

**CS6910: Fundamentals of Deep Learning**

**Programming Assignment-1**

**Function Approximation and Single-Label Classification using**

**MLFFNNs**



**Team Number 19**

**Aditya Mohan ED19B001   Shreyas Singh ED19B030**

**Sujay Srivastava ED19B033**

**Course Instructor:**

**CHANDRA SEKHAR C**

**Dept. Of Computer Science**

**Indian Institute of Technology Madras-600036**

## TASK I: Function approximation on 2-d input

Model Details:

MultiLayer Feedforward Neural Network with 2 hidden layers

Activation Function: Tanh function

Weight Update Rule: Generalised Delta Rule

Learning Mode: Pattern Mode

### 1.1 Plot of an average error on training data vs Epoch

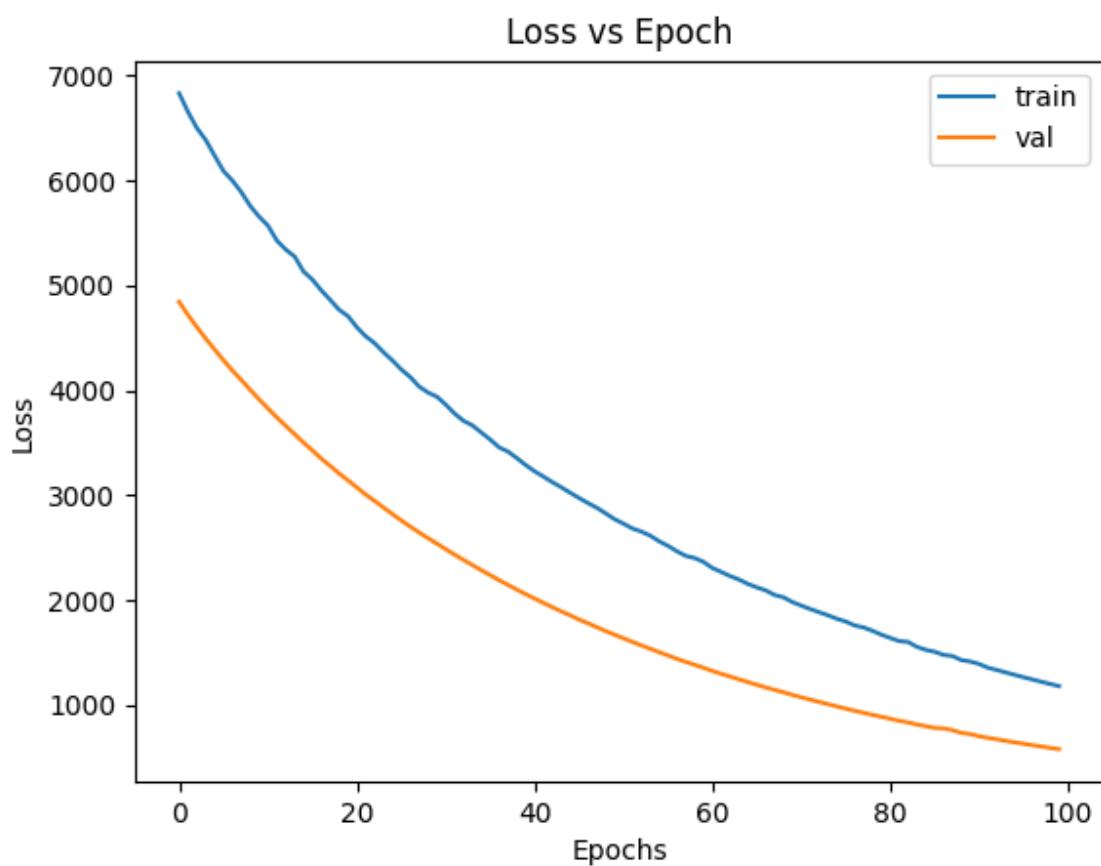
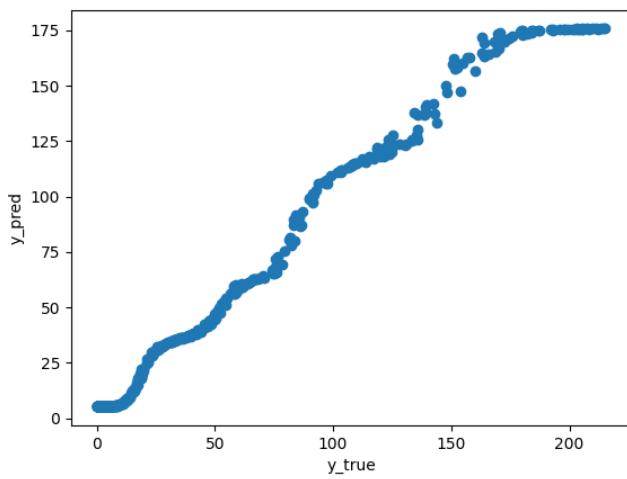
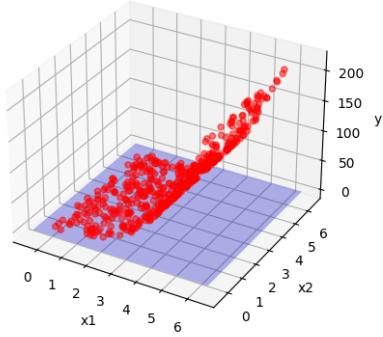


Fig 1.1 MSE error on training data vs epochs

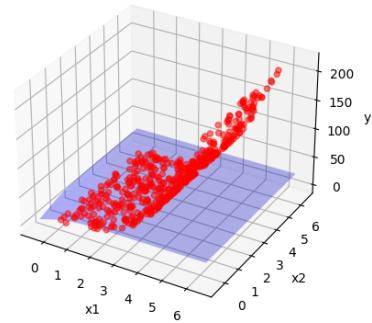
### 1.2 Scatter plot (Model output vs Desired output) for training data after the model is trained



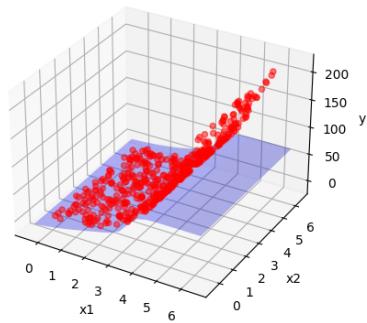
### 1.3 Plots of the desired function and the approximated function



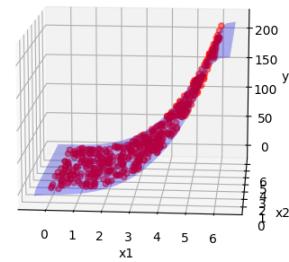
After Epoch 1



After Epoch 10



After 50 Epoch



After convergence

#### **1.4 Observation**

- For a larger learning rate, the error is coming higher
- For the reasonable change in the number of nodes in the hidden layer, the regression function is not heavily impacted
- The approximated function resembles a plane after single epoch and with the increasing number of epochs, the approximated function comes close to the datapoints..

## **TASK II:** Single-label multi-class classification on 2-D input

Model: MLFFNN with softmax output nodes with 2 hidden layers( 5 and 3 nodes respectively)

Hyperparameters:

Activation function: tanh

Weight update rule: Generalized delta rule

Epochs for convergence: 99

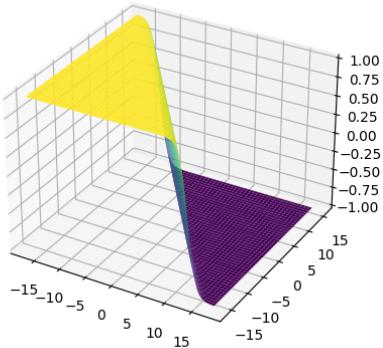
Accuracy at convergence: 97%

### 2.1 Surface Plots of Hidden Layer

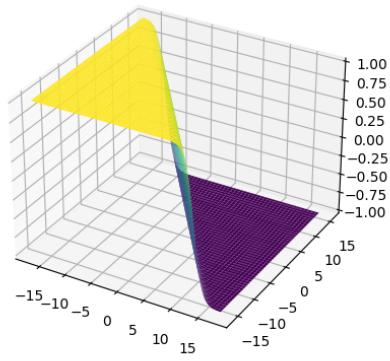
#### 2.1.1 Hidden Layer 1

### a) Node 1

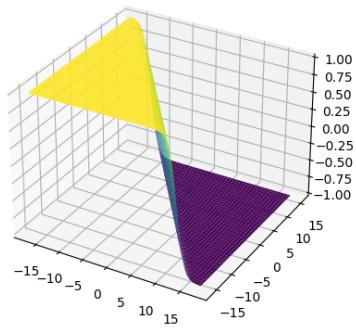
Surface plot layer : 1 node : 1 epoch : 1



