CSL603:Machine Learning

Self Driving Cars using Multi Layer perceptron Model

Aditya Tiwari(2016csb1029)

Introduction

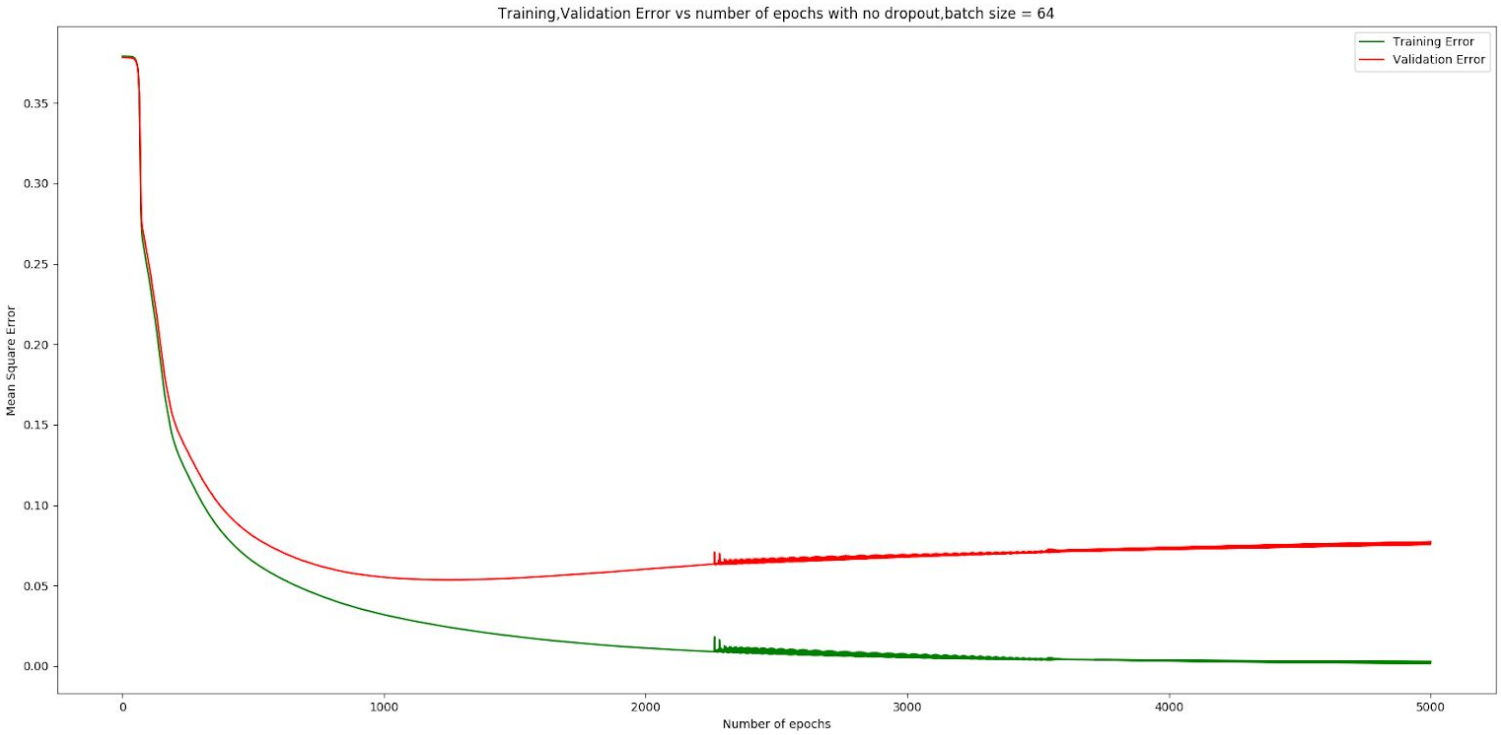
The assignment was to given 32x32 image of roads, task was to predict the steering angle.

DataSet Used:Preprocessed dataset from Udacity's simulator

The model used for the purpose was a basic Artificial Neural Network with some regularization(dropouts)

Experiment 1:

"A plot of sum of squares error on the training and validation set as a function of training iterations (for 5000 epochs) with a learning rate of 0.01. (no dropout,minibatch size of 64)."



