ELL 409 ASSIGNMENT 3 REPORT

Date of submission: 13/11/2022

Group Members:

Perugu Krishna Yadav	2020EE10521
Muvva Srija	2020EE30605
Devarakonda Ronith Kumar	2020EE10486

Q_1)

We know that following three activation functions are widely used

1) Sigmoid
2) Tanhx
3) ReLU

If sigmoid is the activation function, then disadvantages at higher values gradients will be vanished, and sigmoid is not symmetric about origin.

And if we check by keeping activation function as sigmoid, we get 50% accuracy on test data

With tanh and ReLU as activation functions we get around 90% accuracy on test data. Here we are using activation function ReLU in our code to train data because if we use tanh gradient vanishes at higher values.

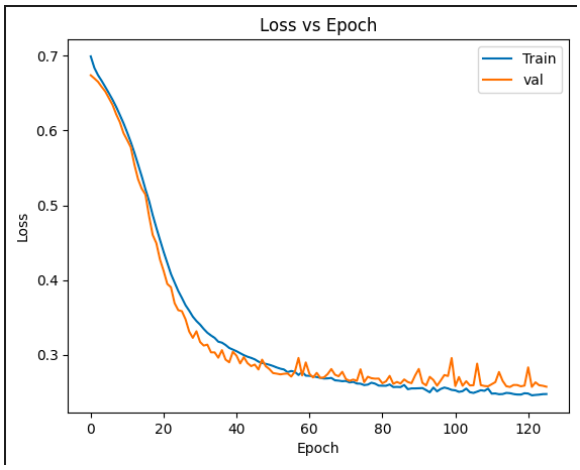
Here we are using fully connected neural network and initially we build model with 2 hidden layers and then we have assigned a variable for it so that we can test for different hidden layers

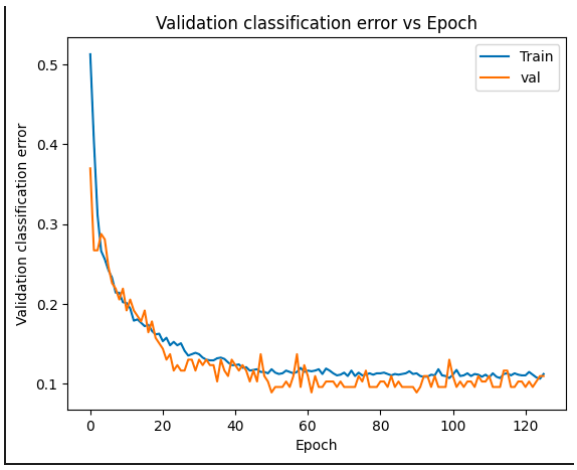
Here we are defining stopping criteria as monitor = val_loss, patience = 10 it will run for a maximum of 200 epochs, and in the middle if the loss is almost same for 10 epochs. And we are using glorot_uniform as initialization method and kept the momentum of stochastic gradient as 0.5 after observing the accuracy by changing the momentum from 0.5 to 0.9 and kept the learning rate update schedule as 0.02 randomly.

Q_2)

Initially we have set learning rate randomly and changed the learning rate based on the fluctuations in validation loss vs epoch graph and on accuracy of the test data with trained model.

Here we found the optimised case with learning rate as 0.02 with around 89% accuracy on test data