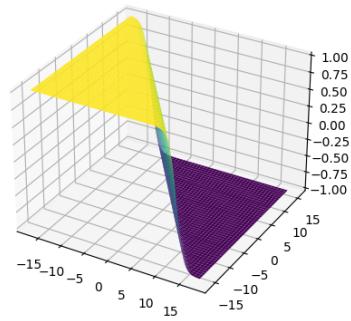
Surface plot layer : 1 node : 1 epoch : 2



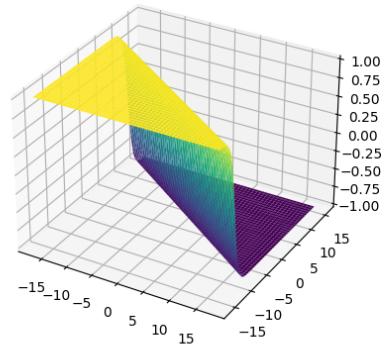
Surface plot layer : 1 node : 1 epoch : 10



Surface plot layer : 1 node : 1 epoch : 50

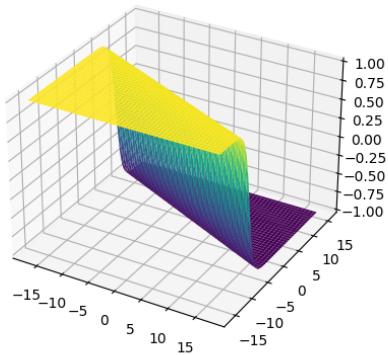


Surface plot layer : 1 node : 1 epoch : 99

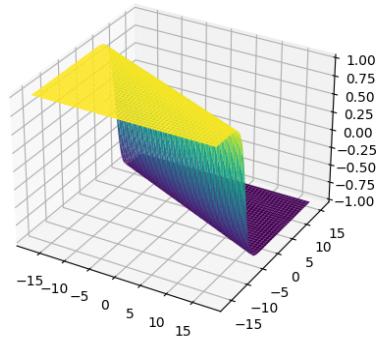


### b) Node 2

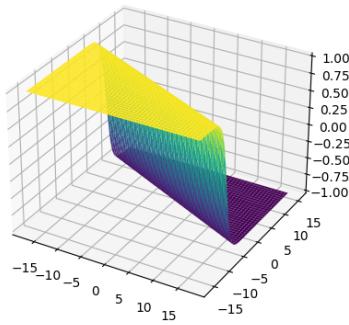
Surface plot layer : 1 node : 2 epoch : 1



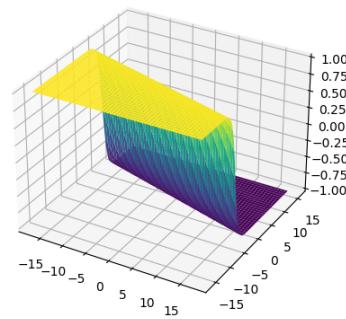
Surface plot layer : 1 node : 2 epoch : 2



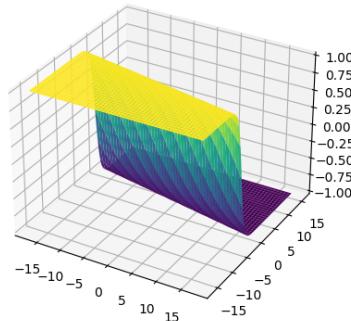
Surface plot layer : 1 node : 2 epoch : 10



Surface plot layer : 1 node : 2 epoch : 50

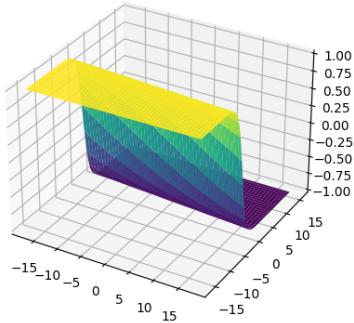


Surface plot layer : 1 node : 2 epoch : 99

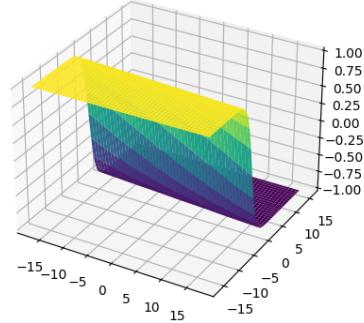


c) Node 3

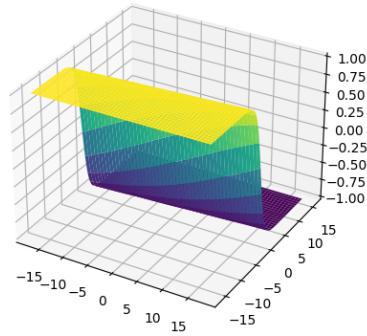
Surface plot layer : 1 node : 3 epoch : 1



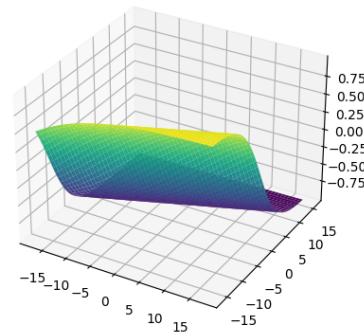
Surface plot layer : 1 node : 3 epoch : 2



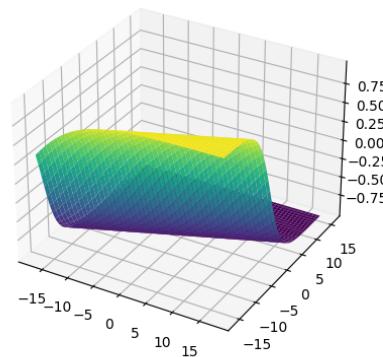
Surface plot layer : 1 node : 3 epoch : 10



Surface plot layer : 1 node : 3 epoch : 50

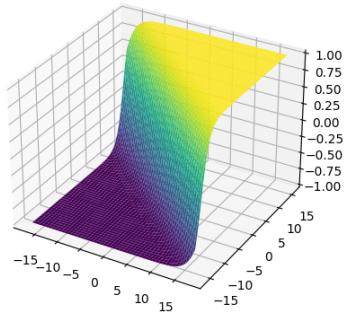


Surface plot layer : 1 node : 3 epoch : 99

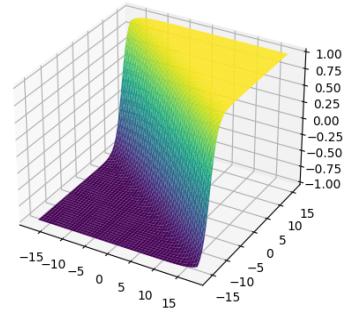


#### d) Node 4

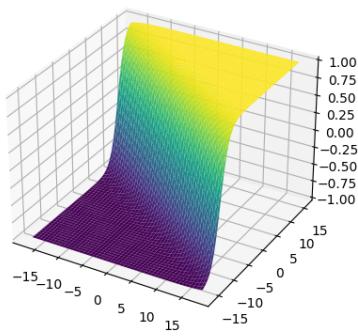
Surface plot layer : 1 node : 4 epoch : 1



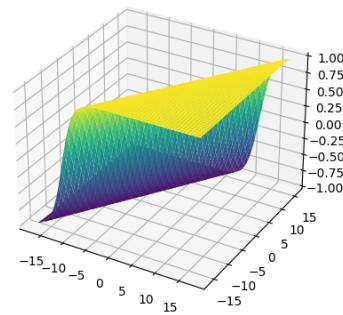
Surface plot layer : 1 node : 4 epoch : 2



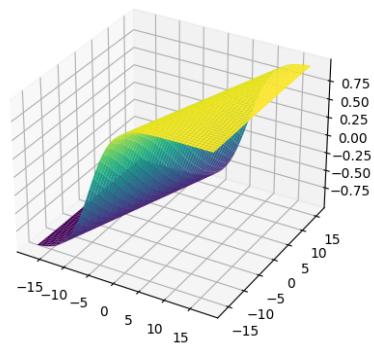
Surface plot layer : 1 node : 4 epoch : 10



