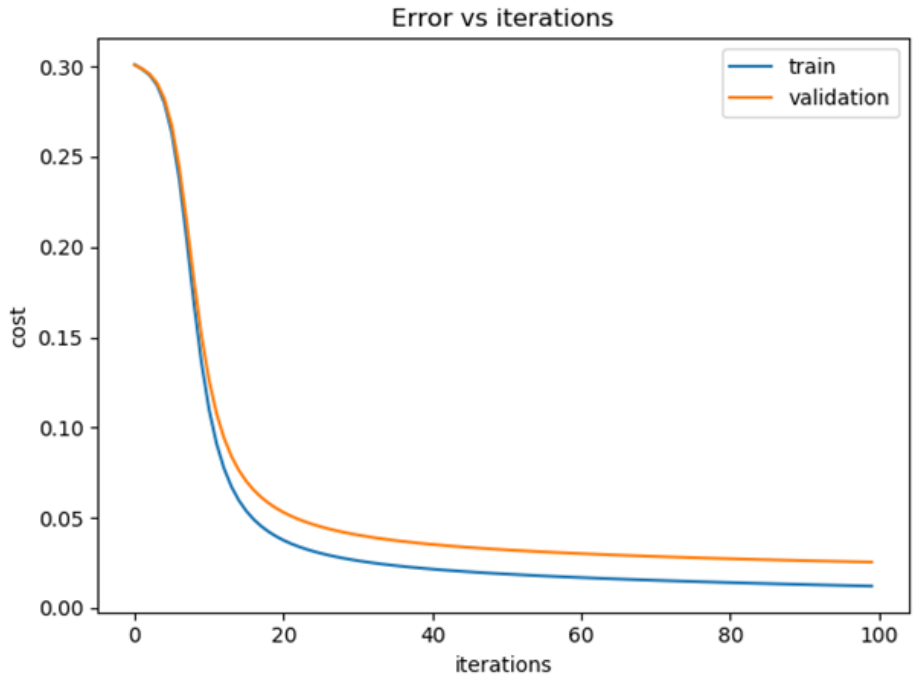
HW3, CSE 569 Fall 2018

Asu id:1215131021

Kaustubh Milindrao Sardar

Q1. **a)N_h=200(no of nodes in hidden layer)**

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure



python twoLayerBinary_starter.py "[784,200]"

Output:

Cost at iteration 0 is: 0.301061

Cost at iteration 100 is: 0.110023

Cost at iteration 200 is: 0.037573

Cost at iteration 300 is: 0.026250

Cost at iteration 400 is: 0.021590

Cost at iteration 500 is: 0.018843

Cost at iteration 600 is: 0.016903

Cost at iteration 700 is: 0.015386

Cost at iteration 800 is: 0.014126

Cost at iteration 900 is: 0.013043

Accuracy for training set is 99.150 %

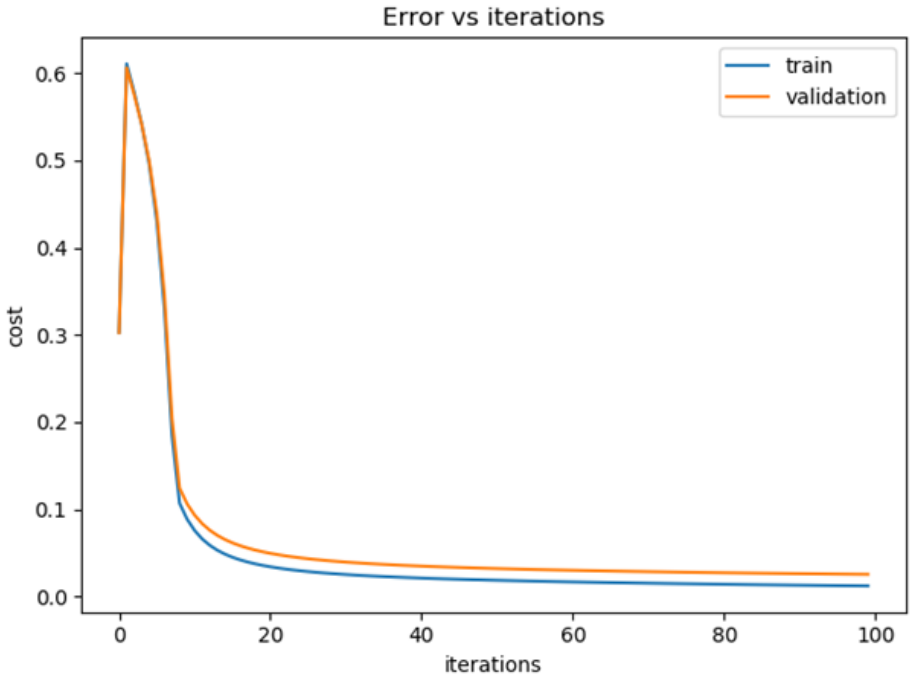
Accuracy for testing set is 98.700 %

Accuracy for validation set is 97.500 %

2) Accuracy for testing set is 98.700 % as given in output

b)N_h: 500

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure



python twoLayerBinary_starter.py "[784,500]"

Output:

Cost at iteration 0 is: 0.302872

Cost at iteration 100 is: 0.075414

Cost at iteration 200 is: 0.033938

Cost at iteration 300 is: 0.025058

Cost at iteration 400 is: 0.020981

Cost at iteration 500 is: 0.018444

Cost at iteration 600 is: 0.016598

Cost at iteration 700 is: 0.015131

Cost at iteration 800 is: 0.013904

Cost at iteration 900 is: 0.012845

Accuracy for training set is 99.150 %

Accuracy for testing set is 98.700 %

Accuracy for validation set is 97.750 %

2) Test accuracy is 98.700 %

c) n_h=100

python twoLayerBinary_starter.py "[784,100]"

output:

Cost at iteration 0 is: 0.300848

Cost at iteration 100 is: 0.133769

Cost at iteration 200 is: 0.040279

Cost at iteration 300 is: 0.027190

Cost at iteration 400 is: 0.022107

Cost at iteration 500 is: 0.019202

Cost at iteration 600 is: 0.017189

Cost at iteration 700 is: 0.015632

Cost at iteration 800 is: 0.014347

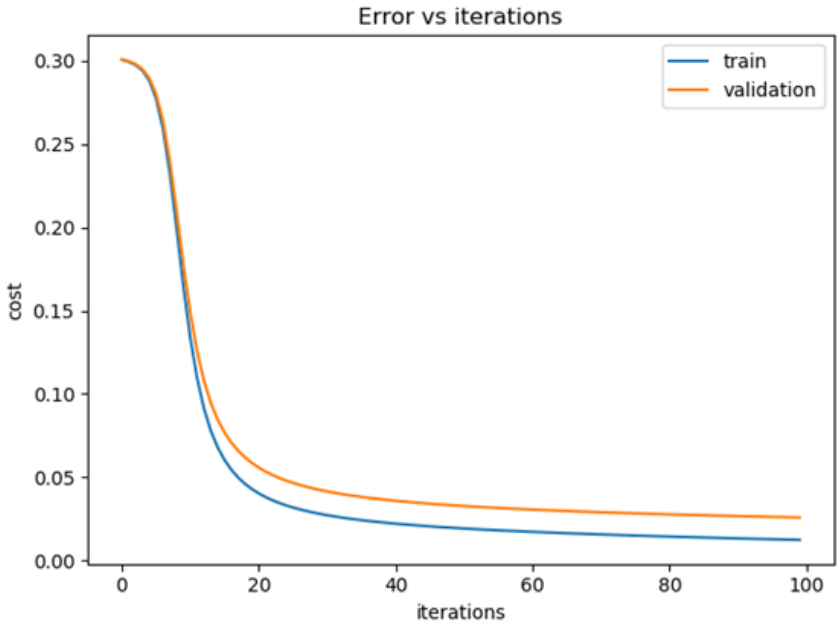
Cost at iteration 900 is: 0.013245

Accuracy for training set is 99.150 %

Accuracy for testing set is 98.700 %

Accuracy for validation set is 97.250 %

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure.

