Multi-layer Perceptrons and self driving car

CSL 603: Machine Learning

KESHAV GARG

PANKAJ VERMA

11/6/2017

Multi-layer Perceptrons and self driving car

CSL 603: Machine Learning

Introduction

In this Lab, We have implemented multi-layer perceptron, which is a class of feedforward artificial neural network.

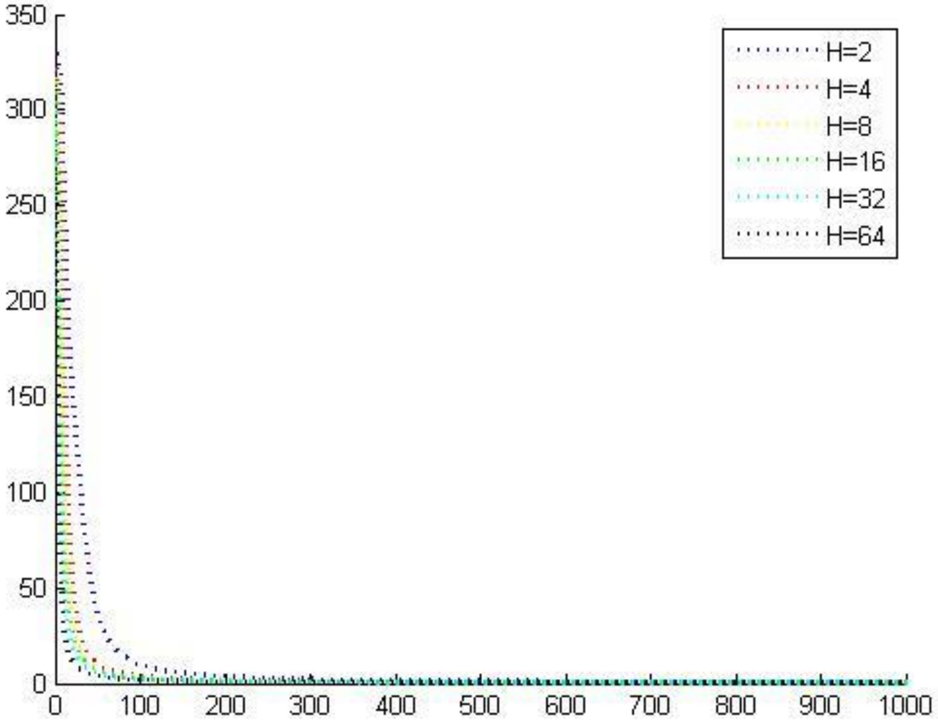
In task 1, our MLP consists of three layers-input layer, hidden layer and output layer. We have used sigmoid function on the inner layers and softmax function on the output layer as the activation functions. For calculating errors, we are using cross entropy function.

In task 2, MLP is of four layers with two hidden layers. We have used the stable sigmoid function on the inner layers and no activation function on the output layer. We are using squared error function to calculate the training and validation errors.

