Pattern Recognition and Machine Learning

Assignment 3
Report

Group No. 19

Ranjith Tevnan	Sai Bandawar	Jay shah
EE18B146	EE18B150	EE18B158

13 May, 2021

1 Dataset 1: 2-dimensional artificial data

1.(a) Linearly seperable data set for static pattern classification

1.(a).1 Perceptron for every pair of classes

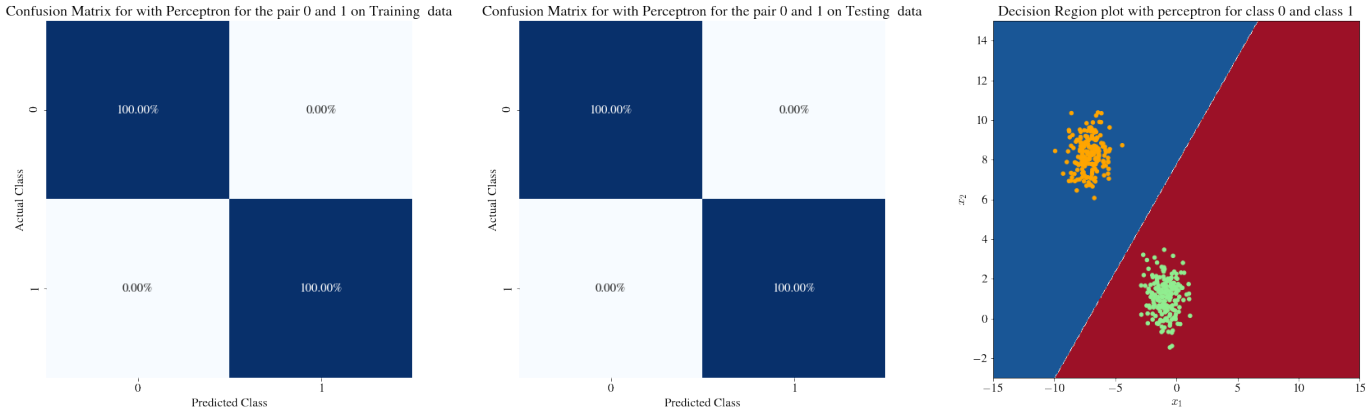


Figure 1: Confusion matrices on train & test data and decision plot between class 0 & class 1

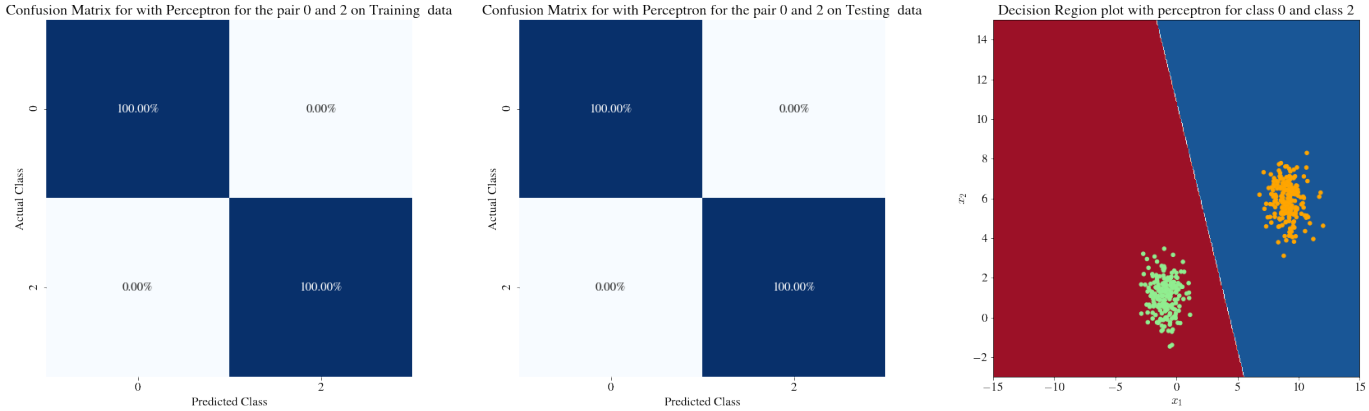


Figure 2: Confusion matrices on train & test data and decision plot between class 0 & class 2

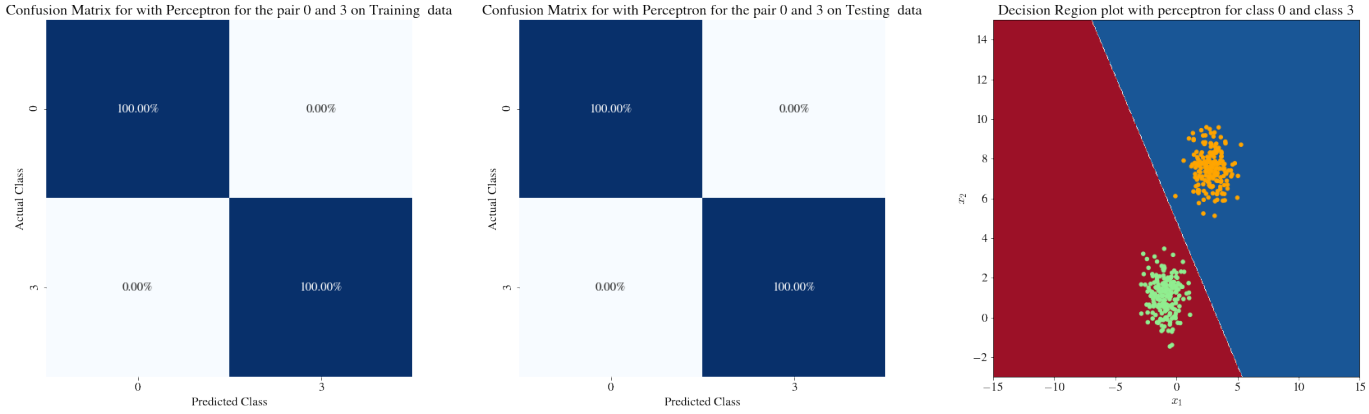


Figure 3: Confusion matrices on train & test data and decision plot between class 0 & class 3

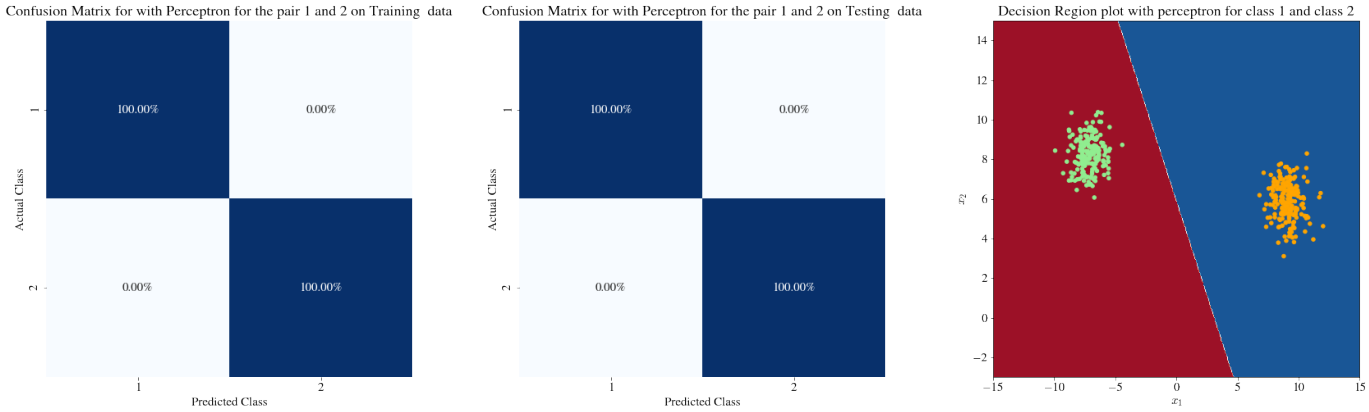


Figure 4: Confusion matrices on train & test data and decision plot between class 1 & class 2

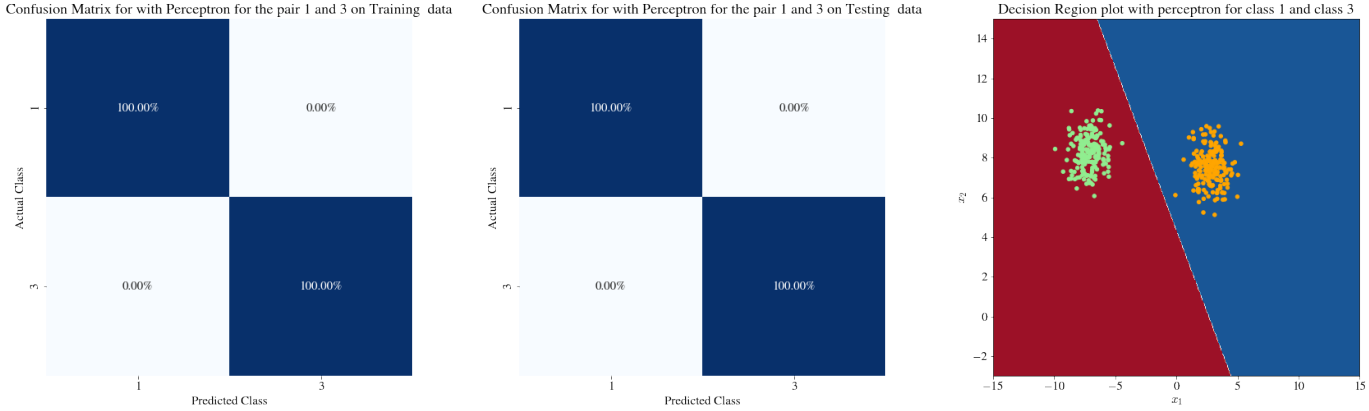


Figure 5: Confusion matrices on train & test data and decision plot between class 1 & class 3

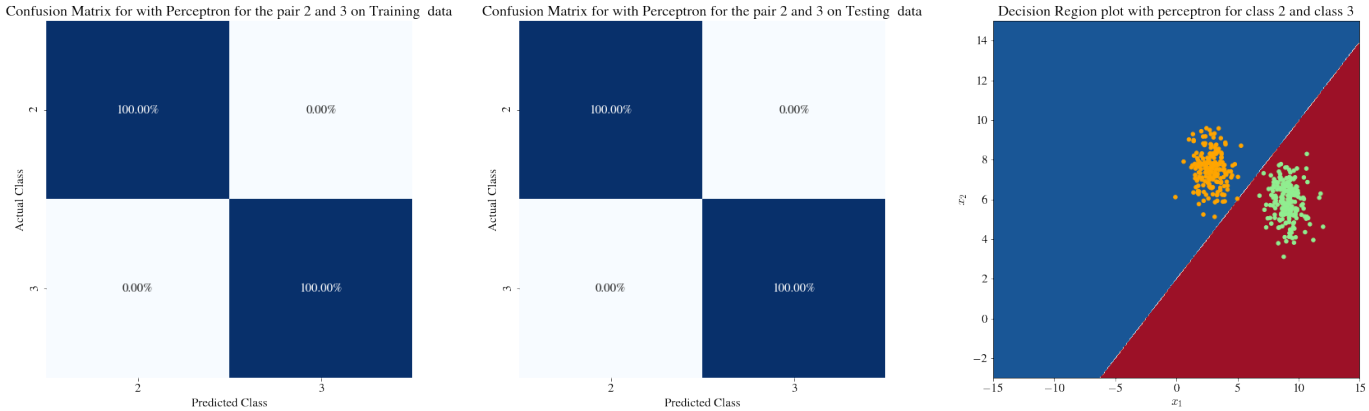
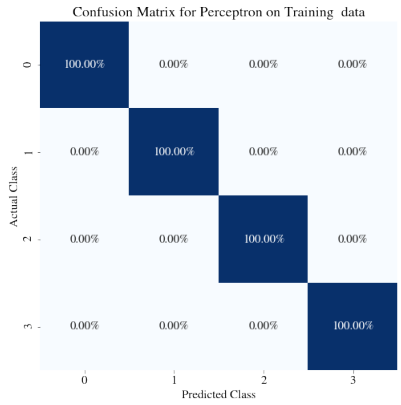
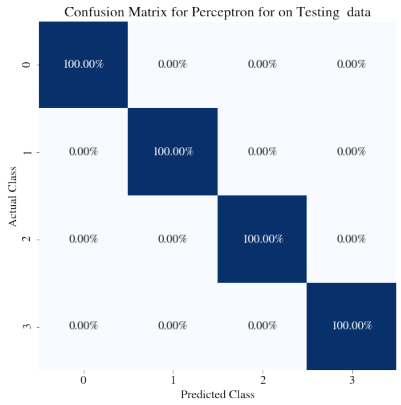


Figure 6: Confusion matrices on train & test data and decision plot between class 2 & class 3





