# Assignment-3

*Abhiraj S. Kanse (16D07006)*
*Devanshu Singh Gaharwar(16D070042)*

## *A) MNIST Dataset*

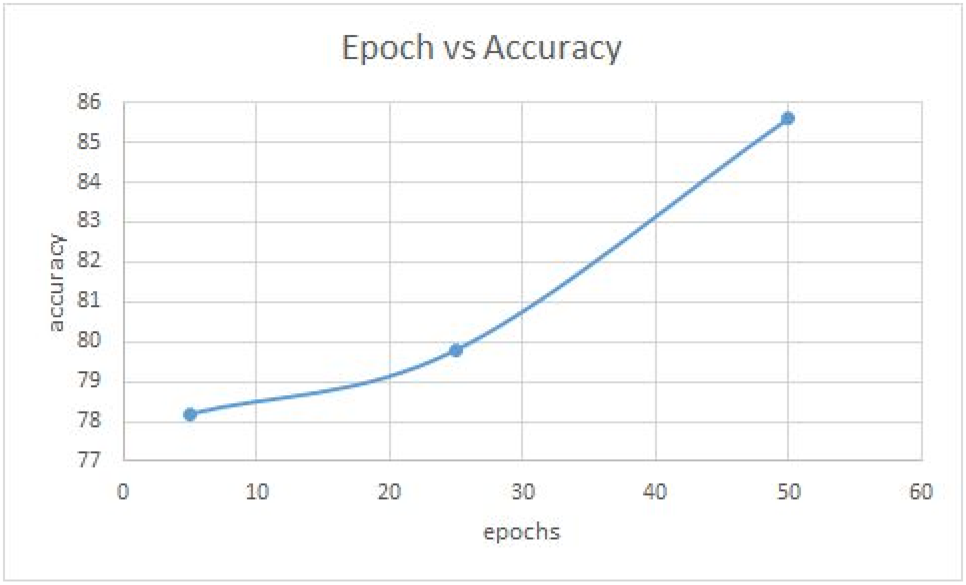
For this dataset we did changes only in the train_mnist file as mentioned. We tweaked various parameters and the observations are compiled below.

1) Number of layers

The observation was pretty obvious in this case. As we increase the number of hidden layers the accuracy increases from 78% to 97.85% and the training time increases due to increased complexity. To see this please uncomment the hidden layer lines keeping all other parameters at there default value.

2) Epoch

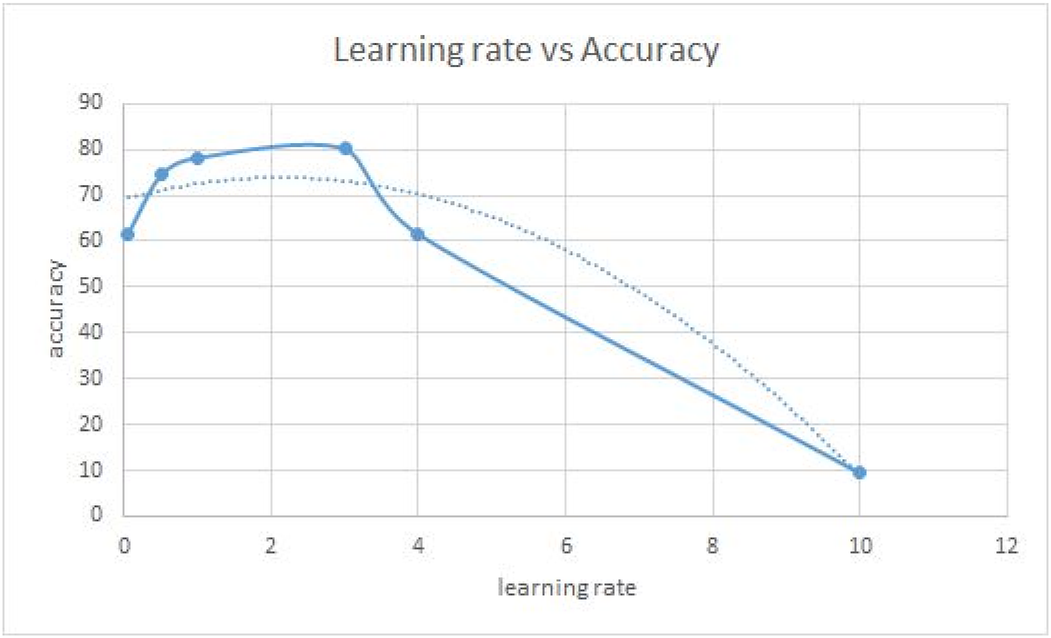
As we increase the number of epochs the accuracy increases significantly although the training time also increases in proportion.
(Observations are reported for single layer with learning rate = 0.5)
Epochs = 5 gives peak accuracy of 78.2%
Epochs = 25 gives peak accuracy of 79.8%
Epochs = 50 gives peak accuracy of 85.6%



