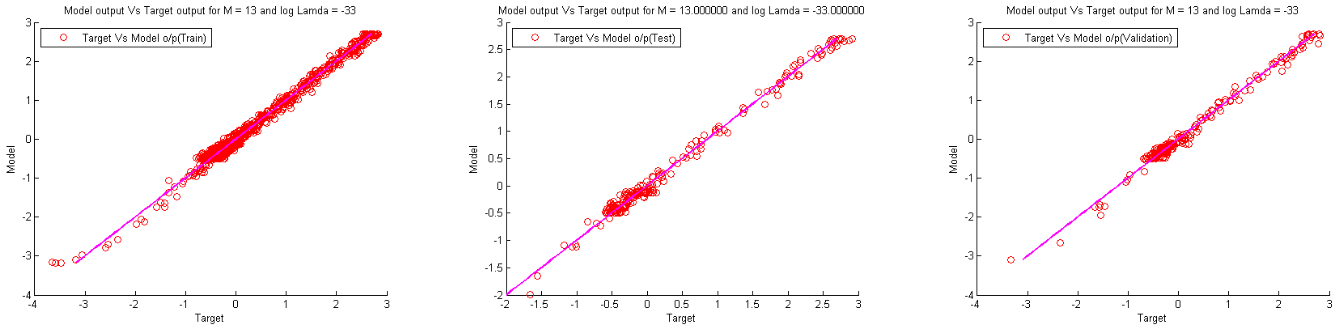
(b) Train data

(d) Test data

(f) Validation data

Figure 36: Realization model output and target output

6.1.3 Scatter Plot of Target versus Model Output



(b) Target vs Model o/p for train data (d) Target vs Model o/p for test data (f) Target vs Model o/p for validation data

Figure 37: Target vs Model output

6.2 Bi-variate Data (Data-set 2)

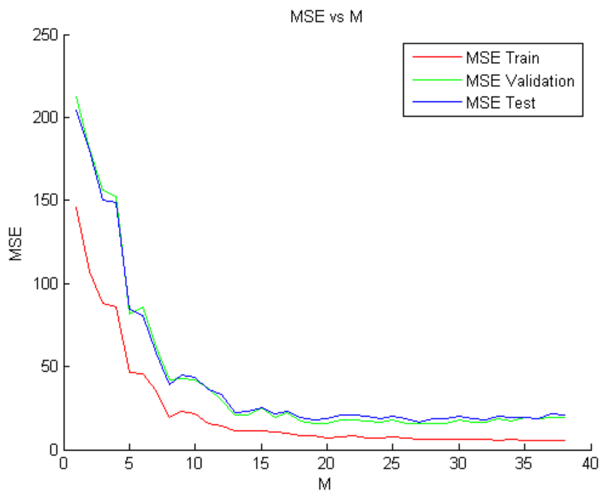
Aim was to build Generalized RBF Model for the given data-set. Data is bi-variate and the regression task was to approximate the underlying function for the test data-set using the trained model.

6.2.1 Results

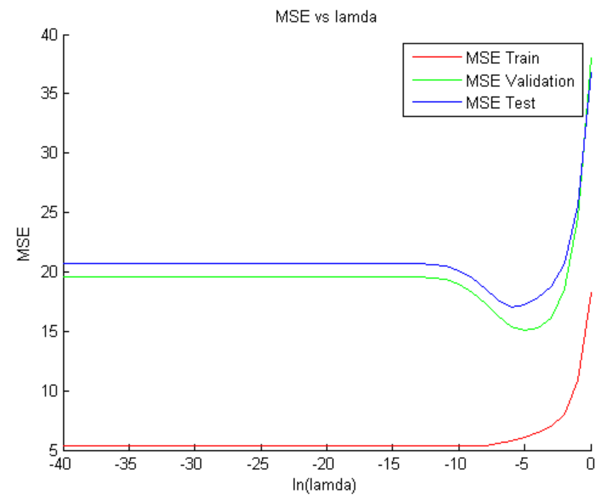
Error on Validation Data	4.517522e+03
Least MSE	1.505841e+01
M	40
log(lambda)	-4
lambda	2.478752e-03
Error on Test Data with best model	3.438185e+03
Least MSE	1.719092e+01

6.2.2 Plots of MSE Error on Training data, Validation data and Test data.

Following are the plots of Mean Squared Error for (a) different model complexities and for (b) different regularization parameter values.