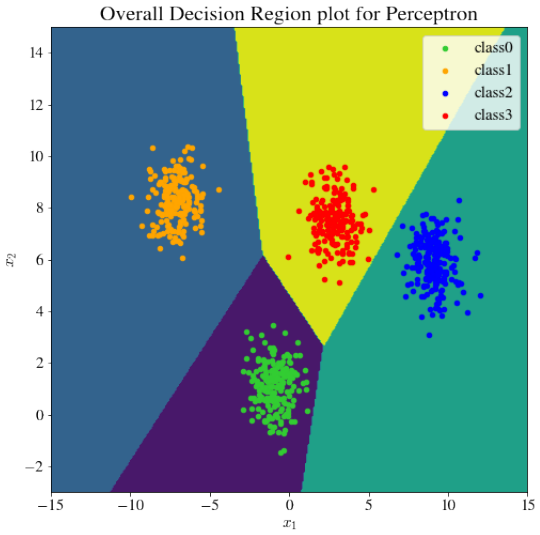


Figure 7: Overall Confusion Matrices on train & test data and Decision plot of all classes

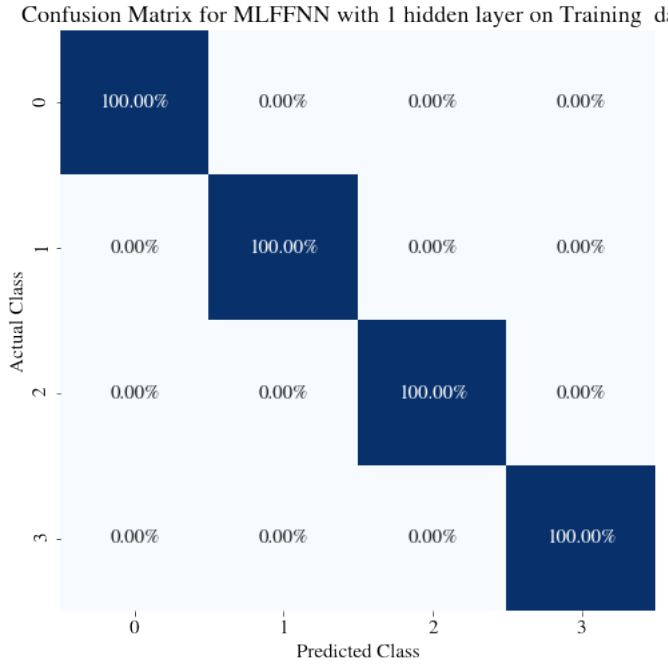
Observations:

- We observe a 100% accuracy in all the pair plots.
- In the pair plot of class 2 and class 3, we can see that the decision boundary is not an optimum boundary as the nearest training example is very close to boundary. This distinguishes a linear classifier with SVM.
- The overall decision region plot was plotted with a neural network having 4 output nodes. This was done to provide an overall accuracy of the dataset.

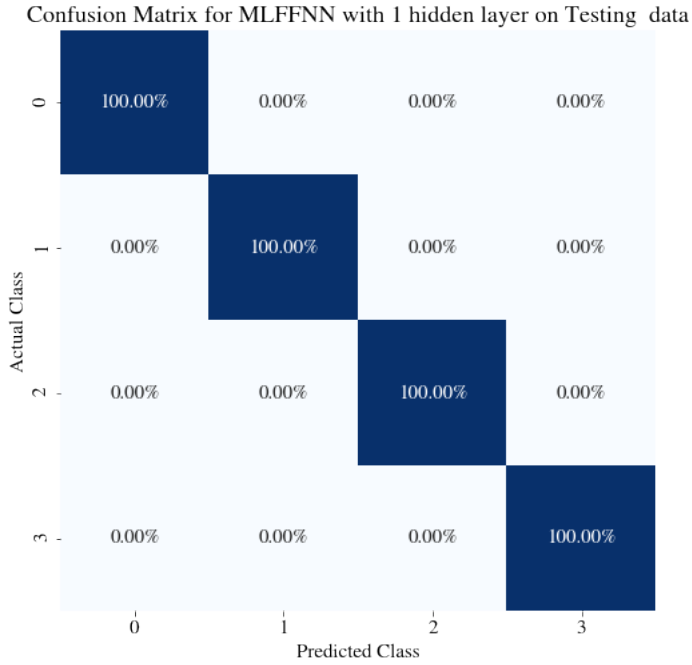
1.(a).2 Multilayer feedforward neural network (MLFFNN) with a single hidden layer for all classes

No of nodes in hidden layer	Accuracy		
	Training (%)	Validation (%)	Testing (%)
1	76.375	71.66	78.33
2	98.375	100	100
3	100	100	100
5	100	100	100
40	100	100	100

Best Model:- Model having **3 hidden layer nodes** is taken to be the best as it is the simplest model having 100% accuracy all throughout



(a) Confusion matrix for training dataset



(b) Confusion matrix for test dataset

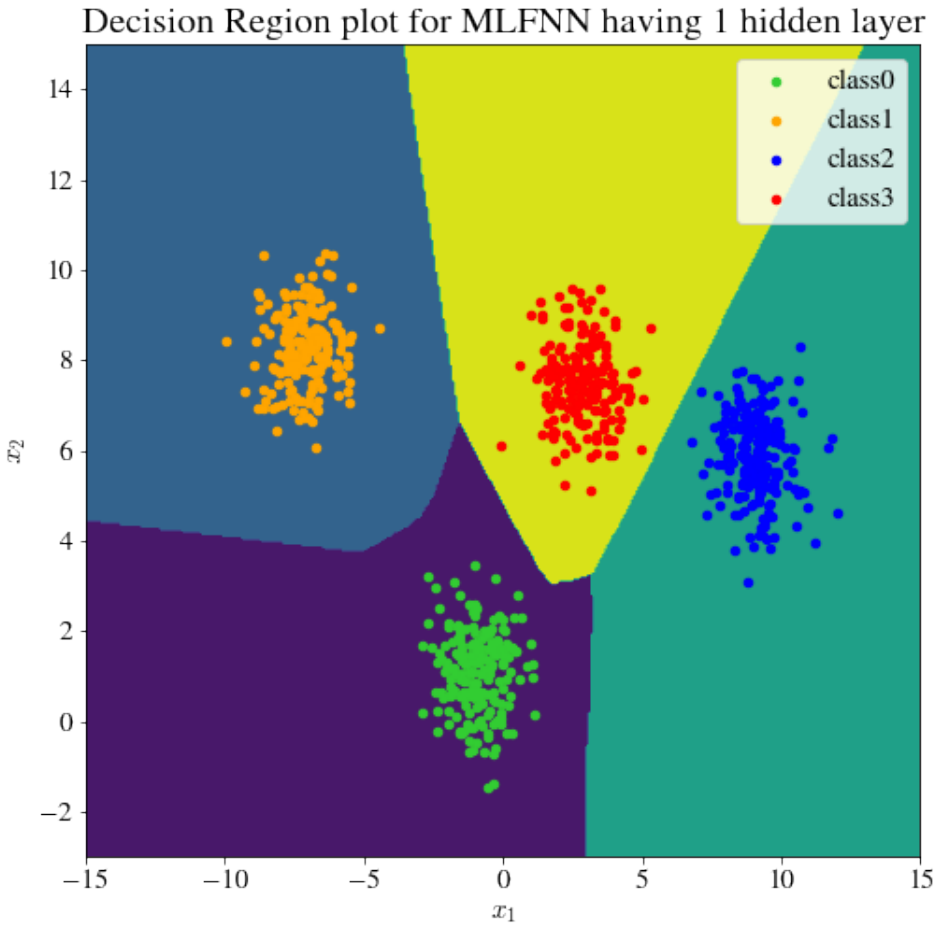


Figure 9: Decision region

Observations:

- We get 100% accuracy all across if the number nodes in the hidden layer is atleast 3
- This is because the data is synthetically generated linearly separable and the MLFFNN can even be used to separate non-linear data.
- If compare the decision boundary of MLFFNN and the Perceptron shown earlier, it can be seen that the decision boundaries are similar in nature.
- The Perceptron had strictly linear boundaries whereas MLFFNN has slightly curved boundaries (though almost a straight line) due to its non-linear activation function.
- Hence the MLFFNN converges faster taking less epochs to achieve 100% accuracy

1.(a).3 Linear SVM classifier for every pair of classes

C	Training(%)	Validation(%)
0.001	100	100
0.01	100	100
0.1	100	100
1	100	100
2	100	100
4	100	97.778

Best Model:- Taken to be **c=1** hereafter has 100% test accuracy

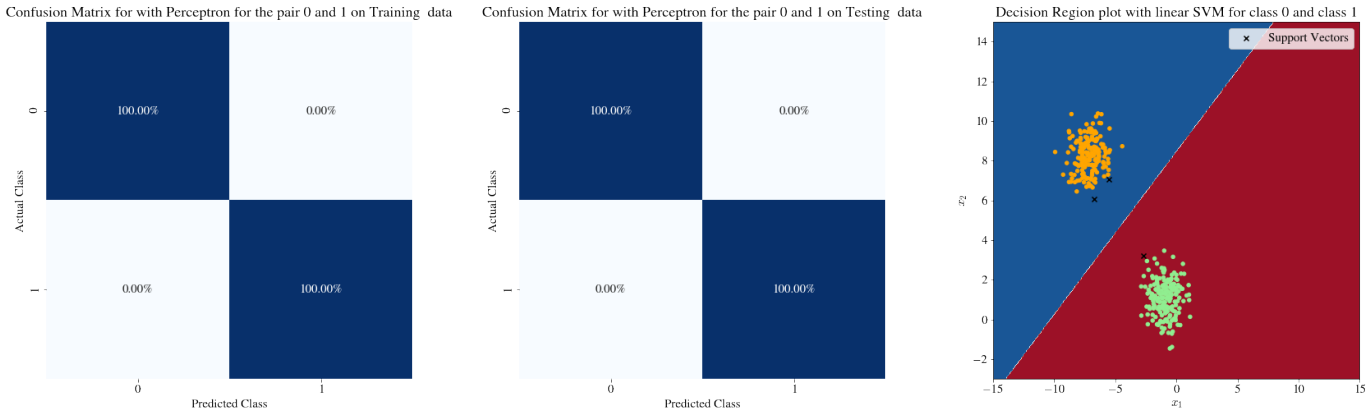


Figure 10: Confusion matrices on train & test data and decision plot between class 0 & class 1

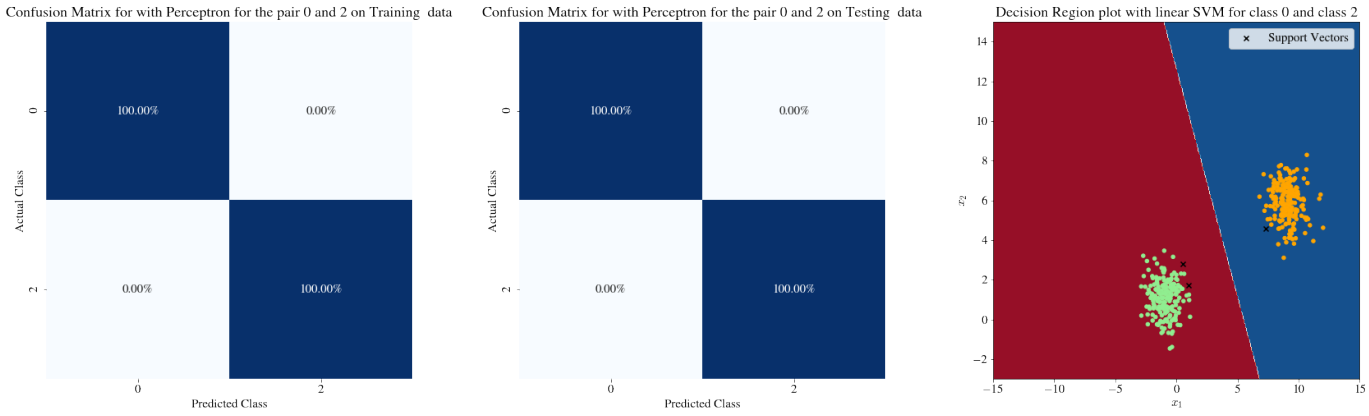


Figure 11: Confusion matrices on train & test data and decision plot between class 0 & class 2

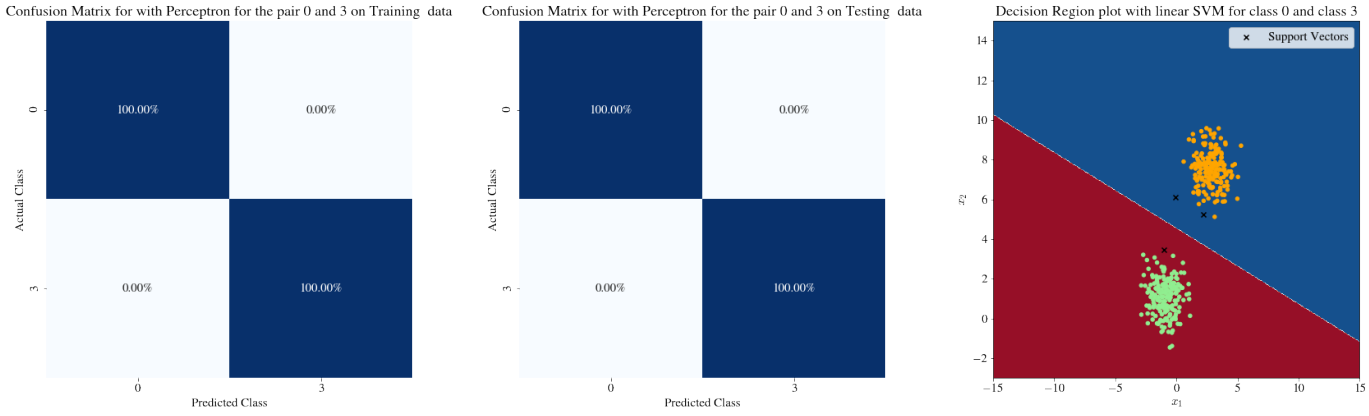


Figure 12: Confusion matrices on train & test data and decision plot between class 0 & class 3