Observations

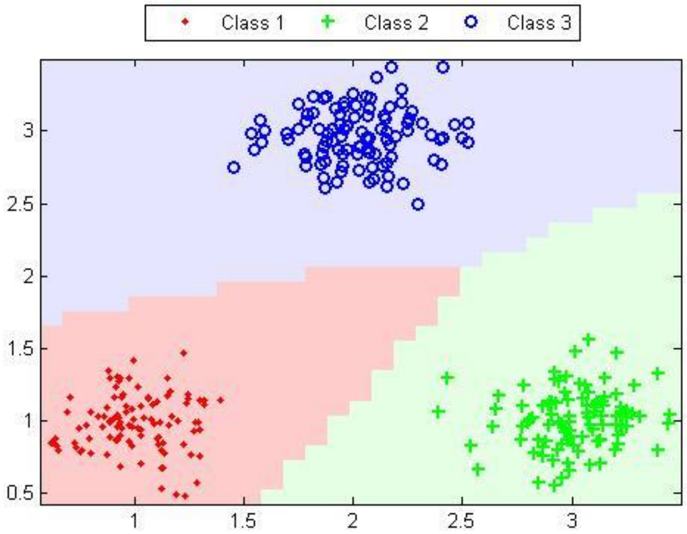
Task 1

A. Hidden layer units =2, 4, 8, 32, 64

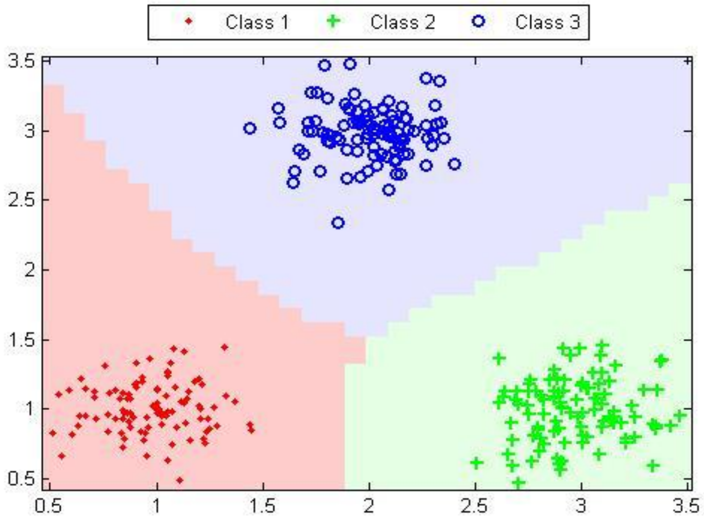


As the no. of hidden layer units increases, training error tends to converge to zero faster.