Surface plot layer : 1 node : 4 epoch : 50

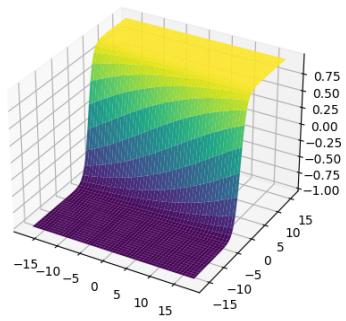


Surface plot layer : 1 node : 4 epoch : 99

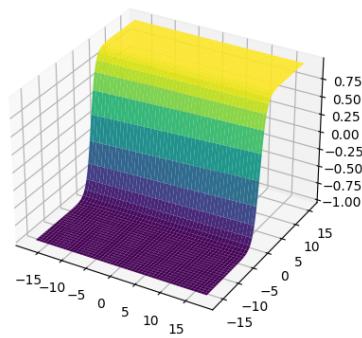


e) Node 5

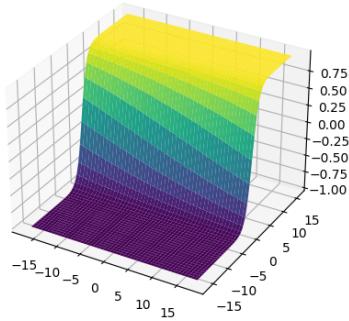
Surface plot layer : 1 node : 5 epoch : 1



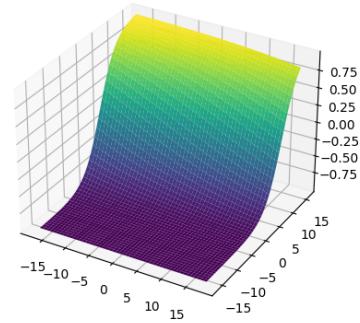
Surface plot layer : 1 node : 5 epoch : 2



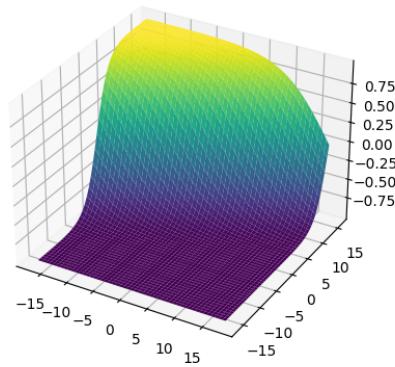
Surface plot layer : 1 node : 5 epoch : 10



Surface plot layer : 1 node : 5 epoch : 50



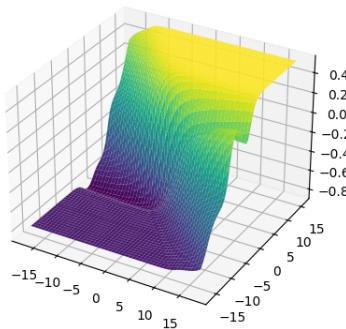
Surface plot layer : 1 node : 5 epoch : 99



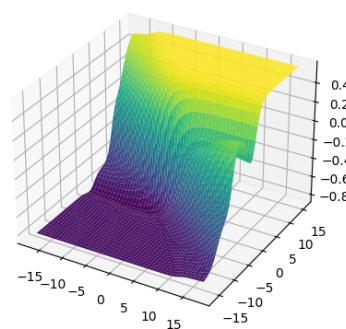
## 2.1.2 Hidden Layer 2

### a. Node1

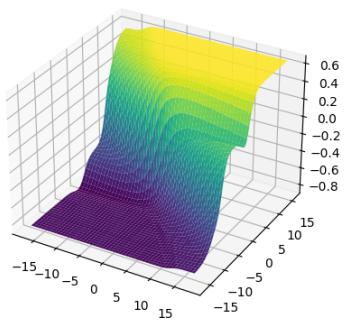
Surface plot layer : 2 node : 1 epoch : 1



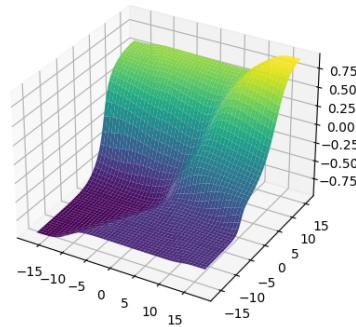
Surface plot layer : 2 node : 1 epoch : 2



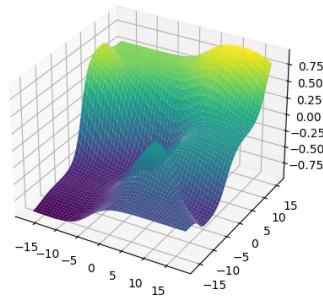
Surface plot layer : 2 node : 1 epoch : 10



Surface plot layer : 2 node : 1 epoch : 50

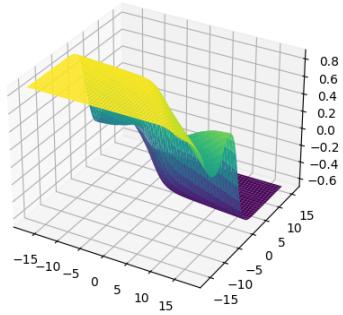


Surface plot layer : 2 node : 1 epoch : 99

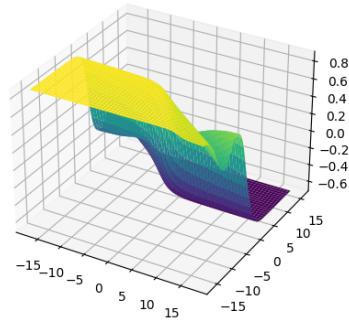


b. Node 2

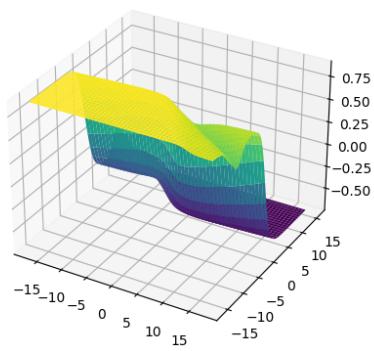
Surface plot layer : 2 node : 2 epoch : 1



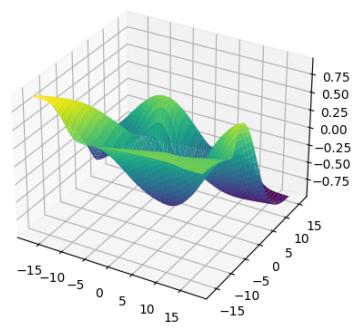
Surface plot layer : 2 node : 2 epoch : 2



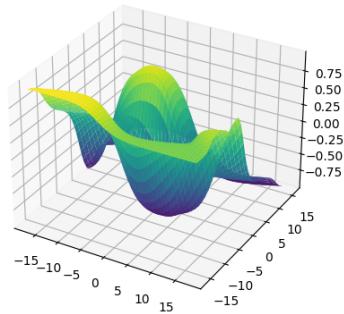
Surface plot layer : 2 node : 2 epoch : 10



Surface plot layer : 2 node : 2 epoch : 50

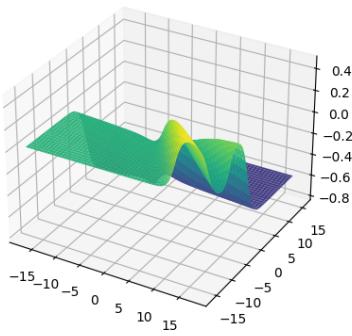


Surface plot layer : 2 node : 2 epoch : 99

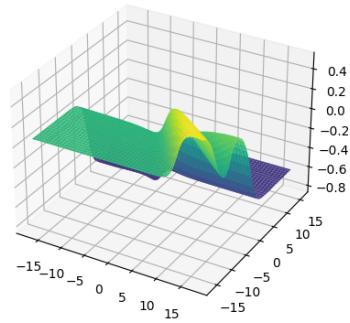


c. Node 3

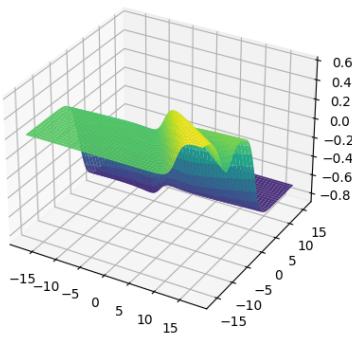
Surface plot layer : 2 node : 3 epoch : 1



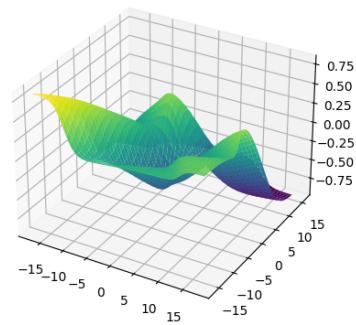
Surface plot layer : 2 node : 3 epoch : 2