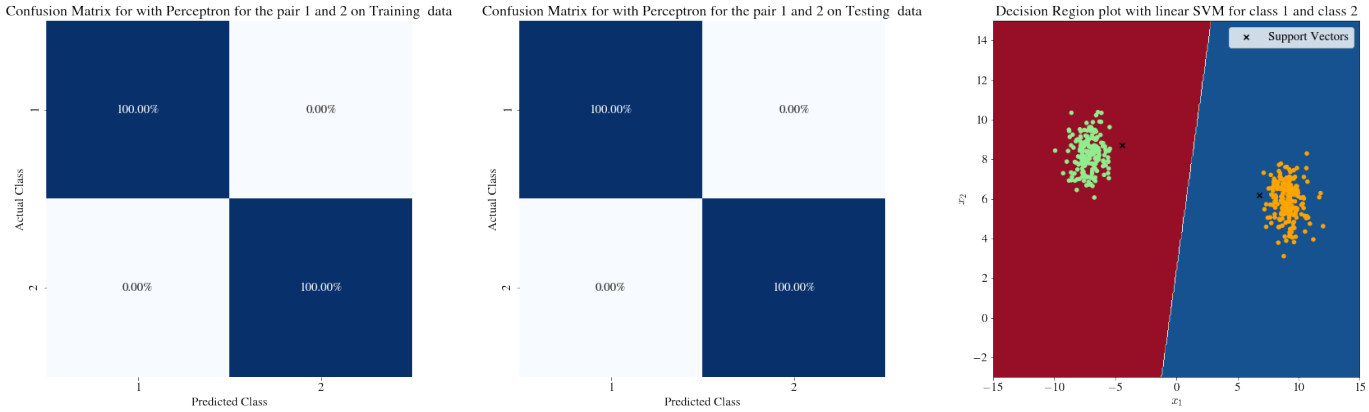


Figure 13: Confusion matrices on train & test data and decision plot between class 1 & class 2

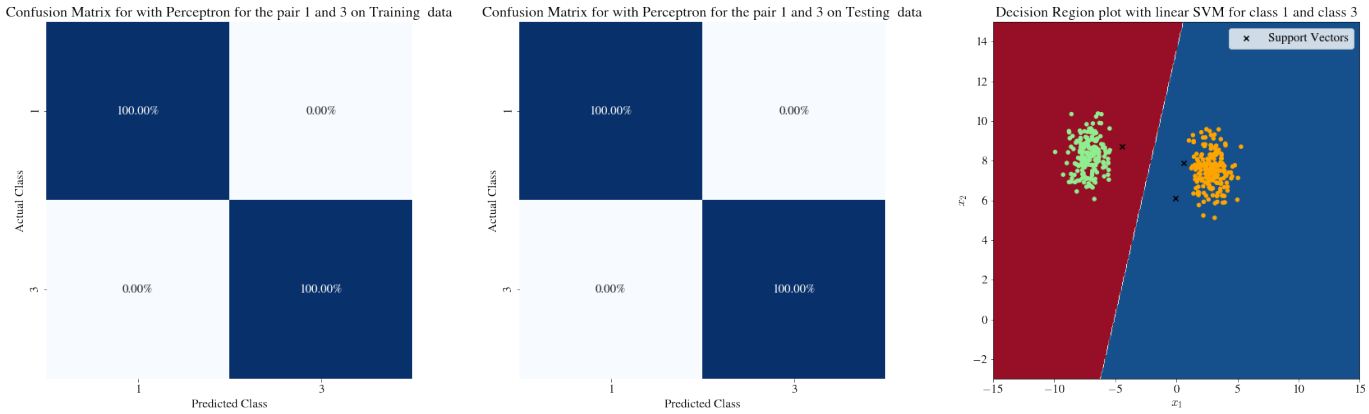


Figure 14: Confusion matrices on train & test data and decision plot between class 1 & class 3

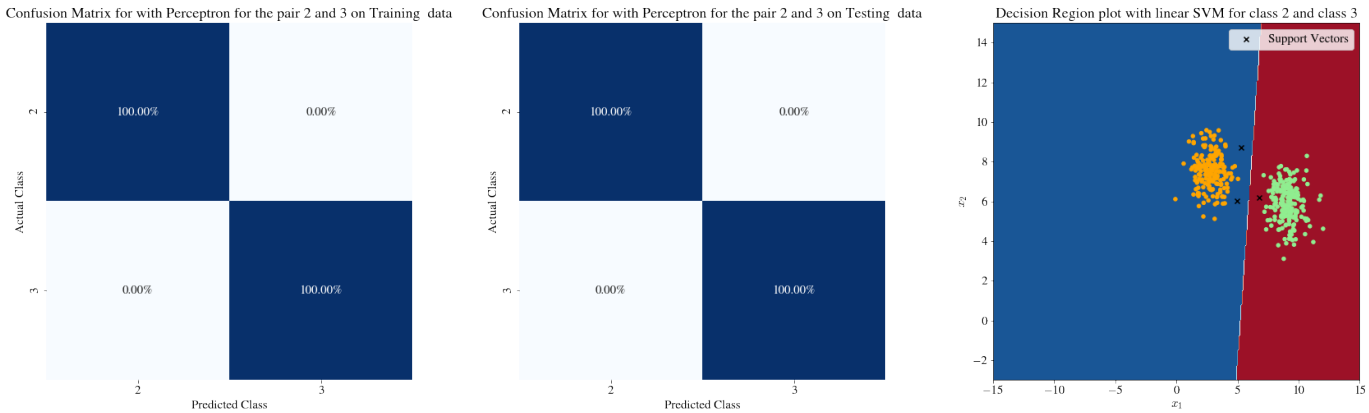
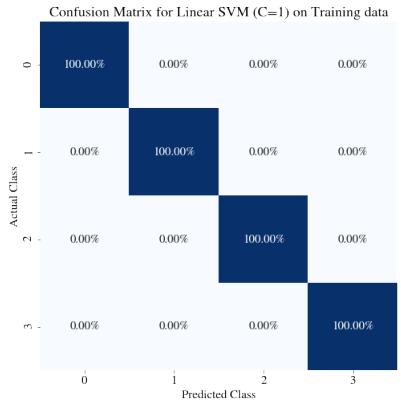
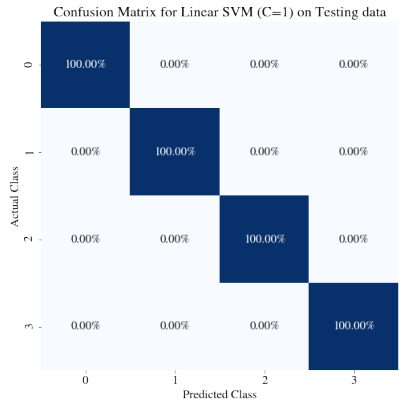


Figure 15: Confusion matrices on train & test data and decision plot between class 2 & class 3





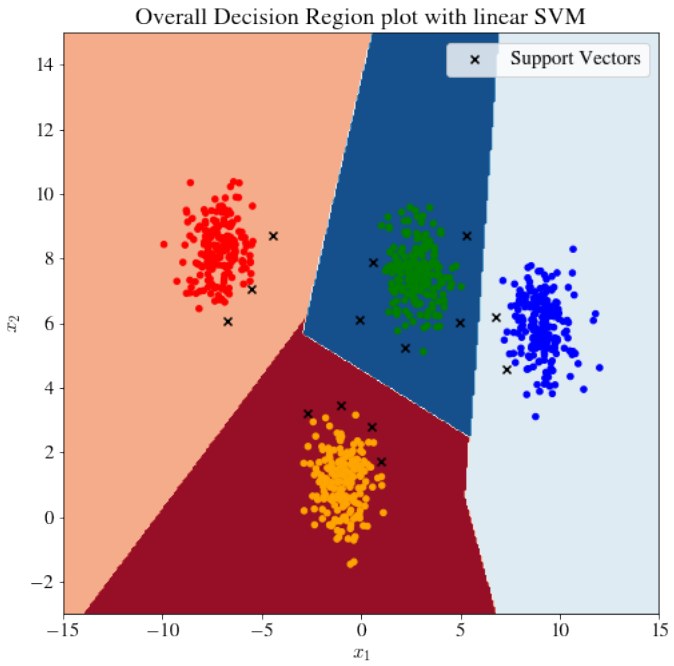


Figure 16: Overall Confusion Matrices on train & test data and Decision plot of all classes

Observations:

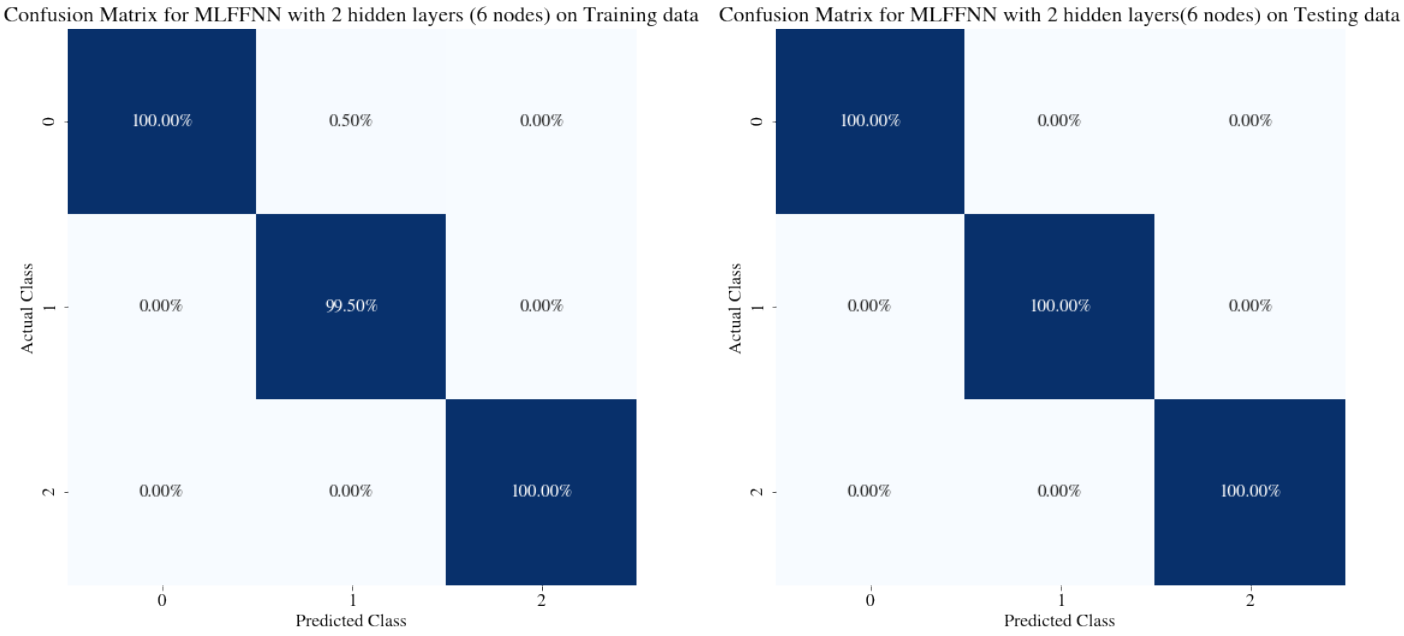
- We get 100% accuracy for C=1. A 100% accuracy was expected as the dataset is linearly seperable.
- We can see in the plots for every pair of classes that the decision region boundary is approximately halfway between the different classes which justifies the SVM's ability to find the optimal linear boundary.
- We can see that for C=4 the validation accuracy is less than 100%. This is because of the overfitting of the model on training set.