Training Error starts from 0.38 and goes on decreasing to 0.002

Validation Error starts from 0.38 and decreases to 0.05 and then starts increasing

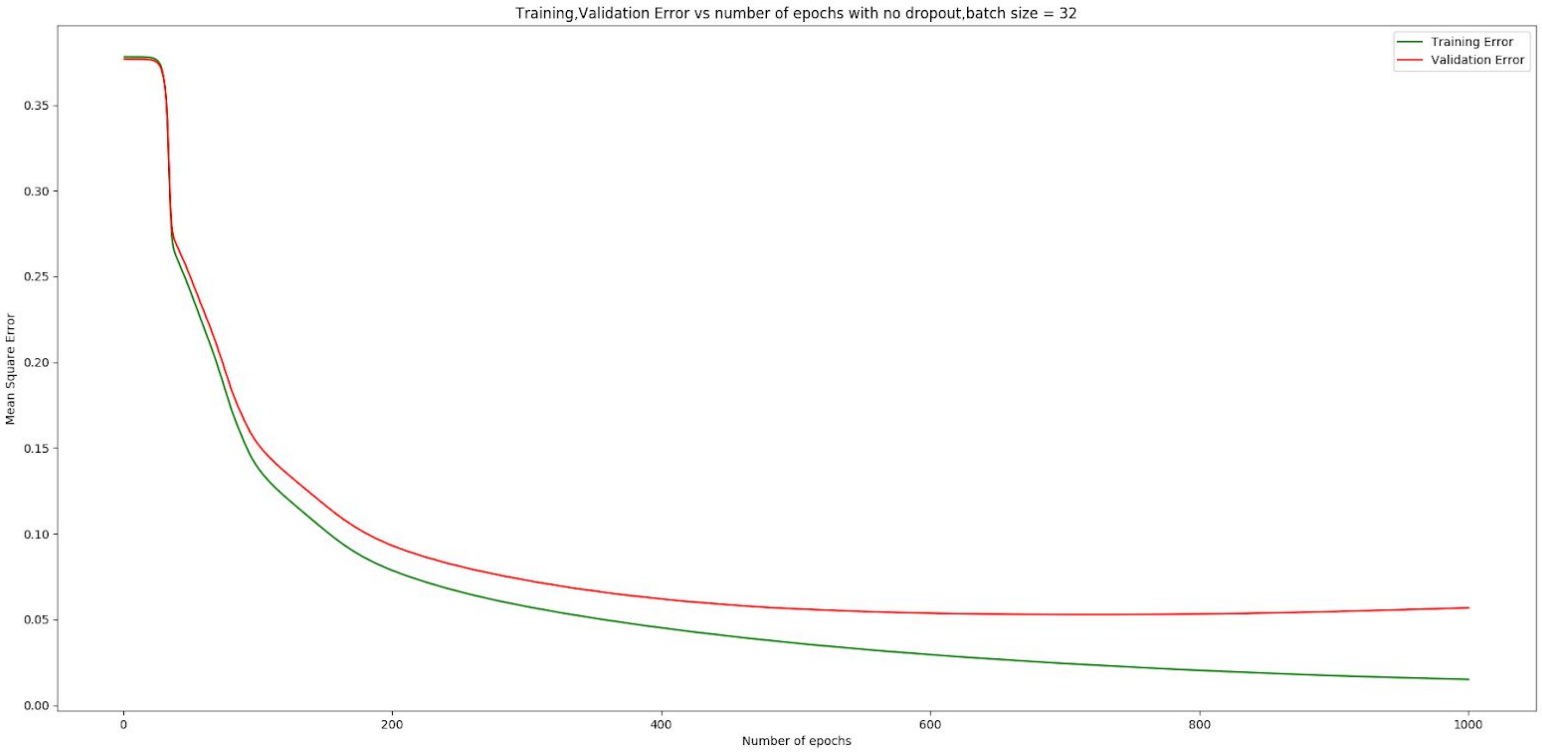
This Increase in error indicates the overfitting in the model

Also the graph is very smooth and there are very less ups and downs. I believe this is due to The error vector that I passed in the back propagation is divided by the batch size. I saw a couple of my friend's graph and they had more ups and downs because they didn't divide it. By dividing the error factor,I have reduced the learning rate by the batch size and hence the gradient descent is smoother