(a) MSE vs Model complexity

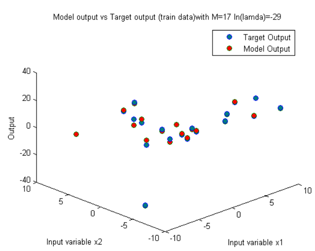


(b) MSE vs Regularization Parameter

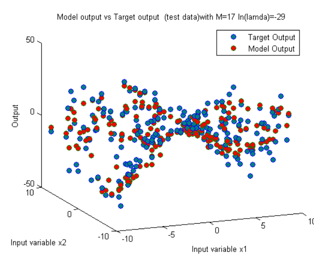
Figure 38: Generalized RBF Model

6.2.3 Plots of model output and target output

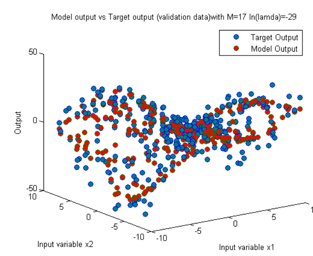
Below are the plots of model output and target output for training data, validation data and test data using Generalized RBF Model.



(b) Train data

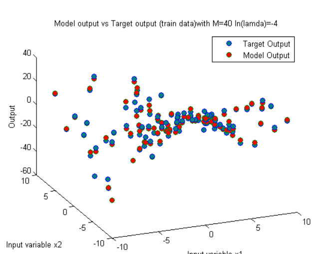


(d) Test data

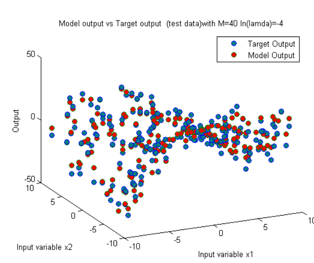


(f) Validation data

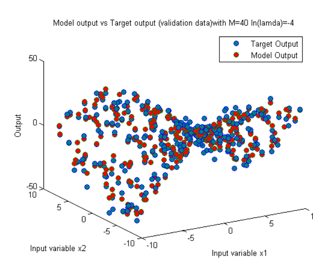
Figure 39: Realization of model output and target output, N=20



(b) Train data

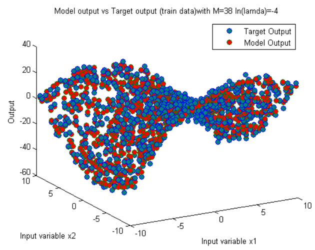


(d) Test data

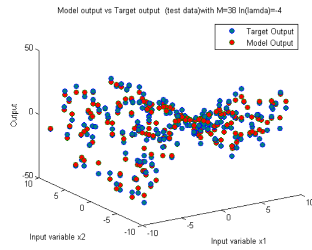


(f) Validation data

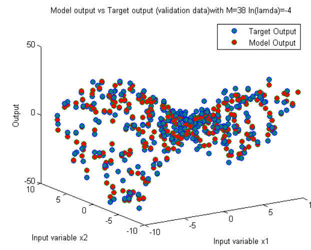
Figure 40: Realization of model output and target output, N=100



