CS671 Assignment-1

GROUP - 8

Akash Karnatak (B19147)
Laishram Ponghtangamba (B19011)
Abhishek Mishra (B19231)

Classification tasks:

Linear separable data:

Perceptron:

For class 0 and class 1

1 The graph of average errors which was obtained. during training and the number of epochs.
For class 0 and class 1

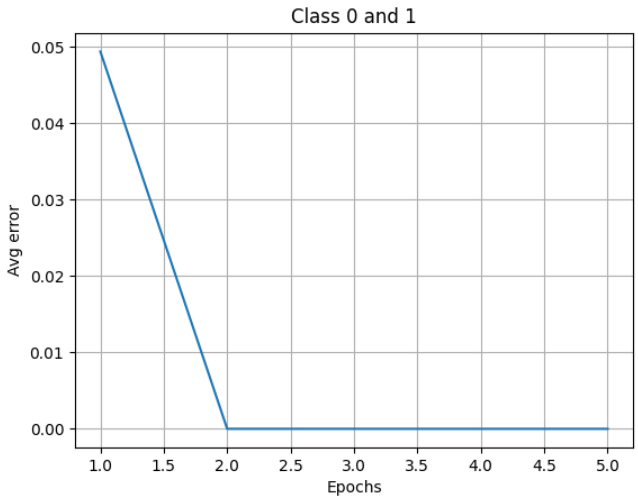

Fig. *Average error vs epochs for perceptron learning algorithm for class 0 and 1*

2.(a) Design region plot of class 0 and class 1 of training data.

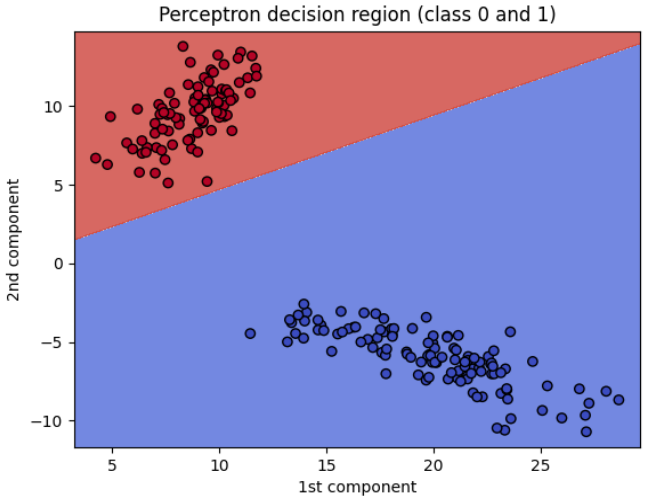

Fig. *Decision region plot for class 0 and 1*

3.Classification Accuracy:
The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class	
		0	**1**
Predicted	**0**	111	0
	1	0	89

For class 0 and class 2

1. The graph of average errors which was obtained. during training and the number of epochs.
For class 0 and class 2

2. a) Design region plot of class 0 and class 2 of training data.

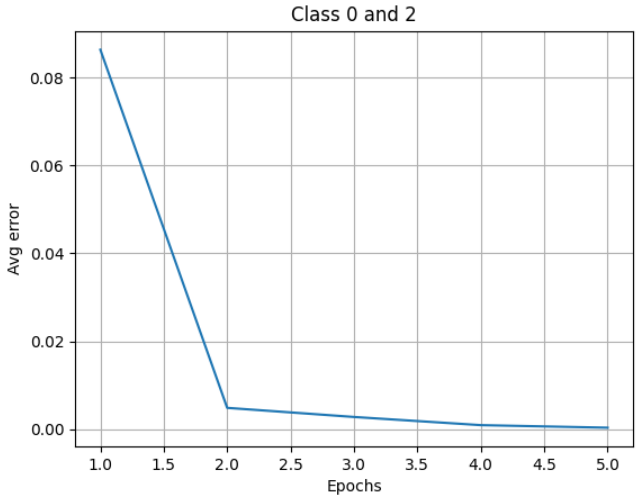

Fig. *Average error vs epochs for perceptron learning algorithm for class 0 and 2*

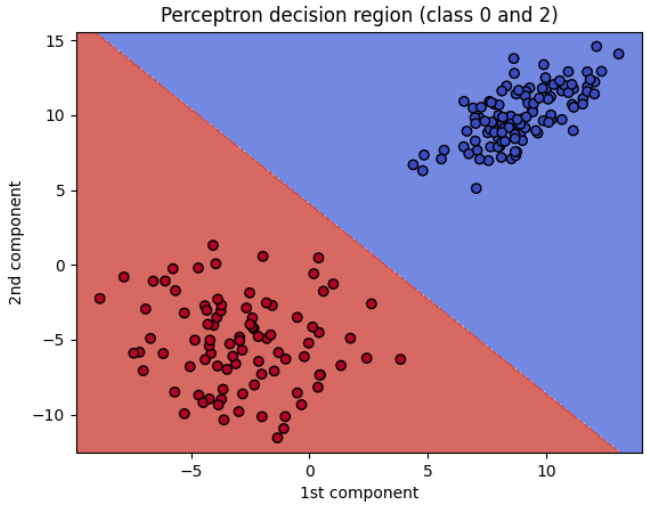

Fig. *Decision region plot for class 0 and 2*

3.Classification Accuracy:
The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class	
		0	**2**
Predicted class	**0**	111	0
	2	0	89

For class 1 and class 2

1 The graph of average errors which was obtained. during training and the number of epochs
For class 1 and class 2

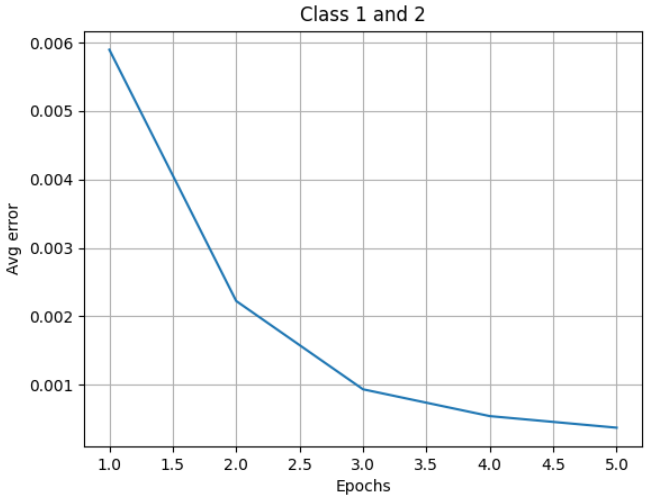

Fig. *Average error vs epochs for perceptron learning algorithm for class 1 and 2*

2. a) Design region plot of class 1 and class 2 of training data.

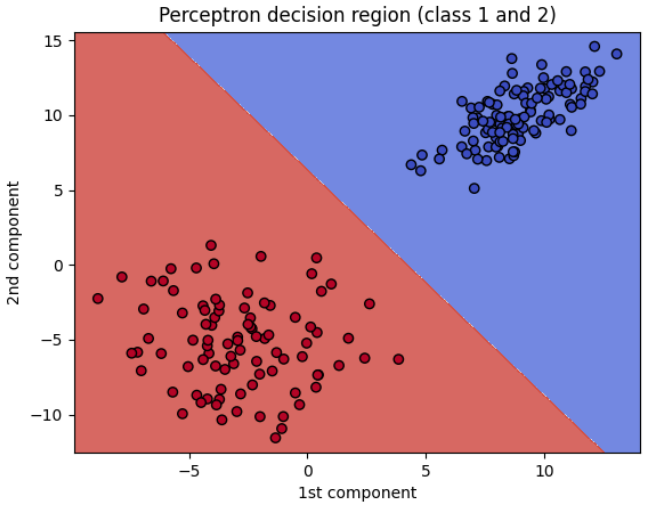

Fig. *Decision region plot for class 1 and 2*

3.Classification Accuracy:

The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class	
		1	**2**
Predicted class	**1**	110	0
	2	0	90

2(a). Decision region for all classes:

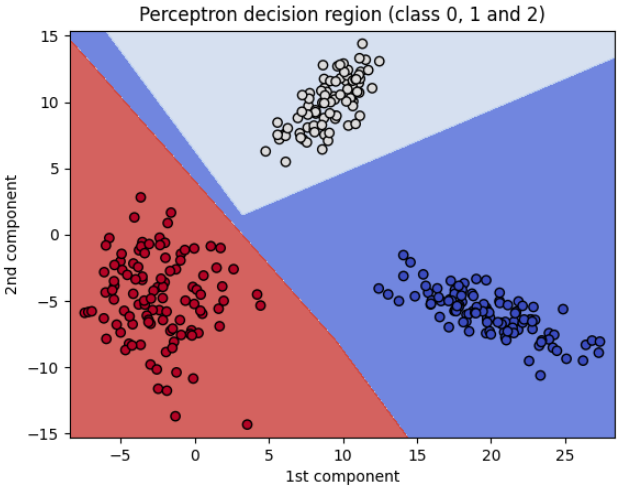

Fig. *Decision region for all classes*

FCNN:

Fully connected neural network with 1 hidden layer of 5 neurons:

1 The graph of average errors which was obtained. during training and the number of epochs.

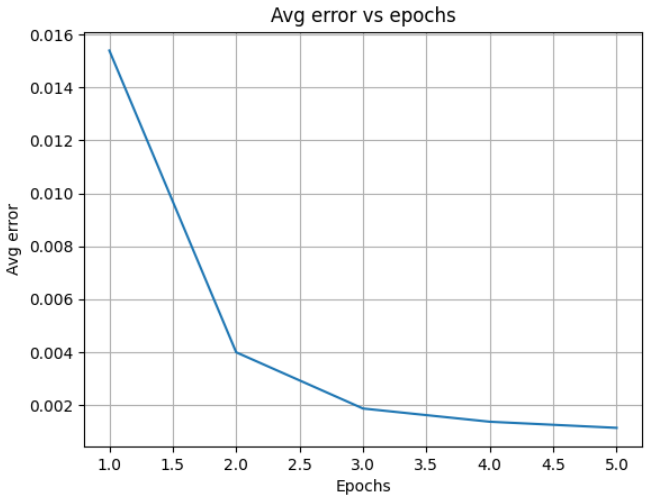

Fig. *Average error vs epochs for FCNN 1 hidden layer of 5 neurons*

2.(b) Design region plot given by superimposition of training data.

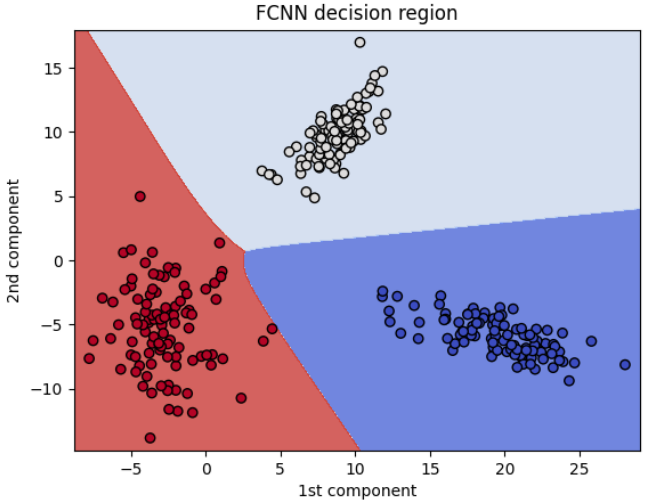

Fig. *Decision region plot for FCNN with 1 hidden layer of 5 neurons*

3. Classification accuracy:

The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	98	0	0
	1	0	102	0
	2	0	0	100

Fully connected neural network with 1 hidden layer of 10 neurons:

1 The graph of average errors which was obtained. during training and the number of epochs.

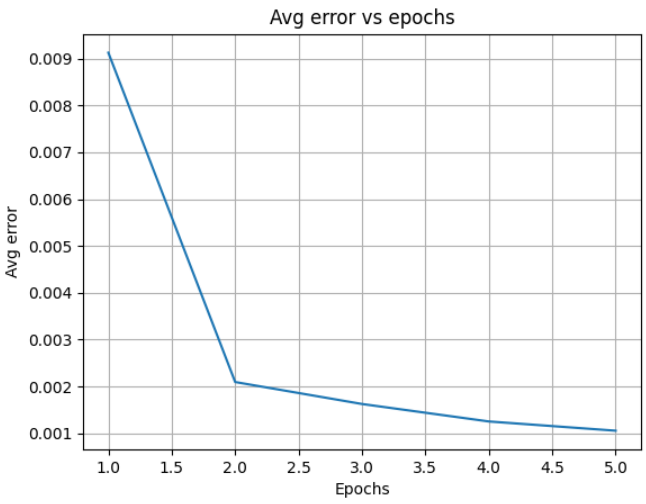

Fig. *Average error vs epochs for FCNN 1 hidden layer of 10 neurons*

2.(b) Design region plot given by superimposition of training data.

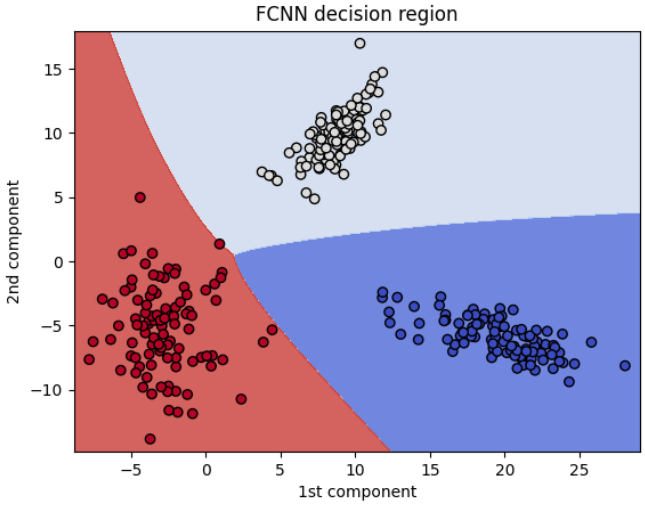

Fig. *Decision region plot for FCNN with 1 hidden layer of 10 neurons*

3. Classification accuracy

The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	98	0	0
	1	0	102	0
	2	0	0	100

Fully connected neural network with 1 hidden layer of 15 neurons:

1 The graph of average errors which was obtained. during training and the number of epochs.

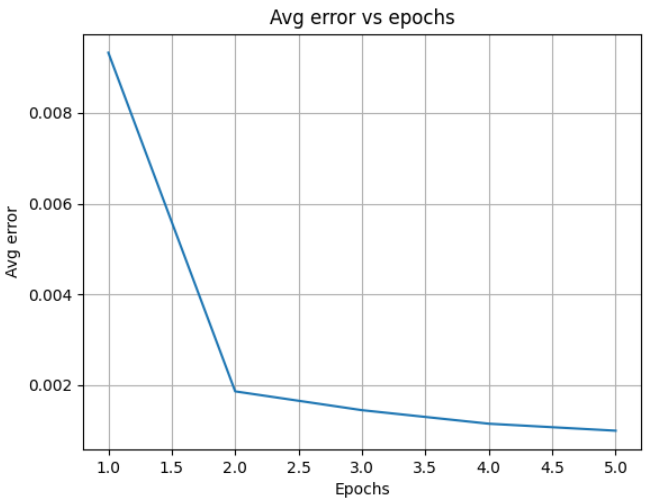

Fig. *Average error vs epochs for FCNN 1 hidden layer of 15 neurons*

2.(b) Design region plot given by superimposition of training data.