Note:Learning rate was 0.01

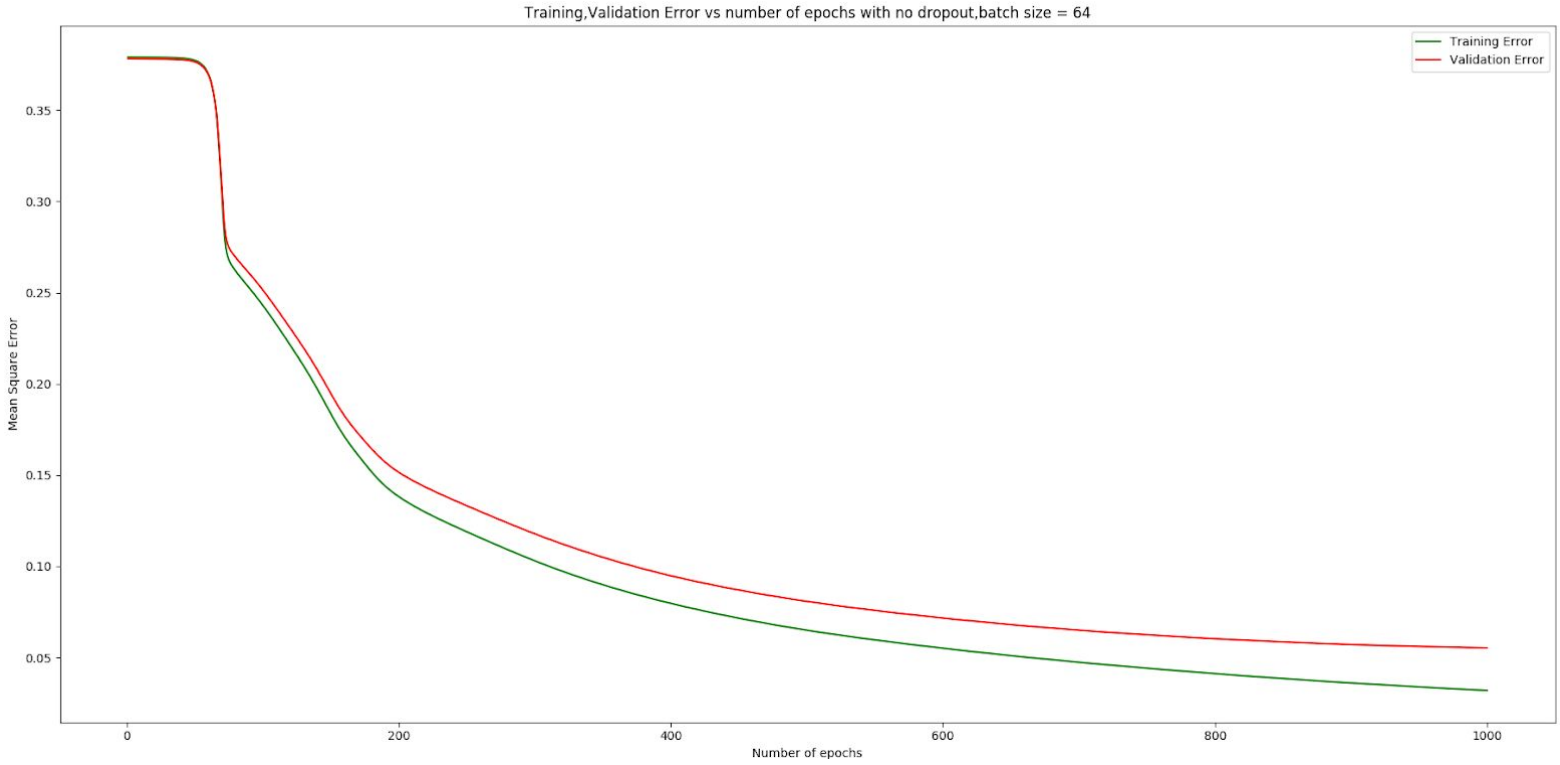
Experiment 2:

"A plot of sum of squares error on the training and validation set as a function of training iterations (for 1000 epochs) with a fixed learning rate of 0.01 for three minibatch sizes – 32, 64, 128."