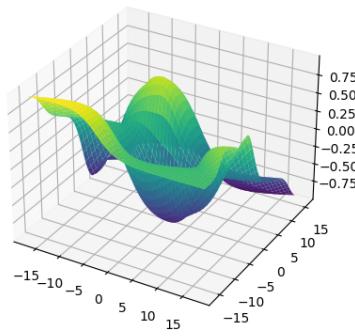
Surface plot layer : 2 node : 3 epoch : 10



Surface plot layer : 2 node : 3 epoch : 50



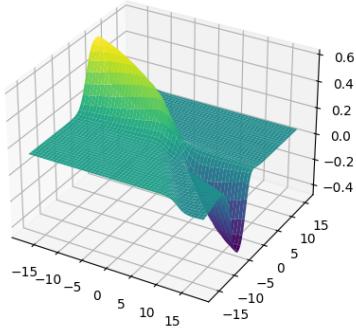
Surface plot layer : 2 node : 3 epoch : 99



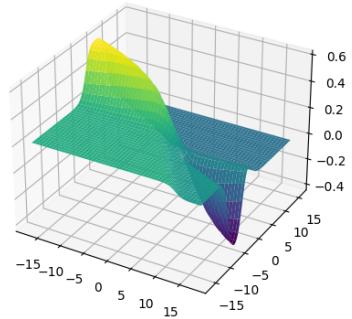
### 2.1.3 Output Layer

## a. Node 1

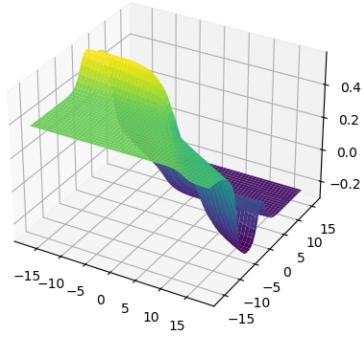
Surface plot layer : 3 node : 1 epoch : 1



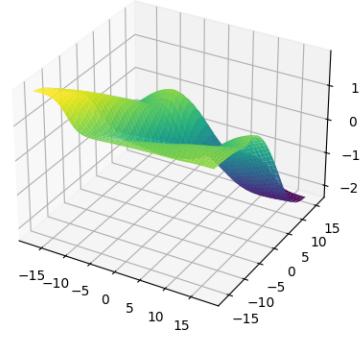
Surface plot layer : 3 node : 1 epoch : 2



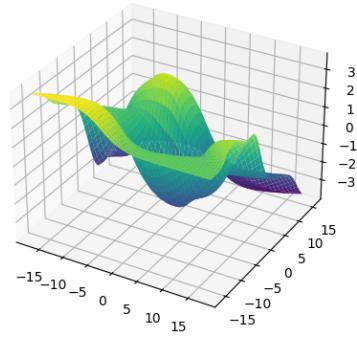
Surface plot layer : 3 node : 1 epoch : 10



Surface plot layer : 3 node : 1 epoch : 50

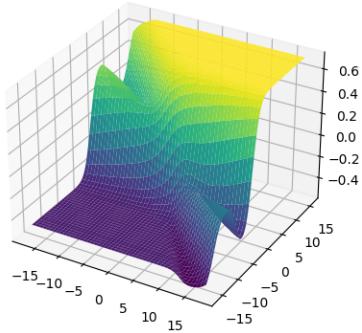


Surface plot layer : 3 node : 1 epoch : 99

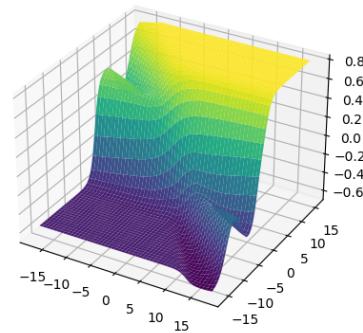


## b. Node 2

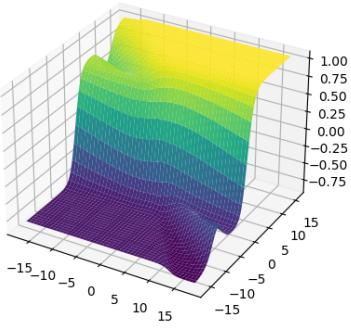
Surface plot layer : 3 node : 2 epoch : 1



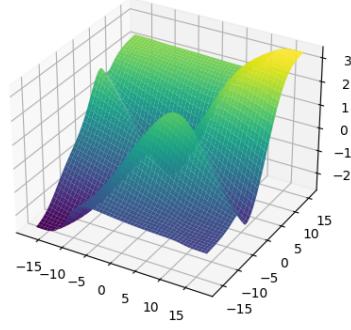
Surface plot layer : 3 node : 2 epoch : 2



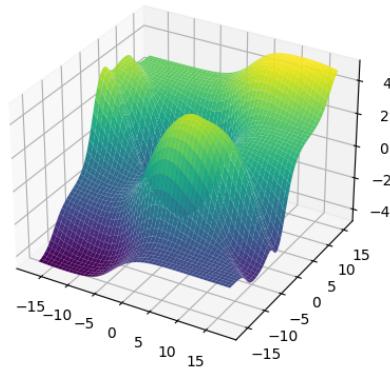
Surface plot layer : 3 node : 2 epoch : 10



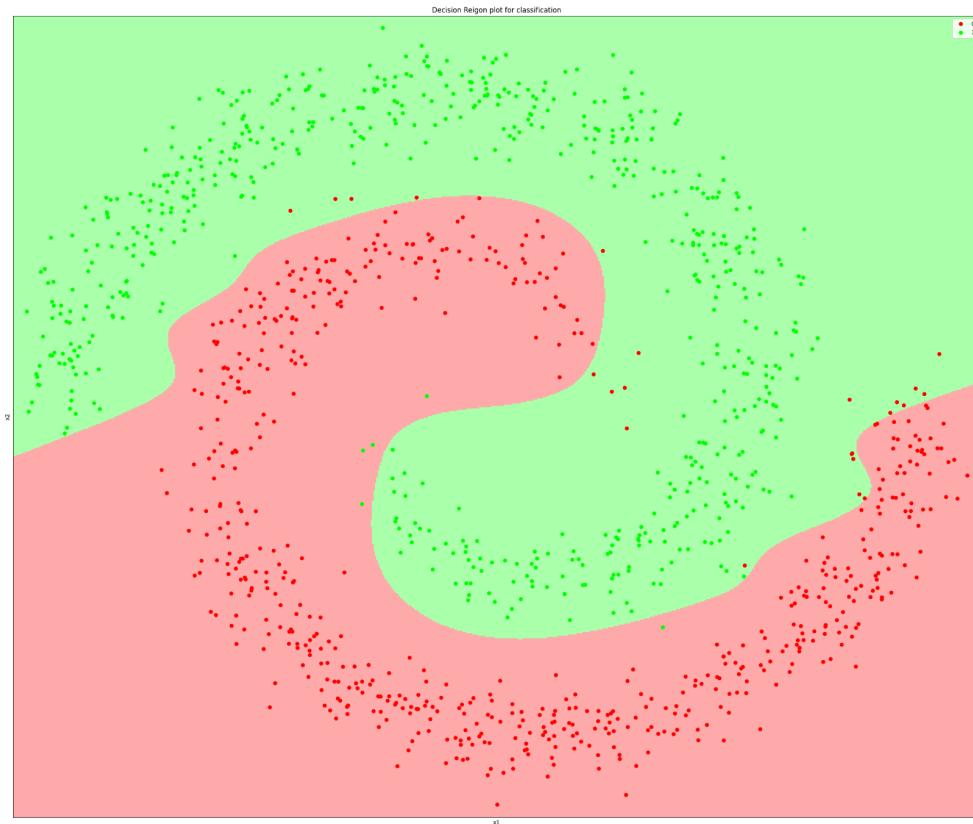
Surface plot layer : 3 node : 2 epoch : 50



Surface plot layer : 3 node : 2 epoch : 99



## 2.2 Decision Region Plot after the Model is Trained



Accuracy of the model after 99 epochs: 97%

## 2.3 Observation

- The surface plots in the hidden layer 1 have uniform representation, while the peaks and valleys, and hence the complexity/non-linearity, in the surface plot increases with the depth of hidden layer.