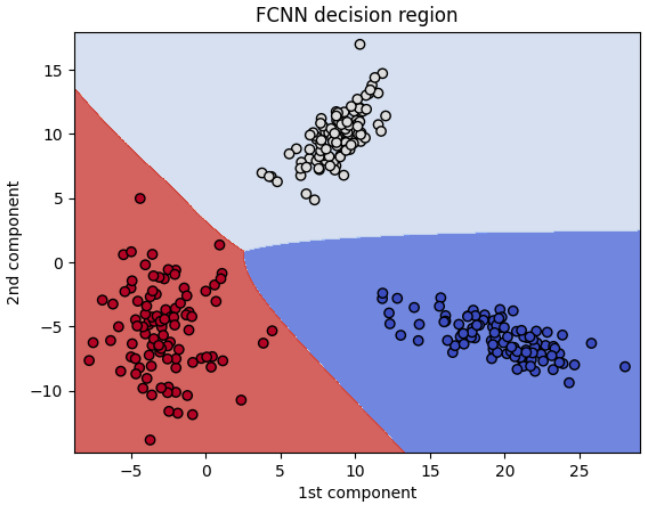

Fig. *Decision region plot for FCNN with 1 hidden layer of 15 neurons*

3. Classification accuracy

The above model achieved accuracy of 100% on both validation data and test data.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	98	0	0
	1	0	102	0
	2	0	0	100

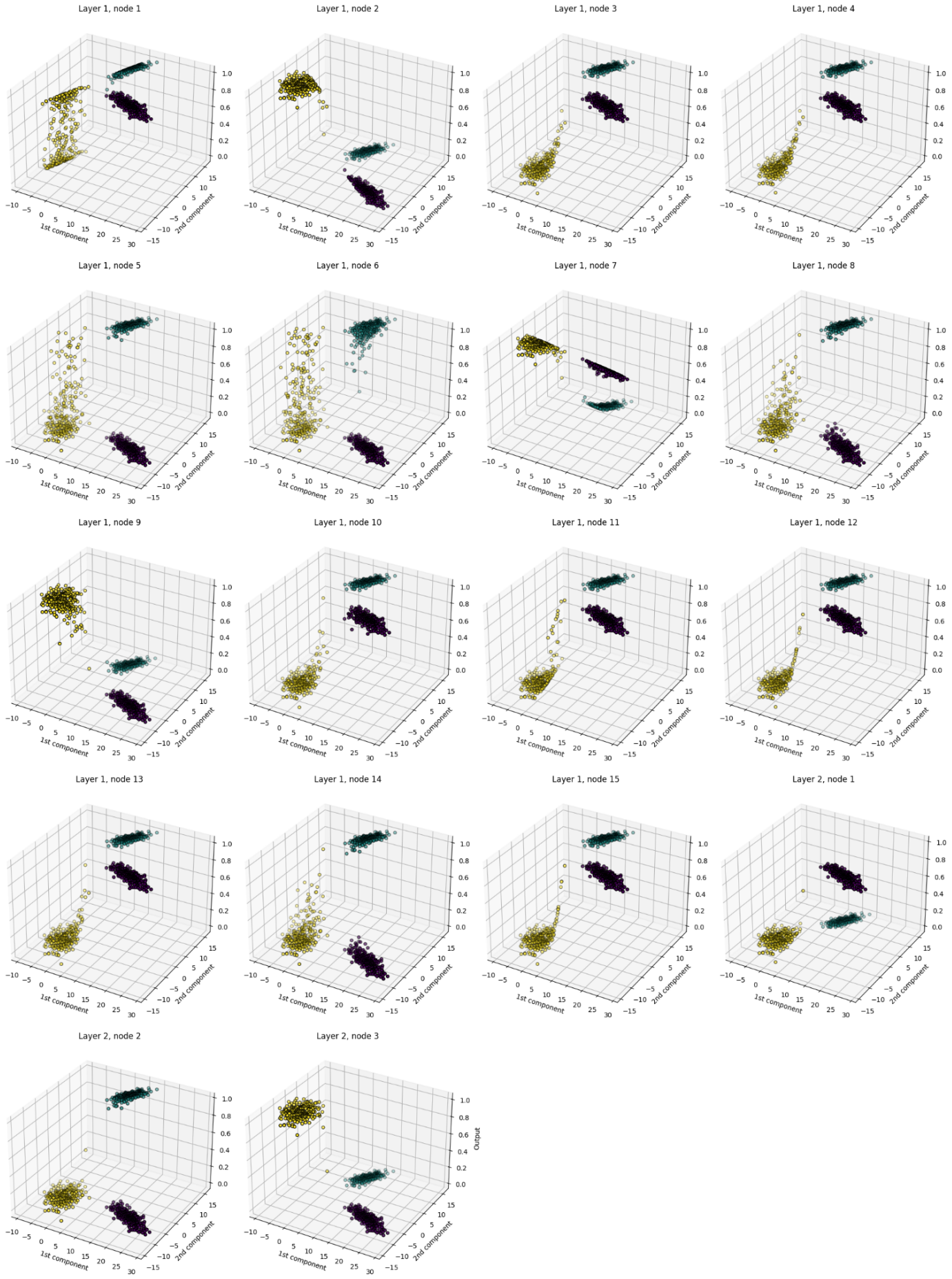

Fig. *Output of each node for training data for FCNN with 1 hidden layer of 15 neurons*

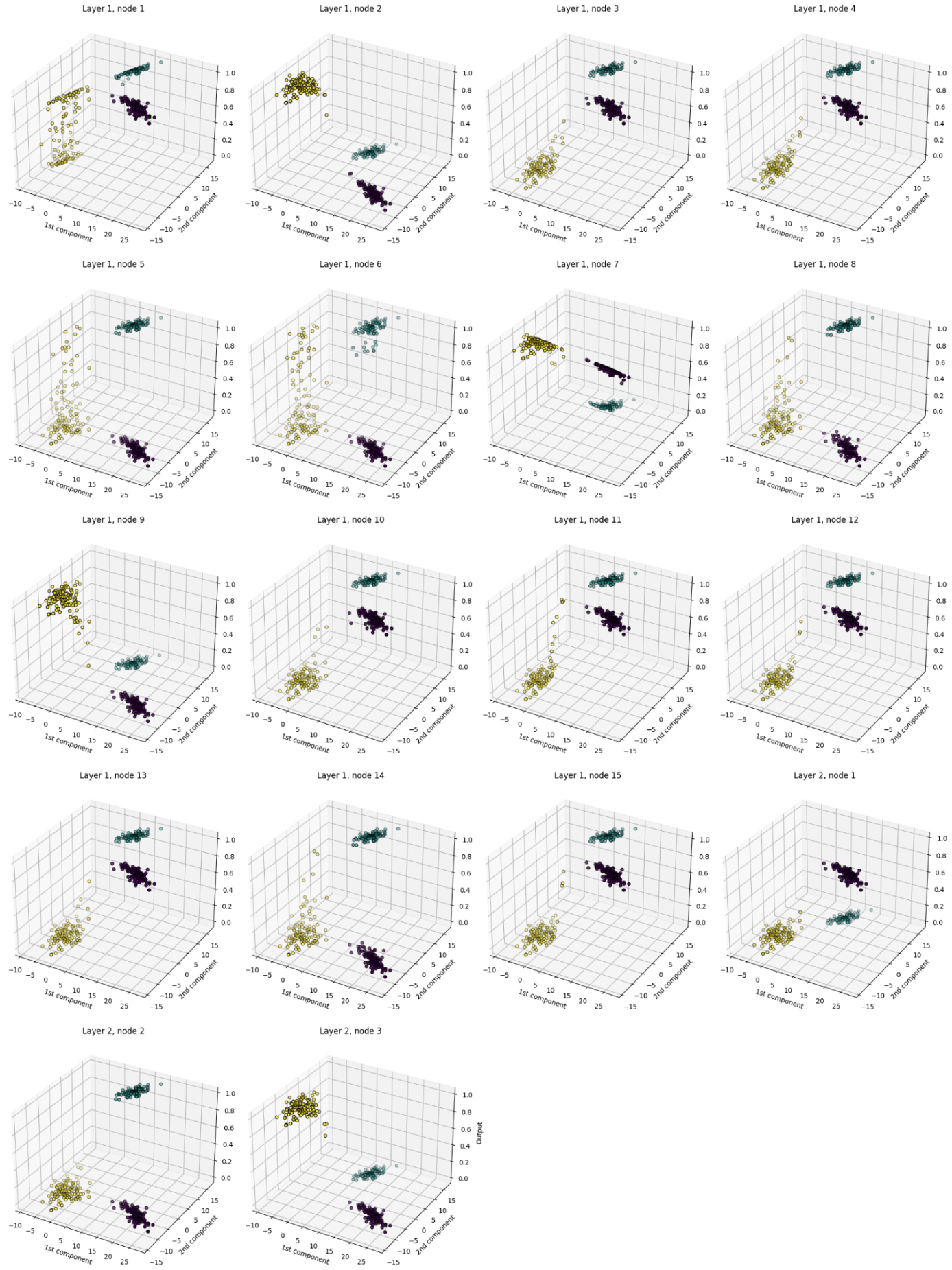

Fig. *Output of each node for validation data for FCNN with 1 hidden layer of 15 neurons*

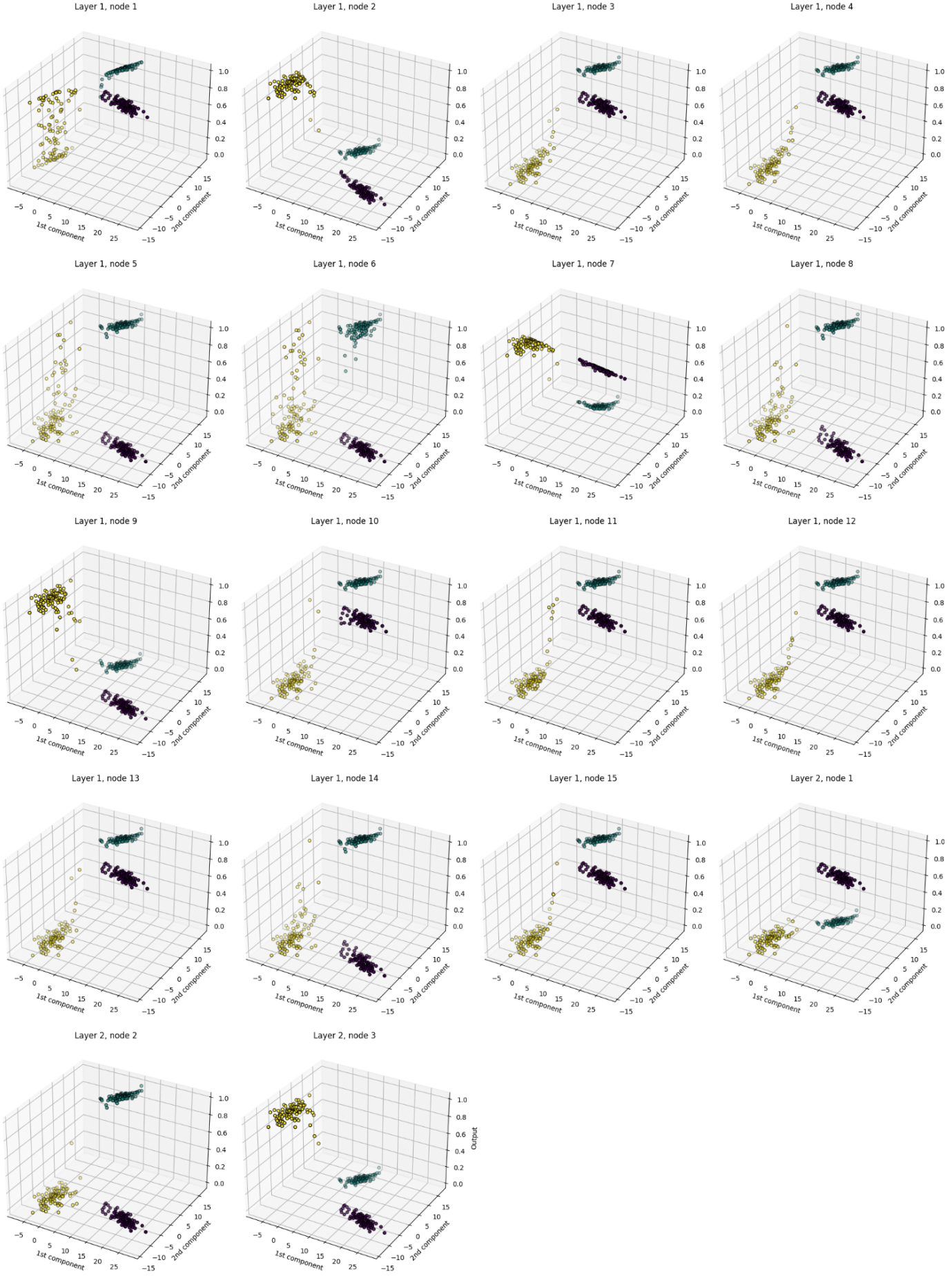

Fig. *Output of each node for test data for FCNN with 1 hidden layer of 15 neurons*

4. Comparison of performance of different models for each dataset:

Both perceptron and FCNN were able to perfectly classify the linearly separable data.

Non linearly separable data:

Perceptron:

For class 0 and class 1:

1 The graph of average errors which was obtained. during training and the number of epochs.
For class 0 and class 1

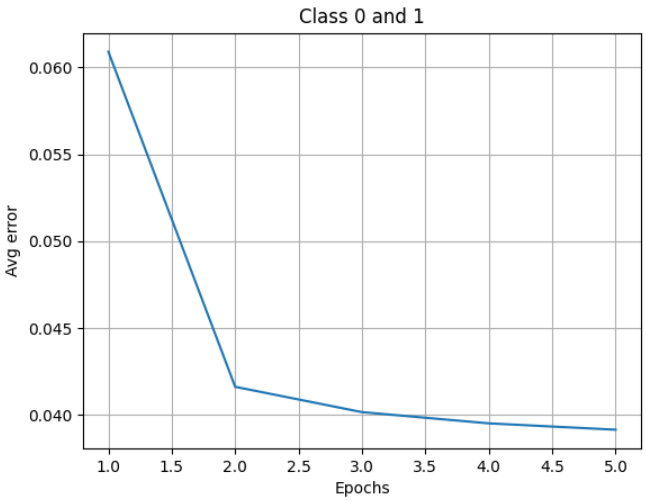

Fig. *Average error vs epochs for perceptron learning algorithm for class 0 and 1*

2.(a) Design region plot of class 0 and class 1 of training data.

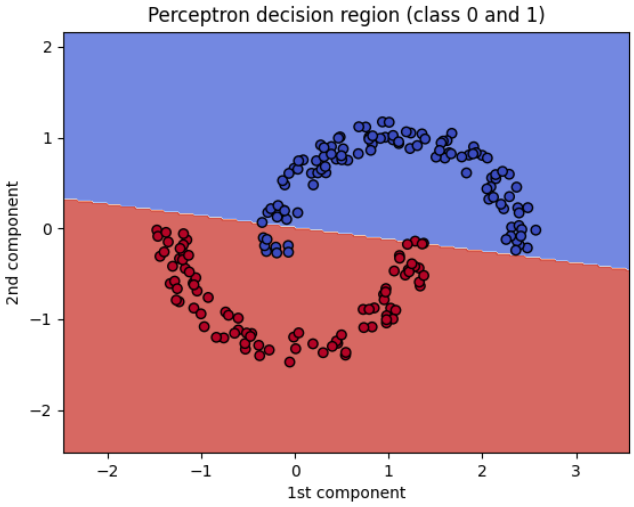

Fig. *Decision region plot for class 0 and 1*

3. Classification accuracy
The above model achieved validation accuracy of 94.0% and test accuracy of 95.0%.

Confusion matrix:

		Actual class	
		0	**1**
Predicted	**0**	104	8
	1	2	86

For Class 0 and class 2

1 The graph of average errors which was obtained. during training and the number of epochs.
For class 0 and class 2

2.(a) Design region plot of class 0 and class 2 of training data.

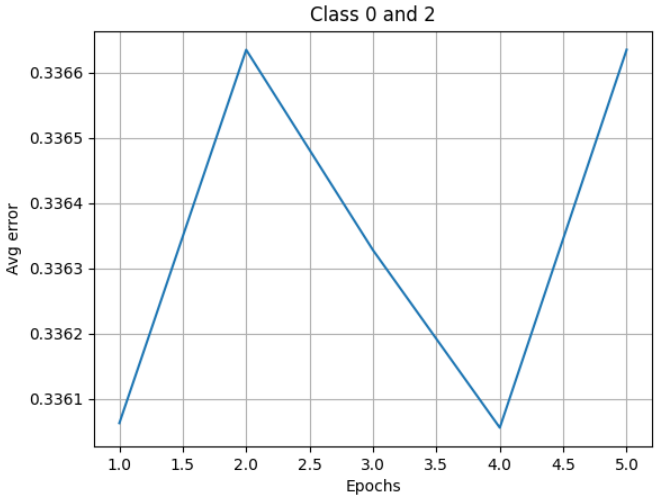

Fig. *Average error vs epochs for perceptron learning algorithm for class 0 and 2*

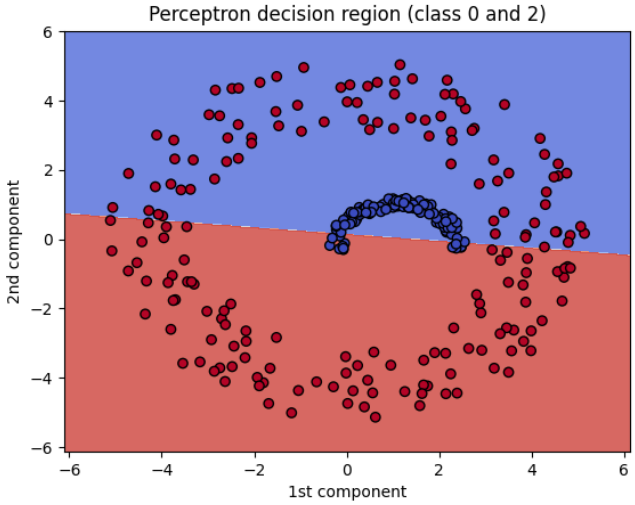

Fig. *Decision region plot for class 0 and 2*

3. Classification accuracy:

The above model achieved validation accuracy of 62.33% and test accuracy of 63.4%.

Confusion matrix

		Actual class	
		0	**2**
Predicted class	**0**	98	16
	2	87	99

For Class 1 and class 2

1 The graph of average errors which was obtained. during training and the number of epochs. for class 1 and class 2

2.(a) Design region plot of class 1 and class 2 of training data.