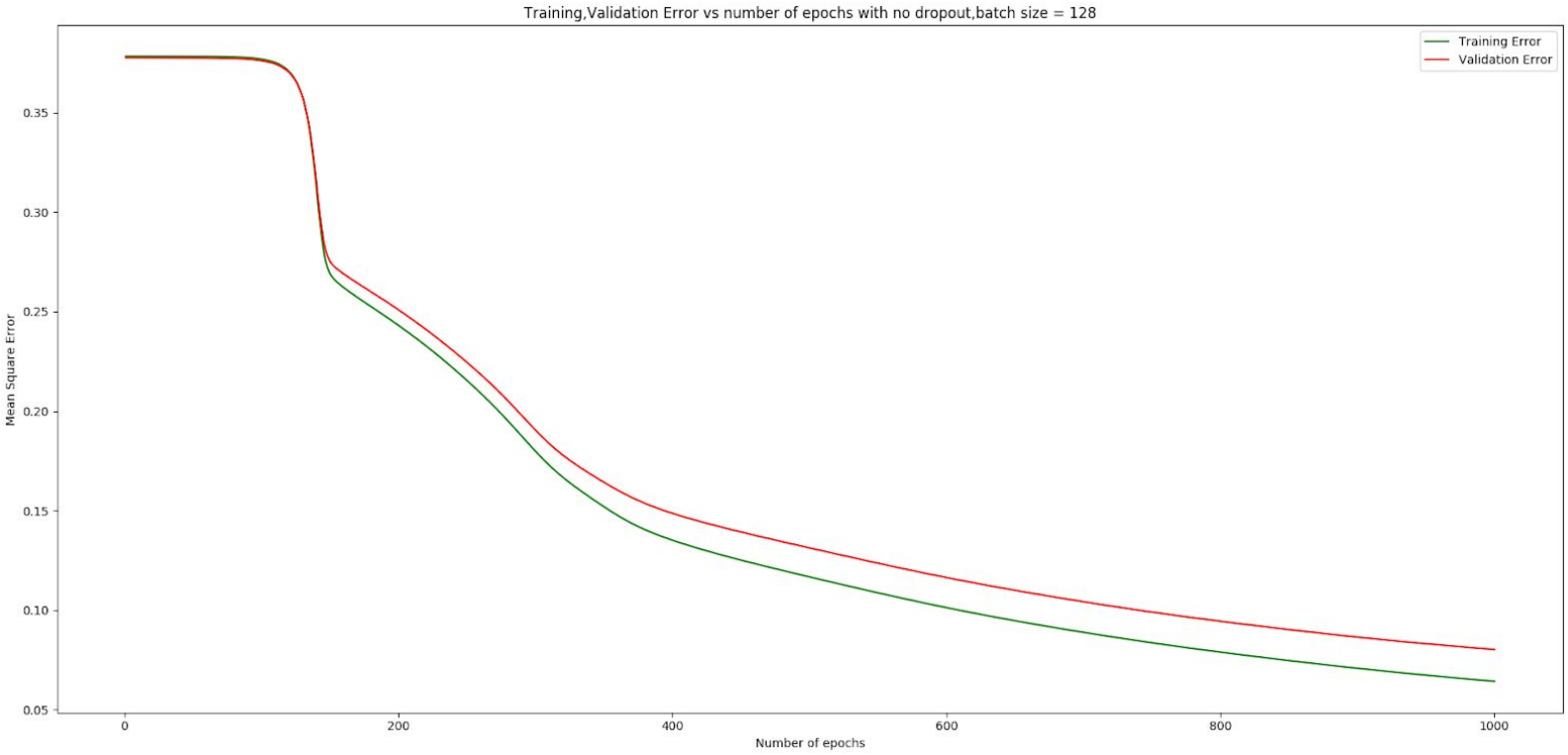


Observation:

The Error starts from 0.38 and goes till 0.03 for Training Error where validation error goes starts increasing after 0.05 indicating overfitting. The curve is very Smooth though



The error starts from 0.38 and goes till 0.03 for Training and a little more for Validation(Overfitting)



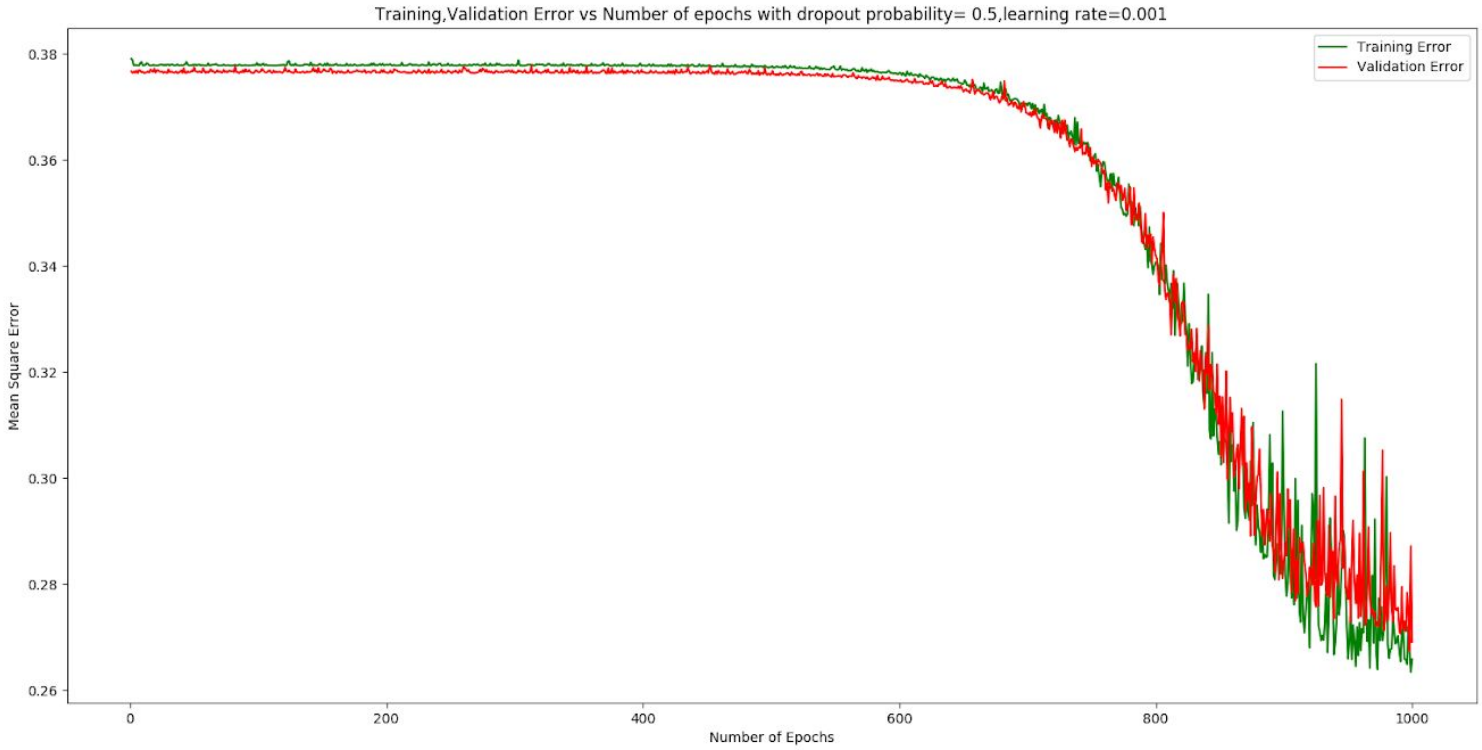
The curve starts from 0.38 and goes till 0.07 for Training and a little greater for Validation.