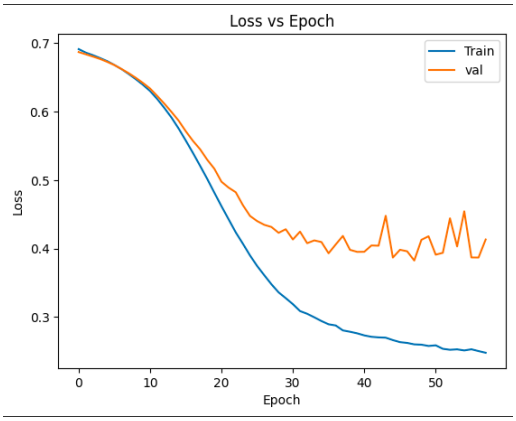




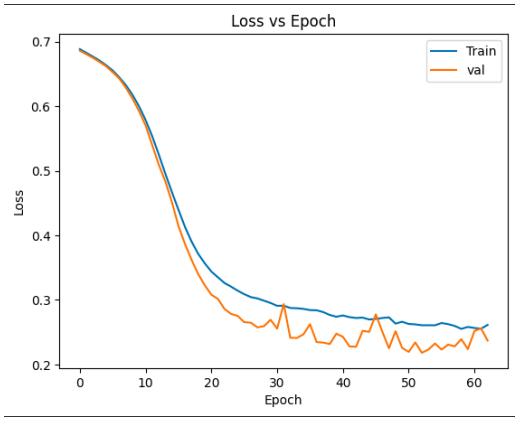
Q_3)

Learning rate update schedule	Accuracy on Validation data
0.02	0.8219178318977356
0.03	0.8835616707801819
0.05	0.8698630332946777
0.009	0.8972602486610413
0.005	0.8493150472640991