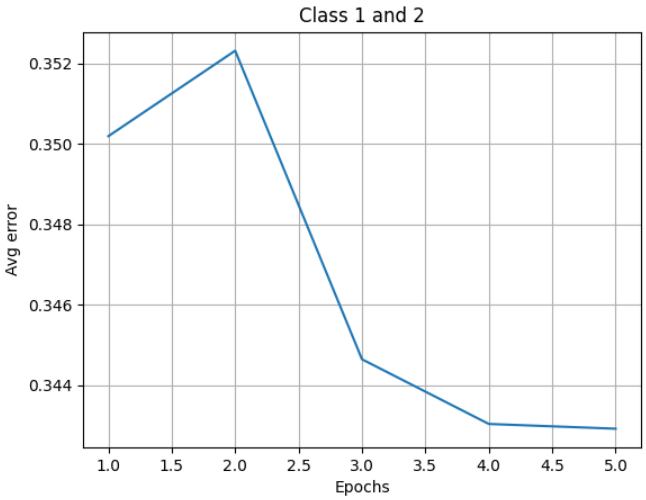

Fig. *Average error vs epochs for perceptron learning algorithm for class 1 and 2*

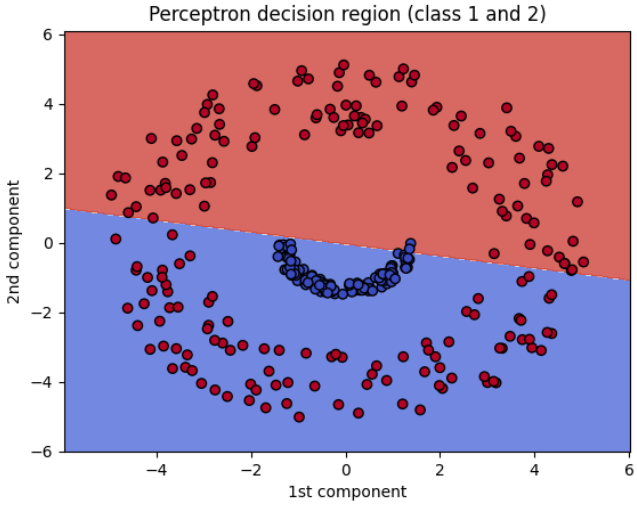

Fig. *Decision region plot for class 1 and 2*

3. Classification accuracy:

The above model achieved validation accuracy of 64.33% and test accuracy of 65.67%.

Confusion matrix:

		Actual class	
		1	**2**
Predicted class	**1**	103	2
	2	93	102

2.(a) Decision region for all classes

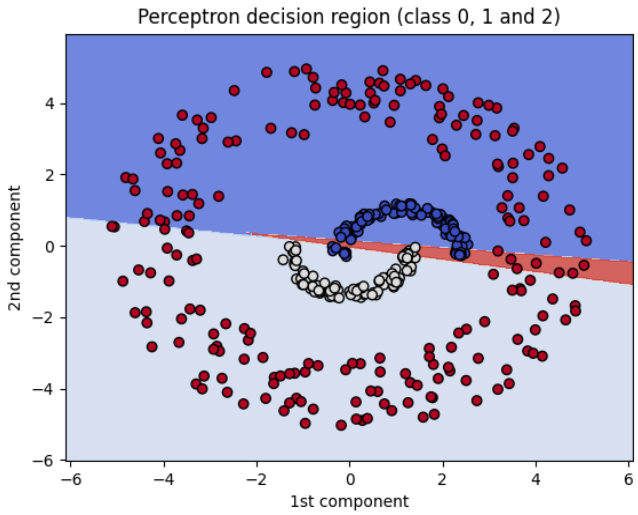

Fig. *Decision region for all classes*

FCNN:

Fully connected neural network with 1 hidden layer of 5 neurons:

1 The graph of average errors which was obtained. during training and the number of epochs.

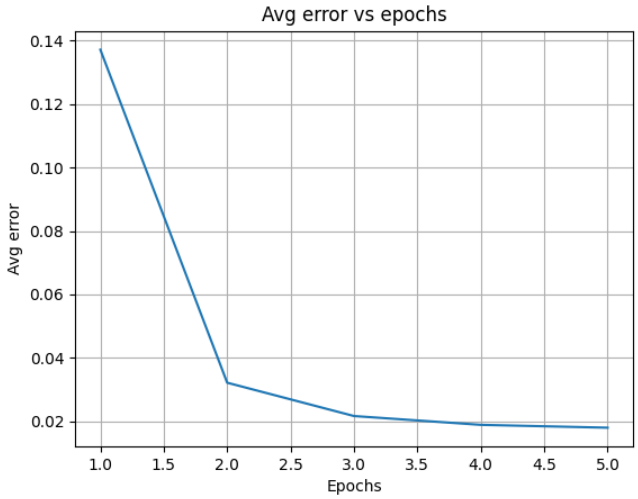

Fig. *Average error vs epochs for FCNN 1 hidden layer of 5 neurons*

2.(b) Design region plot given by superimposition of training data.

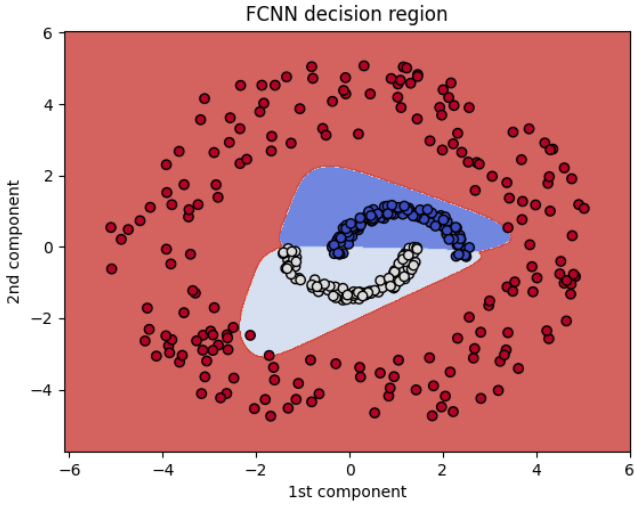

Fig. *Decision region plot for FCNN with 1 hidden layer of 5 neurons*

3. Classification accuracy

The above model achieved validation accuracy of 96.25% and test accuracy of 95.75%.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	101	12	0
	1	3	95	0
	2	0	2	187

Fully connected neural network with 1 hidden layer of 10 neurons:

1 The graph of average errors which was obtained during training and the number of epochs.

2.(b) Design region plot given by superimposition of training data.

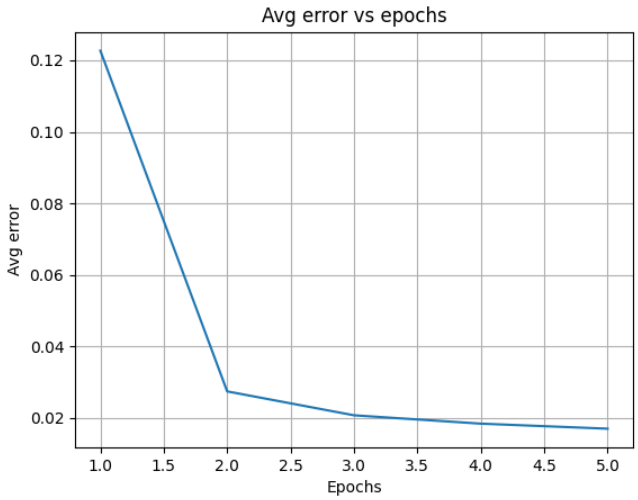

Fig. *Average error vs epochs for FCNN 1 hidden layer of 10 neurons*

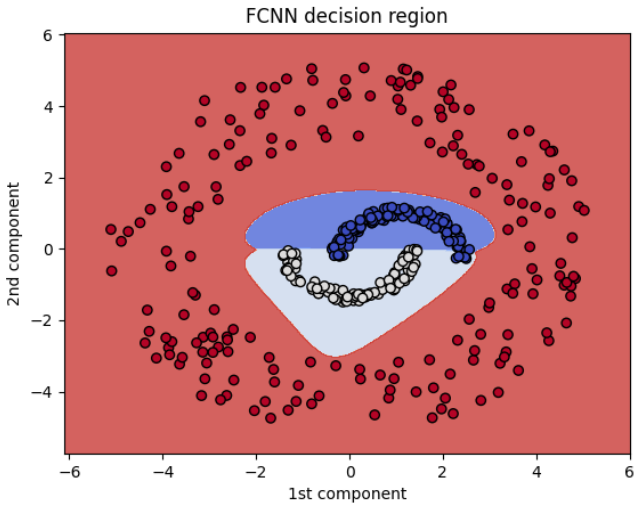

Fig. *Decision region plot for FCNN with 1 hidden layer of 10 neurons*

3. Classification accuracy

The above model achieved validation accuracy of 97.25% and test accuracy of 96.75%.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	100	13	0
	1	2	96	0
	2	0	0	189

Fully connected neural network with 2 hidden layers of 15 neurons each:

1 The graph of average errors which was obtained. during training and the number of epochs.

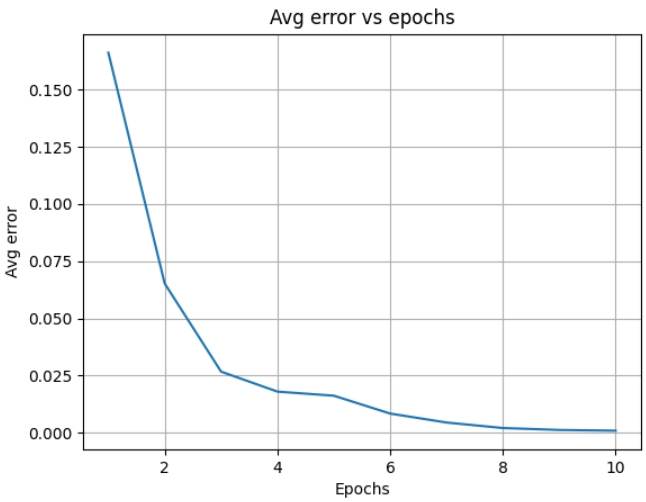

Fig. *Average error vs epochs for FCNN 2 hidden layers of 15 neurons each*

2.(b) Design region plot given by superimposition of training data.

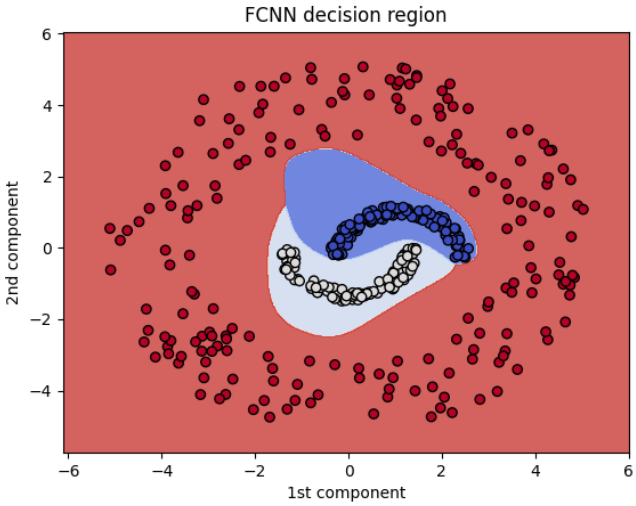

Fig. *Decision region plot for FCNN with 2 hidden layers of 15 neurons each*

3.Classification accuracy

The above model achieved validation accuracy of 100% and test accuracy of 100%.

Confusion matrix:

		Actual class		
		0	**1**	**2**
Predicted class	**0**	113	0	0
	1	0	98	0
	2	0	0	189

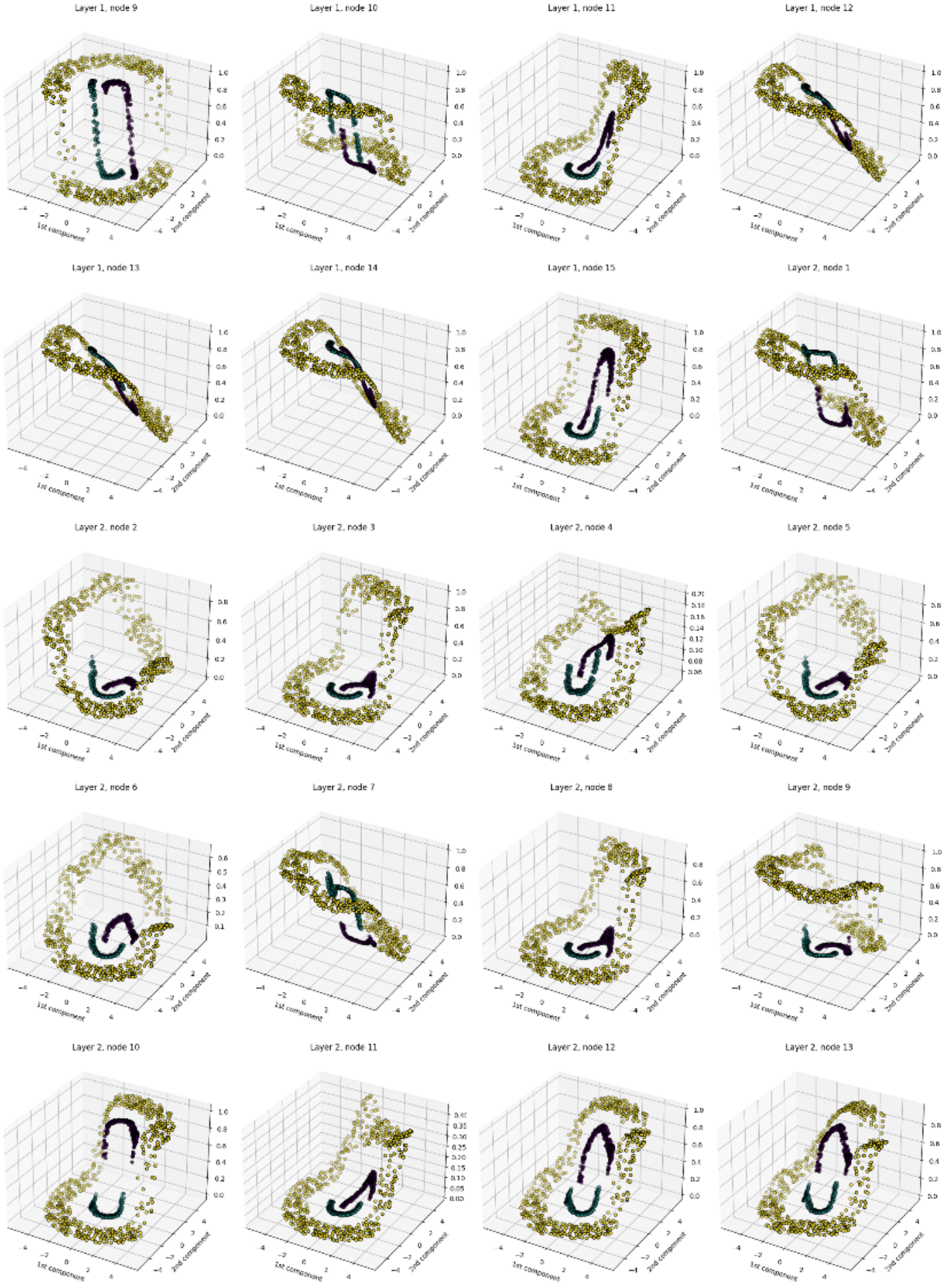

Fig. *Output of each node for training data for FCNN with 2 hidden layer of 15 neurons each*

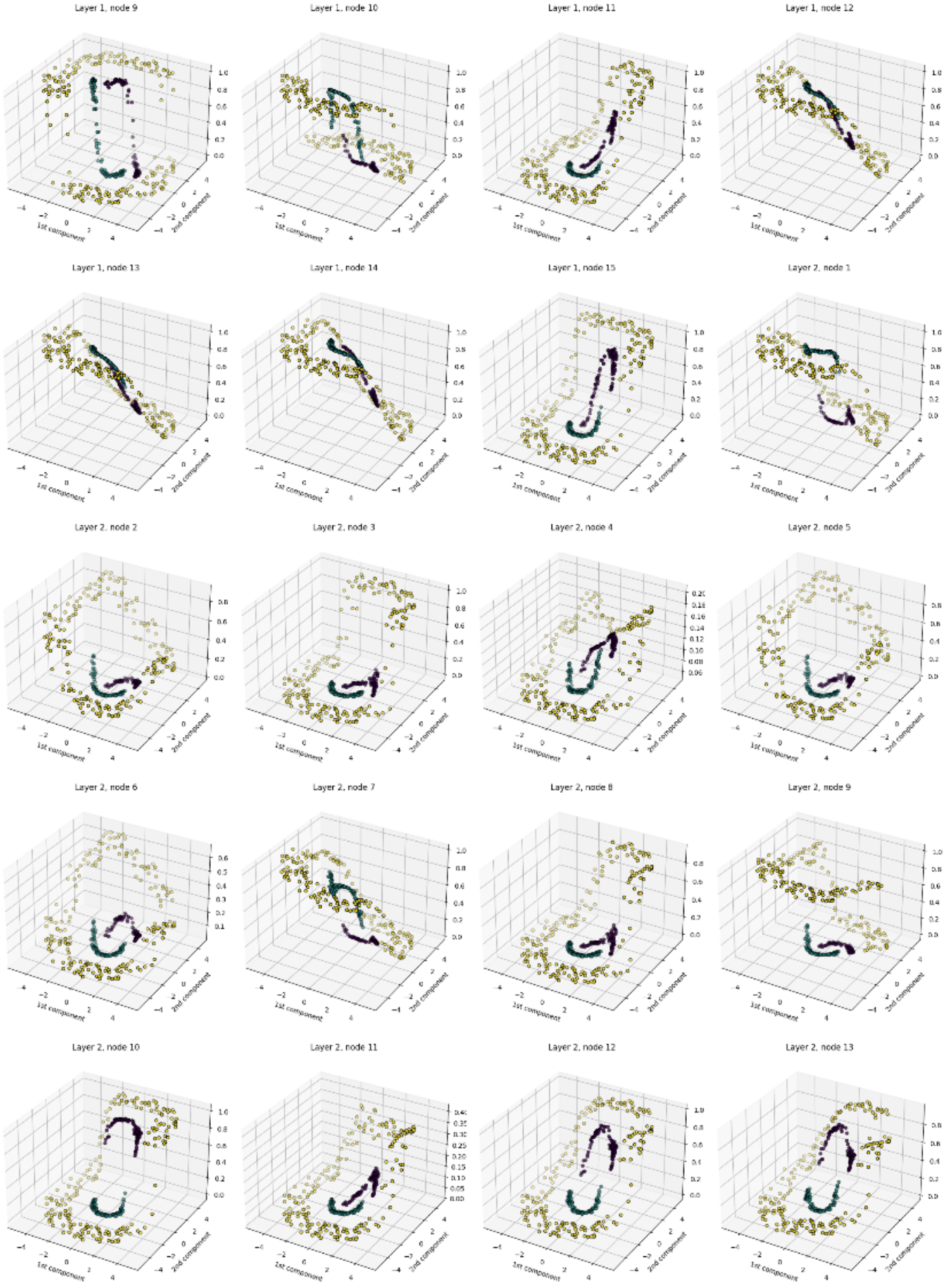

Fig. *Output of each node for validation data for FCNN with 2 hidden layer of 15 neurons each*