(b) Train data



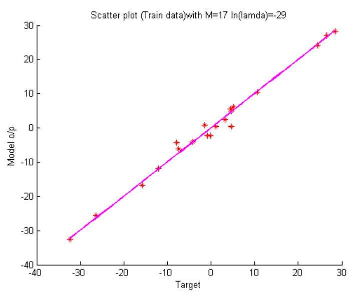
(d) Test data



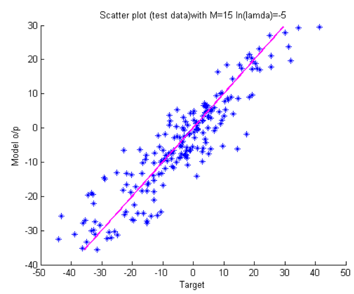
(f) Validation data

Figure 41: Realization of model output and target output,N=1000

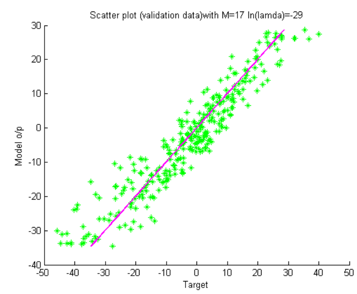
6.2.4 Scatter Plot of Target versus Model Output



(b) Target vs Model o/p for train data

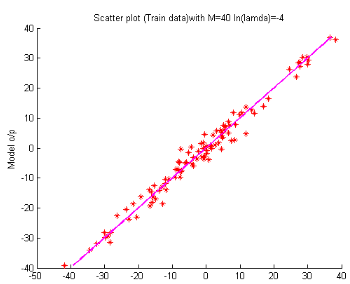


(d) Target vs Model o/p for test data

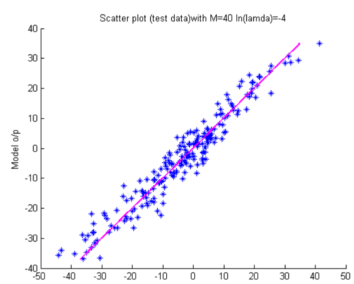


(f) Target vs Model o/p for validation data

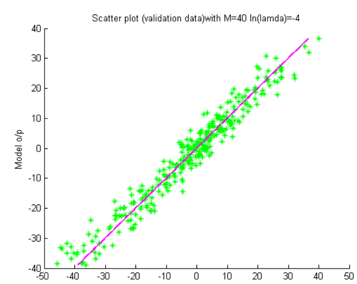
Figure 42: Target vs Model output, N=20



(b) Target vs Model o/p for train data

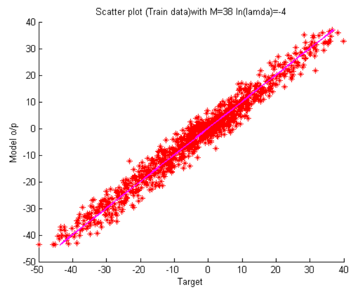


(d) Target vs Model o/p for test data

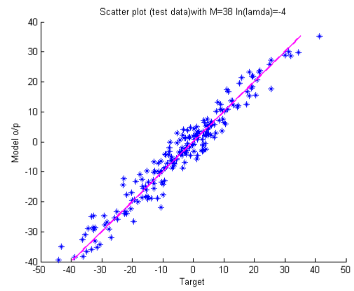


(f) Target vs Model o/p for validation data

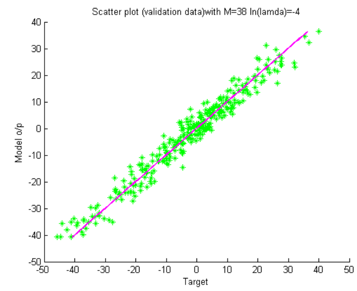
Figure 43: Target vs Model output, N=100