Inference: The gap between the validation and Training decreases as the batch size increases. I.e as the batch size is increased overfitting(this is like regularization)

Note:Learning rate is 0.01 in all three cases

Experiment 3:

"A plot of sum of squares error on the training and validation set as a function of training iterations (for 1000 epochs) with a learning rate of 0.001, and dropout probability of 0.5 for the first, second and third layers."



Insight:

The Error starts from 0.38 and slowly decreases with ups and downs in between indicating presence of local minima in gradient descent , and decreases till 0.26

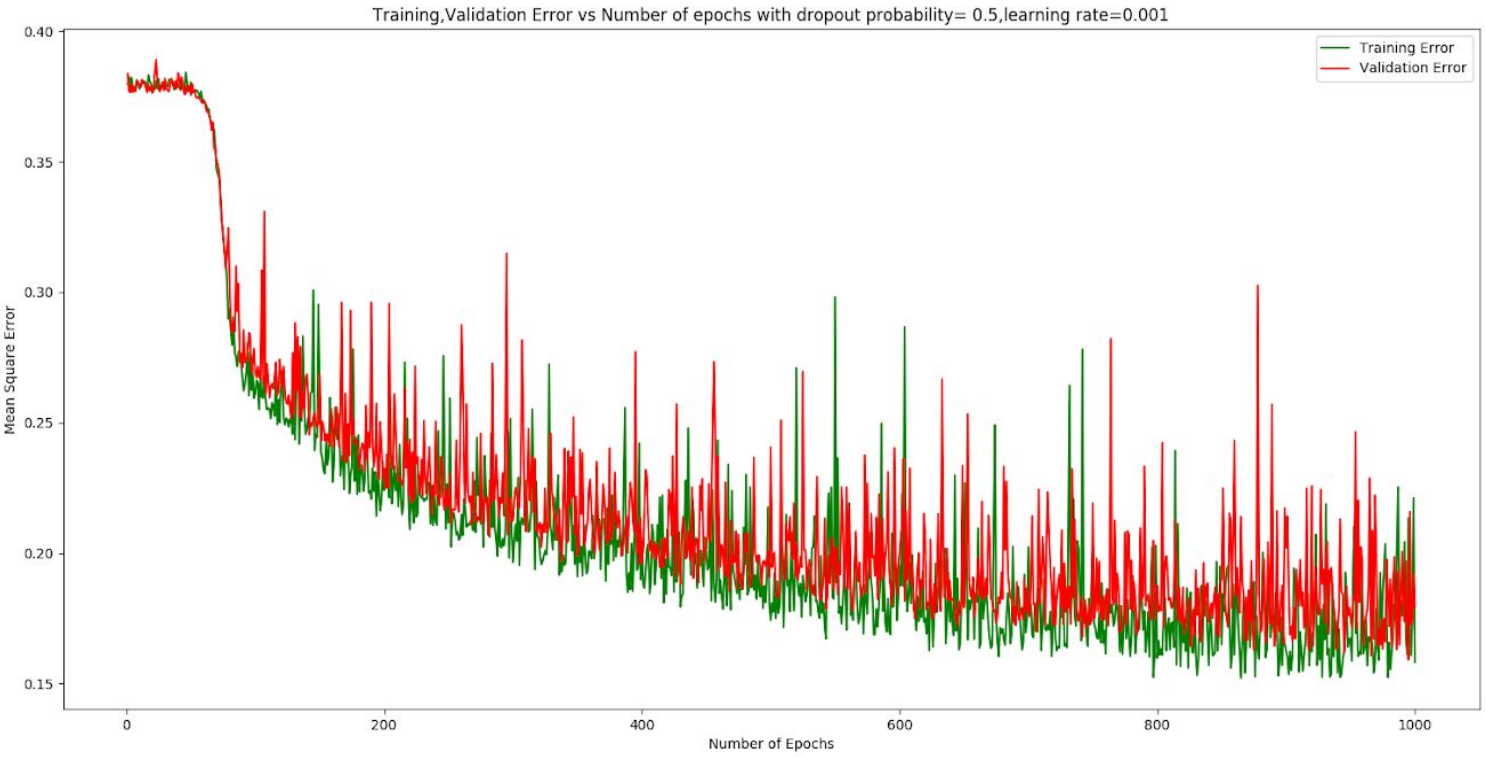
Validation Error is initially less than training but eventually becomes greater than former,maybe due to too much regularization.