1.(b) Nonlinearly separable data set for static pattern classification

1.(b).1 MLFFNN with two hidden layers

No of nodes in the hidden layer	Accuracy		
	Training (%)	Validation (%)	Testing(%)
3	91.5	86.667	88.889
4	97.1667	100	95.556
5	99.667	100	97.667
6	99.834	100	100
7	99.50	100	97.778

Best Model:- It can be seen that hidden layers having **6 nodes each** is the best model. It also has 100% test data accuracy



(a) Confusion matrix for training dataset

(b) Confusion matrix for test dataset

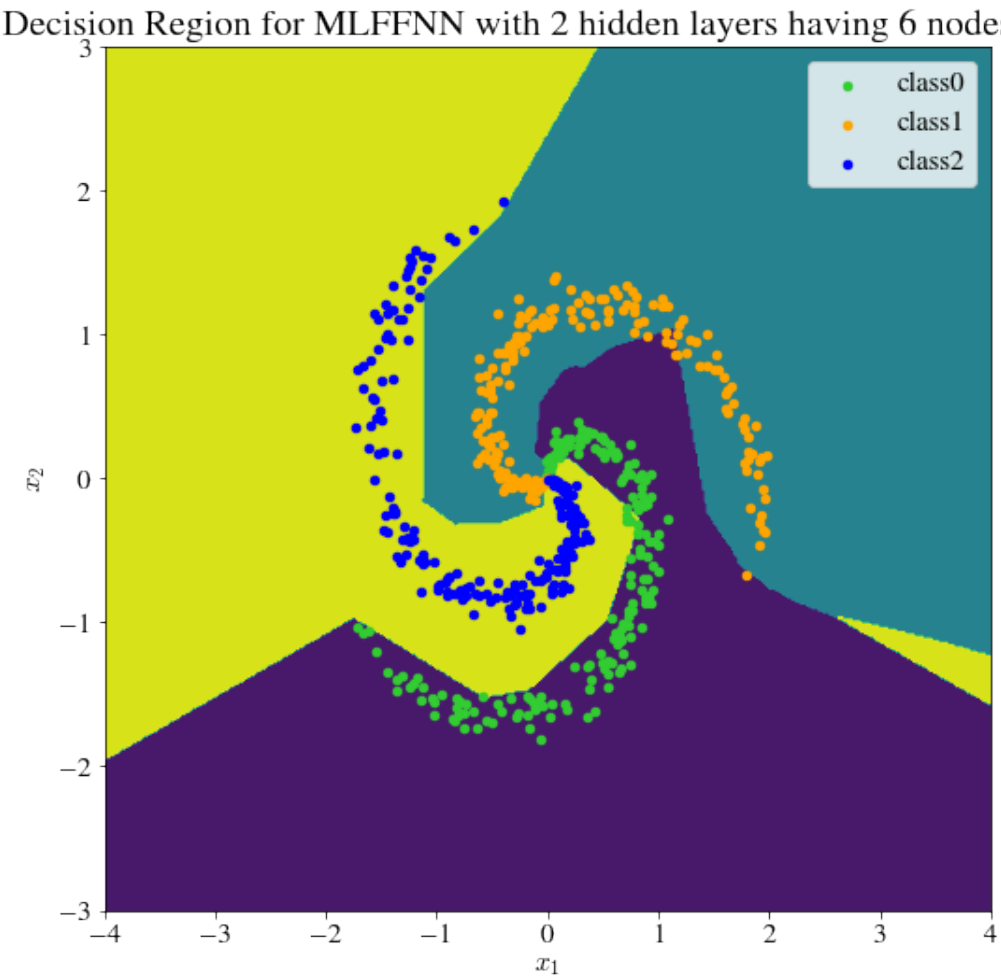


Figure 18: Decision region

Comments on Model Configuration and Convergence Condition

- The best model in this particular instance (i.e. fixing random seed value to 42) is the one which has 6 nodes
- The optimizer is taken to be "adam" and the activation function for the hidden layer is "relu".
- The convergence criteria is when the validation loss is below a certain threshold (10e-4) with a patience (buffer) of 5 epochs
- The model takes 112 epochs to converge

Observations

- It can be seen that the validation accuracy with this particular model configuration has a maxima at 6 nodes and decreases thereafter.
- It is clear from the above plot that the decision boundary is non-linear which is characteristic of MLFFNN.

1.(b).2 Outputs of nodes in layer after convergence - 1 epoch

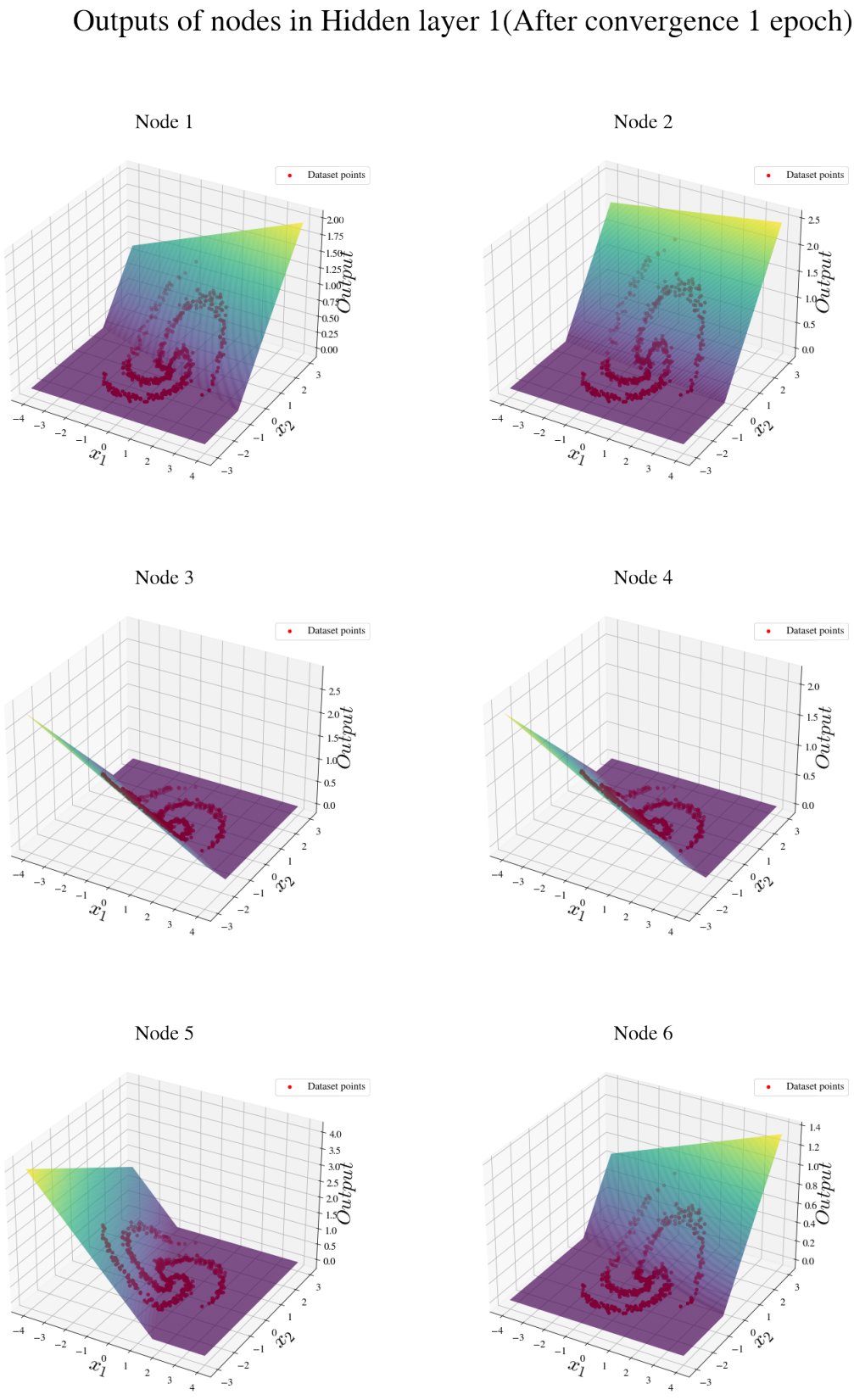


Figure 19: Surface plot for output of each node in hidden layer 1

Outputs of nodes in Hidden layer 2(After convergence 1 epoch)

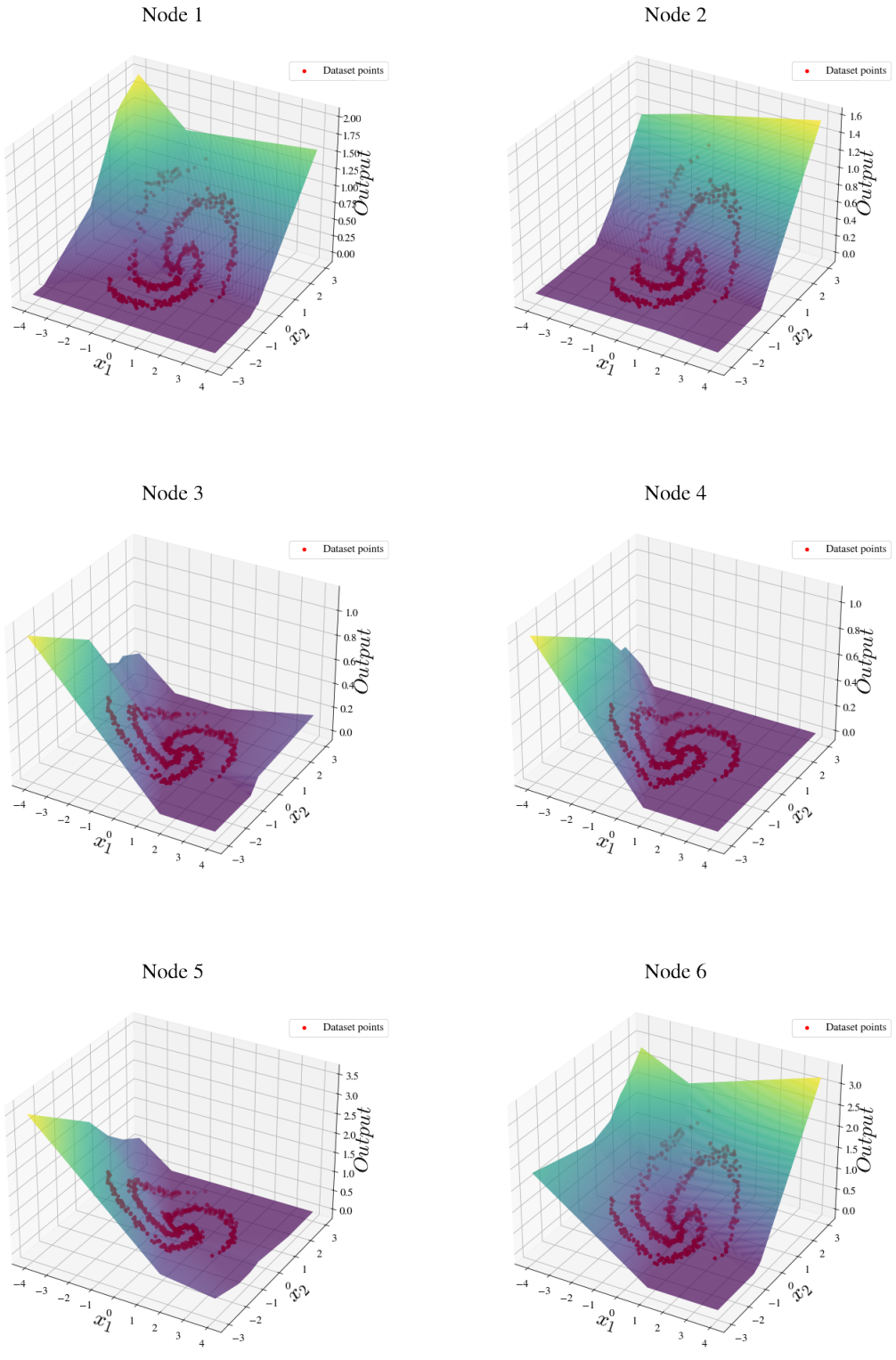


Figure 20: Surface plot for output of each node in hidden layer 2

Outputs (posterior probability estimates) of nodes in the ouput layer after convergence (1 epoch)