(b) Target vs Model o/p for train data



(d) Target vs Model o/p for test data



(f) Target vs Model o/p for validation data

Figure 44: Target vs Model output, N=1000

6.3 Multivariate Data (Data-set 3)

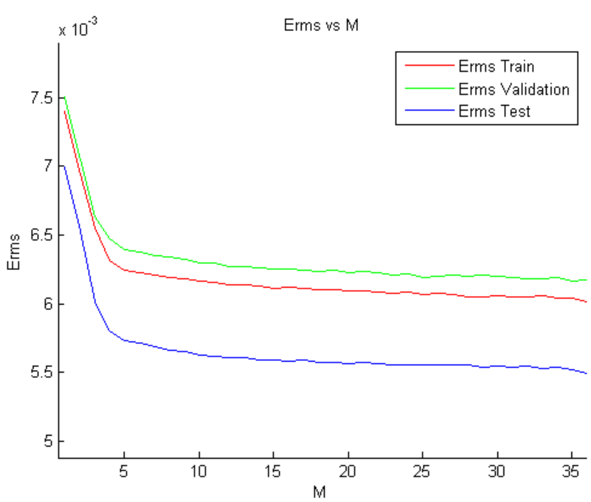
Aim was to build Generalized RBF Model for the given data-set. Data is multivariate with input vector of dimension 18 and the regression task was to approximate the underlying function for the test data-set using the trained model.

6.3.1 Results

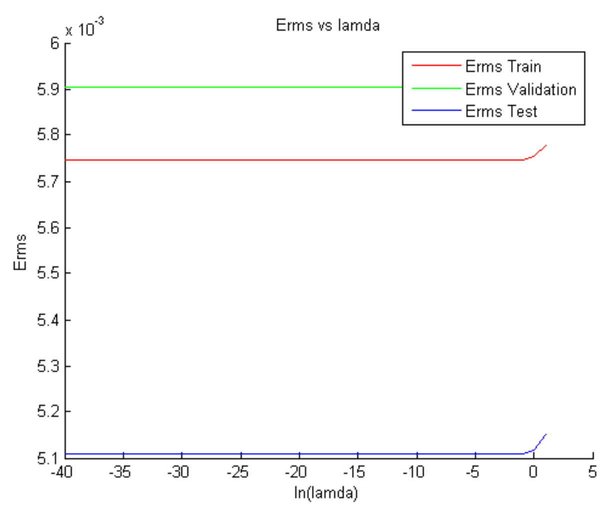
Error on Validation Data	5.226922e-02
Least MSE	5.903062e-03
M	36
log(lambda)	-32
lambda	1.713908e-15
Error on Test Data with best model	3.587622e-02
Least MSE	5.108013e-03

6.3.2 Plots of MSE Error on Training data, Validation data and Test data.

Following are the plots of Mean Squared Error for (a) different model complexities and for (b) different regularization parameter values.



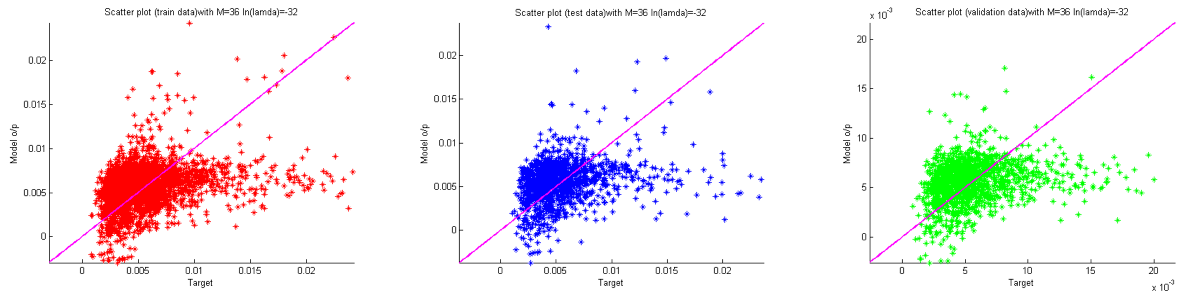
(a) MSE vs Model complexity



(b) MSE vs Regularization Parameter

Figure 45: Generalized RBF Model

6.3.3 Scatter Plot of Target versus Model Output



(b) Target vs Model o/p for train data (d) Target vs Model o/p for test data (f) Target vs Model o/p for validation data

Figure 46: Target vs Model output

6.4 Observations

- MSE is lesser as compared to linear regression models with Gaussian basis function for all the given data-sets.
- Best possible values of Model Complexity are 13 for uni-variate data, 40 for Bi-variate data and 36 for Multivariate data.