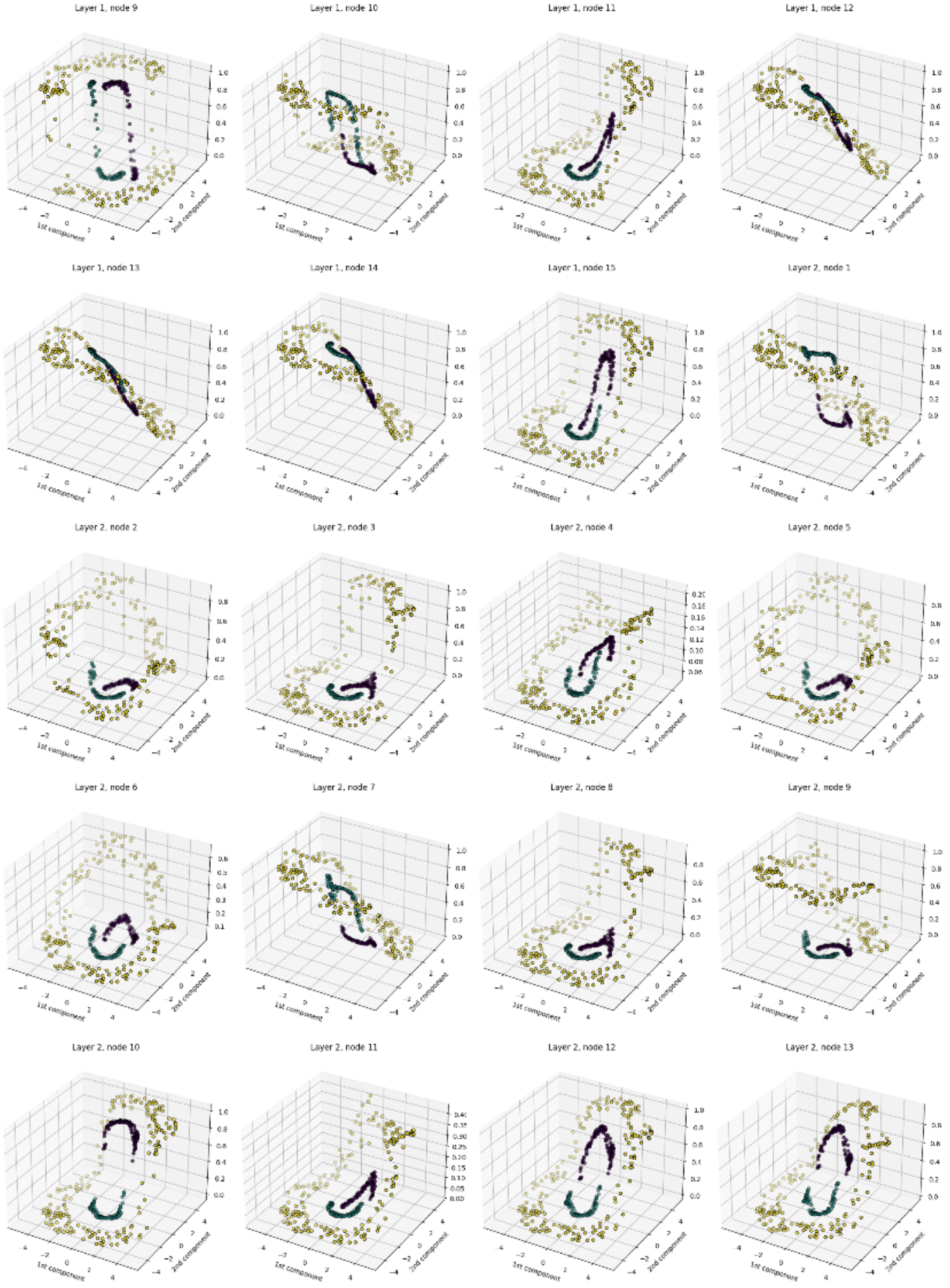

Fig. *Output of each node for test data for FCNN with 2 hidden layer of 15 neurons each*

4. Comparison of performance of different models for each dataset:

Perceptron wasn't able to correctly classify the non linear data which resulted in very poor accuracy. FCNN on the other hand classified the data perfectly.

Regression tasks:

Expression For Change Of Weights:

The network is [P,I, J , K]
dwjk= rate of change of final cost when weight wjk is changed (weight associated with jth neuron of previous layer and kth neuron of present layer); zk = logit value for kth node/Input to kth node; ak= predicted value by kth node; tk = target value for kth node; aj = output of node j of previous layer; g'(zk) : differentiation of activation function for kth neuron in present layer.
$\delta k = (ak - tk) * (g'(zk))$
For Output Layer : $dwjk = (\delta k) * (aj)$
For Jth hidden Layer : $dwik = (\delta j)*(ai)$, where $\delta j = (\Sigma \delta k * wjk) * g'(zj)$ (summation is over k).
For Ith hidden Layer : $dwpi = (\delta i)*(ap)$, where $\delta i = (\Sigma(\Sigma \delta k * wjk) * wij) * g'(zi)$

Univariate Input Data:

The given univariate data for regression are plotted below. They are non-linear sinusoidal in nature.

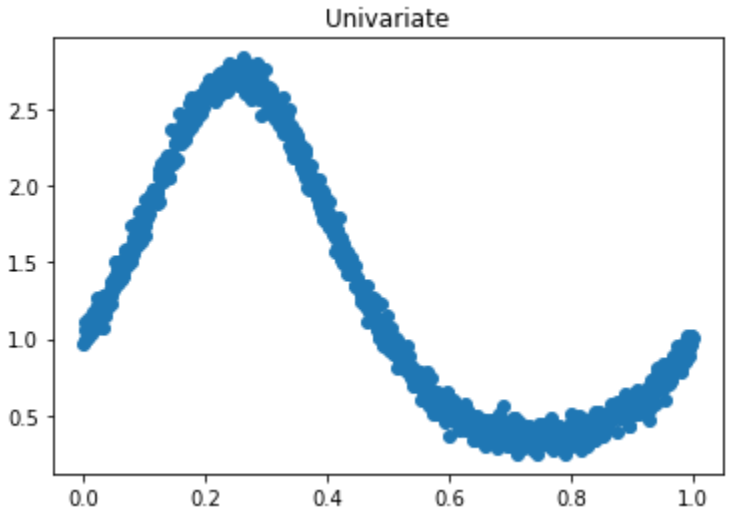

Fig. *The univariate data*

Perceptron:

A perceptron learning model is implemented to perform the regression of the given univariate data. In this model only one input and one output neuron are used. So there is only one weight and bias.

The results obtained from the experiment are given below:

1 The graph of average errors which was obtained. during training and the number of epochs.

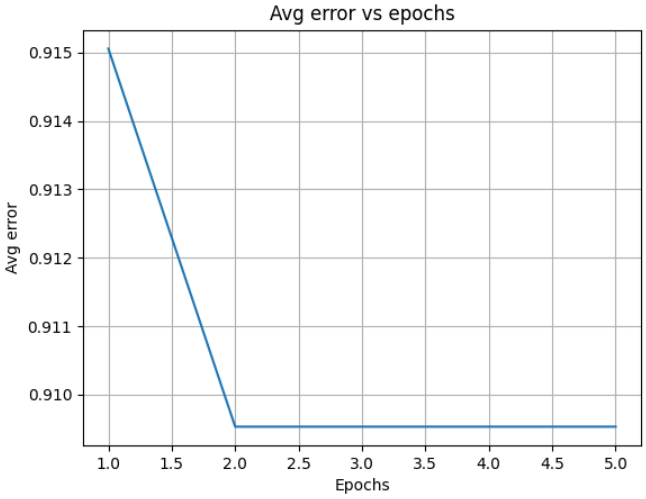

Fig. *Average error vs epochs for perceptron learning algorithm for univariate data.*

2. The bar graphs of Mean Square Errors Of training , validation and testing.

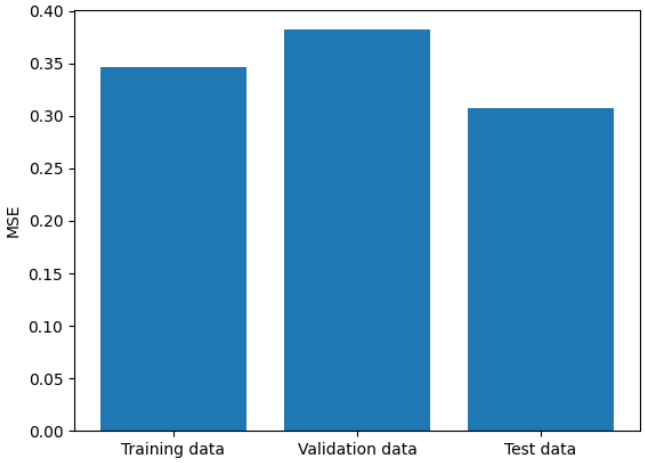

Fig. *Mean Square Errors of training,validation and test data for a perceptron learning algorithm for univariate data.*

Inference:

In Avg error vs epochs graph , the value of error gets its minimum value at 2 epochs and remains constant afterwards. So, the first 2 epochs are only useful. In the bar graph , validation data gives the highest MSE.

3. The target output and predicted output given by the model for various datasets - training, validation and testing data. The blue data points represent the target output and orange data points represent the predicted output given by the perceptron model.

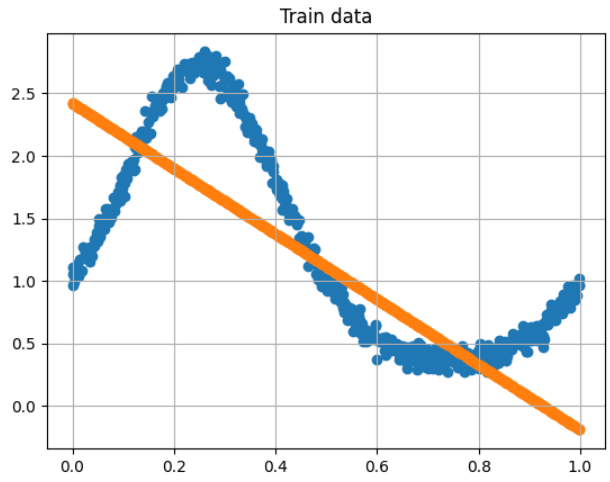

Fig. Training data : *Green points represent targeted output and Orange points represent model output for perceptron learning algorithm (univariate).*

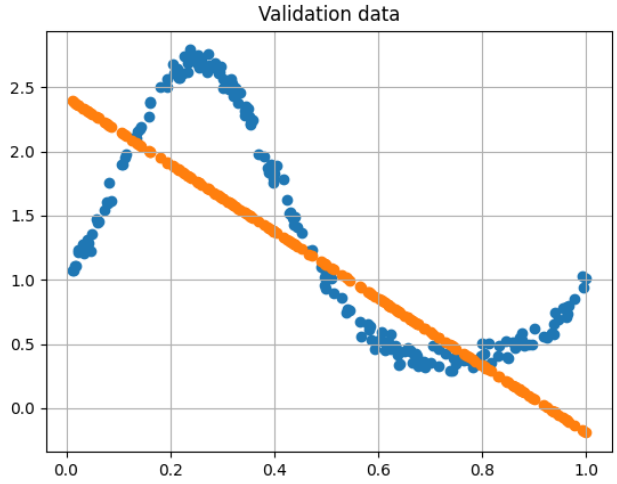

Fig. *Validation data : Green points represent targeted output and Orange points represent model output for perceptron learning algorithm (univariate) .*

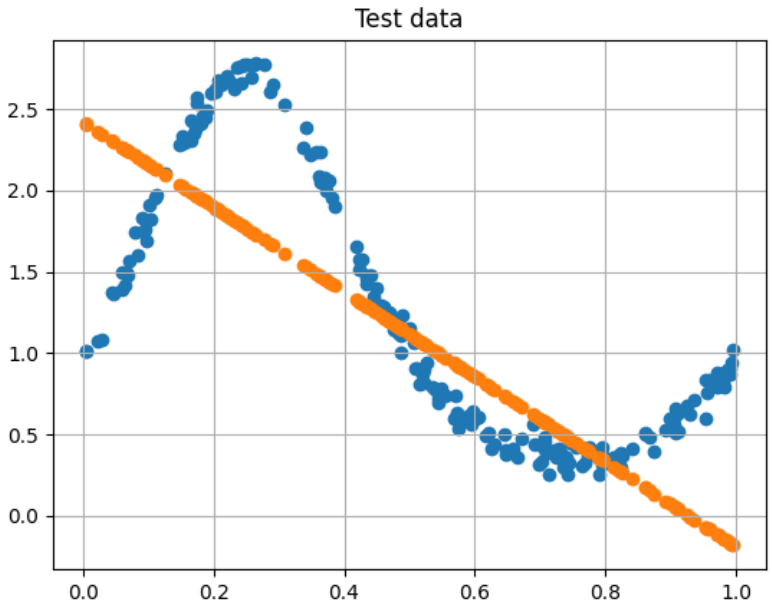

Fig. Testing data : *Green points represent targeted output and Orange points represent model output for perceptron learning algorithm (univariate).*

Inference:

The output predicted by the model can't fit the target test output properly. It is because since the given data is non-linear , a perceptron can't do the regression of the data properly.