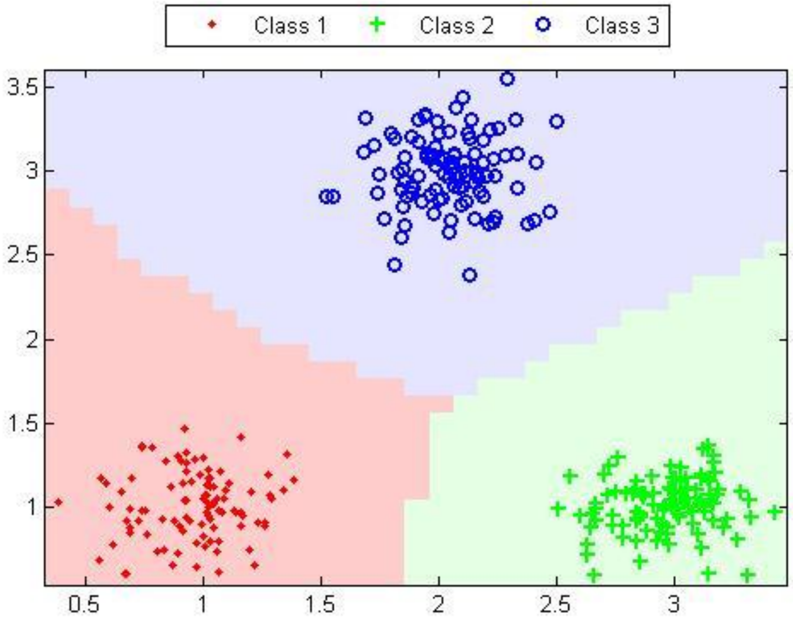
B. Decision boundary for H=2



C. Decision boundary for H=4



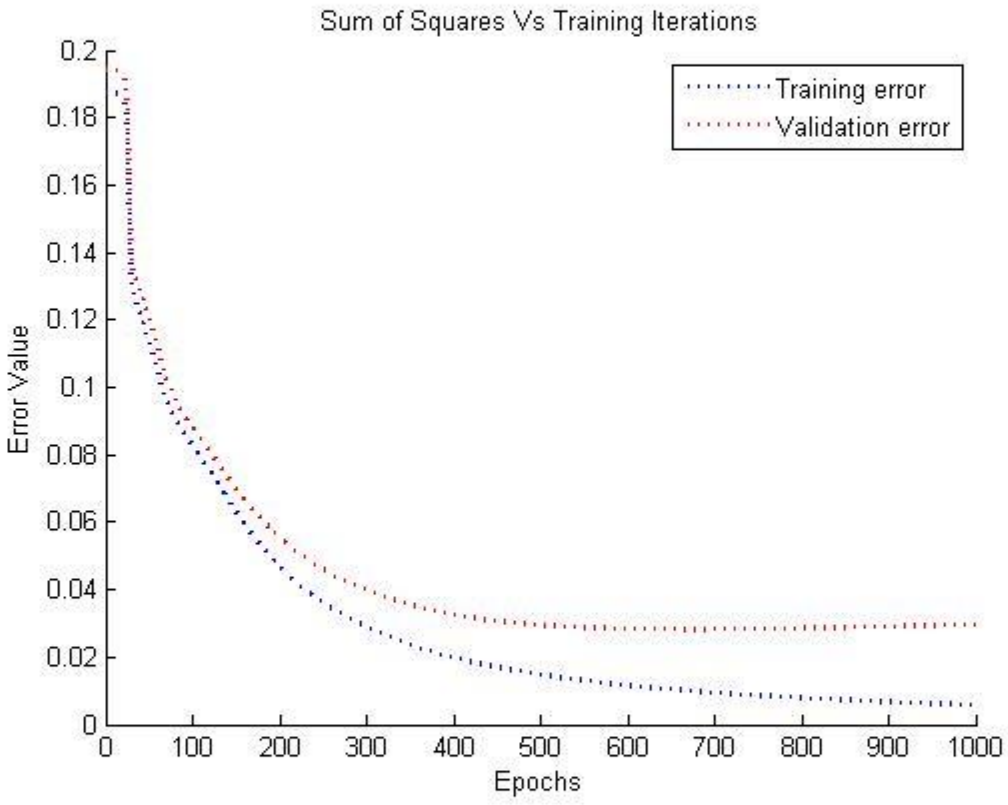
D. Decision boundary for H=64



As the hidden layer units increases, decision boundary seems to adjust itself in a better proportion respective to each class.

Task 2

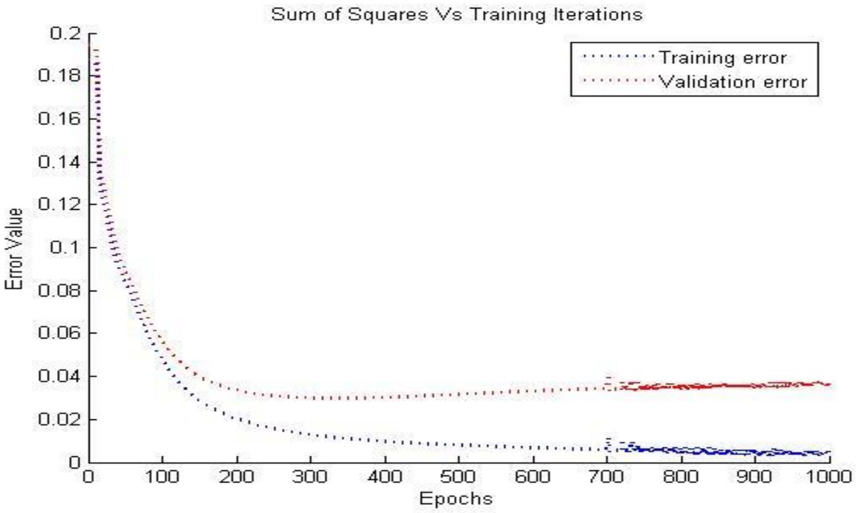
1.) 5000 Epochs, learning rate=0.01, batch size=128.