This observation is reasonable since epoch represent the number of passes over the complete training set and as we increase it the accuracy is bound to increase as our model fits better and better, although if we increase it to a very large value overfitting will start to occur. And the increasing time observation is obvious since we are increasing the number of passes over dataset.

3) Learning Rate

We see that there exists an optimal learning rate i.e. at this learning rate the accuracy obtained is maximum.
(Observations are reported for single layer with Epochs = 5)
Learning Rate = 0.05 gives peak accuracy of 61.5%
Learning Rate = 0.5 gives peak accuracy of 74.43%
Learning Rate = 1 gives peak accuracy of 78.2%
Learning Rate = 3 gives peak accuracy of 80.12%
Learning Rate = 4 gives peak accuracy of 61.5%
Learning Rate = 10  gives peak accuracy of 9.4%

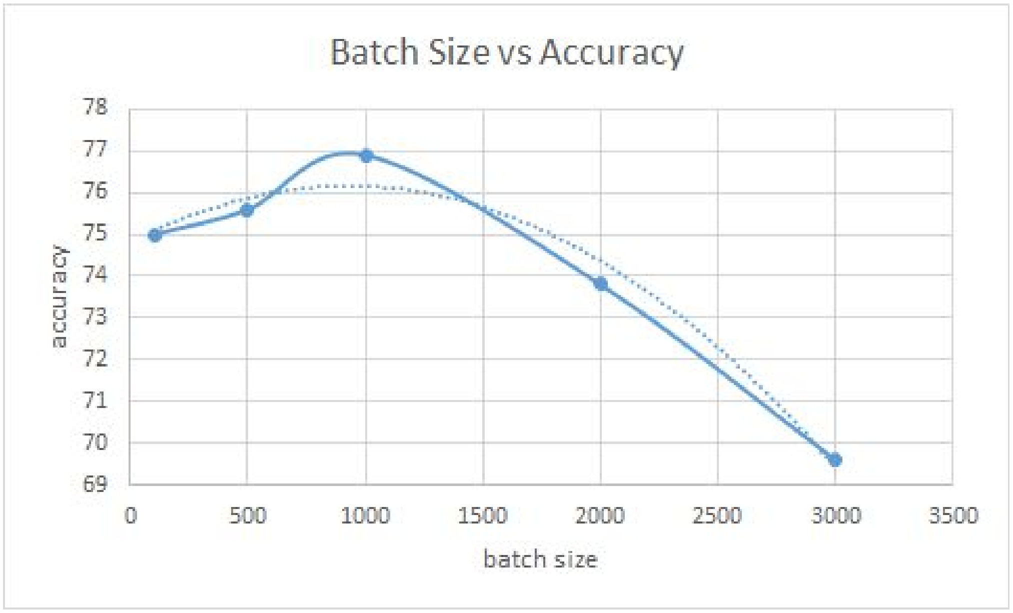


This observation is also justifiable because if we have very small learning rate then the gradient-descent algorithm will not converge by the time of the termination. And thus we don't even reach our optimal value. And if we have a very large learning rate we will take larger steps and thus the time to reach the optimal value decreases but our precision decreases since we have to take larger steps and thus we don't reach very close to the optimal value. And as we

See that in the extreme case when learning rate is increased to 10 the accuracy decreases as much as 9.4% and the training time also decreases significantly.