4. The graph of predicted output on y-axis and target output on x-axis.

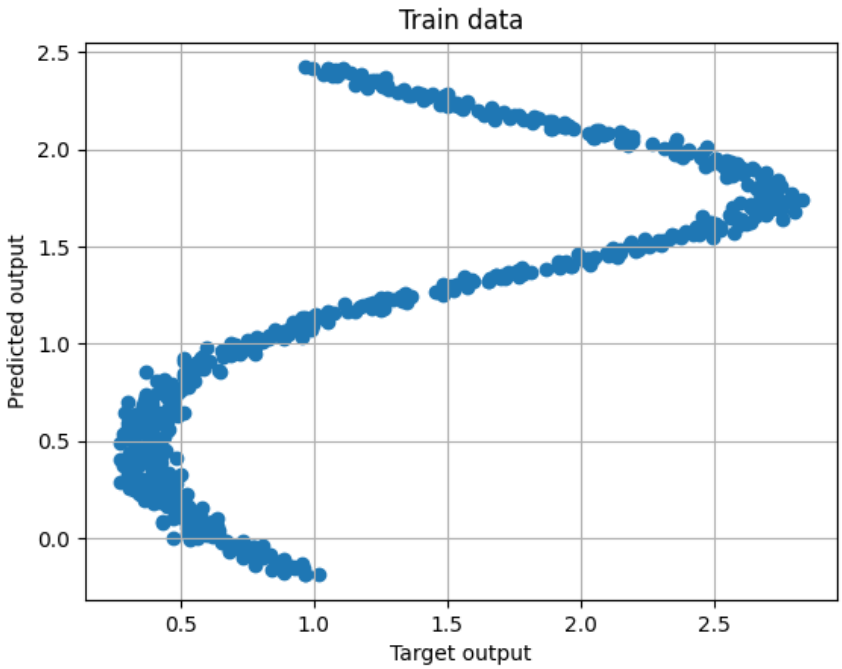

Fig. *Training data : Target output (x-axis) vs model output (y-axis). (univariate data , model-perceptron)*

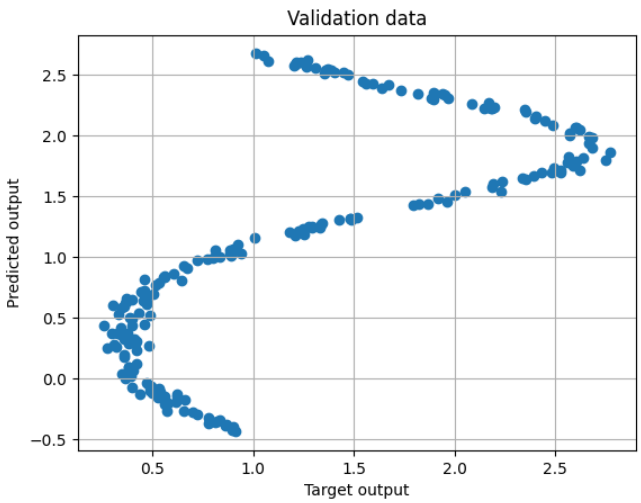

Fig. *Validation data : Target output (x-axis) vs model output (y-axis). (univariate data , model-perceptron)*

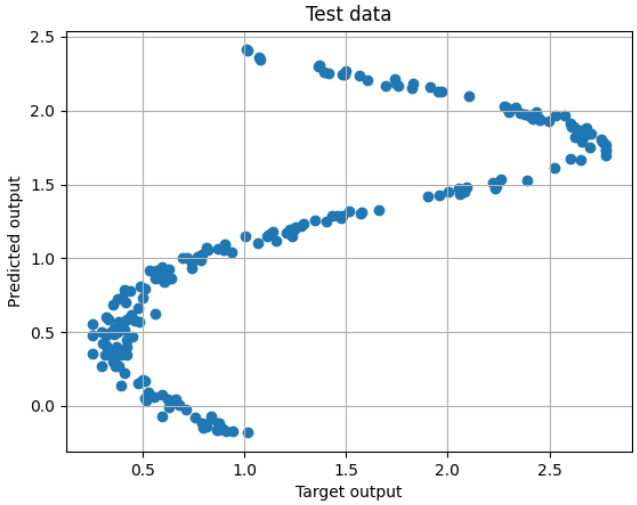

Fig. Test data : Target output (x-axis) vs model output (y-axis). (univariate data , model-perceptron)

Inference:

Target outputs are nonlinear and the predicted outputs are linear. So , a sinusoidal graph is obtained when a graph against each other is plotted.

Fully Connected Neural Networks:

A fully connected neural network (FCNN) is implemented to make a regression model for the given univariate data. The FCNN has 2 input nodes, one hidden layer with 6 nodes and 1 output node.

1. The graph of average errors which was obtained. during training and the number of epochs.

2. The bar graphs of Mean Square Errors Of training , validation and testing.

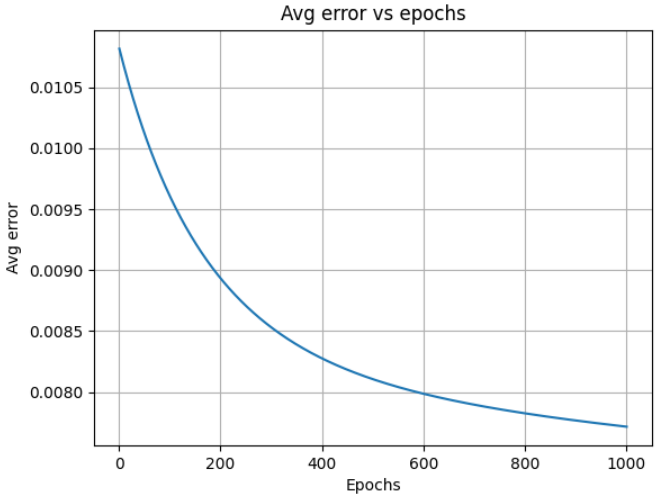

Fig. *Average error vs epochs for FCNN learning algorithm for univariate data.*

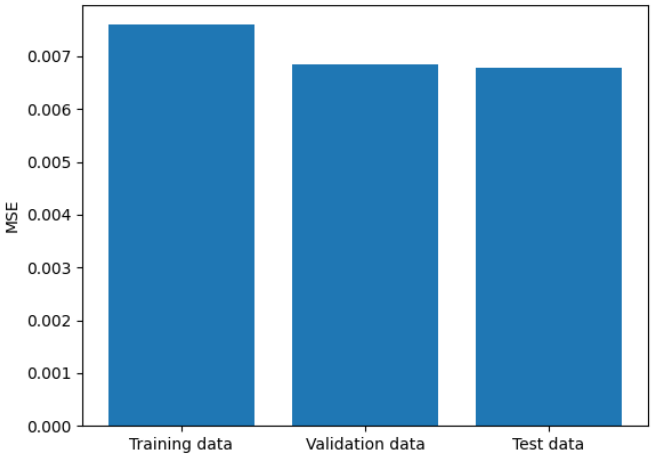

Fig. *Mean Square Errors of training,validation and test data for a FCNN learning algorithm for univariate data.*

Inference:

In Avg error vs epochs graph , the value of error decreases exponentially with the increasing number of epochs . So, more epochs will result in less error but if we do the training repeatedly , overfitting may result.

3. The target output and predicted output given by the model for various datasets - training, validation and testing data. The blue data points represent the target output and
orange data points represent the predicted output given by the perceptron model.

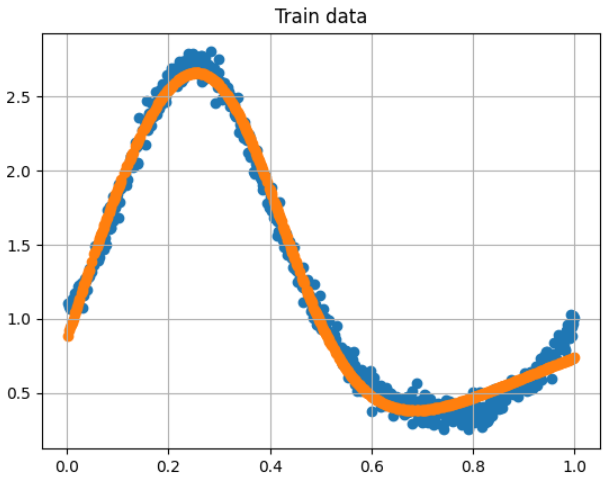

Fig. Training data : *Green points represent targeted output and Orange points represent model output for FCNN learning algorithm (univariate).*

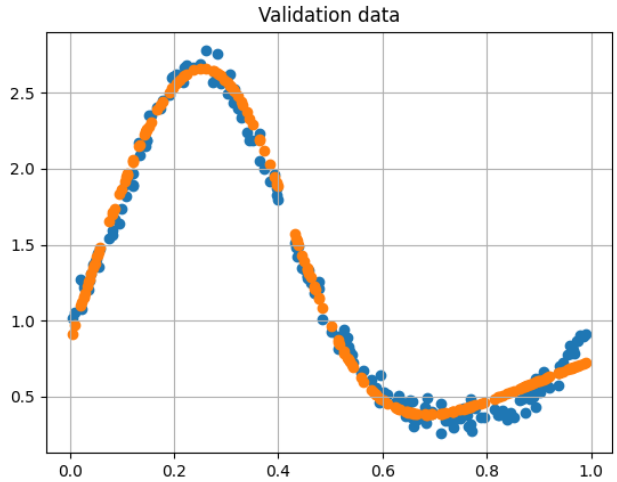

Fig. *Validation data : Green points represent targeted output and Orange points represent model output for FCNN learning algorithm (univariate) .*

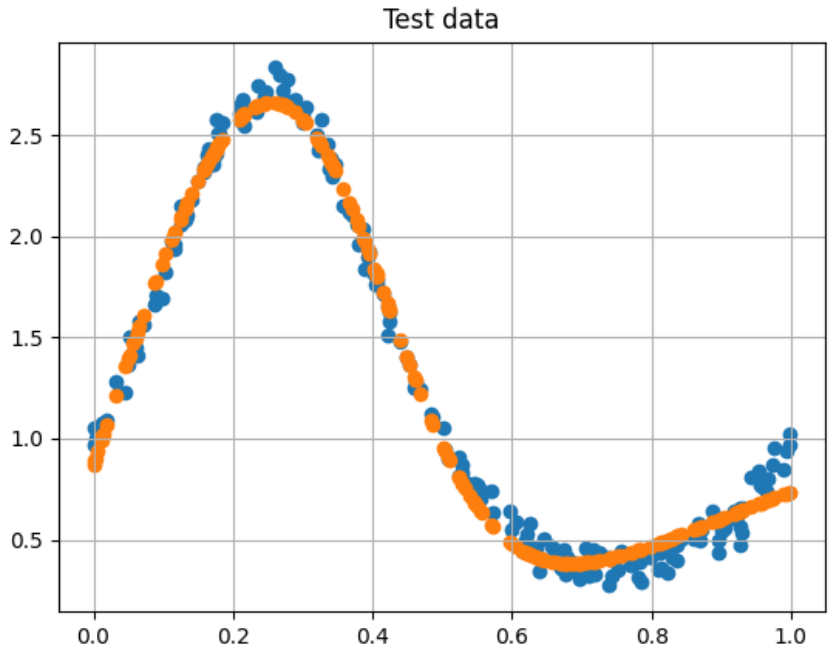

Fig. Testing data : *Green points represent targeted output and Orange points represent model output for FCNN learning algorithm (univariate).*

Inference:

The output predicted by the FCNN model can fit the target test output properly. It is because since the given data is non-linear , a FCNN with non-linear and linear activation can do the regression of the data properly.

4. The graph of predicted output on y-axis and target output on x-axis.

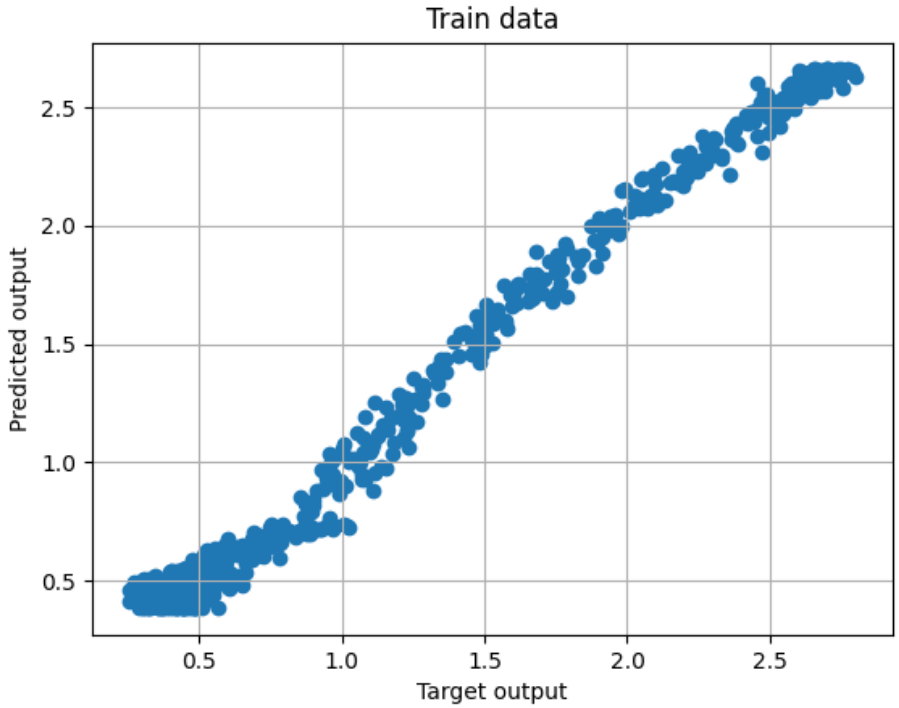

Fig. *Training data : Target output (x-axis) vs model output (y-axis). (univariate data , model-FCNN)*

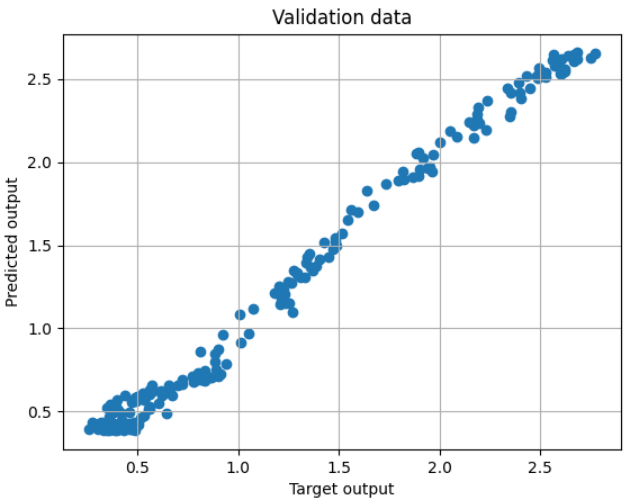

Fig. *Validation data : Target output (x-axis) vs model output (y-axis). (univariate data , model-FCNN)*

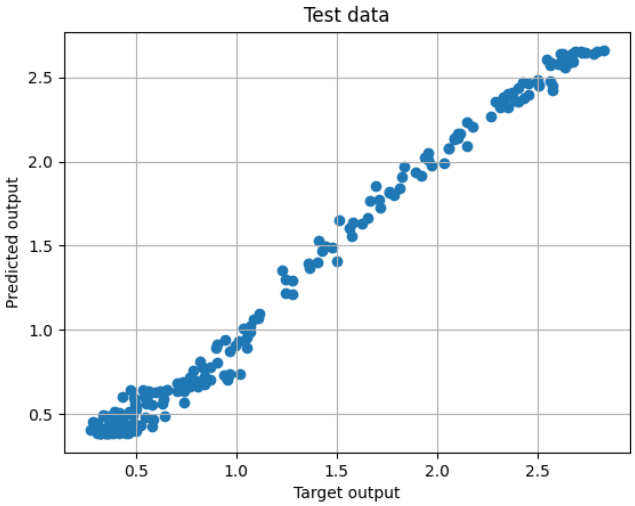

Fig. Test data : Target output (x-axis) vs model output (y-axis). (univariate data , model-FCNN)

Inference:

Target outputs are nonlinear and the predicted outputs are nonlinear and the error of the prediction is very less. So , an almost linear graph is obtained when a graph against each other is plotted.

5. Plots of outputs for each of the hidden nodes and output nodes in FCNN.

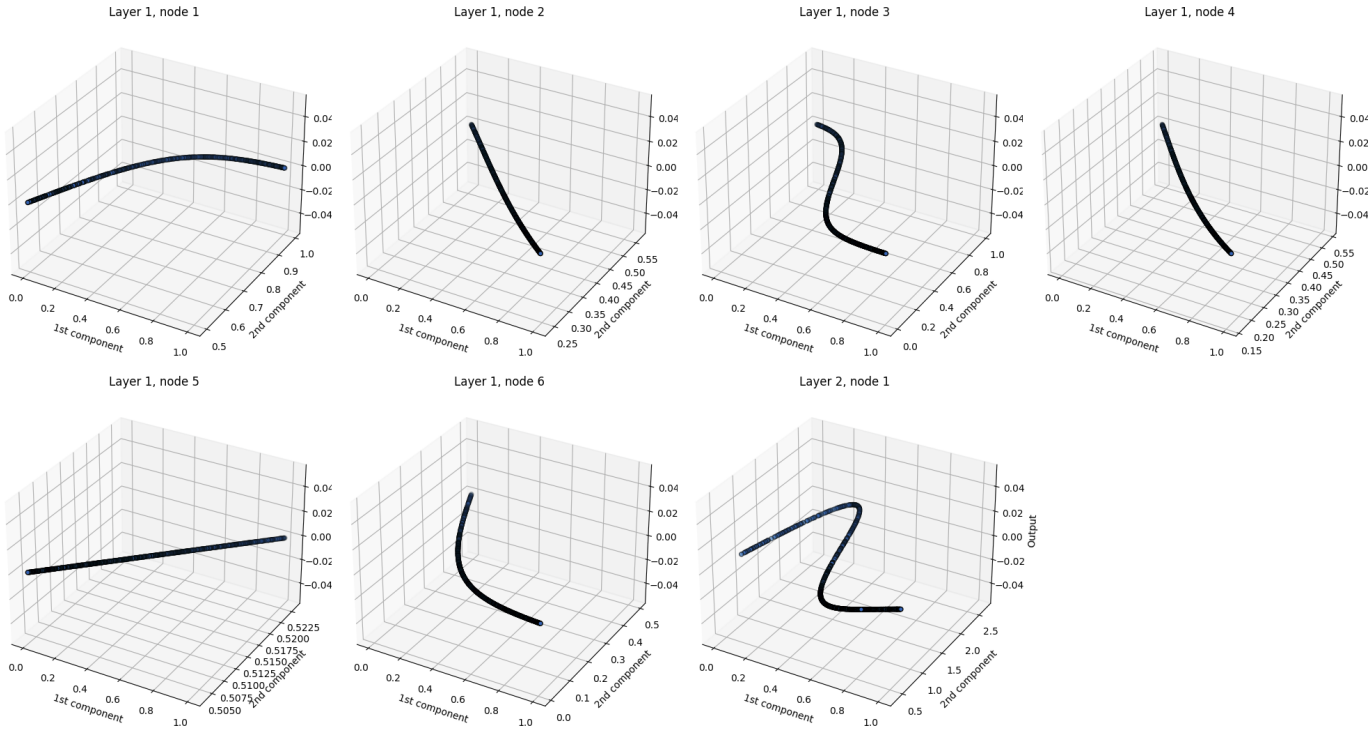

Fig. *Training data : Output on z-axis and inputs in x-axis and y-axis of all hidden layers and one output layer.*