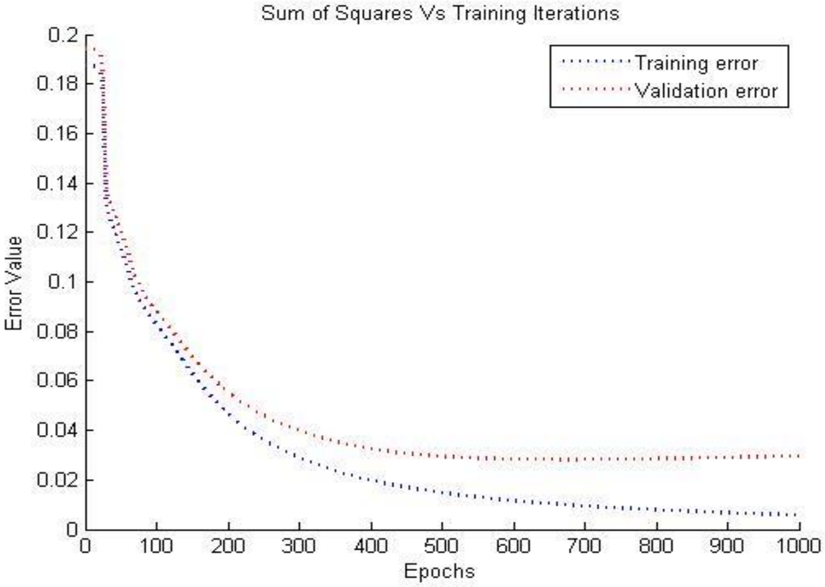
With more epochs, it converges to a minimum value and then remains constant.

2.)

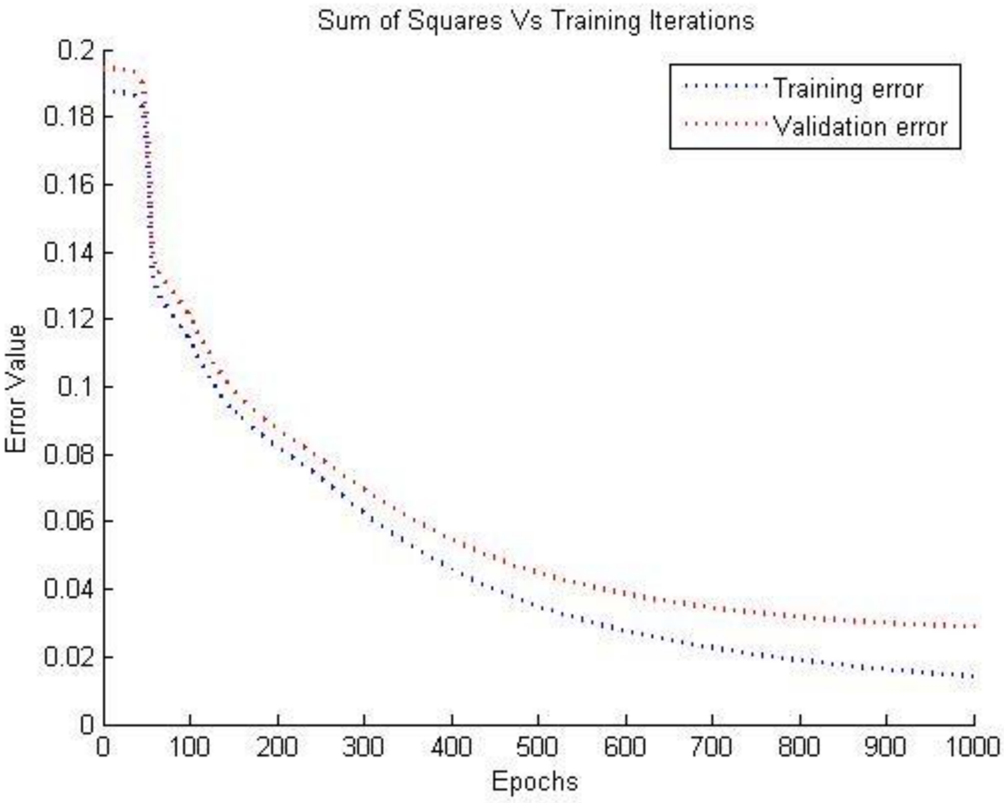
A. 1000 Epochs, learning rate =0.01, batch size = 32