“We dropped half of the nodes in each layer”

Alter Experiment:

My hunch was maybe learning rate is too small and the convergence is very slow, so I increased the learning rate to 0.01

And the result is:



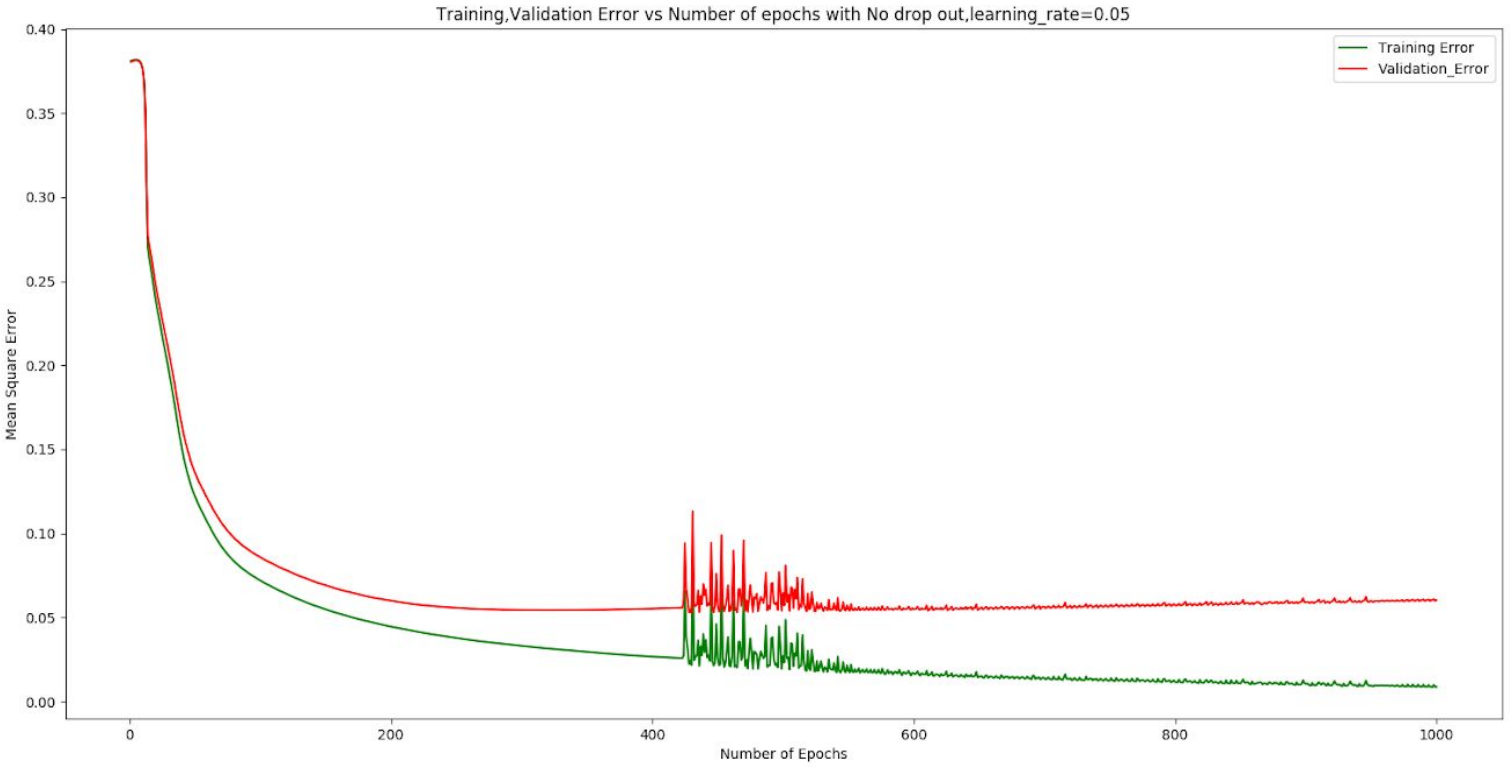
Error started from 0.37 and converged further below to 0.15. The spikes are the effect of increase in learning rate indicating bouncing over local minimas in gradient descent