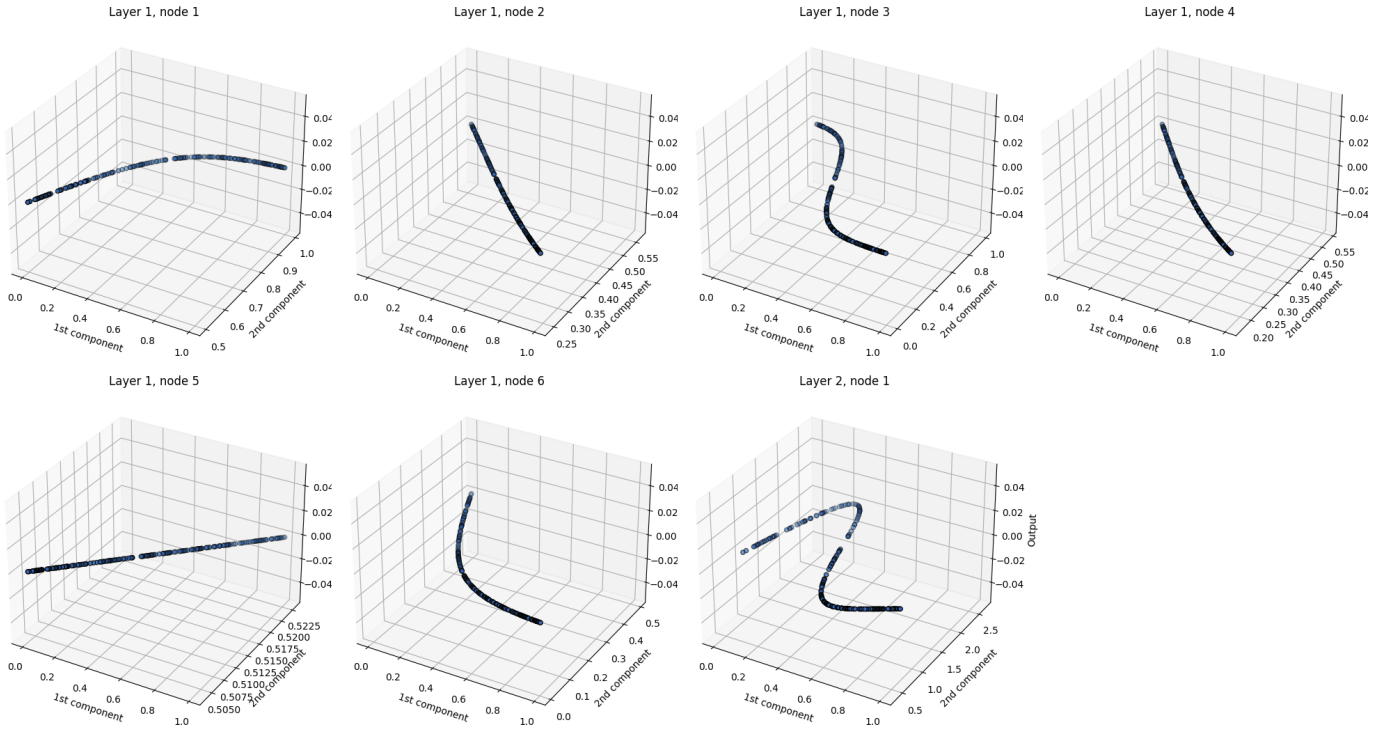

Fig. *Validation data : Output on z-axis and inputs in x-axis and y-axis of all hidden layers and one output layer.*

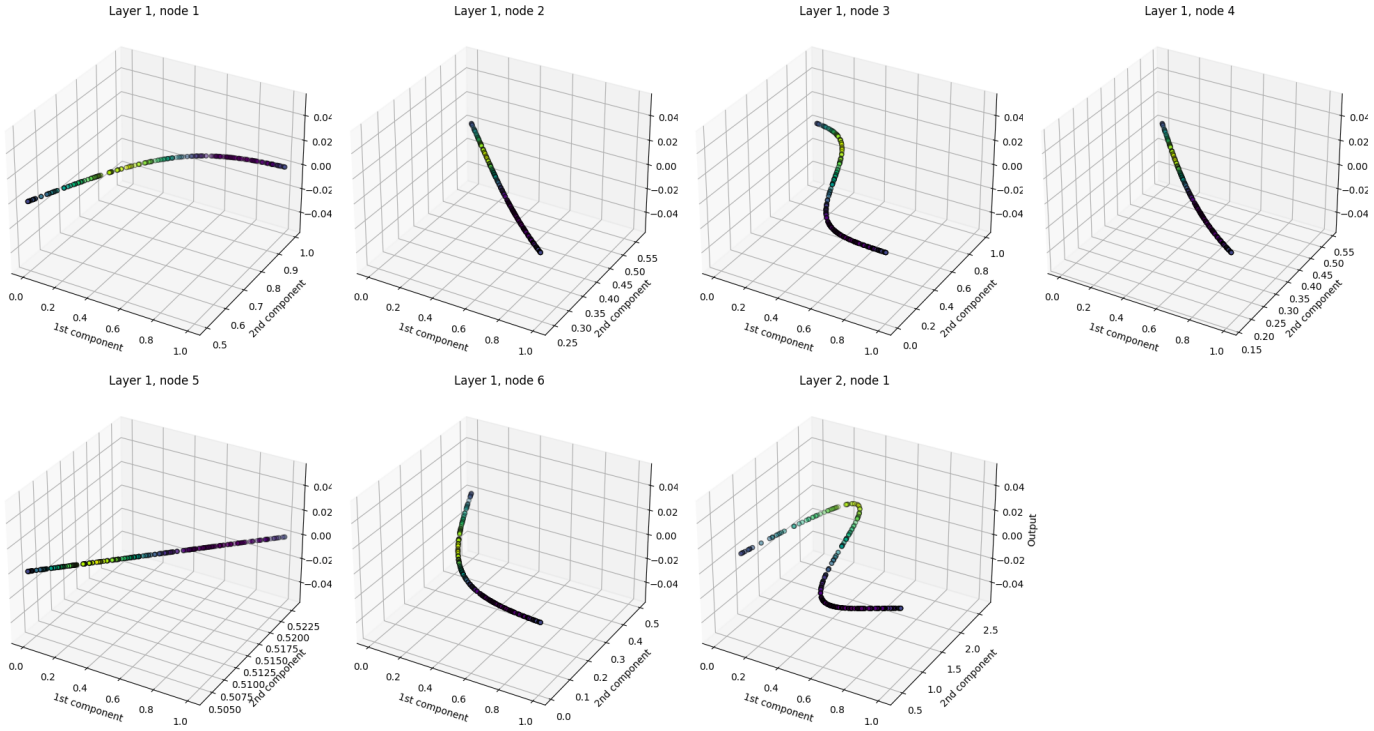

Fig. *Test data : Output on z-axis and inputs in x-axis and y-axis of all hidden layers and one output layer.*

Bivariate Input Data:

PERCEPTRON :

Avg Error Vs epochs : We see from the graph that the error decreases drastically for the first 3 epochs but remains somewhat constant thereafter. So, our model learned the optimal parameters quite fast.

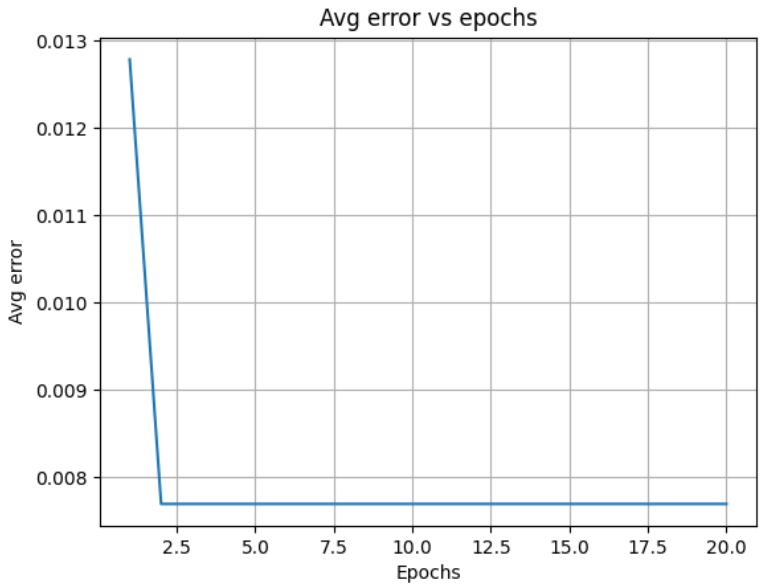

Fig. *Avg Error Vs Epochs Graph*

MSE Errors On Various Sets: As is visible from the graph, the MSE is almost the same for every set. This indicates that our model isn't overfitting on the train data. This is further supported by the accuracies obtained for various sets. Accuracy of 81.43% was obtained for the train data, 82.01% was obtained for the validation set, and 80.54% was obtained for the test set. These accuracies indicates that our model isn't properly generalizing the dataset.

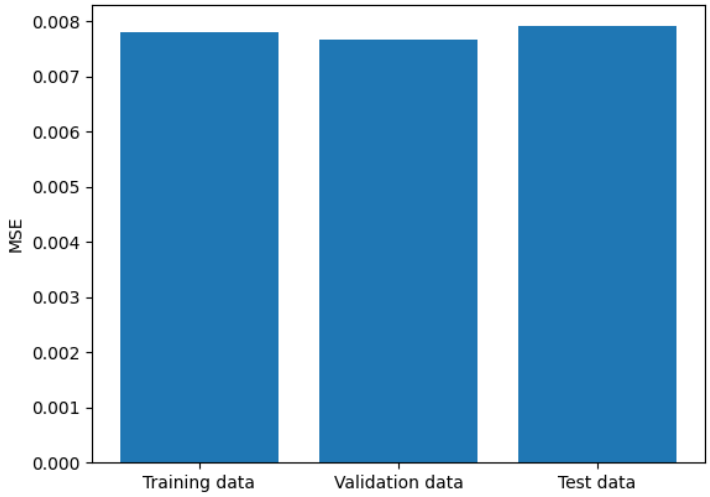

Fig. *MSEs for various sets*

Predicted Output Vs Target Output: In LHS graphs a distorted thick straight line (y = x) is being obtained, which indicates that our model is predicting the values that slightly differ from the target output. This is further confirmed by the RHS scatter plots and accuracies of around 80% for various sets.

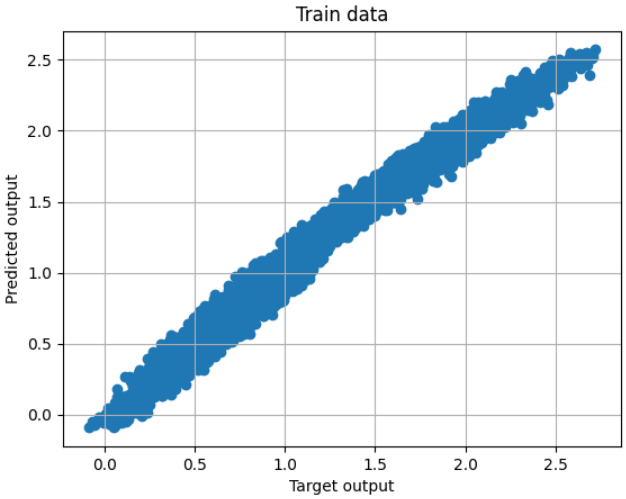

Fig. *Predicted Vs Target output for train data*

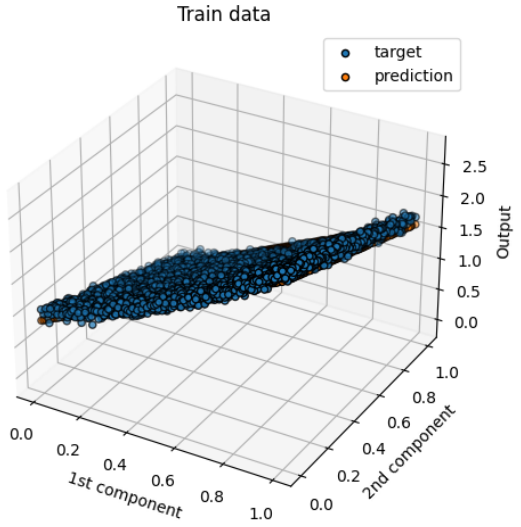

Fig. *Predicted Vs Target output for train data*

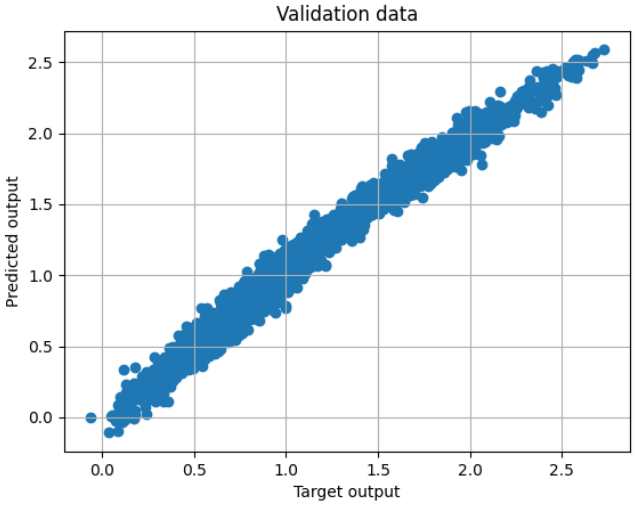

Fig. *Predicted Vs Target output for validation data*

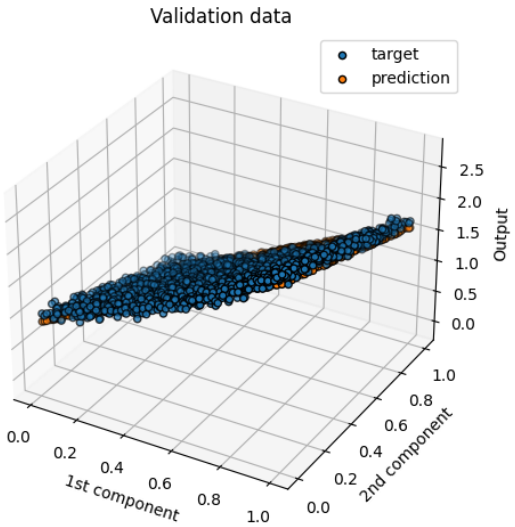

Fig. *Predicted Vs Target output for validation data*

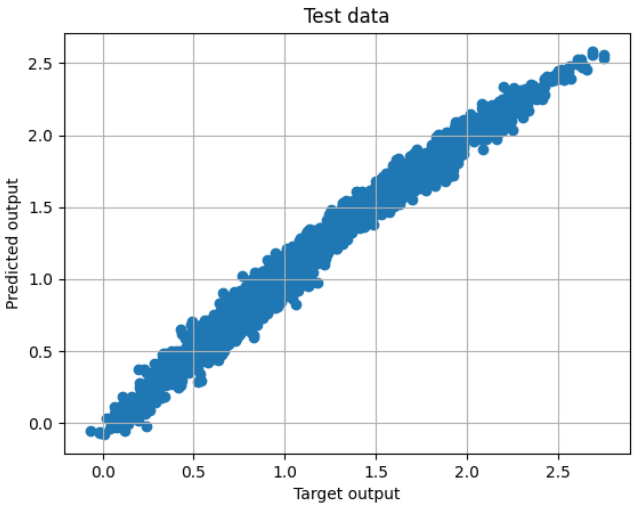

Fig. *Predicted Vs Target output for test data*

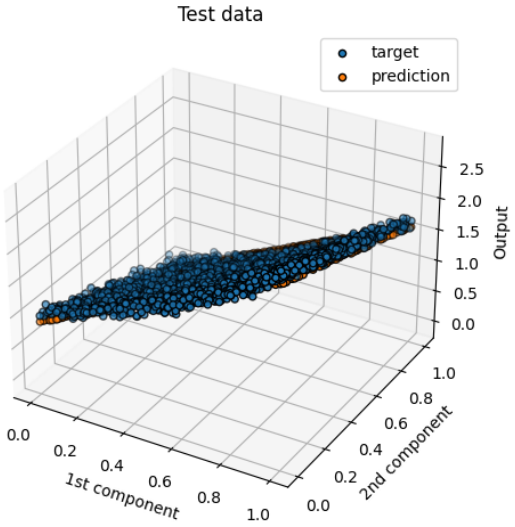

Fig. *Predicted Vs Target output for test data*

FCNN WITH 1 HIDDEN LAYER :

We used one hidden layer with 3 nodes. A learning rate of 0.05 was used and 20 epochs were used to train the data.

Avg Error Vs epochs : We see from the graph that the error is decreasing continuously. It decreases drastically for the first 3 epochs, and then decreases gradually thereafter. Also as compared to Perceptron model , average error is much lower which indicates that our model is performing better than the perceptron model.