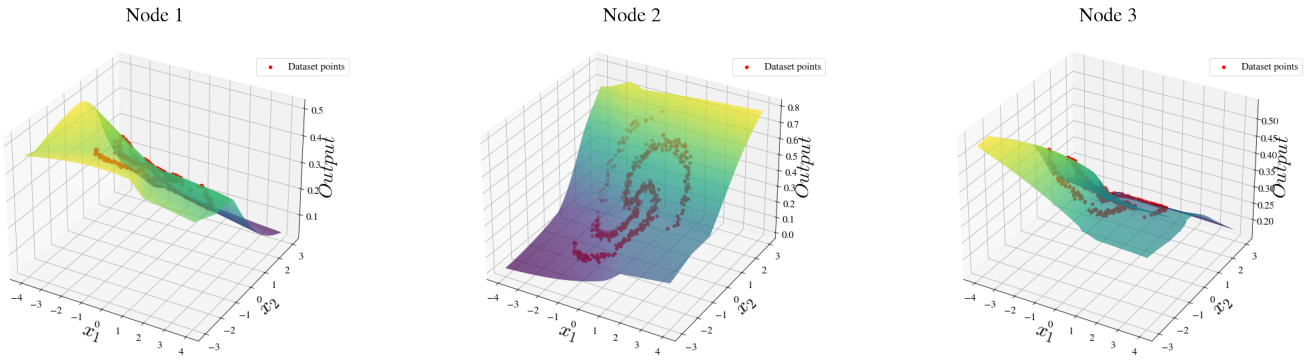


Figure 21: Surface plot for output of each node in output layer after 1 epoch

1.(b).3 Outputs of nodes in layer after convergence - 5 epochs

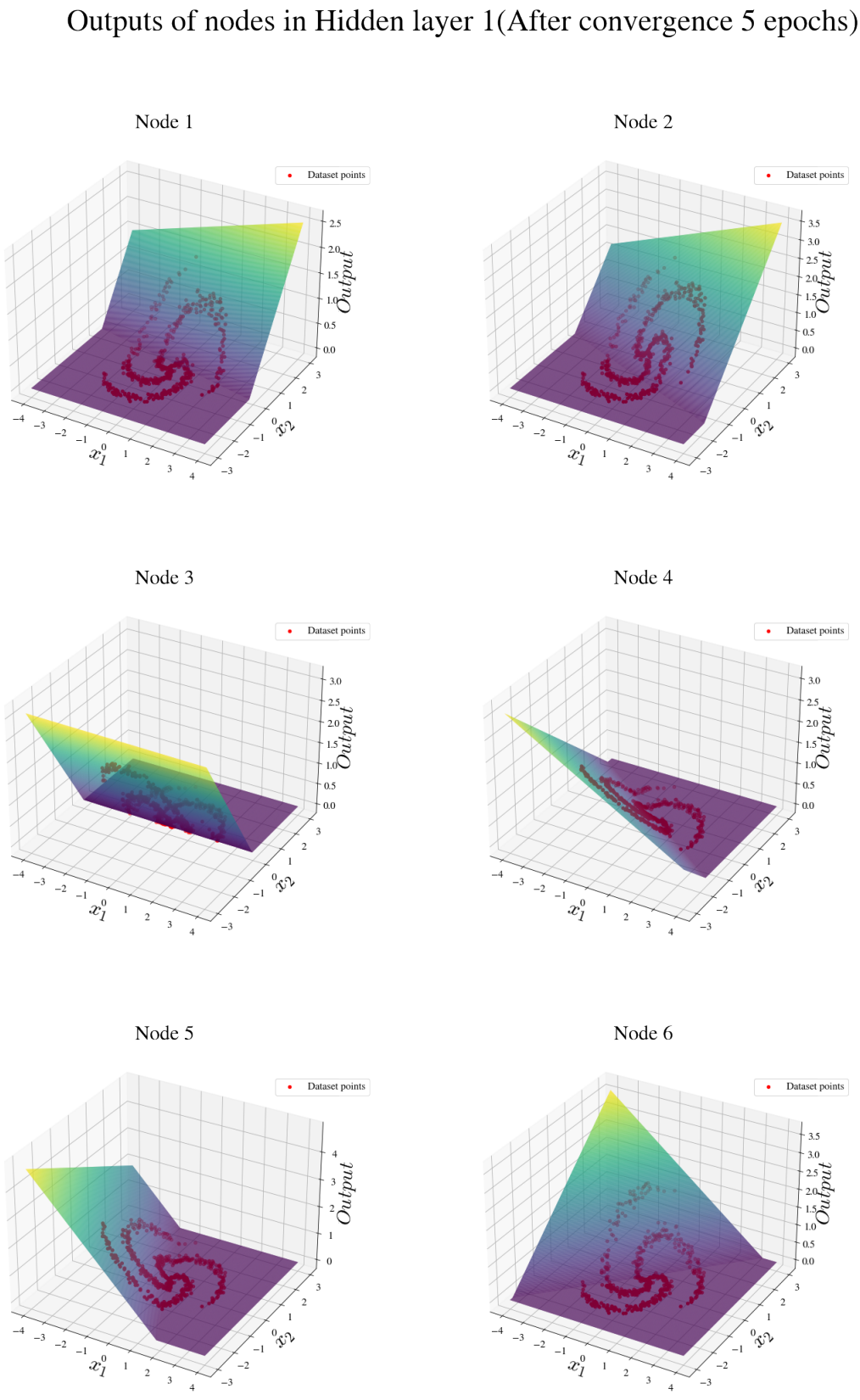


Figure 22: Surface plot for output of each node in hidden layer 1

Outputs of nodes in Hidden layer 2(After convergence 5 epochs)

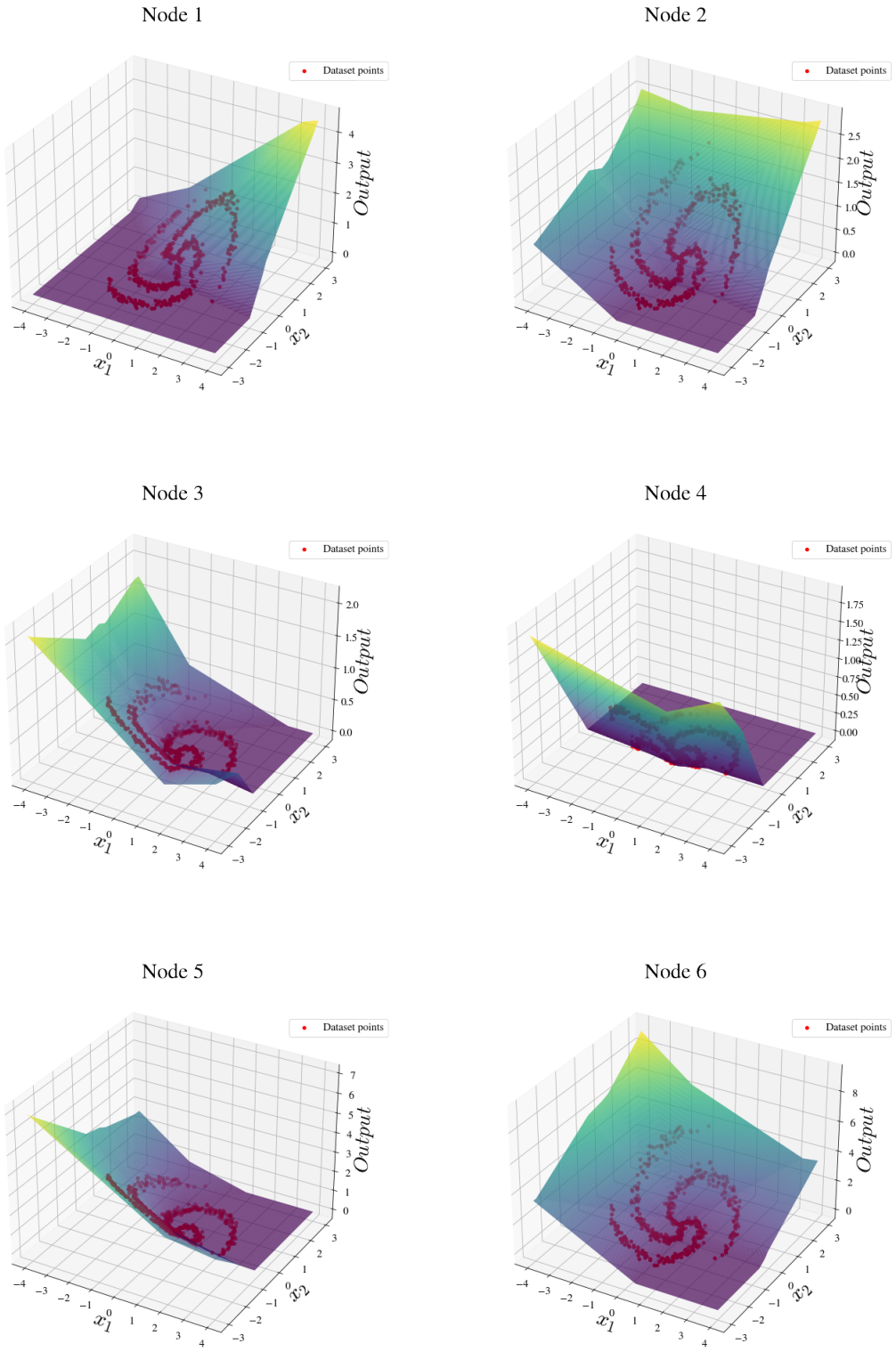


Figure 23: Surface plot for output of each node in hidden layer 2

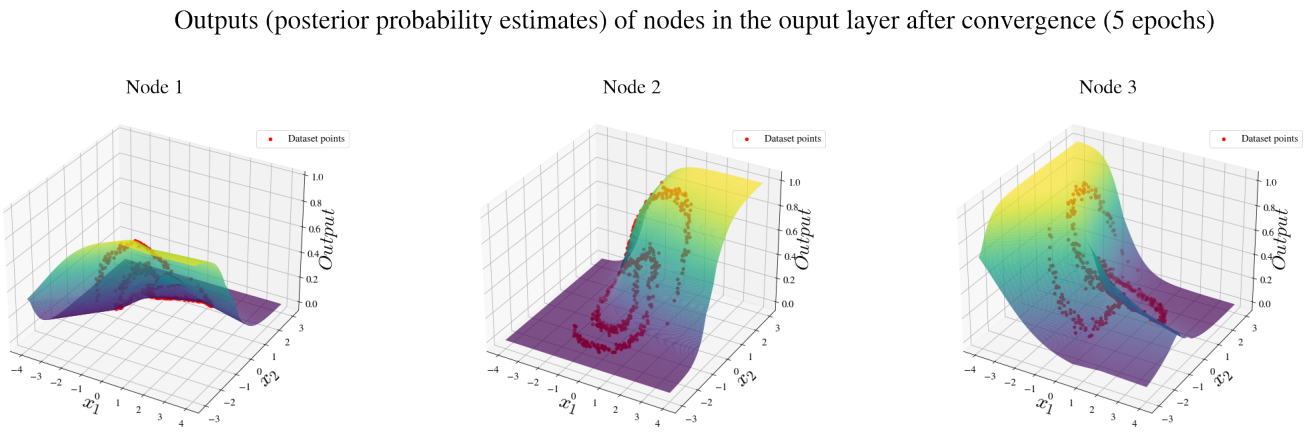


Figure 24: Surface plot for output of each node in output layer after 5 epochs

1.(b).4 Outputs of nodes in layer after convergence - 20 epochs

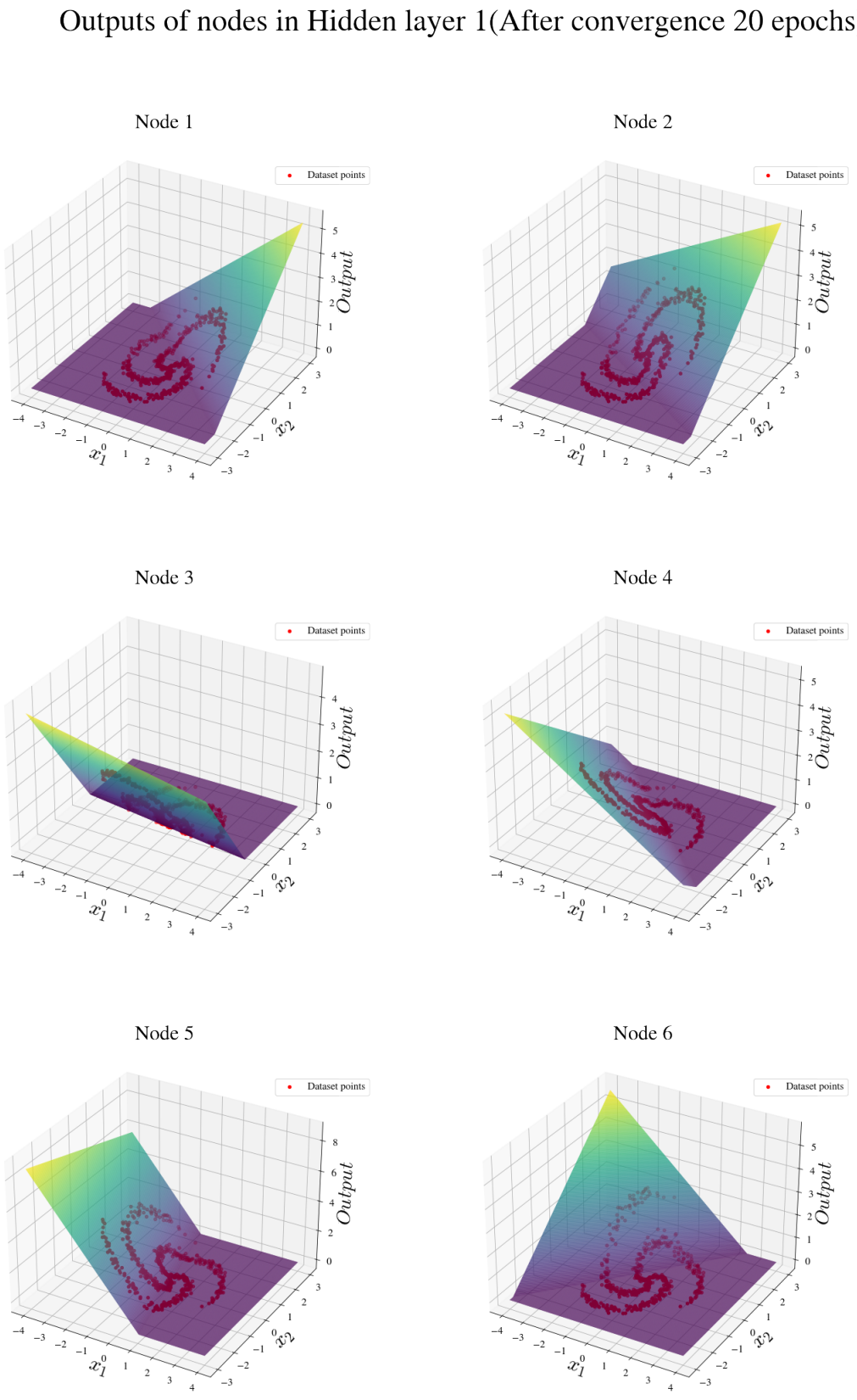


Figure 25: Surface plot for output of each node in hidden layer 1

Outputs of nodes in Hidden layer 2(After convergence 20 epochs)