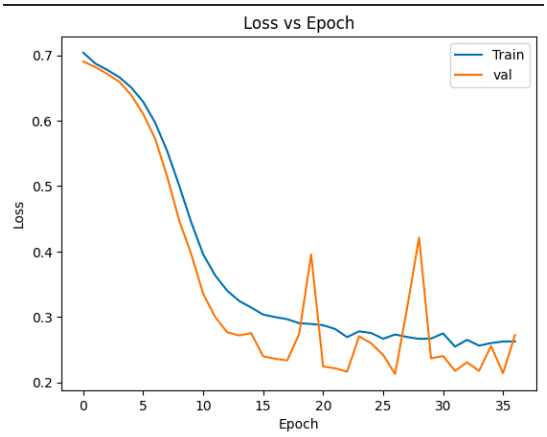
Learning rate update schedule – 0.02



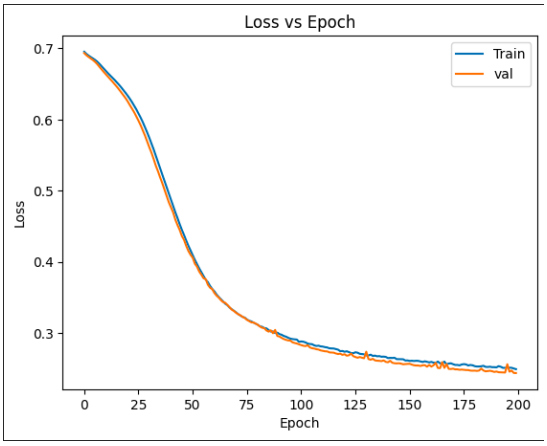
Learning rate update schedule -0.03



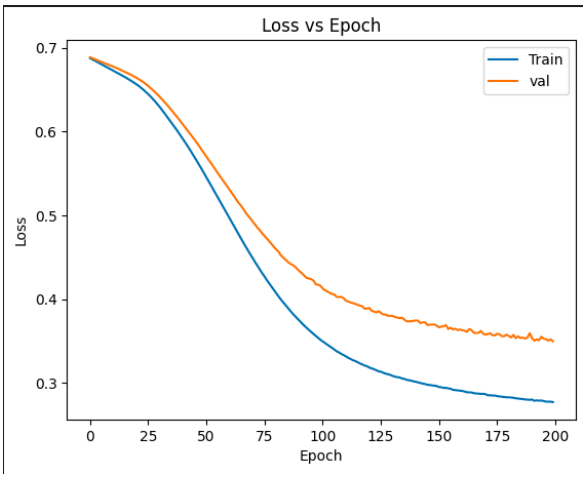
Learning rate update schedule -0.05



Learning rate update schedule -0.009



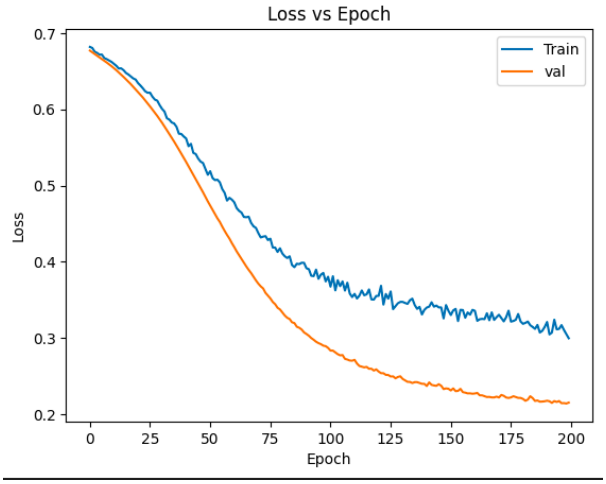
Learning rate update schedule-0.005



Here we observe that accuracy is better and even fluctuations are less if learning rate update schedule is at around 0.009 and if we increase or decrease from 0.009 we observe either more fluctuations or less accuracy.

Q_4)

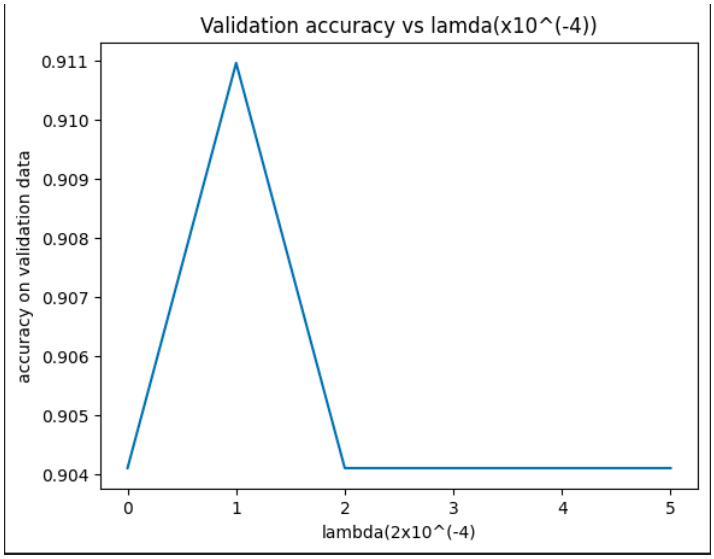
If we consider 34% dropout then our accuracy has been improved to 0.9246575236320496 with parameters same as above optimised values (i.e., learning rate=0.02 and learning rate schedule =0.009) and also we can observe that when we use the dropout the fluctuations in graph of loss of train data vs epoch increased and loss of validation data vs epoch is decreased



Q_5)

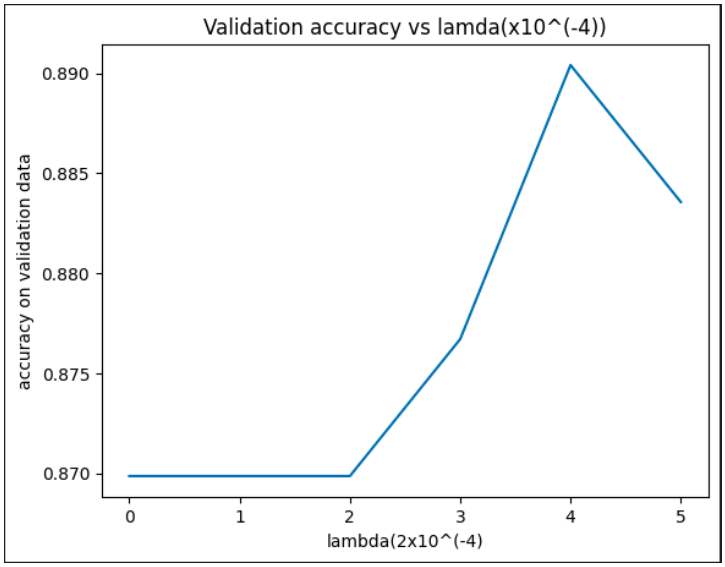
Here we used L1 weight regularizer that is lasso regression with dropout and other parameters same as previous case. Below is the graph of accuracy on validation data vs lambda used in regularisation.