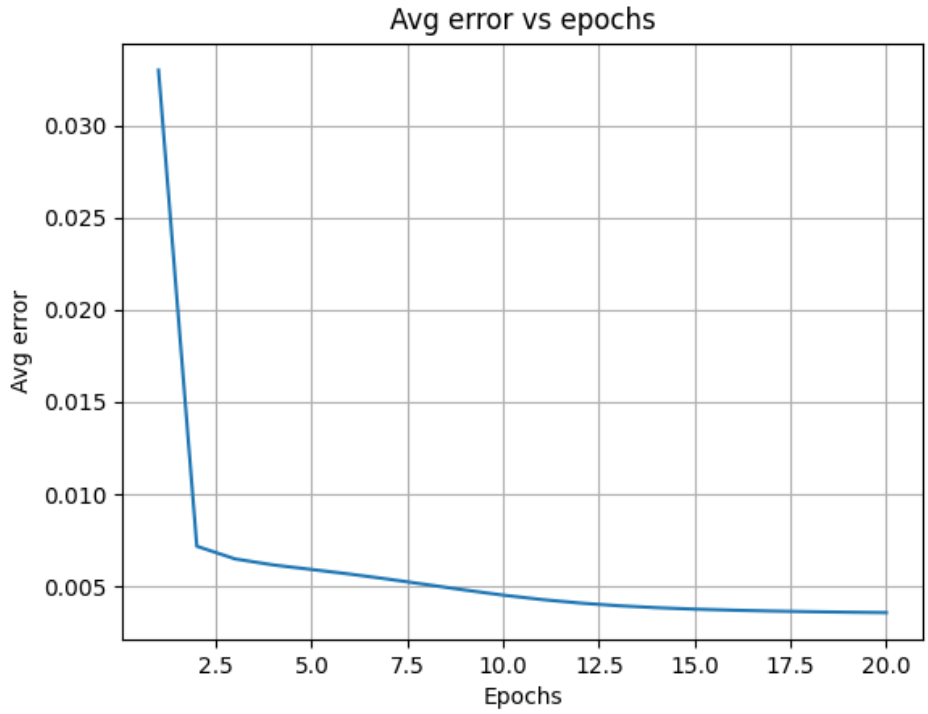

Fig. *Avg Error Vs Epochs Graph*

MSE Errors On Various Sets: As is visible from the graph, the MSE is almost the same for every set. This indicates that our model isn't overfitting on the train data. Also a much lower MSEs are obtained as compared to the perceptron model which indicates that this FCNN model is performing better than the perceptron model. This is further supported by the accuracies obtained for various sets. Accuracy of 92.19% was obtained for the train data, 91.52% was obtained for the validation set, and 90.88% was obtained for the test set. These accuracies indicate that although our model is performing better than the perceptron model, there is still some scope for improvement.

Fig. *MSEs for various sets*

Predicted Output Vs Target Output: We observe that the thickness of the lines in LHS graphs have reduced as compared to the perceptron model, which indicates that our model is getting closer to the y=x line. This is further confirmed by the RHS scatter plots and accuracies of fitting of around 91% for various sets.

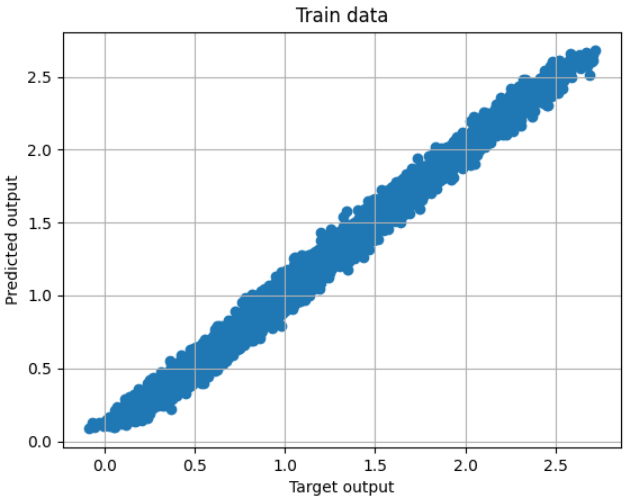

Fig. *Predicted Vs Target output for train data*

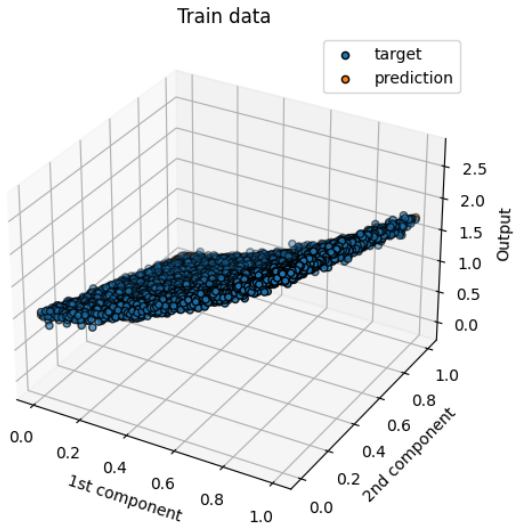

Fig. *Predicted Vs Target output for train data*

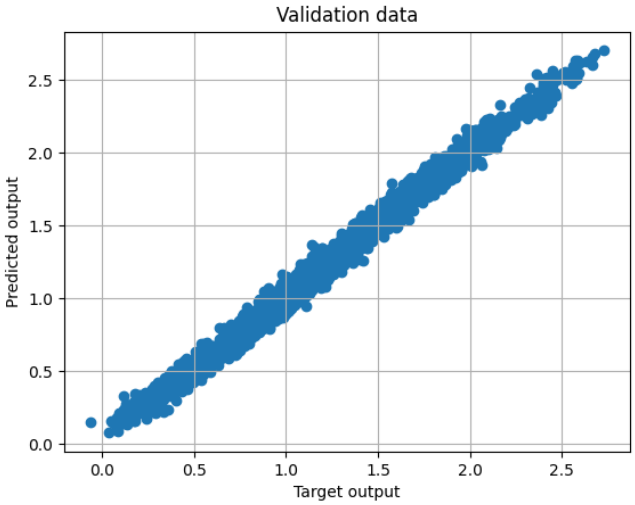

Fig. *Predicted Vs Target output for validation data*

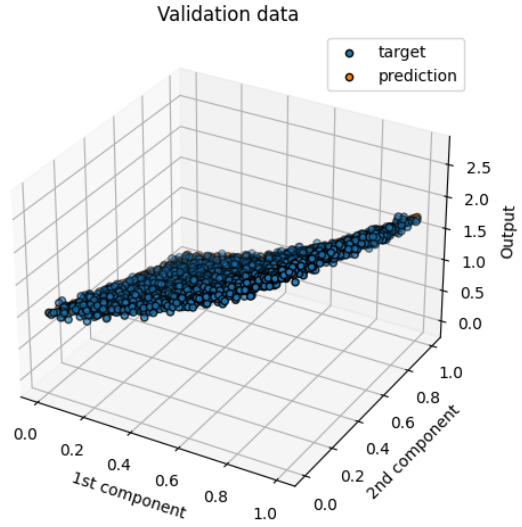

Fig. *Predicted Vs Target output for validation data*

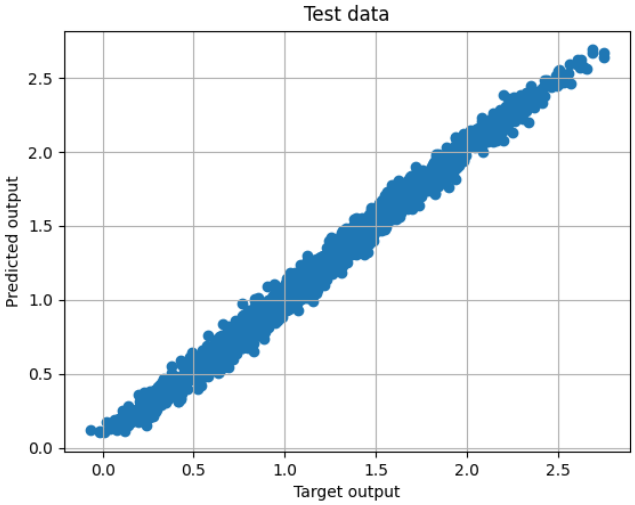

Fig. *Predicted Vs Target output for test data*

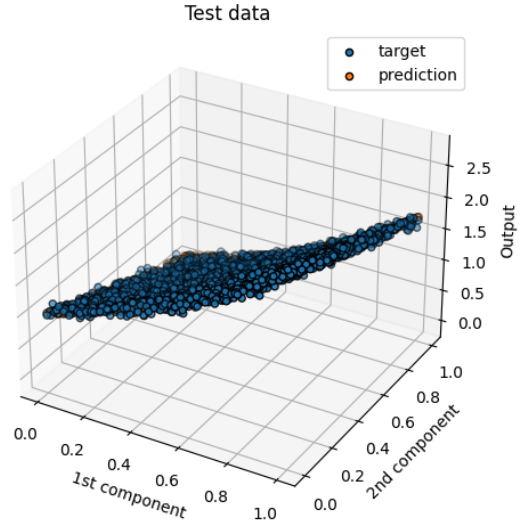

Fig. *Predicted Vs Target output for test data*

All Node Outputs :

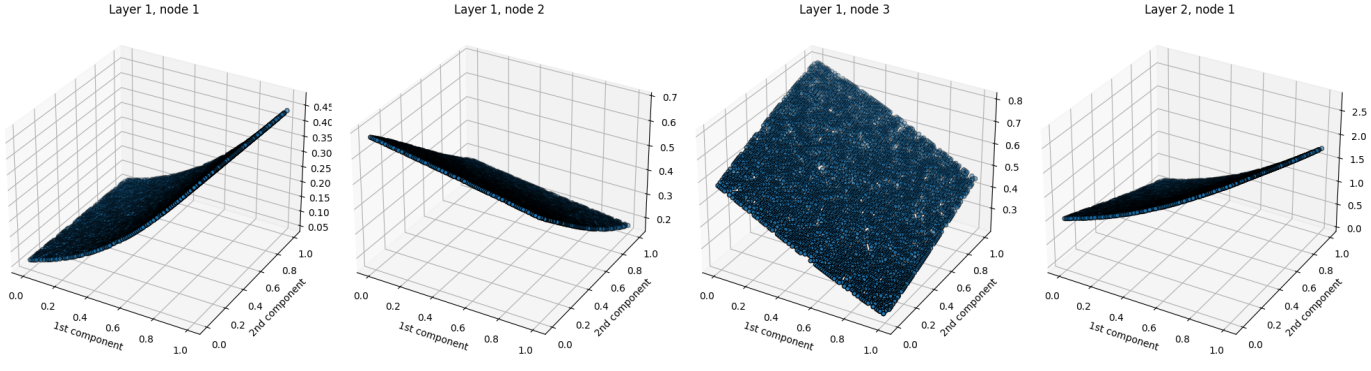

Fig. *Outputs of various nodes for Train data*

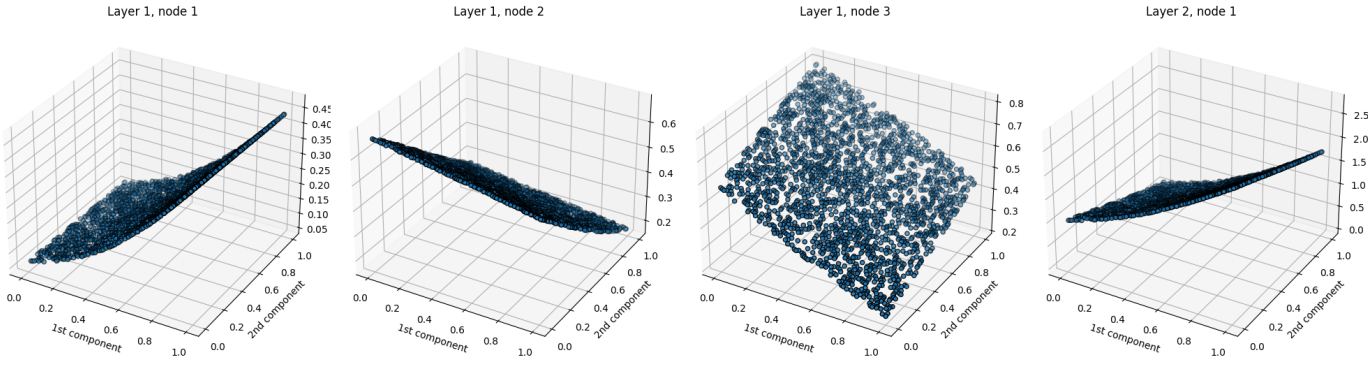

Fig. *Outputs of various nodes for Validation data*

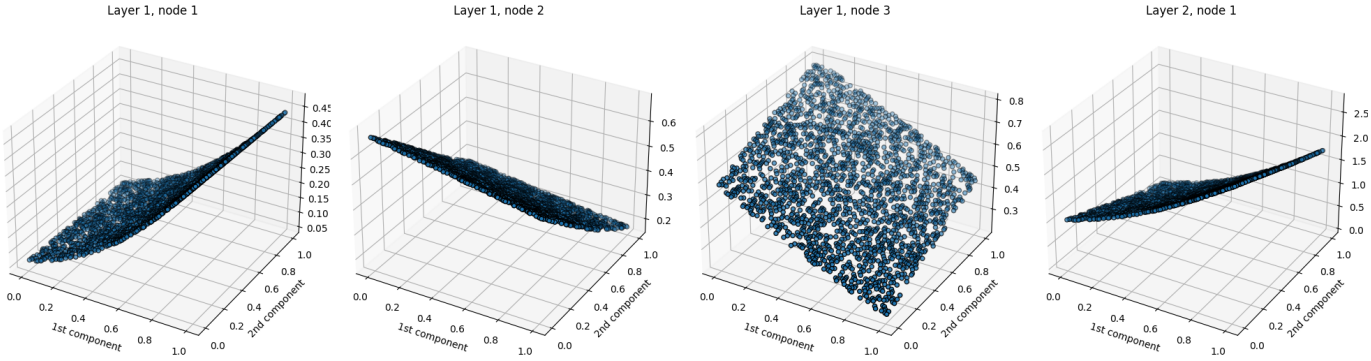

Fig. *Outputs of various nodes for Test data*

FCNN WITH 2 HIDDEN LAYER :

We used two hidden layers consisting of 3 and 4 nodes, so that the effective neural network becomes [2,3,4,1]. A learning rate of 0.05 was used and 300 epochs were used to train the data.

Avg Error Vs epochs : We see from the graph that the error decreases drastically for around first 20 epochs, and then remains almost constant. We also noticed that the error almost becomes 0 after 250 epochs.

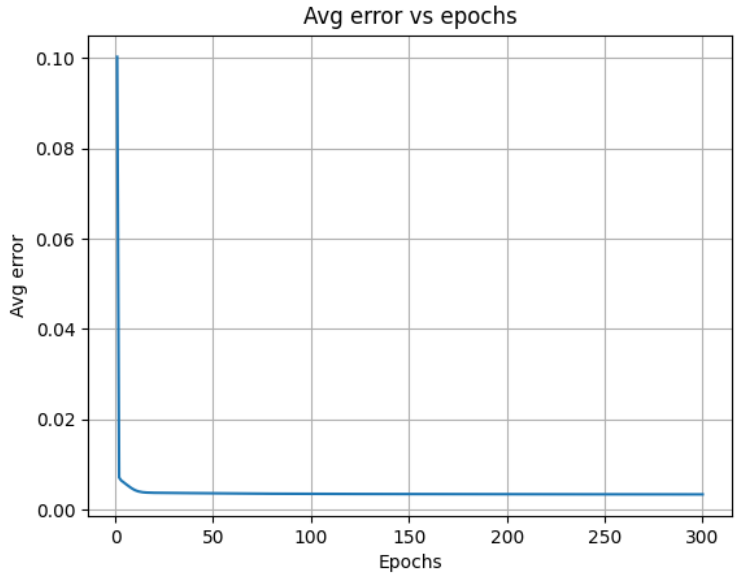

Fig. *Avg Error Vs Epochs Graph*

MSE Errors On Various Sets: As is visible from the graph, the MSE is almost the same for every set. This indicates that our model isn't overfitting on the train data and is generalizing well over the data. Accuracy of 92.14% was obtained for the train data, 91.32% was obtained for the validation set, and 90.78% was obtained for the test set. These accuracies are slightly less than as compared to the FCNN with 1 hidden layer , but as we see the difference between different accuracies is much smaller here as compared to the above models which indicates that this model is more general.