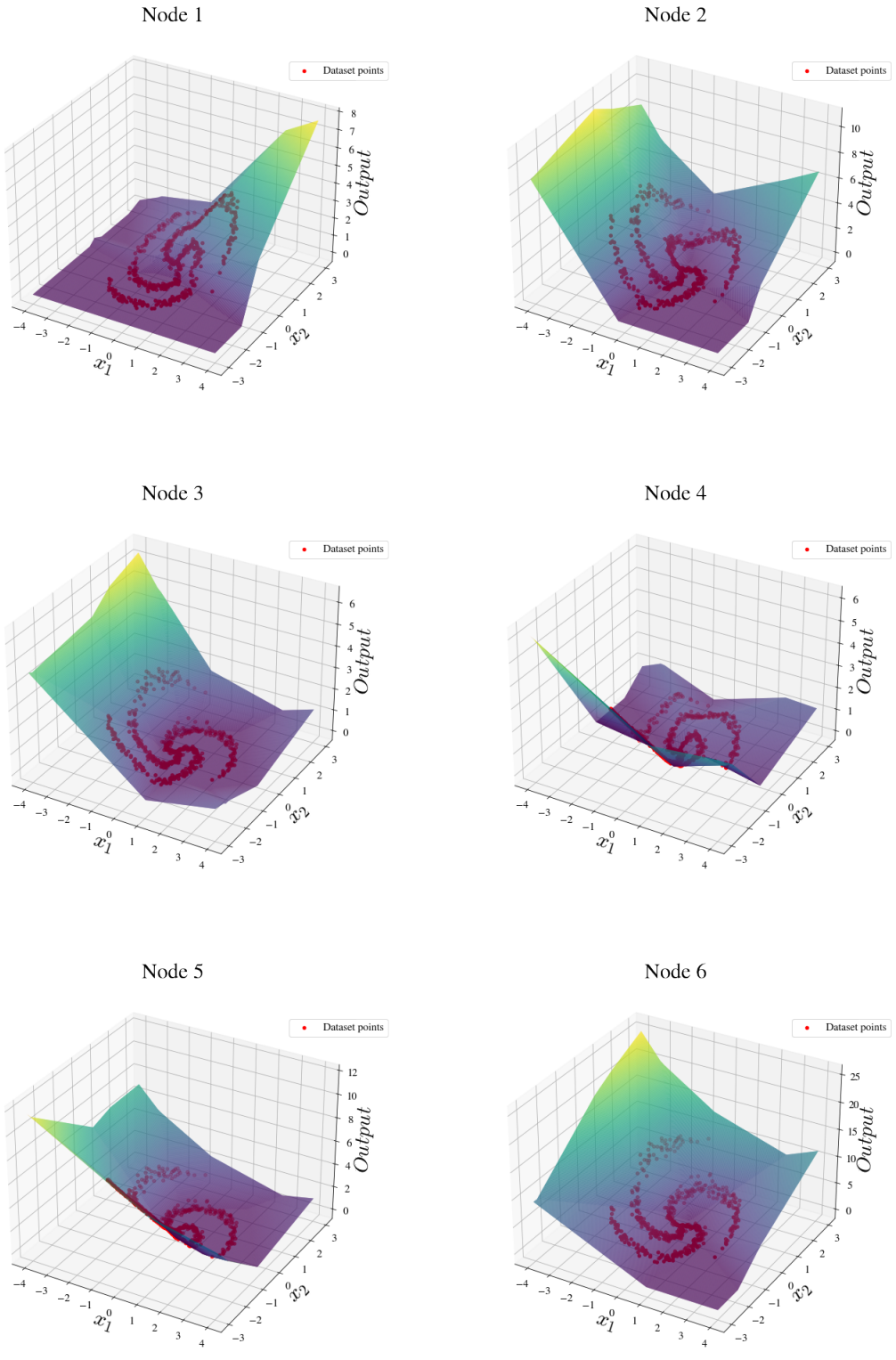


Figure 26: Surface plot for output of each node in hidden layer 2

Outputs (posterior probability estimates) of nodes in the ouput layer after convergence (20 epochs)

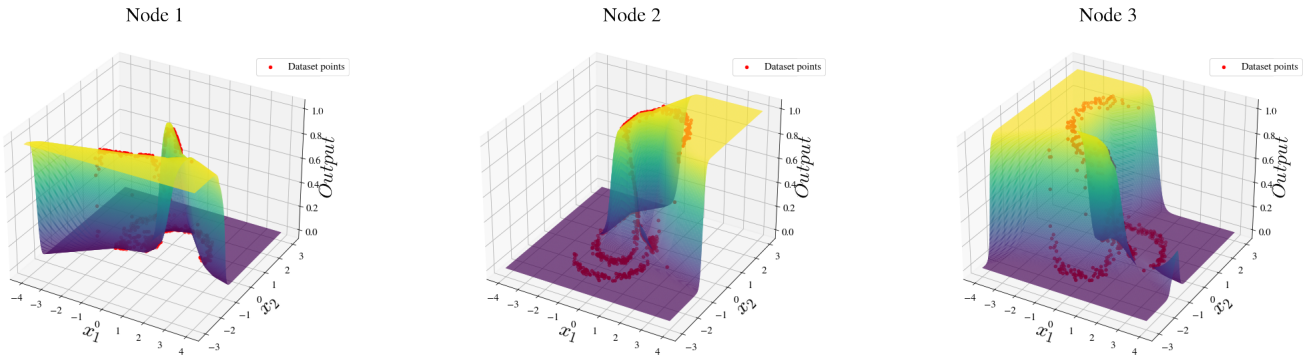


Figure 27: Surface plot for output of each node in output layer after 20 epochs

1.(b).5 Outputs of nodes in layer after convergence - 112 epochs

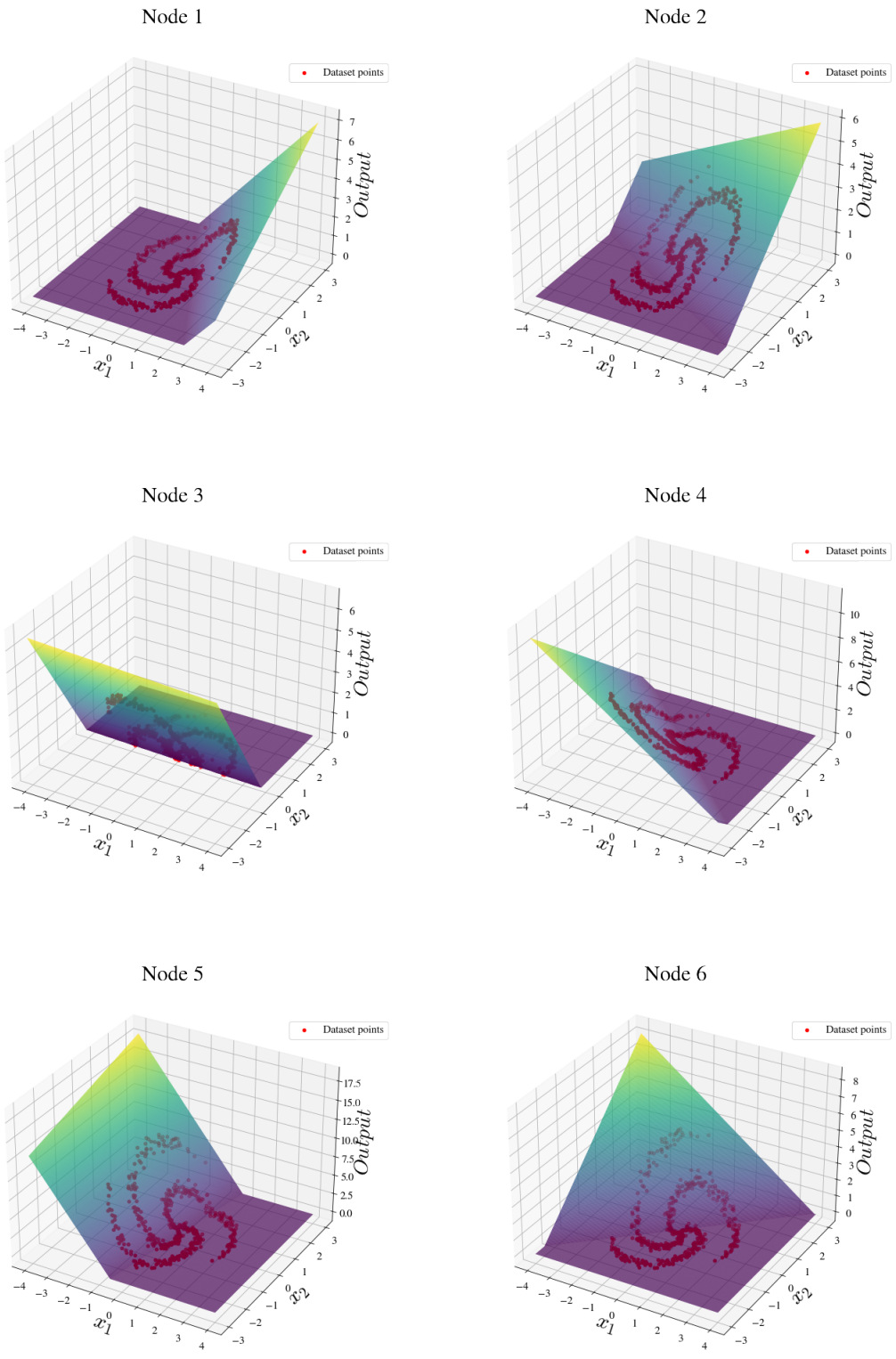


Figure 28: Surface plot for output of each node in hidden layer 1

Outputs of nodes in Hidden layer 2(After convergence 112 epochs)

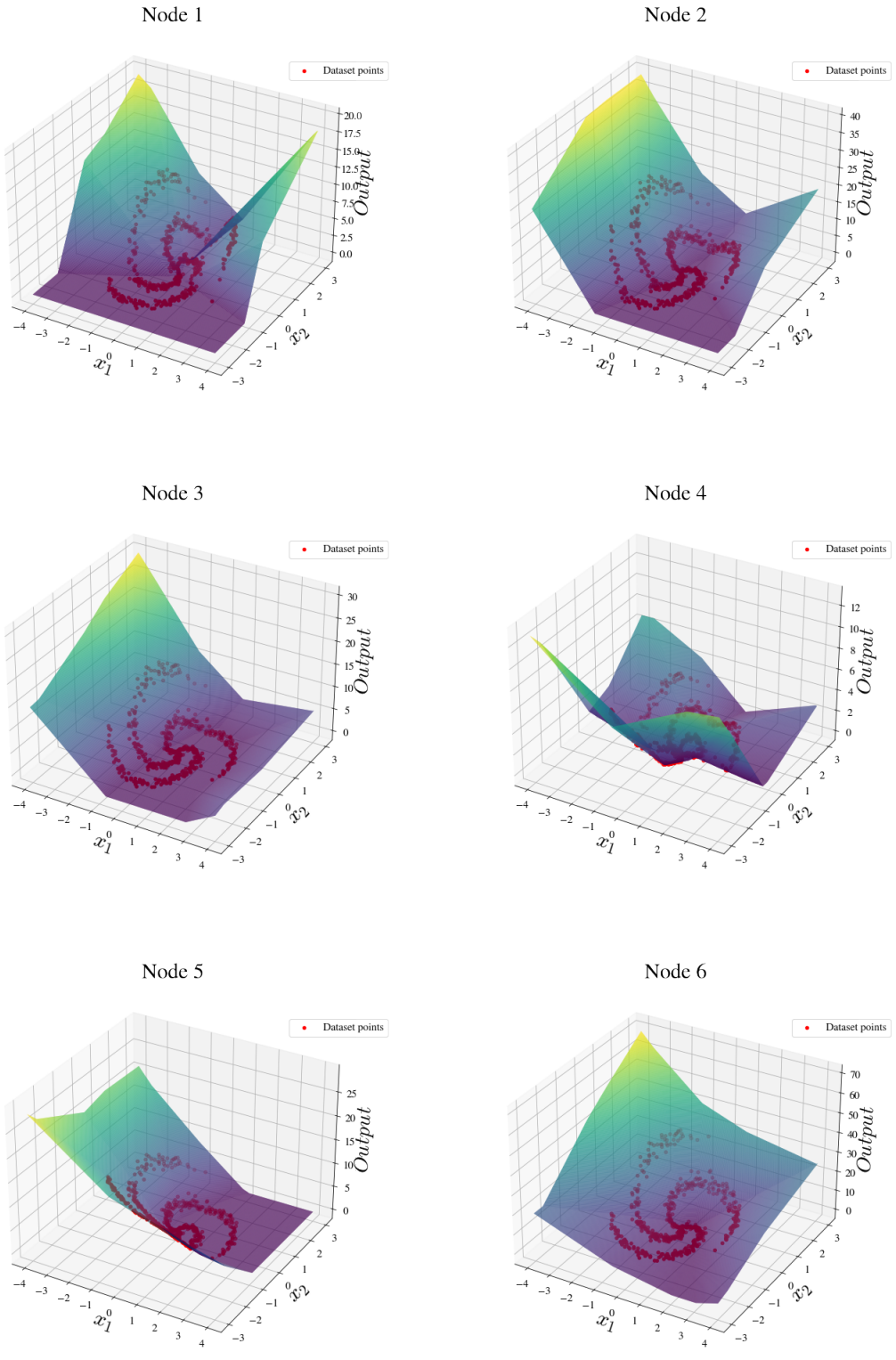


Figure 29: Surface plot for output of each node in hidden layer 2

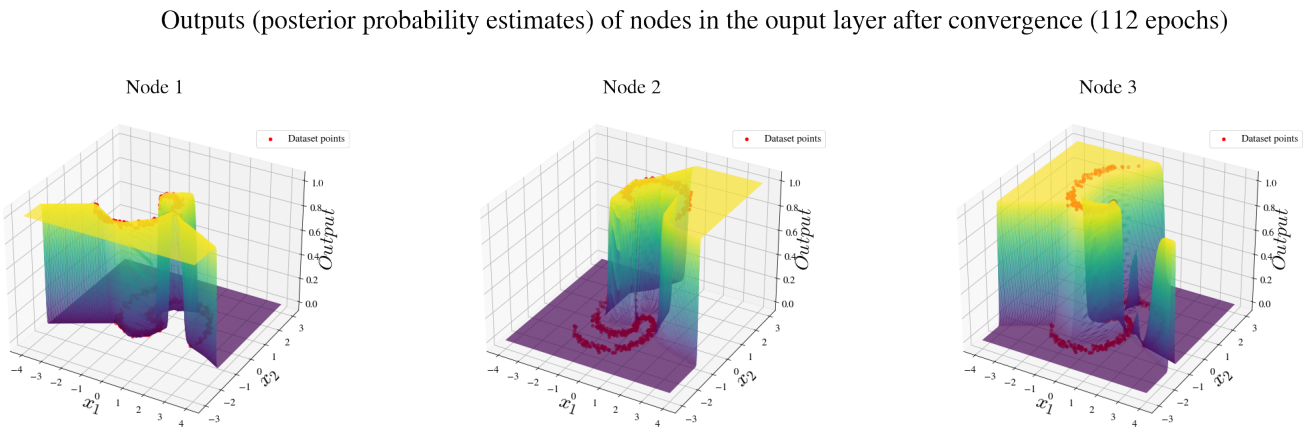


Figure 30: Surface plot for output of each node in output layer after convergence