Note: The batch size taken was 64 in both the experiments

Experiment 4:

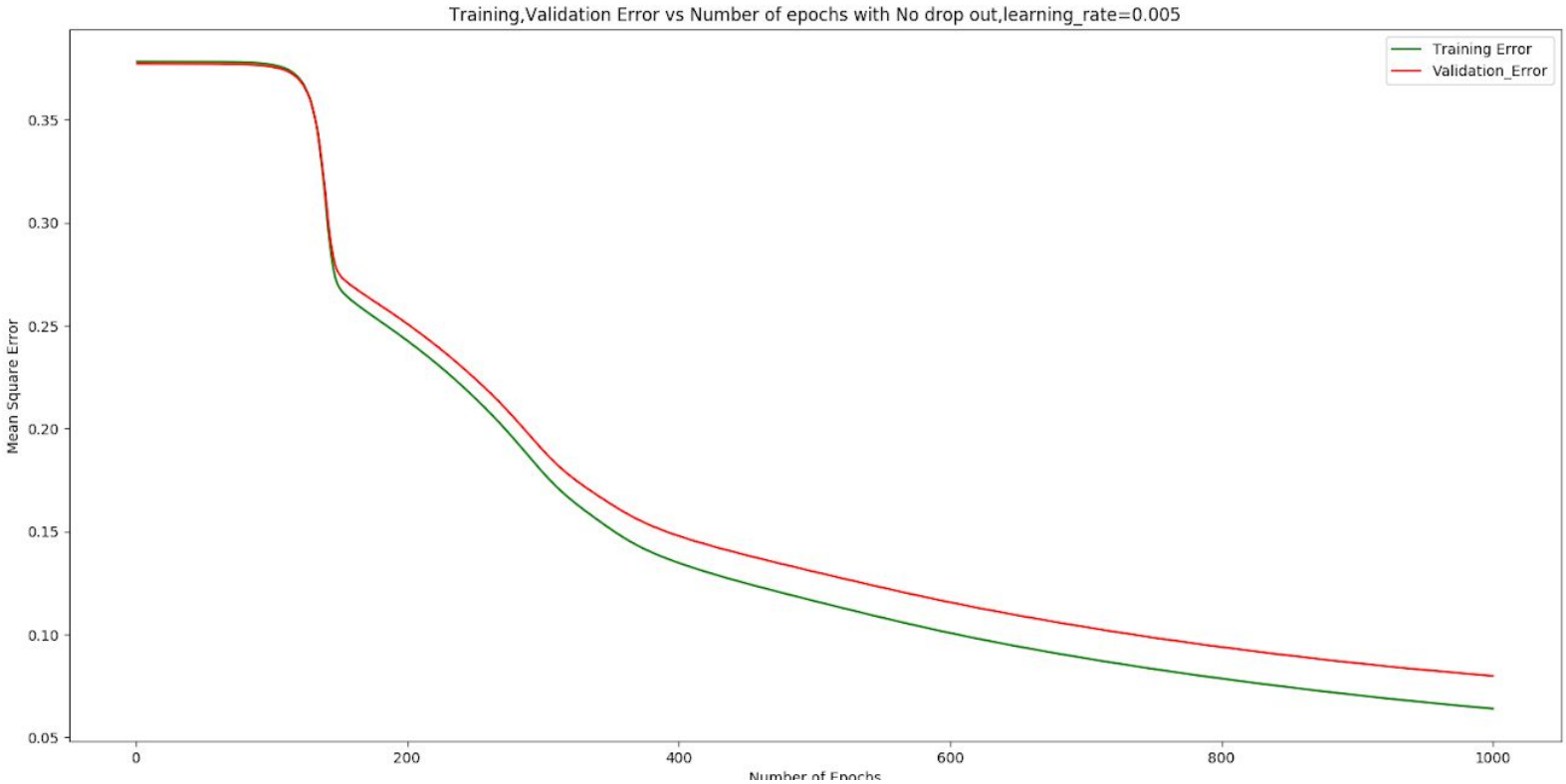
"A plot of sum of squares error on the training and validation set as a function of training iterations (for 1000 epochs) with different learning rates – 0.05, 0.001, 0.005 (no drop out, minibatch size – 64)"