## Task III: Classification task for image data

Hyperparameters:

Learning Rate : 0.01

Same initial seed for the weights

Model: MLFFNN with 2 Hidden Layer( )

Activation Function: tanh

### 3.1 Loss on training data vs Epochs

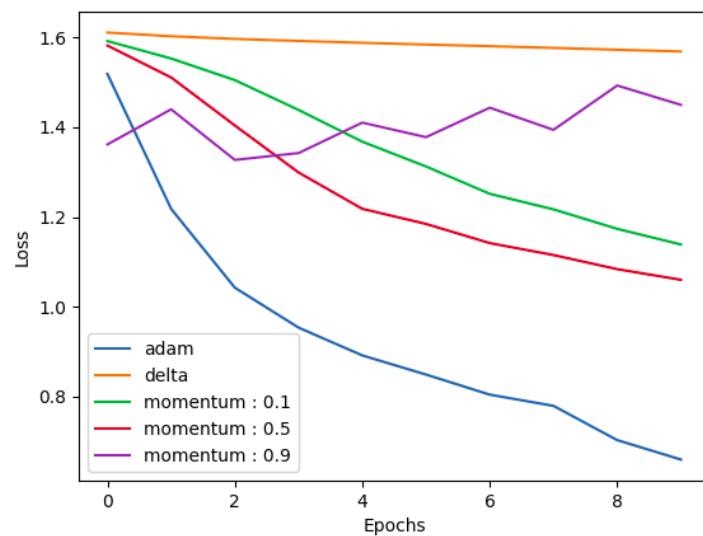


Fig 3.1 Loss on training data v/s epochs for different weight update rules

### 3.2 Accuracy on Validation Data vs Epochs

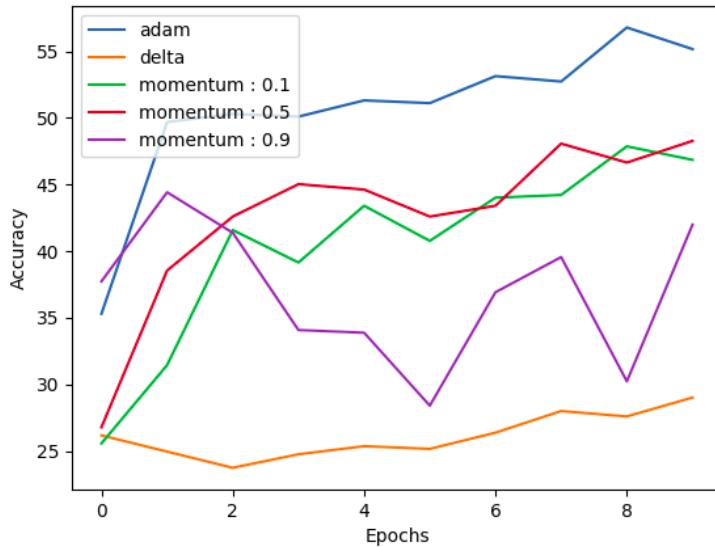


Fig 3.2 Accuracy v/s epochs graph for different weight update rules

Weight Update Rule	Accuracy on Validation Data
Adam	56%
Delta	29%
Generalised Delta Rule	
- Momentum 0.1	46%
- Momentum 0.5	48%
- Momentum 0.9	41%

Table 3.1 Weight Update Rule vs Accuracy on Validation Data

### Observations:

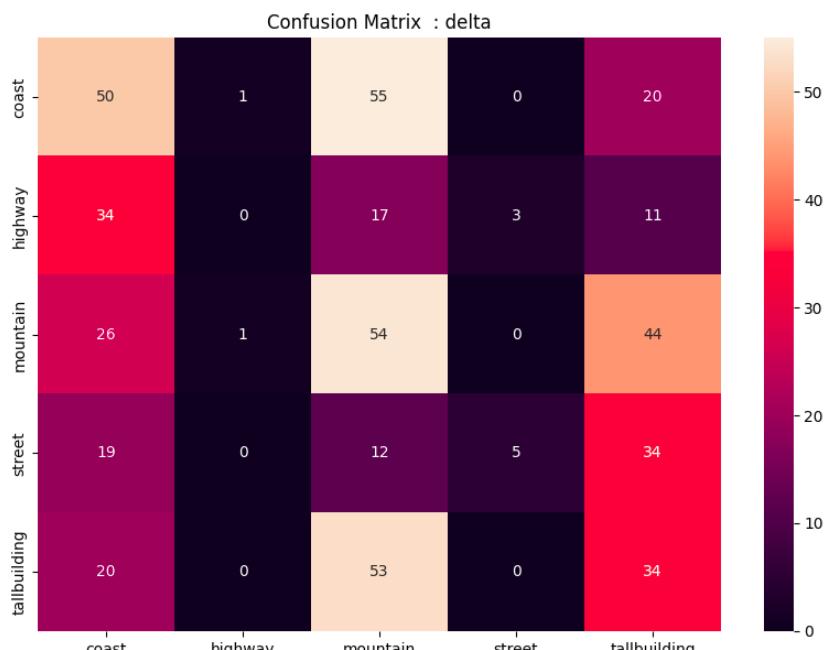
- We notice that Loss decreases maximum for Adam optimiser.
- Delta without momentum doesn't reduce loss significantly
- With momentum we can see that there is a reduction of loss
- But with higher momentum (0.9) the loss tends to oscillate
- We can see that adam has highest accuracy while Delta with momentum of 0.5 and 0.1 perform worse but still have good accuracy.

- Delta with momentum of 0.9 oscillates and doesn't give a good accuracy.
- Delta without momentum performs worst and has least accuracy.