4) <u>Batch Size</u>

As we increase the number of epochs the accuracy increases slightly initially. (Observations are reported for single layer with Epochs = 5)
Batch Size = 100 gives peak accuracy of 75.0%
Batch Size = 500 gives peak accuracy of 75.6%
Batch Size = 1000 gives peak accuracy of 76.9%
Batch Size = 2000 gives peak accuracy of 73.8%
Batch Size = 3000 gives peak accuracy of 69.6%



We know that batch size refers to the number of training examples utilized in one iteration. As we increase the batch size the the number of iterations thus decreases since the number of training examples are fixed. Thus we see that there is a trade-off between accuracy in a single iteration and the number of iteration. And due to this we see that initially our accuracy increases with batch size and later it starts to decrease. The training time has no observable difference with change in batch size.

5) Activation Function

We see that among the 3 activation functions Relu, sigmoid and tanh; Relu gives the optimum results keeping the value of other parameters fixed. And tanh and sigmoid are giving very close values of accuracy. This can be argued because we know that both the tanh and sigmoid functions have same nature of curve, which saturates later on, but in the case of Relu we know that it doesn't saturate.

6) Regularizer

(Results are reported for L2 regularizer)

We see that using the optimum value of hyperparameter our accuracy slightly Increases from the base accuracy. The observations are as follows.

(Observations are reported for single layer with Epochs = 5 and batch size = 500)

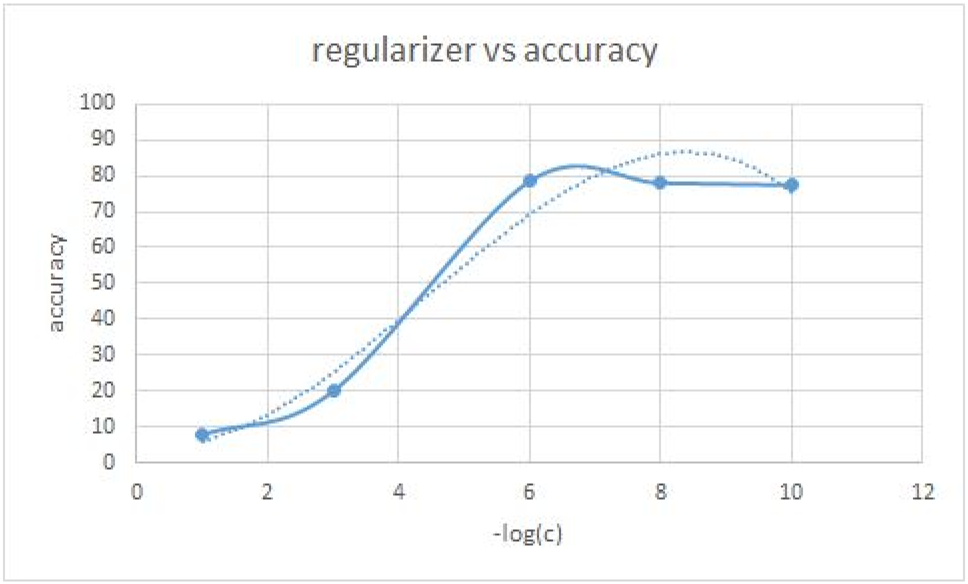
Hyperparameter = 1e-1 gives peak accuracy of 7.9%

Hyperparameter = 1e-3 gives peak accuracy of 20.03%

Hyperparameter = 1e-6 gives peak accuracy of 78.57%

Hyperparameter = 1e-8 gives peak accuracy of 78.02%

Hyperparameter = 0 gives peak accuracy of 77.44%



This graph is as expected initially. We see that there exists an optimal value of the hyperparameter above which the accuracy decreases due to increases weightage of regularizer and below which the accuracy decreases due to possible overfitting because of very less weightage of regularizer.