Observation:

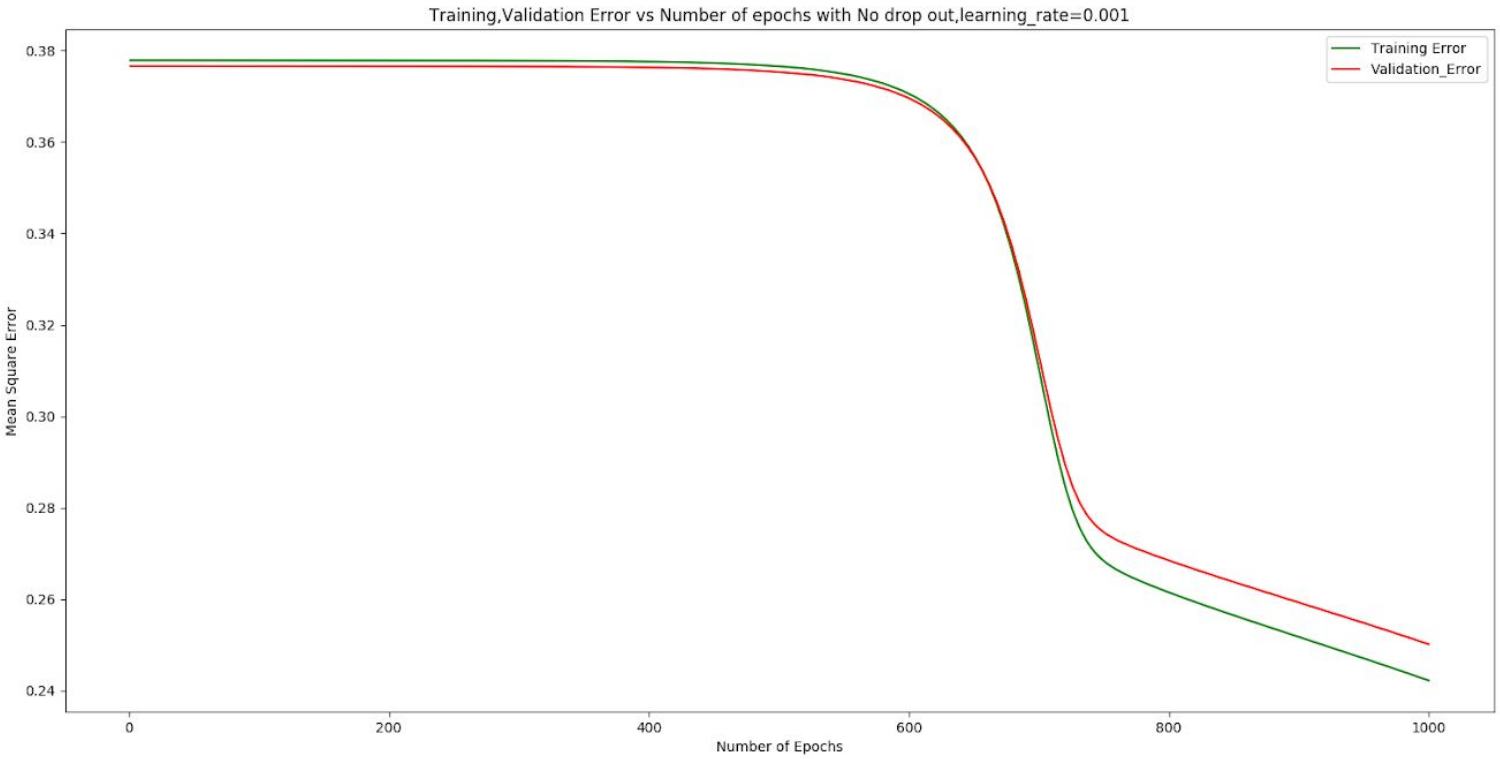


Error starts from 0.37 and goes till 0.03 and validation error is always greater than training error. This is result of overfitting model.

The Zig Zags in between is due to high learning rate=0.05



Error Starts from 0.43 and goes till 0.05 and is very smooth because of smaller learning rate 0.005 which is 10 times smaller than 0.05



The Graph is much more smoother compared to learning rate 0.05. Here learning Rate is 0.001 which is 50 times smaller than 0.05 ,but due to that descent is only from 0.38 to 0.24. Maybe more iterations would have converged it to further smaller value but Validation Error would still be greater than Training again indicating overfitting

Inference: 0.005 would be the best value of learning rate as the descent is smooth and it converges to a pretty good value

Note : Batch size was 64 in all the three experiments