Observations on the surface plots:

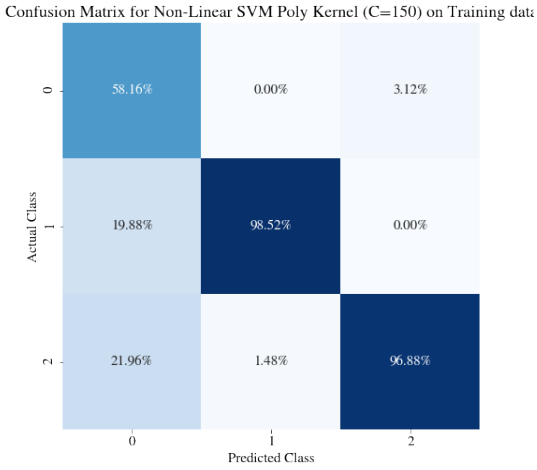
- We have plotted surface plots for epochs 1,5 ,20 & 112 (convergence)
- The surface plots are seen to become more accurate (especially) the ouput layer as the number of epochs increases.
- It can be seen from the above plot that the output layer values which are nothing but estimates of posterior probabilities (softmax function) are almost spot on after convergence (112 epochs).

1.(b).6 Nonlinear SVM using one-against-the-rest approach : (a) Polynomial kernel, (b) Gaussian kernel

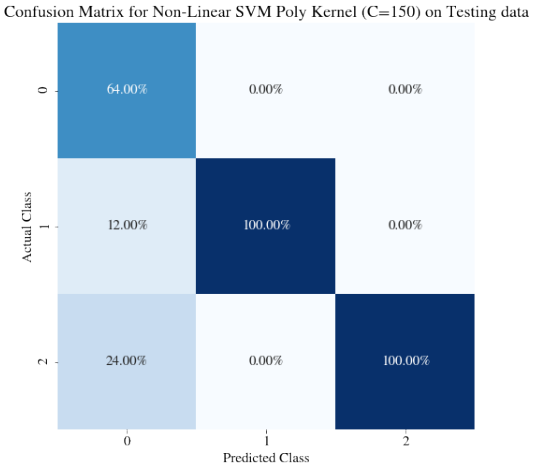
(a) Polynomial Kernel

C	Training(%)	Validation(%)
0.01	59.16	57.77
4	73.33	73.33
150	75.5	77.77
500	75.5	77.77
1000	75.5	77.77

Best Model:- c= 150 is the best model having testing accuracy 80.00%



(a) Confusion matrix for training dataset



(b) Confusion matrix for test dataset

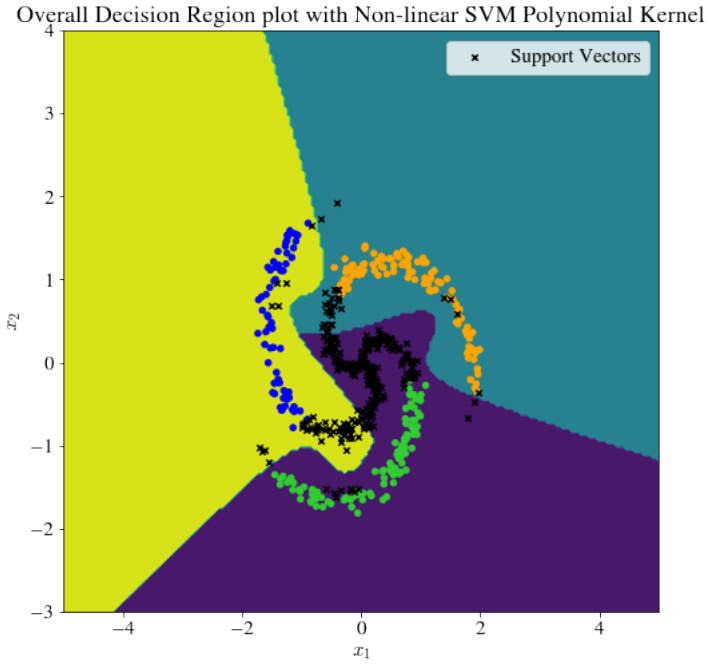


Figure 32: Decision region

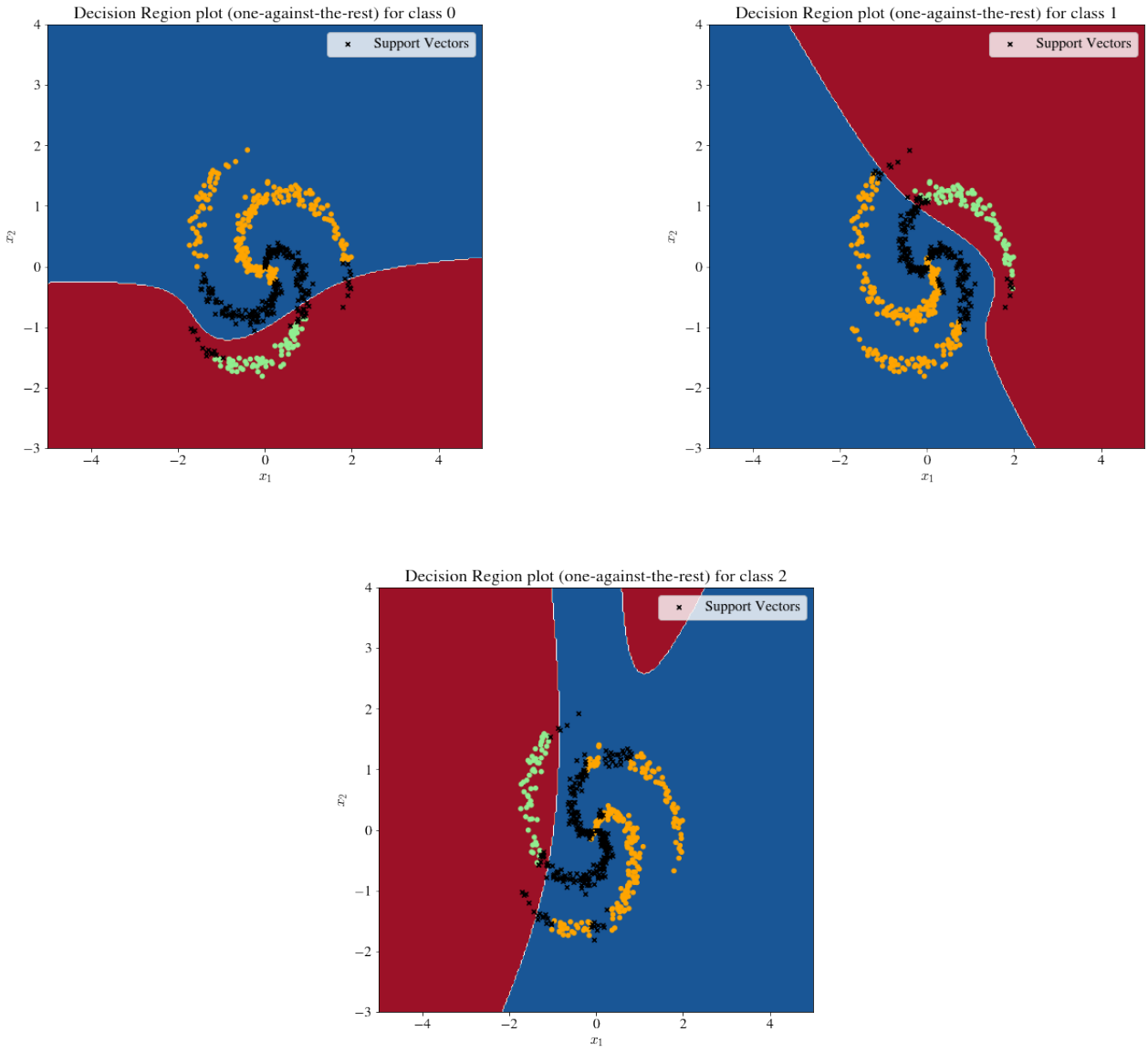


Figure 33: Decision Region plot (one-against-the-rest) for each class

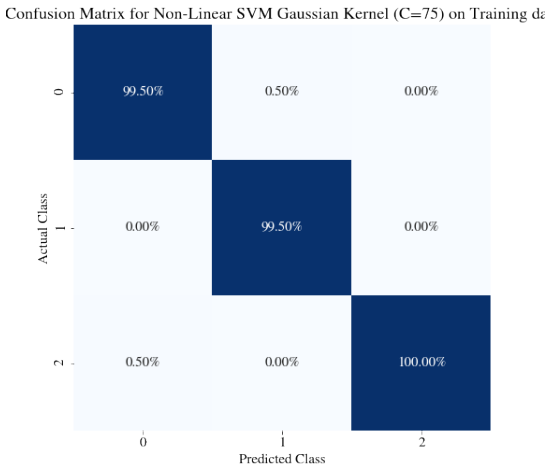
Observations:

- C=150 was chosen as the best Model as there were no further improvements in validation accuracy.
- The polynomial Kernel struggles to classify the non-linear spiral data compared to Gaussian SVM Kernel.
- The polynomial Kernel doesn't give a high accuracy because no polynomial function could separate the spiral data properly.This is clearly evident from the one-against-the-rest plots.
- Among all the degrees, degree=3 was found to be the best power for classification.

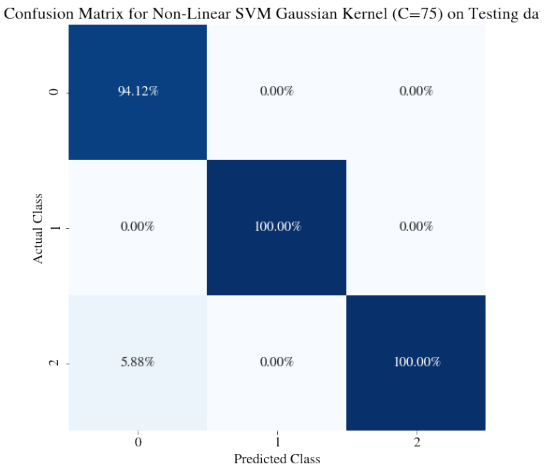
(b) Gaussian Kernel

C	Training(%)	Validation(%)
0.01	70.83	66.66
4	99.33	100
50	99.5	100
75	99.67	100
150	99.67	100