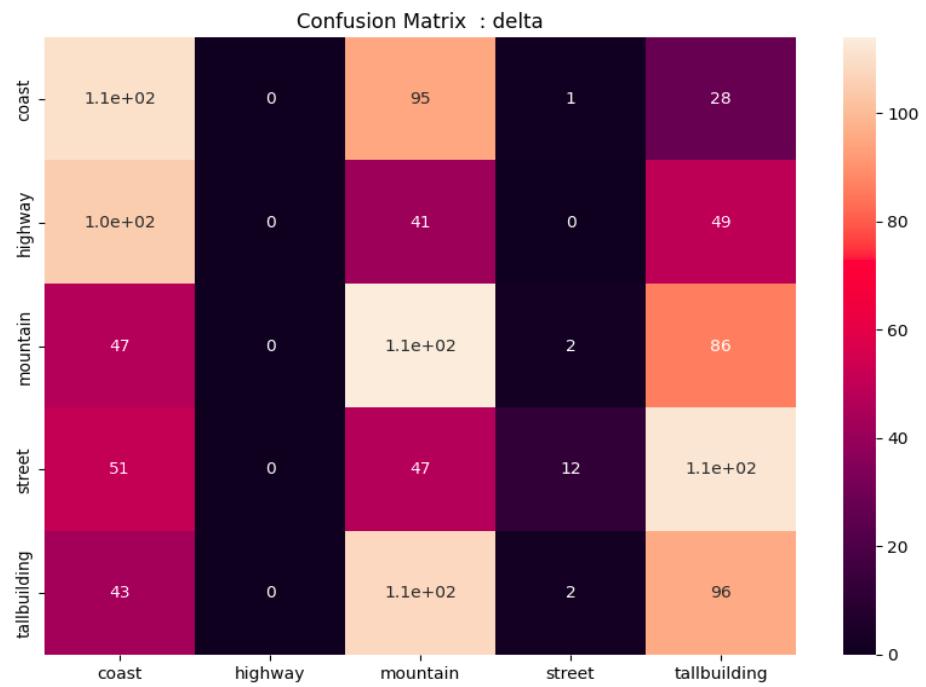
### 3.3 Confusion Matrix

#### 1. Delta Rule -

##### a. Confusion Matrix (Test Data)



##### b. Confusion Matrix (Training Data)

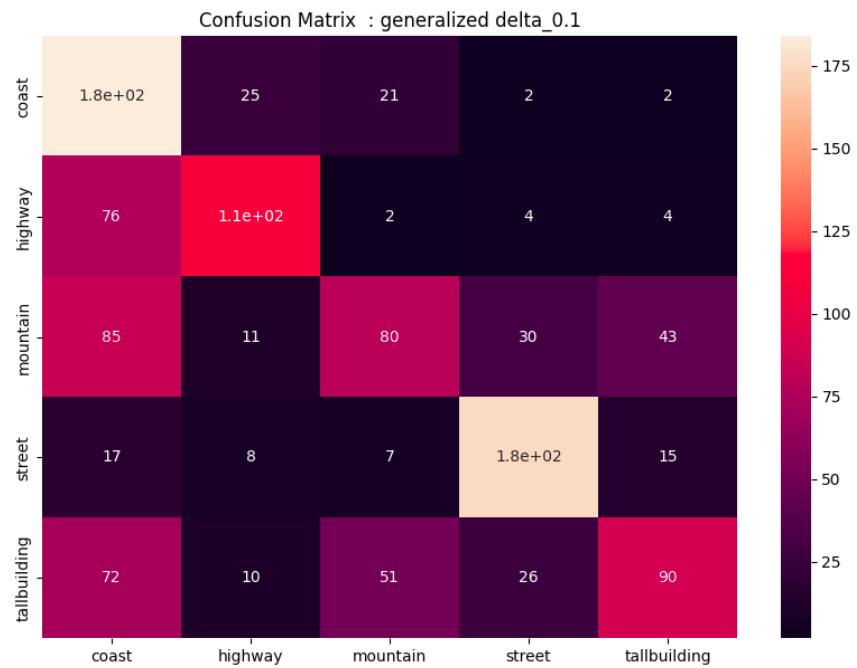


## 2. Generalized Delta Rule - (momentum = 0.1)

### a. Confusion Matrix (Test Data)

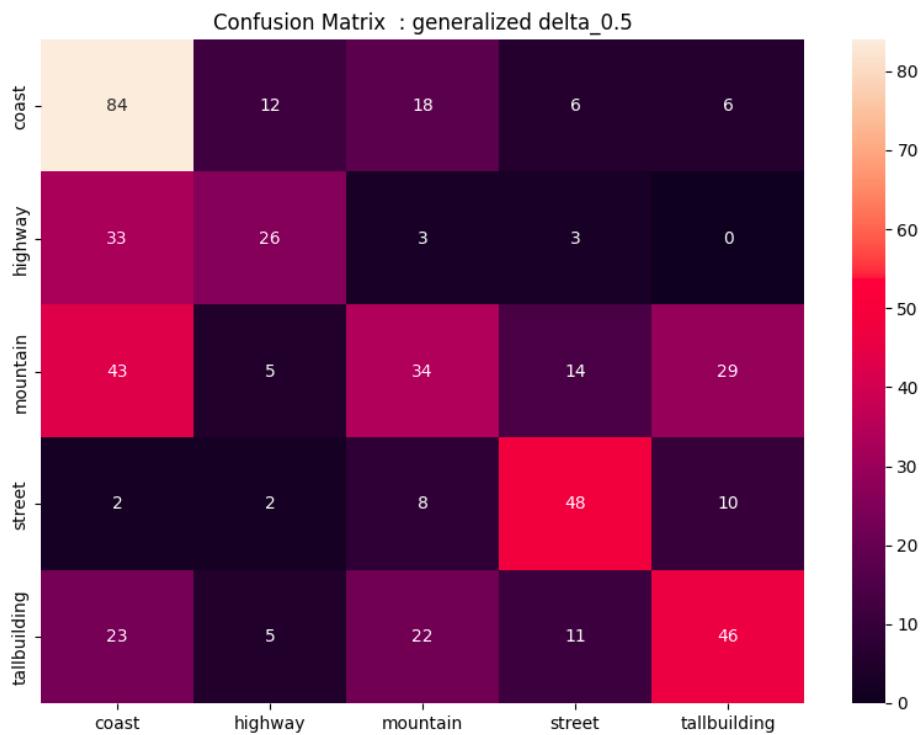


### b. Confusion Matrix (Training Data)

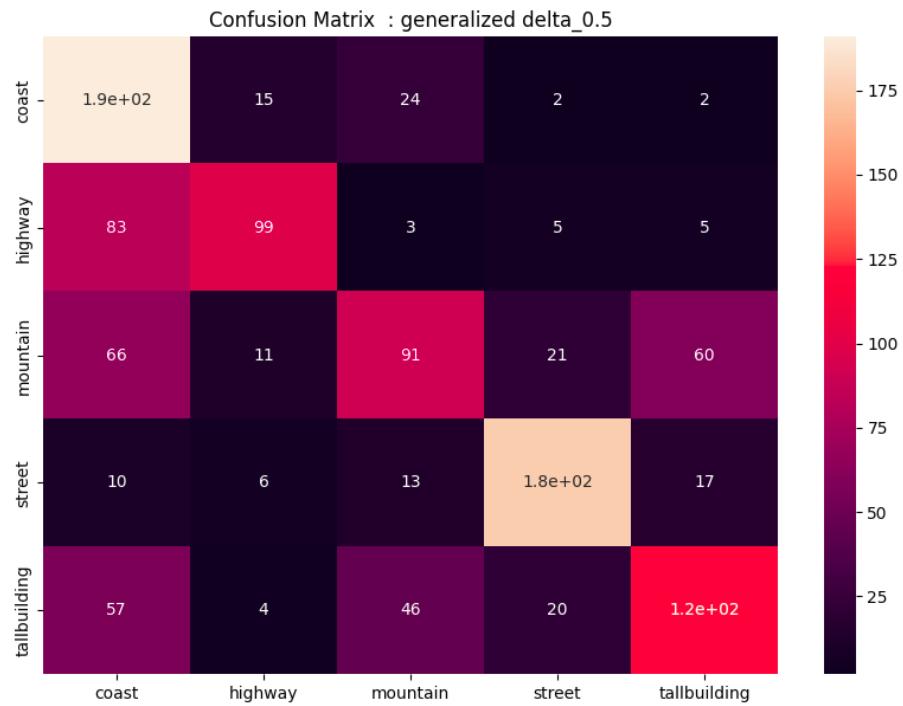


### 3. Generalized Delta Rule - (momentum = 0.5)

#### a. Confusion Matrix (Test Data)

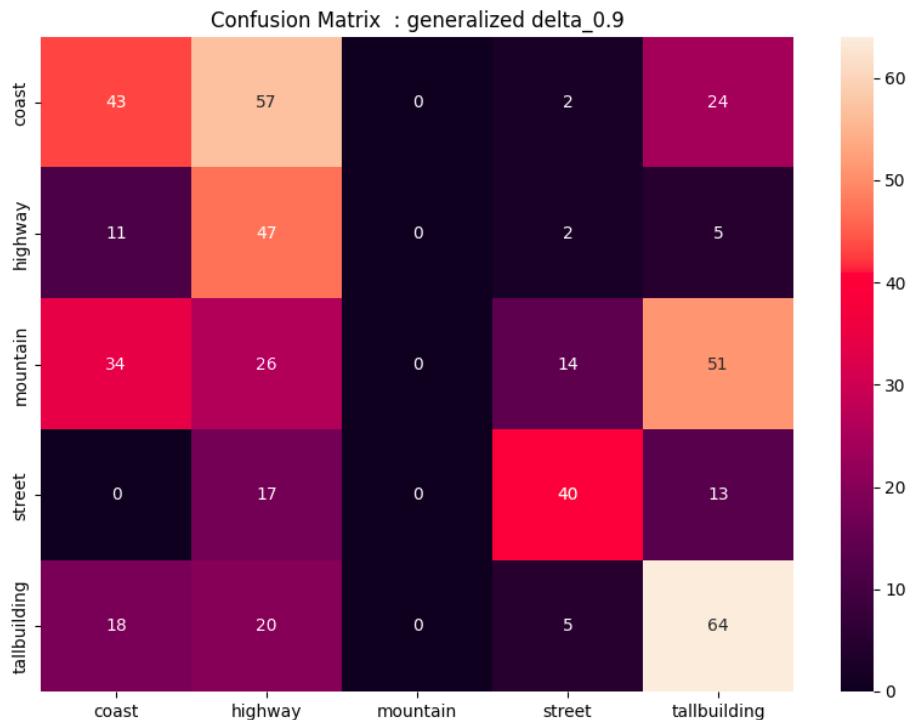


b. Confusion Matrix (Training Data)

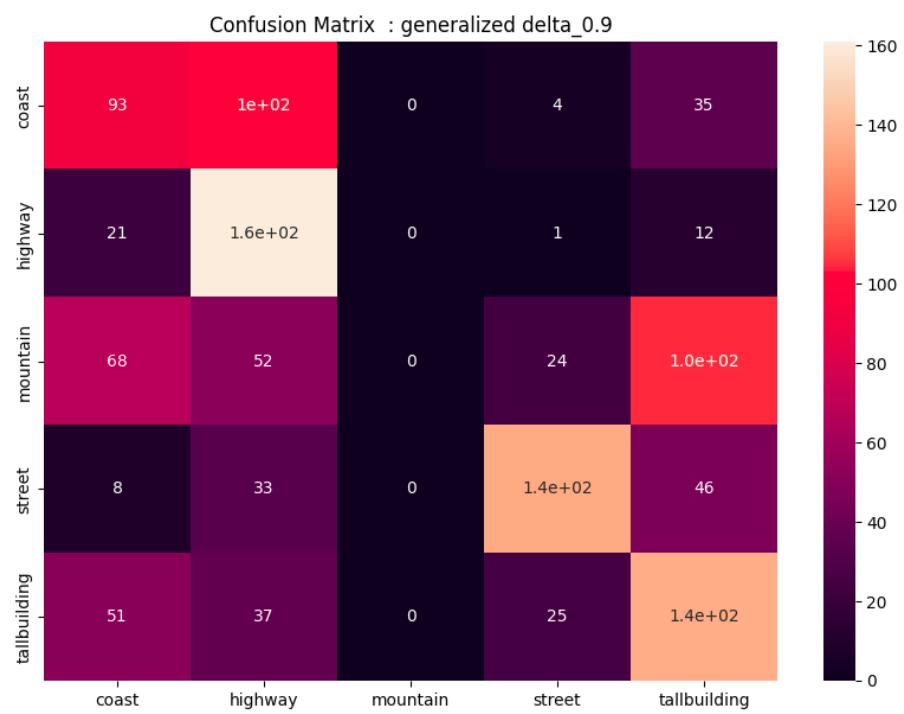


#### 4. Generalized Delta Rule - (momentum = 0.9)

a. Confusion Matrix (Test Data)