B. 1000 Epochs, learning rate =0.01, batch size = 64

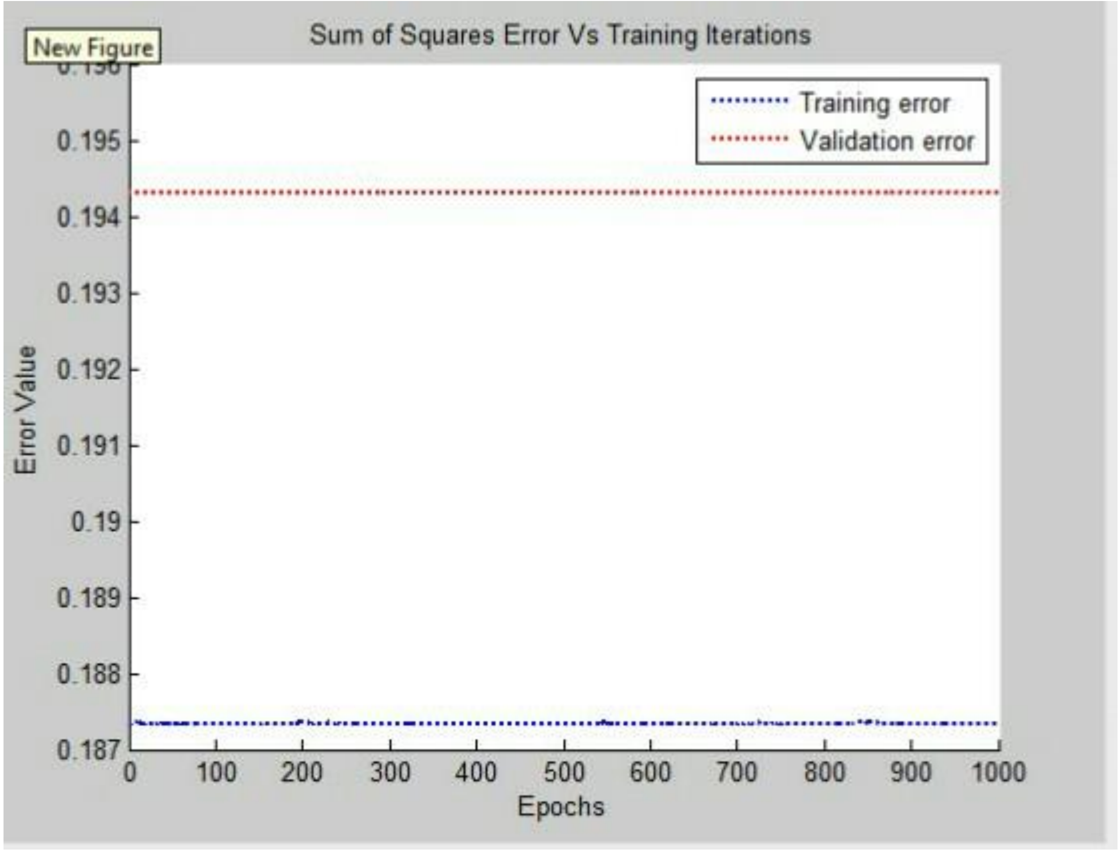


C. 1000 Epochs, learning rate =0.01, batch size = 128



As batch size increases, error takes more epochs to converge to zero but the processing time is small for larger batch size as compared to the smaller ones.