Best Model:- c= 75 is the best model having testing accuracy 97.77%



(a) Confusion matrix for training dataset



(b) Confusion matrix for test dataset

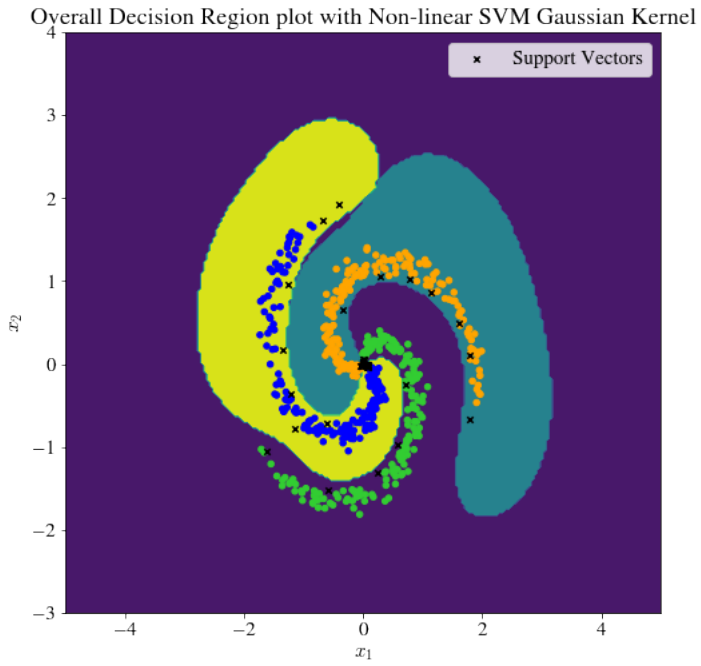


Figure 35: Decision region

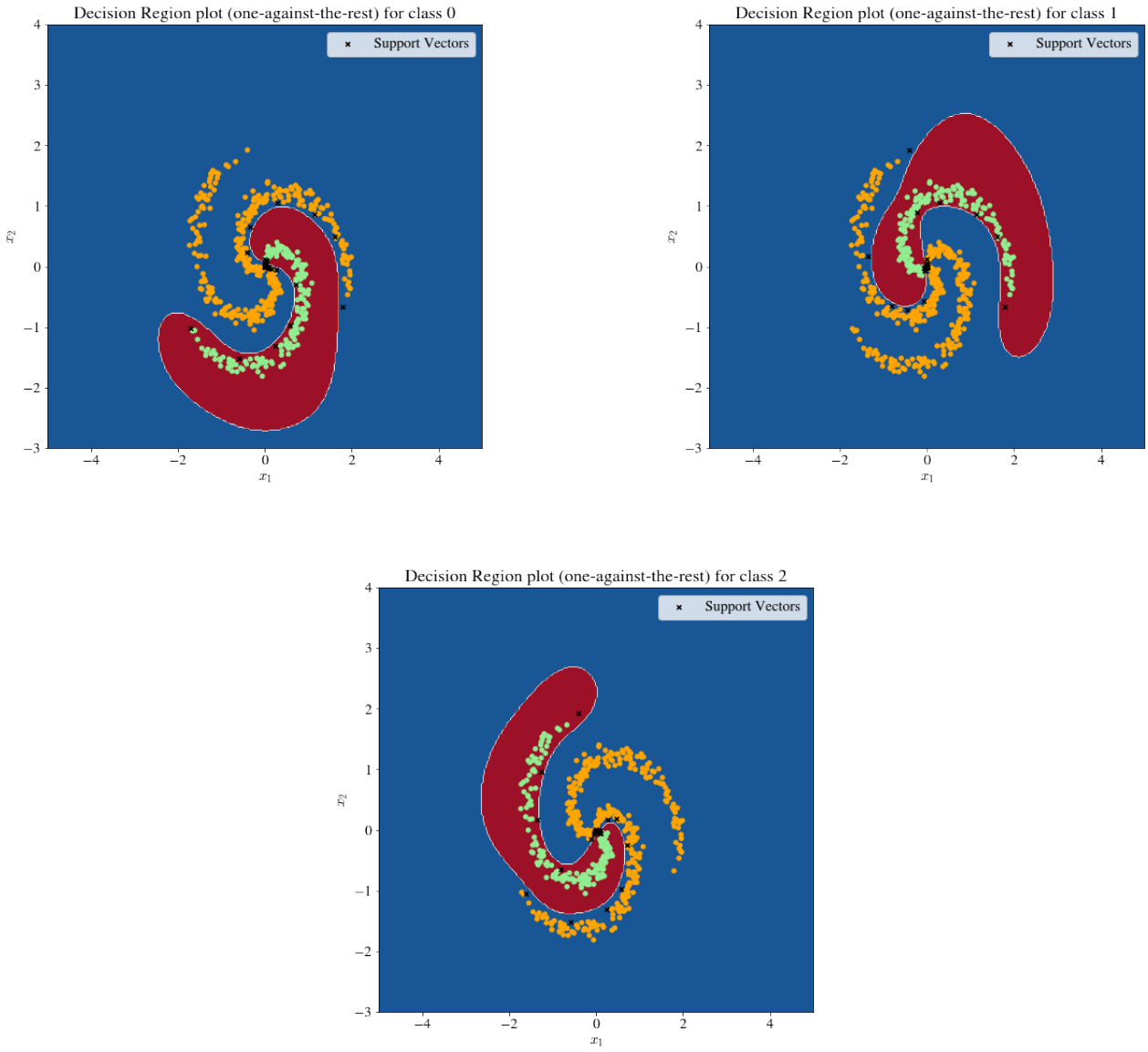


Figure 36: Decision Region plot (one-against-the-rest) for each class

Observations:

- C=75 was chosen as the best model as there were no further improvements in validation accuracy.
- The Gaussian Kernel classifies the non-linearly separable spiral data with almost 100% accuracy.
- The Gaussian Kernel gives a very high accuracy. We can see in one-against-the-rest plots that the decision region for each class is very distinctively and clearly chosen by the Gaussian SVM.

2 Dataset 2: Real World Data Sets

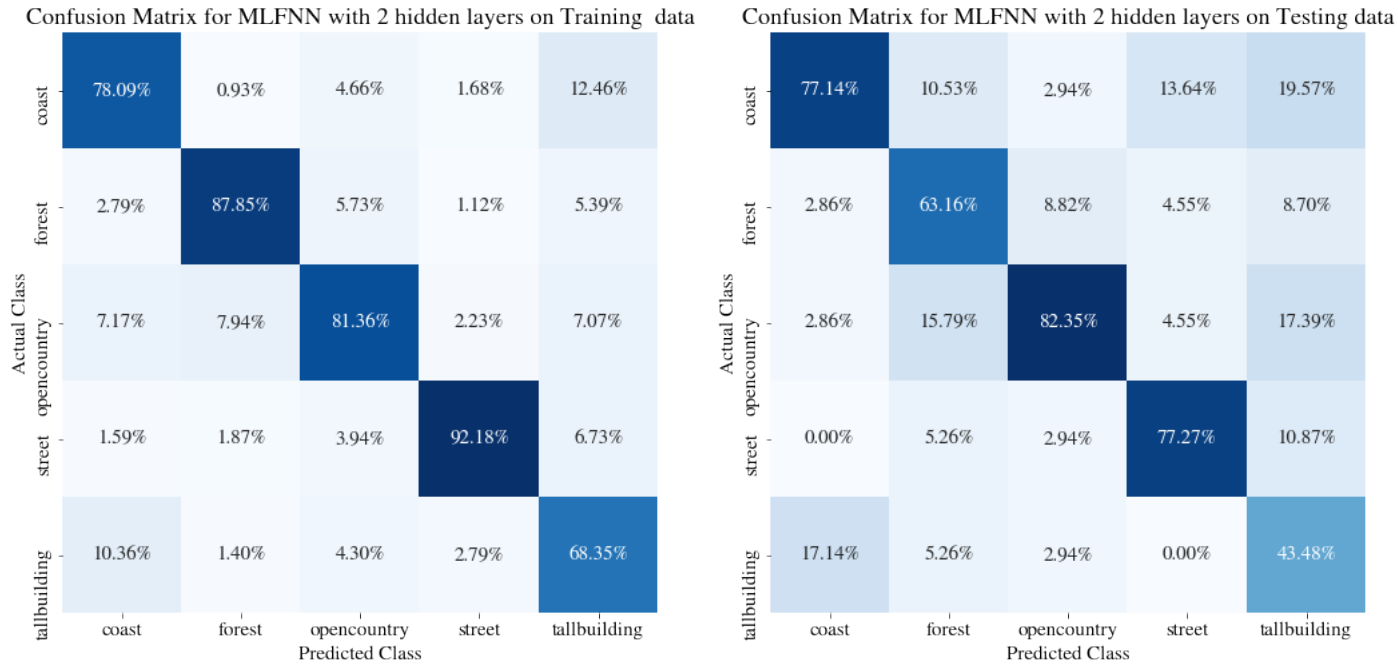
2.(a) Image data set for static pattern classification

2.(a).1 MLFFNN with two hidden layers

Table of Classification Accuracies

No of nodes		Accuracy		
Hidden Layer 1	Hidden Layer 2	Training (%)	Validation (%)	Testing (%)
5	5	58.11	59.09	52.55
10	10	64.18	62.98	50.43
25	25	76.39	59.64	60.4
37	31	77.31	64.32	63.12
41	47	87.29	68.12	62.56
55	51	82.09	72.57	66.21
67	27	87.95	69.142	62.46
68	41	88.41	68.29	64.54
75	75	77.950	65.49	65.363
150	150	85.57	65.49	64.24

Best Model:- The optimum model has hidden layer nodes in the range 40-70. In this particular instance (random seed state) node no. (55,51) gave the best classification accuracies as seen from the table. The max testing accuracy came out to be around 66%



(a) Confusion matrix for training dataset

(b) Confusion matrix for test dataset

Figure 37: Confusion Matrices for MLFFNN using 2 hidden layer for the best model instance

Comments on Model Configuration

- The loss function taken is categorical-crossentropy,optimizer is taken to be "adam" and the activation function for the hidden layer is "relu"

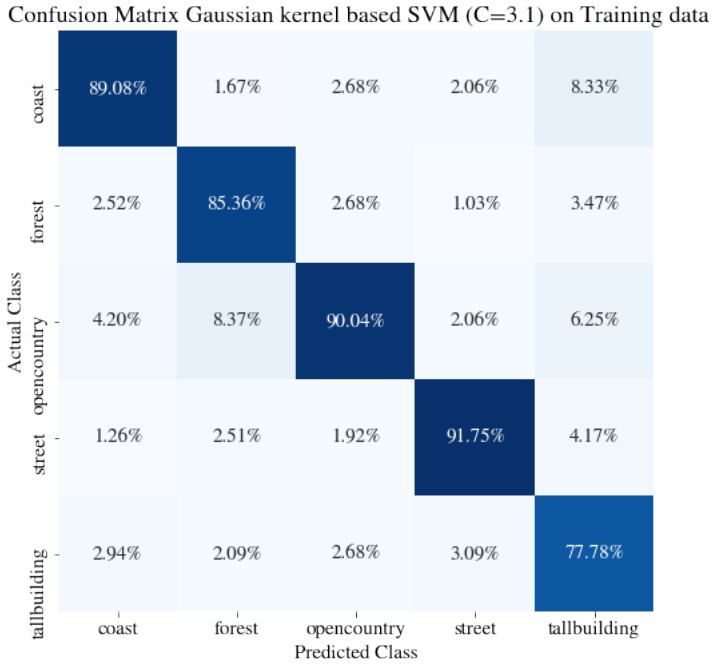
Comments Hyperparameter tuning to find Optimum Hidden Layer nodes

- Random search has been used to find the optimum parameters. In this particular case the optimum nodes for the 2 hidden layers were found out to be (55,51).
- However, we need to keep in mind that this validation accuracy of 72.57% was achieved in that particular random state. As the weight initializations are random results differ even within a particular model configuration.
- It was observed that on avergae the validation accuracy saturates to around 65-69% if the hidden layer nodes are taken within the range 45-70.
- **However MLFFNN is a reasonable (6-10%) improvement over the GMM model considered in the previous assignment (i.e from 60 to 68)**. In MLFFNN no assumption regarding class conditional pdf is being made. It directly attacks the decision boundary, hence it may give better results

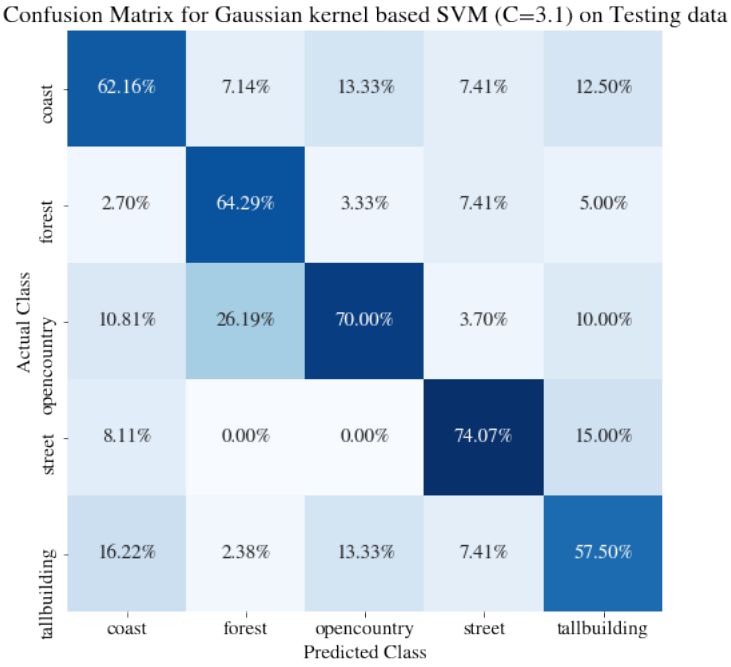
2.(a).2 Gaussian kernel based SVM using one-against-the-rest approach

C	Training(%)	Validation(%)
0.01	23.52	23.56
2.5	84.75	62.068
3.1	86.311	62.068
75	99.84	57.47
100	100	58.62

Best Model:- c= 3.1 is the best model having testing accuracy 64.772%



(a) Confusion matrix for training dataset



(b) Confusion matrix for test dataset

Figure 38: Confusion Matrices for Gaussian kernel based SVM using one-against-the-rest approach using the best model c=3.1

Observations:

- C=3.1 was chosen as the best model as there were no further improvements in validation accuracy and it gave the best training accuracy.
- The training accuracy increases with increase in C indicating overfitting on training Dataset with no consistent improvement in validation set.