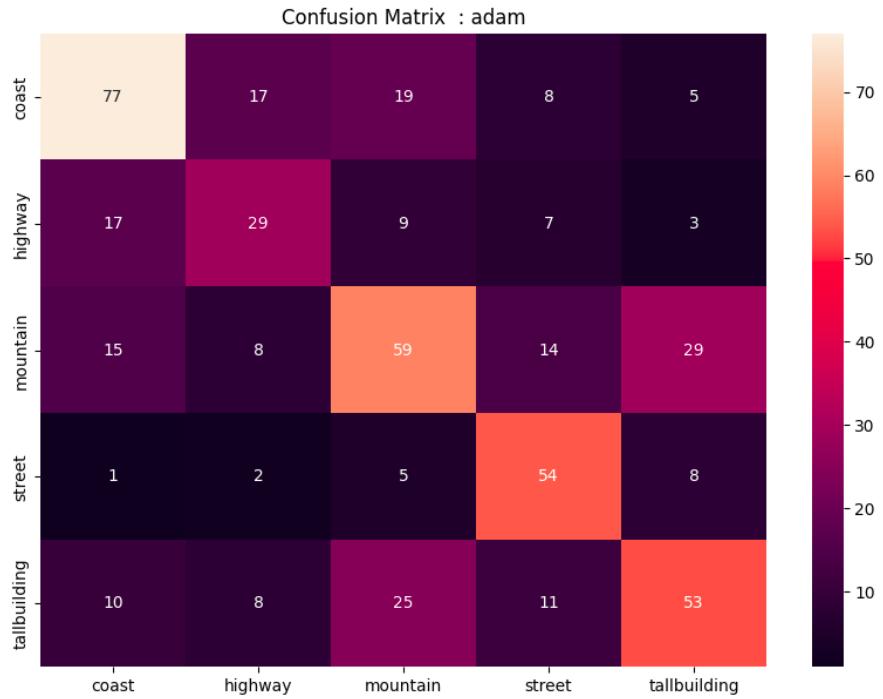


b. Confusion Matrix (Training Data)

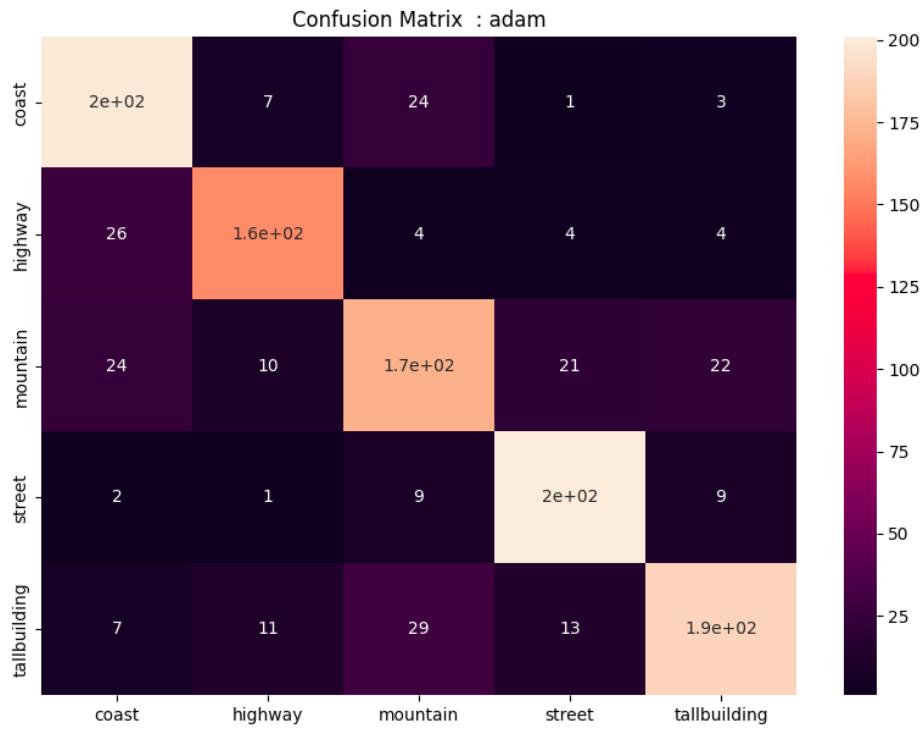


## 5. Adam Optimization

### a. Confusion Matrix (Test Data)



### b. Confusion Matrix (Train Data)

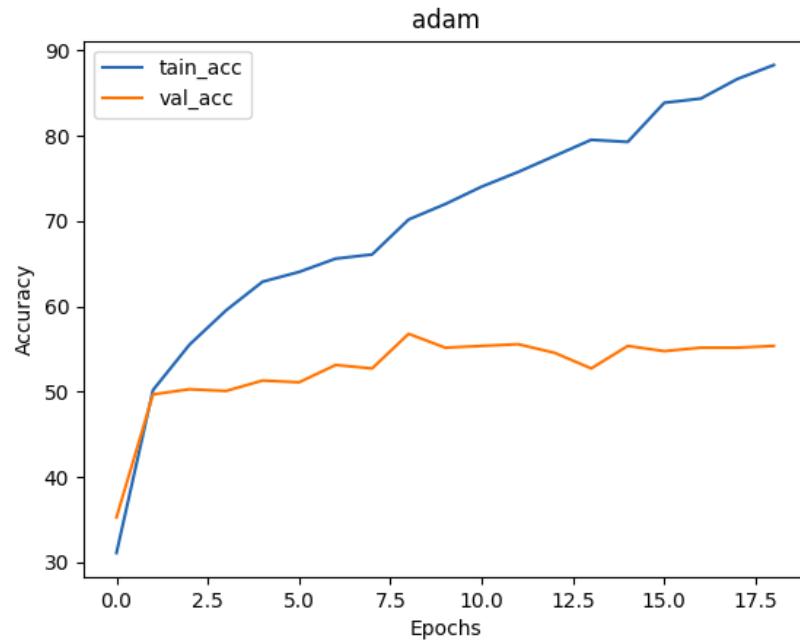


### 3.4 Epochs for Convergence -

**Convergence has been taken to be the number of Epochs for which the validation accuracy has a 1% change in accuracy for 3 continuous epochs.**

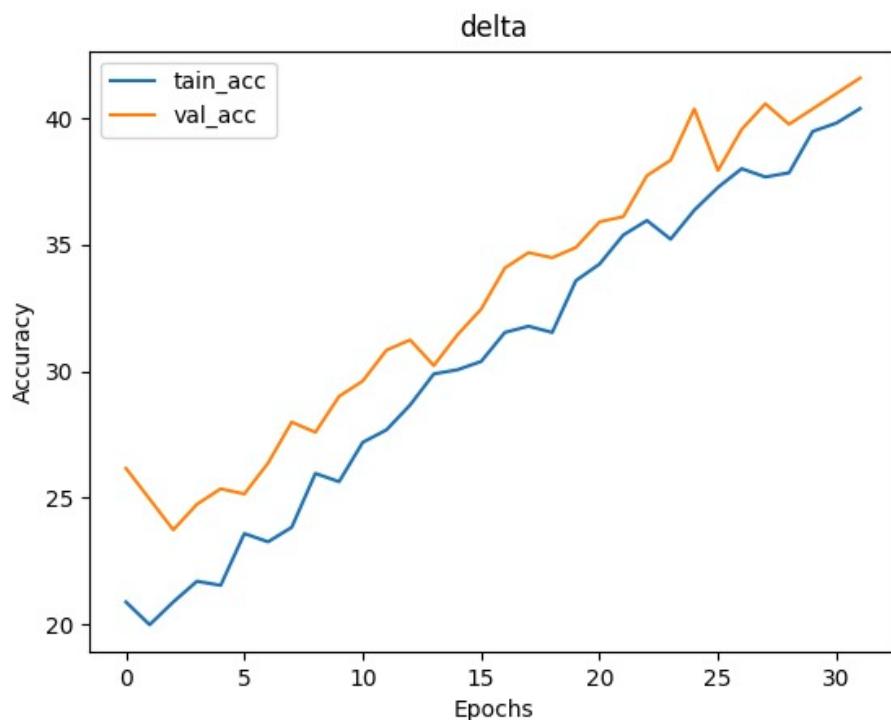
Weight Update Rule	Epochs
Adam	18
Delta	31
Generalized Delta Momentum: 0.1 0.5 0.9	35 39 oscillating , hence do not converges

