Task 2:

Learning Rate	Using Learning Rate decay	No. of epochs	Num/Size of hidden Layers	lambda (regu)	Training loss	Training accuracy	Test Accuracy (on test,csv)
0.001	No	10000	1(100)	0	1.018183691	0.7655	0.54766

Task 3:

Learning Rate	Using Learning Rate decay	No. of epochs	Num/Size of hidden Layers	lambda (regu)	Training loss	Validation loss	Training accuracy	Validation accuracy
0.1	Yes	4000	2(100,100)	0.01	1.2411	1.227	0.7297	0.7415
0.1	Yes	4000	1(100)	0.01	1.2139	1.201	0.734	0.748
0.1	Yes	2200	3(96,48,96)	0.0001	0.958	0.967	0.7728	0.7778
0.1	Yes	2200	3(48,96,48)	0.0001	0.9512	0.9722	0.7754	0.778
0.01	Yes	3000	2(50,50)	0	0.9644	0.9696	0.7721	0.7768

For Learning rate decay, I am calculating standard deviation of last 5(calculation at every 100th epoch) validation loss values. If the standard deviation comes out to be less than a particular threshold this means the model has bottomed out on that learning rate and it is not able to improve further. I reduce the learning rate and then again weight for convergence on next learning rate.

The first element in the table gives best f1 score of .55200 on test set.

Task 4:

I have used relu and tanh for hidden layers

Learning Rate	No. of epochs	Num/Size of hidden Layers	Training loss	Validation loss	Training accuracy	Validation accuracy	Test Accuracy	Activation
0.01	300	1(100)	0.9826712	0.77065	0.77885	0.771	0.5503	Relu
0.01	1500	1(100)	0.983503	0.98337	0.7751	0.7743	0.5510	Tanh

.

There is no regularization used. I have used learning rate decay here as well. Test accuracy is on test.csv

Task 5:

I have incorporated dropouts on the first layer and also early stopping(500 epochs only). Dropout value at first layer is 0.3.

Below is the description of the network

1 hidden layer. Hidden Units=100. Learning rate=0.01. Train loss=1.212181, train accuracy=0.7347, Validation loss=1.09724, Validation accuracy=0.74375. Test Accuracy=0.55066