We observe that at lambda=0.0002 we get optimised accuracy



Here we used L2 weight regularizer that is ridge regression with dropout and other parameters same as previous case. Below is the graph of accuracy on validation data vs lambda used in regularisation

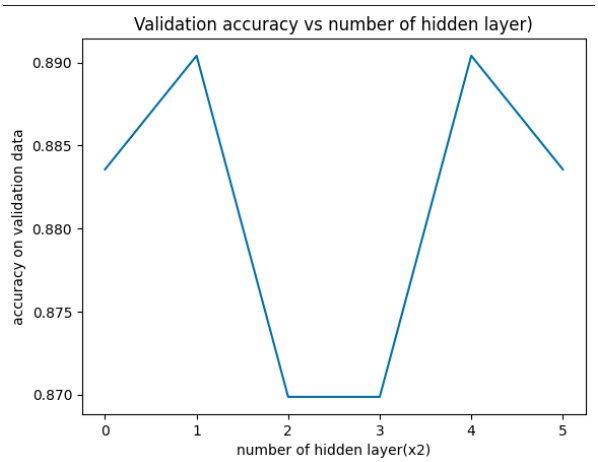
We observe that at lambda=0.0008 we get optimised accuracy

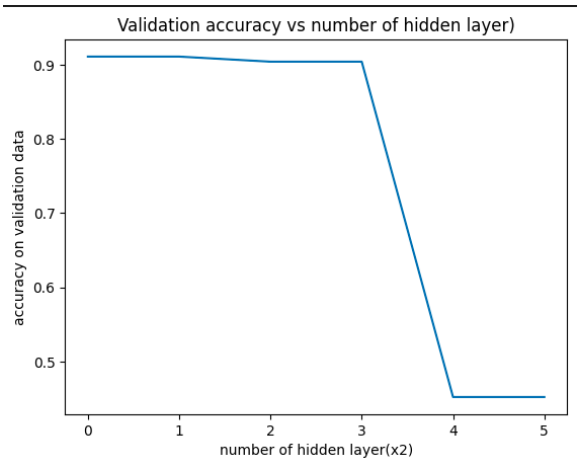


Here we used two different regularizations to overcome overfitting while training the model.we can observe that we are getting better accuracy with the above regularisations when compared with using only dropout.

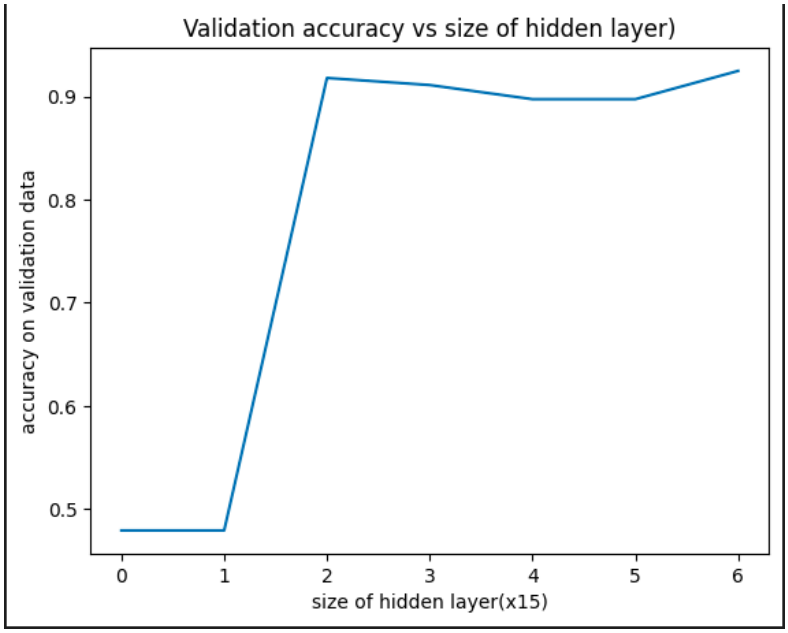
Q_6)