## 1. Adam



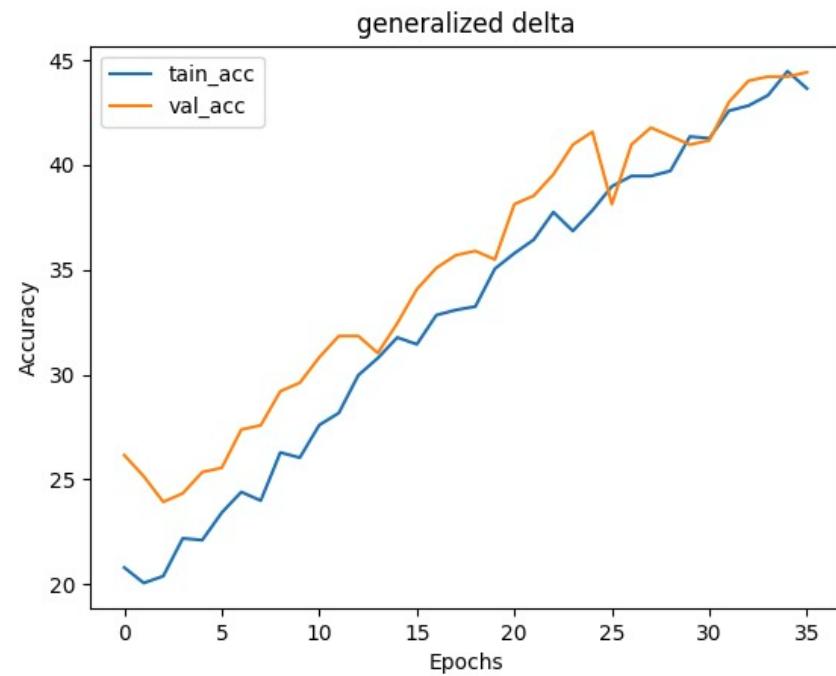
Final Accuracy - 56.17%

## 2. Delta



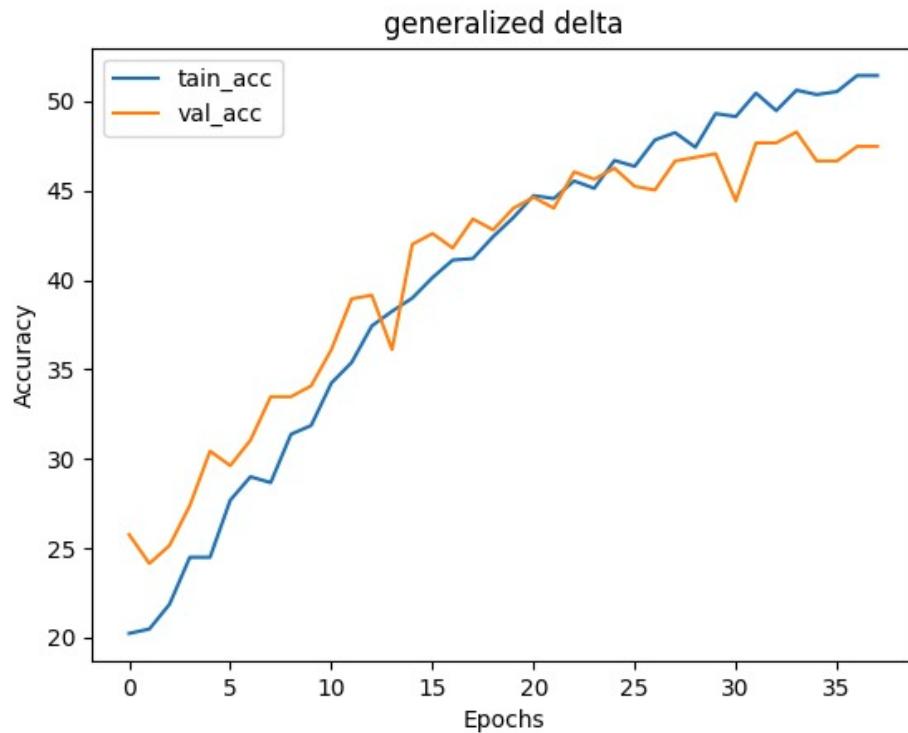
Final Accuracy - 29.006%

### 3. Generalized Delta (0.1)



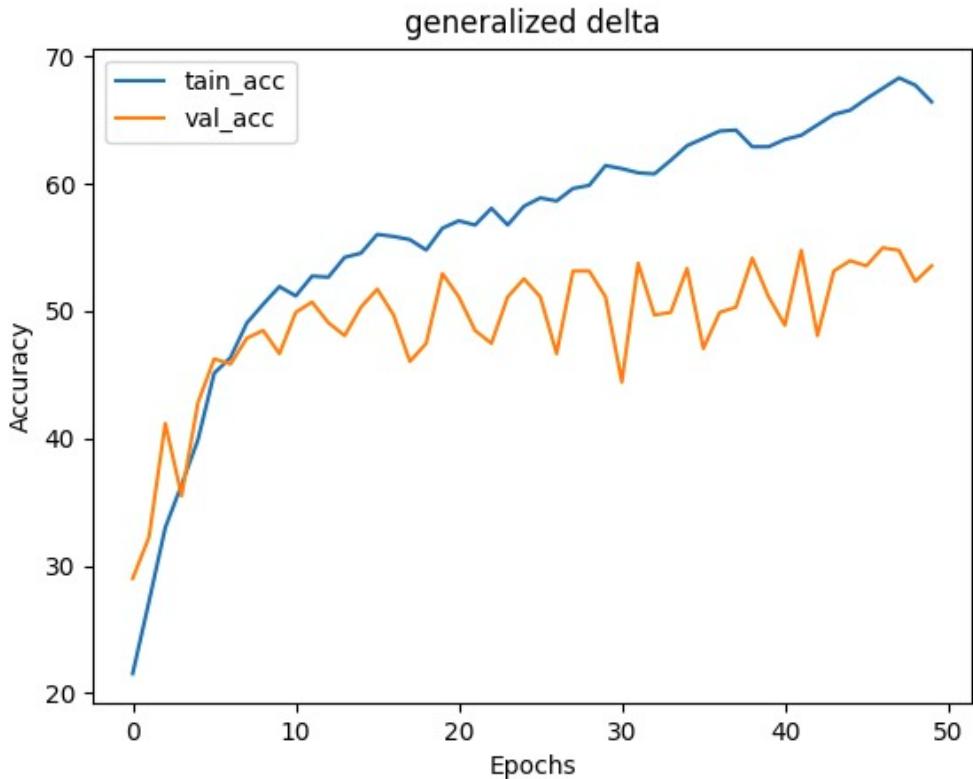
Final Accuracy - 46.85%

### 4. Generalized Delta (0.5)



Final Accuracy - 48.25%

## 5. Generalized Delta (0.9)



Final Accuracy - 48.98%

Observations:

- Adam converges the quickest at 18 epochs and provides accuracy of . Adam is focussed on fastest computation time.
- Delta is able to converge in 31 epochs since the path taken by the algorithm is full of noise. Thus, SGD uses a higher number of iterations to reach the local minima. Due to an increase in the number of iterations, the overall computation time increases. Additionally, it is possible for SDG to get stuck at local minima thus we get very low accuracy of 29.006%
- Generalized Delta with Momentum of 0.1 is able to get much better accuracy of 46.85% with 35 epochs
- Generalized Delta with Momentum of 0.5 is able to get better accuracy of 48.25% with 39 epochs around the same as momentum 0.1 but with better accuracy.

- Generalized Delta with Momentum of 0.9 takes greater than 50 epochs to get an accuracy of 48.98%, the accuracy oscillates around the minima due to the higher value of momentum, thus doesn't converge.