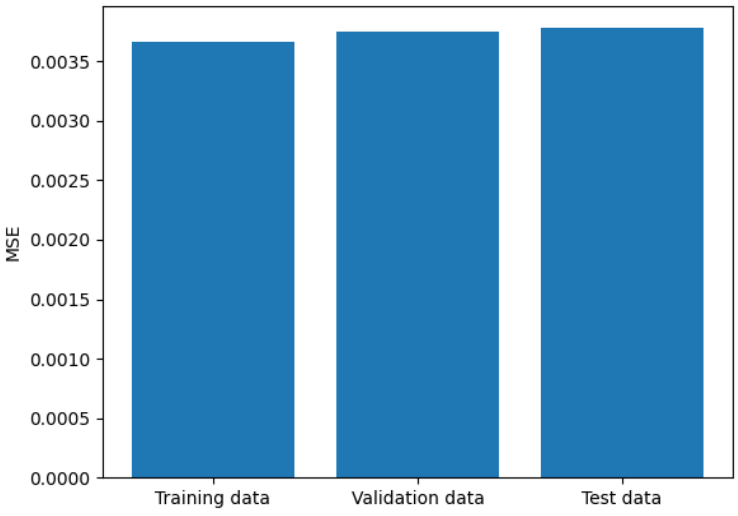

Fig. *MSEs for various sets*

Predicted Output Vs Target Output: The thickness of line is comparable to that of the FCNN with 1 hidden layer. Most of the data points lie around the y=x line, which indicates that most of the data points are being correctly predicted. This is further confirmed by the RHS scatter plots and accuracies of fitting of around 91% for various sets.

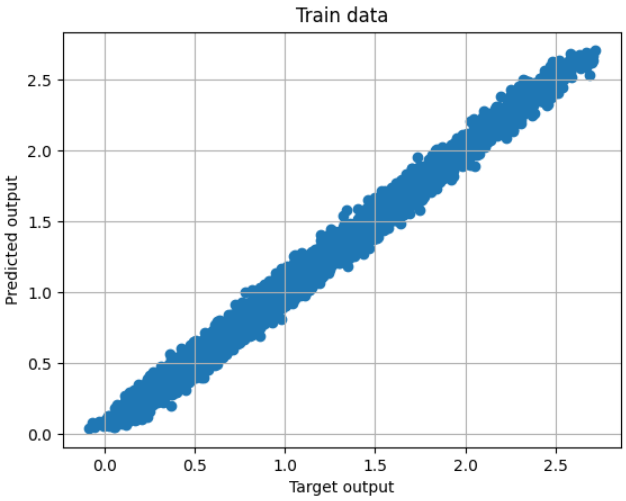

Fig. *Predicted Vs Target output for train data*

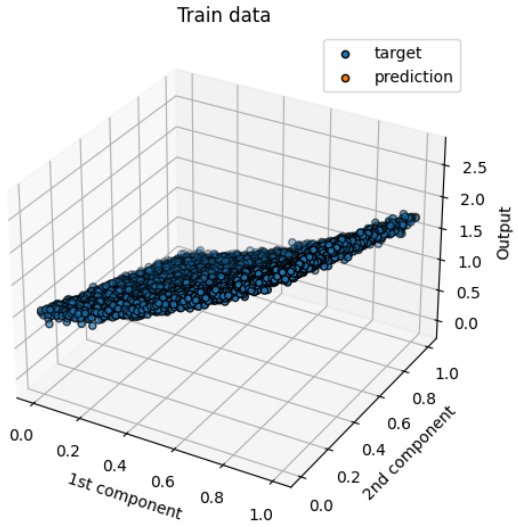

Fig. *Predicted Vs Target output for Train data*

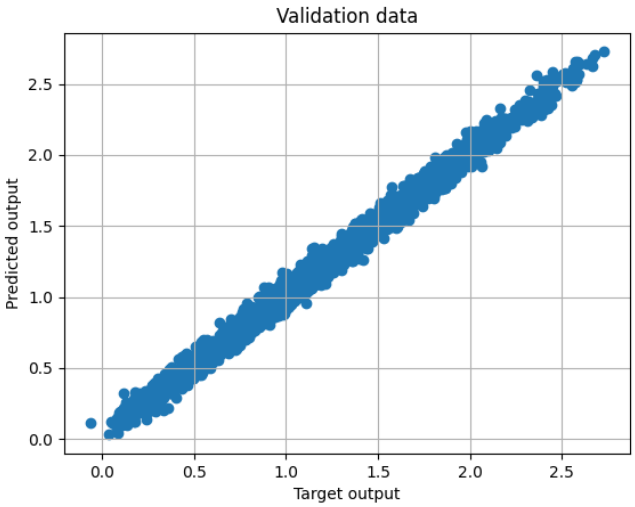

Fig. *Predicted Vs Target output for Validation data*

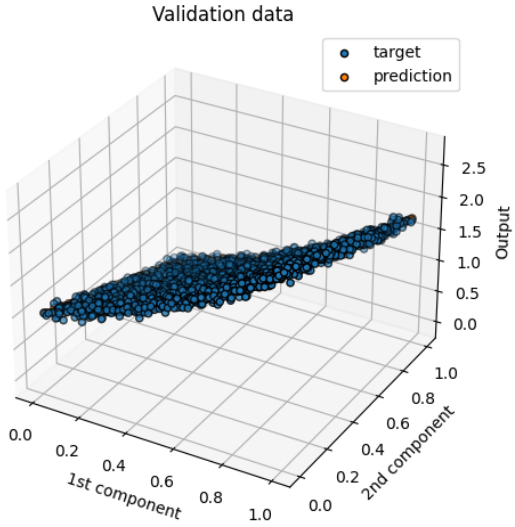

Fig. *Predicted Vs Target output for validation data*

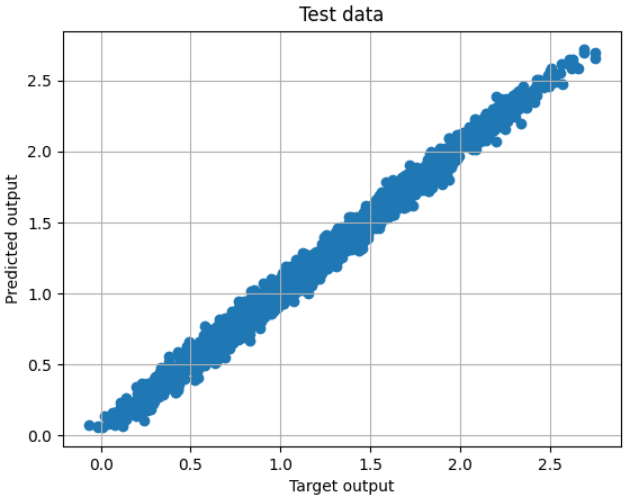

Fig. *Predicted Vs Target output for test data*

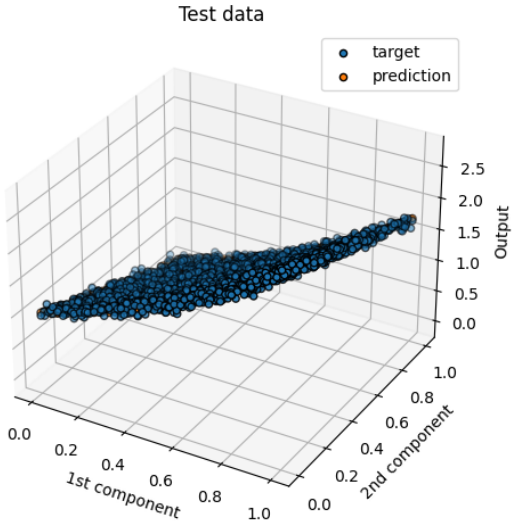

Fig. *Predicted Vs Target output for test data*

All Node Outputs

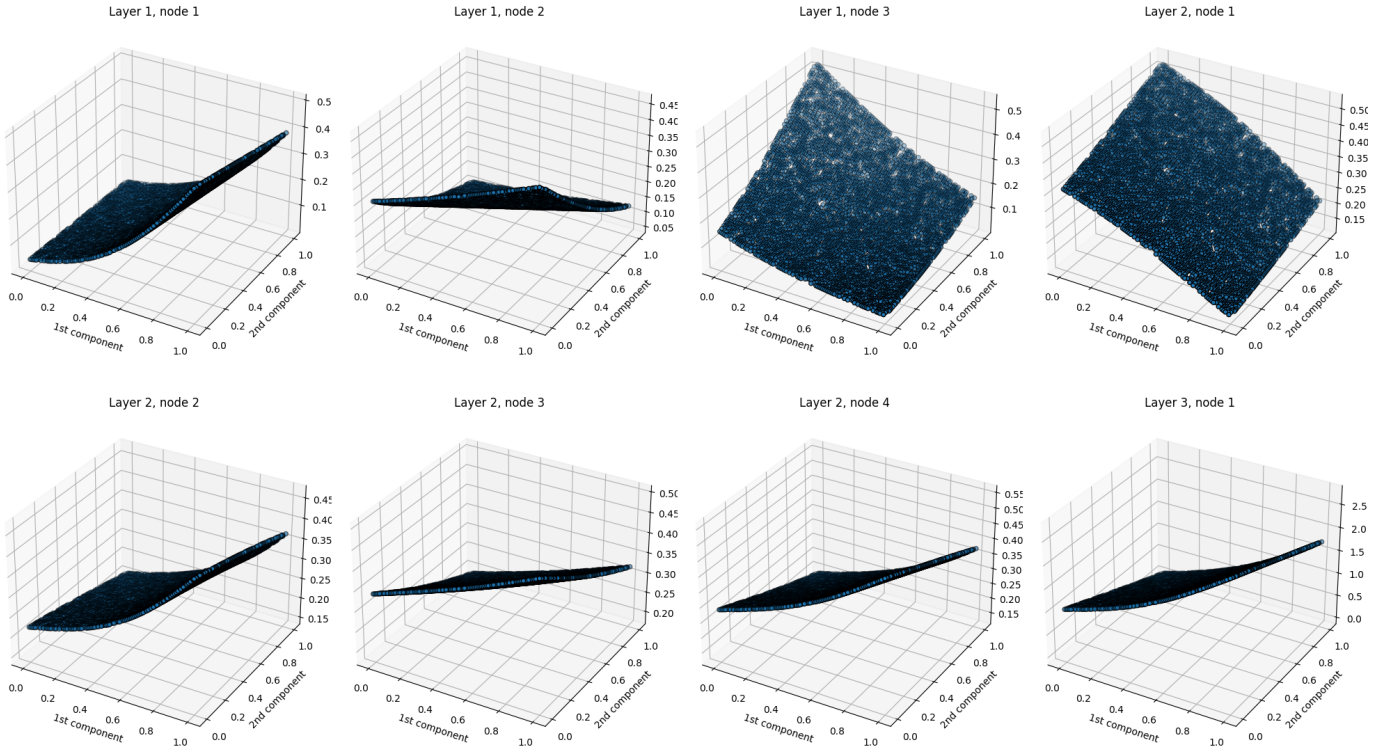

Fig. *Outputs of various nodes for Train data*

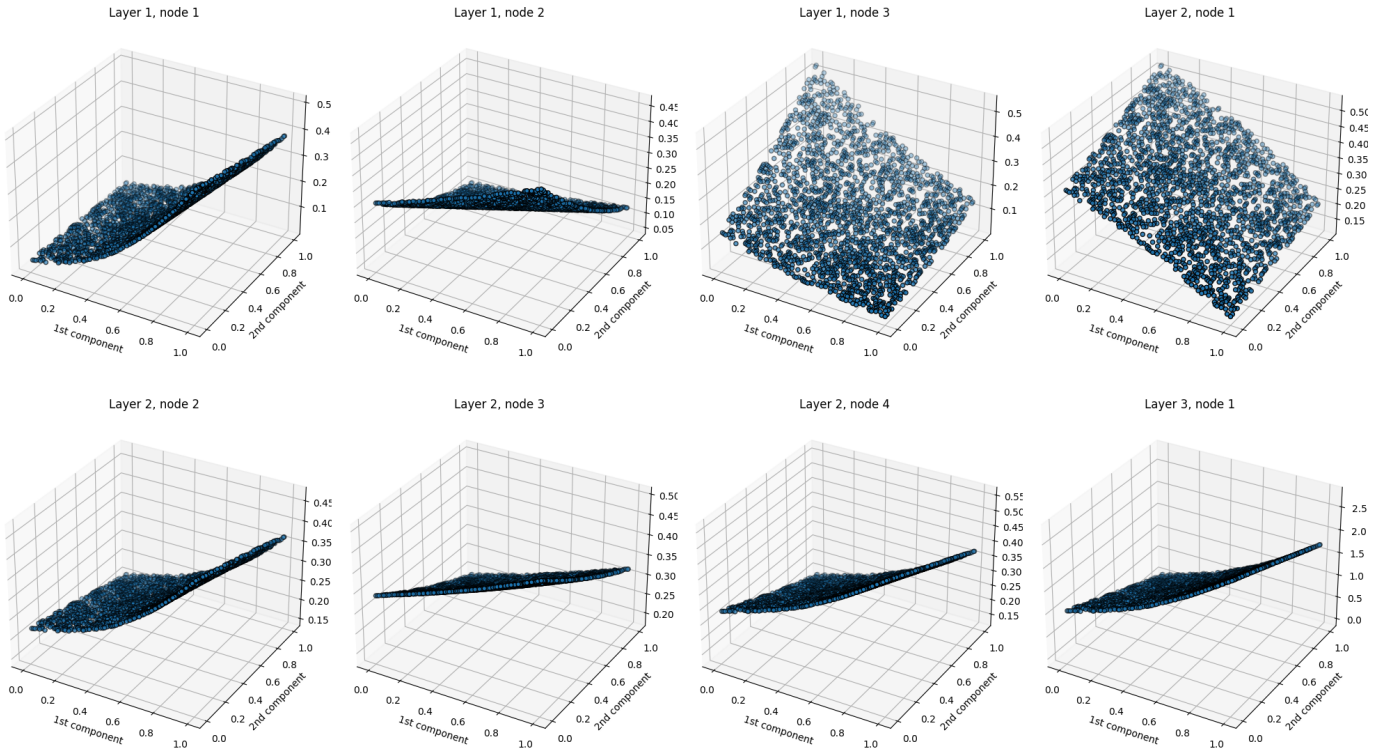

Fig. *Outputs of various nodes for Validation data*

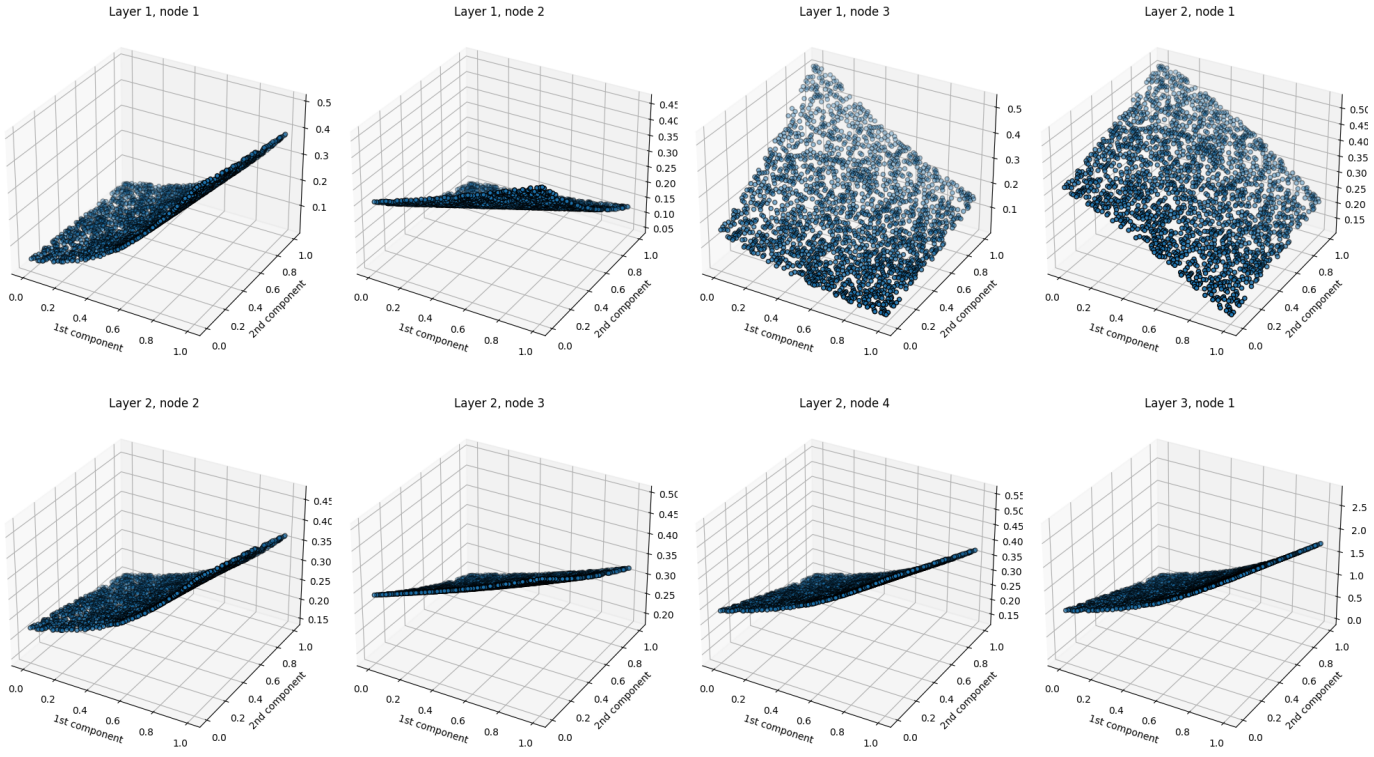

Fig. *Outputs of various nodes for Test data*