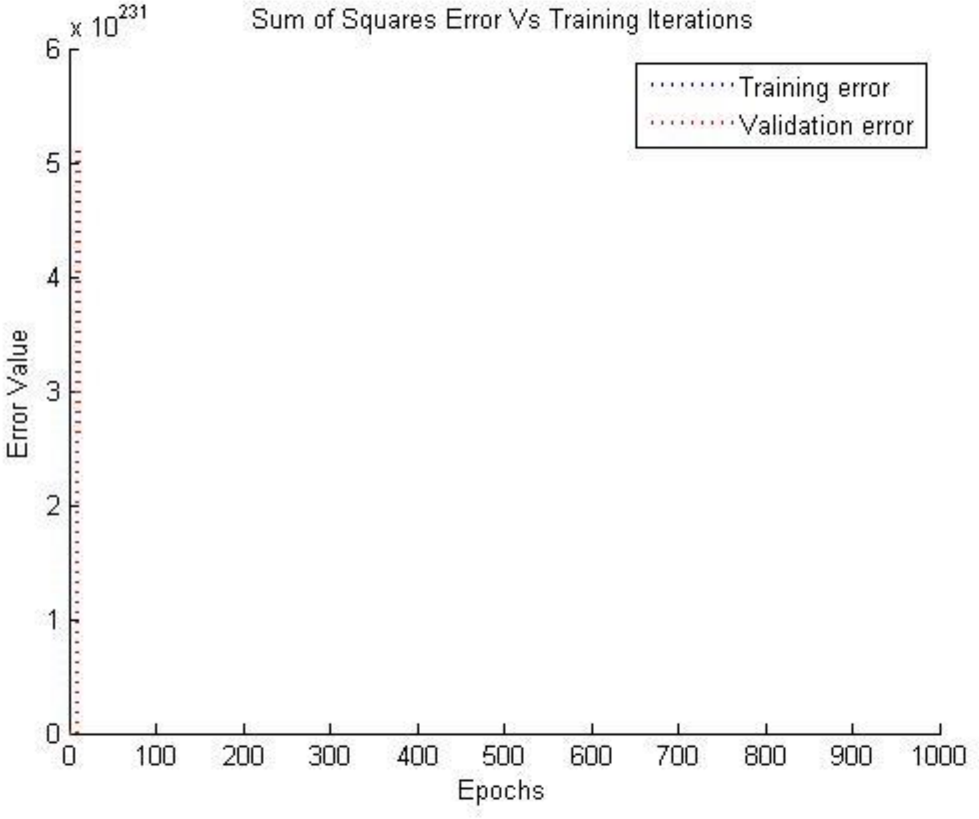
3.) 1000 Epochs, learning rate =0.001, batch size = 64, dropout probability of the first, second and third layer = 0.5



With dropout percentage 0.5, our error remains almost constant

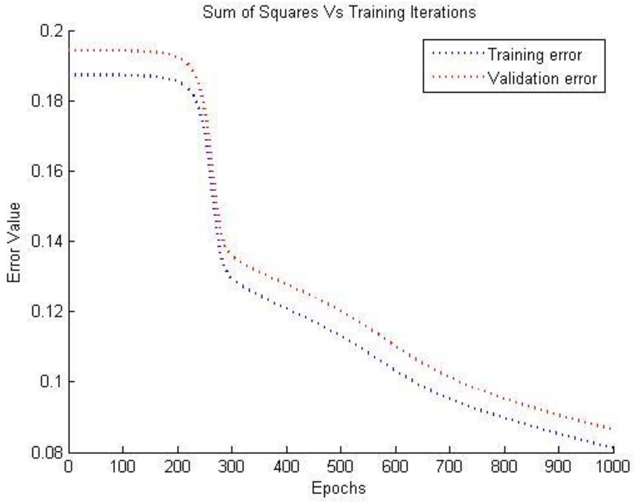
4.)

A. 1000 Epochs, learning rate =0.05, batch size = 64



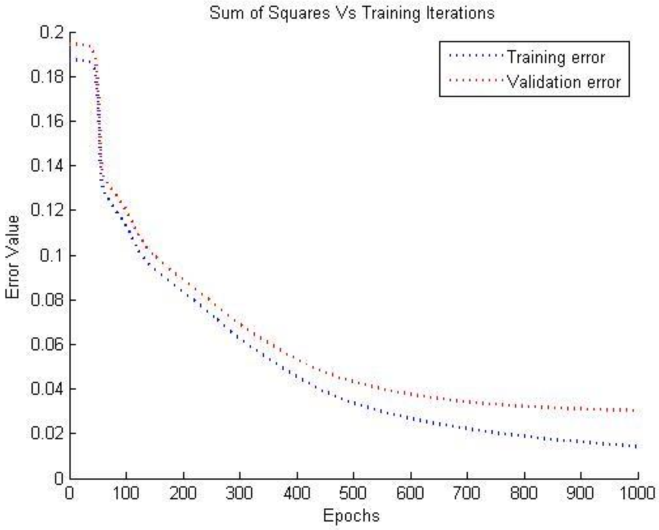
As the value of learning rate is quite high, it overshoots the minima and reaches infinity within initial 20 epochs.

B. 1000 Epochs, learning rate =0.001, batch size = 64



Learning rate is too small, so error remains almost constant in the starting but converges as iterations increases.

C. 1000 Epochs, learning rate =0.005, batch size = 64



Learning rate is small but our error converges to zero as iterations increases.