Here we plotted graph between accuracy on validation data vs no of hidden layers in neural network. By running code different times we are getting different plots but always the accuracy is more when the number of hidden layers are 2





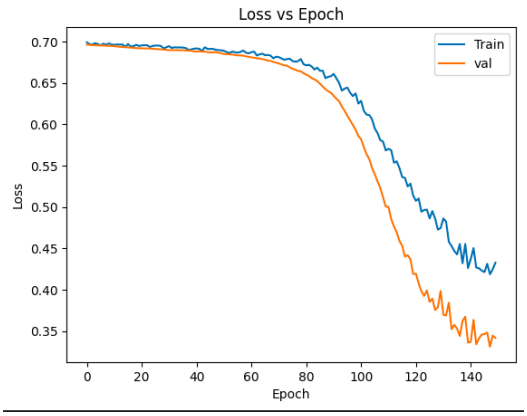
Here we plotted graph between accuracy on validation data vs size of hidden layers(no of neurons in hidden layer) in neural network. From below plot we can observe that at 30,90 We get optimised accuracy on validation data



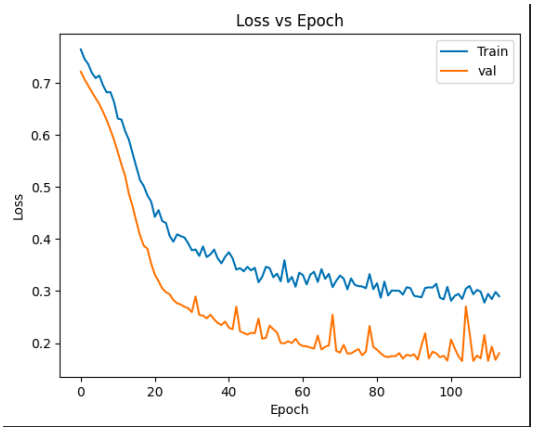
Q_7)

With ReLu as activation function, we get 0.9246575236320496 as accuracy whereas when tanhx as activation function we get accuracy as 0.801369845867157 and with tanh we can observe more fluctuations in the graph.

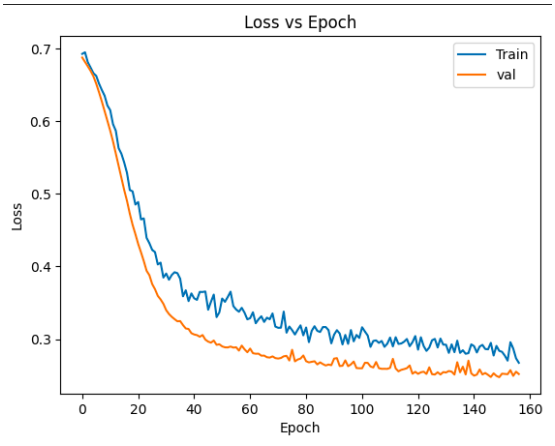
ReLu

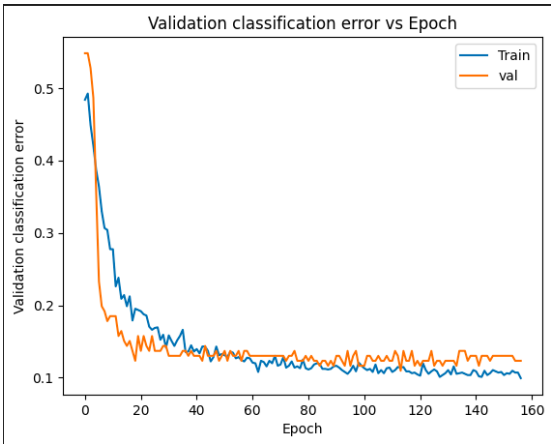


tanhx



Q_8)





If we observe the above figures and that in Q_2) we can observe so much variation in the error when epochs are increasing in the figures in Q_2). In the above figures error in validation set is almost same at high epochs.

For the optimized value we are getting the above confusion matrix with the accuracy of 92%.