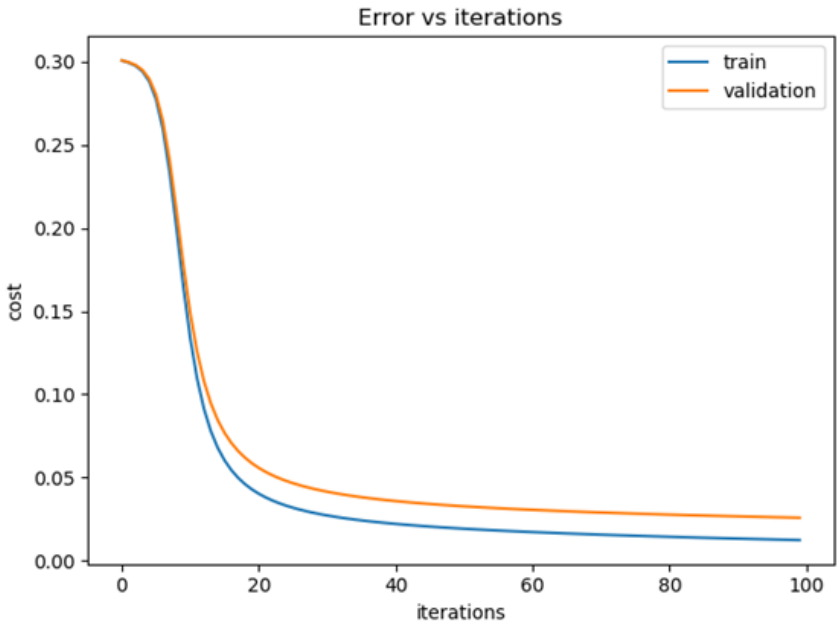


2)test accuracy as given in output is: 98.700 %

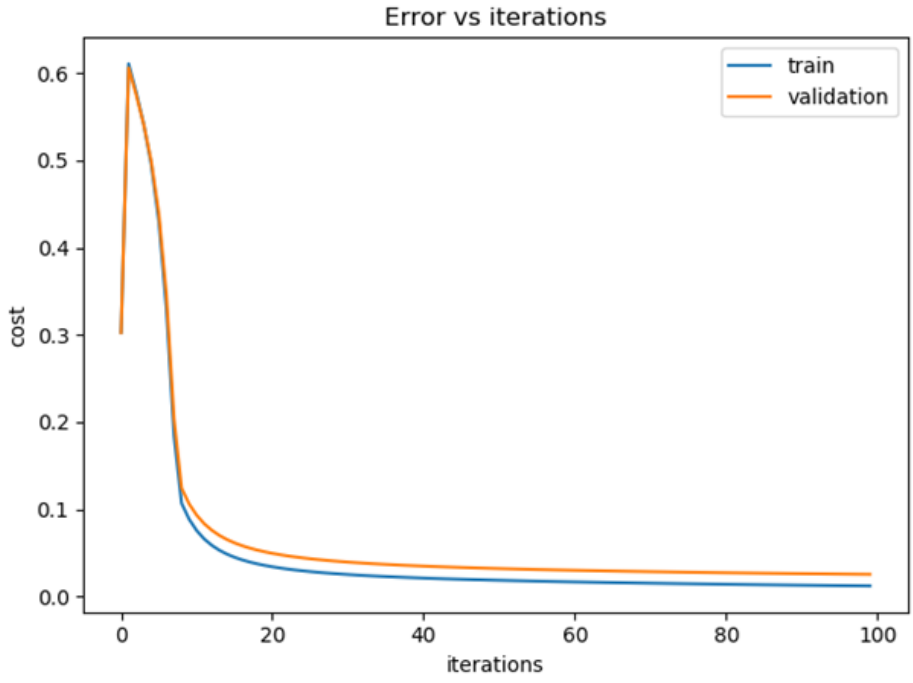
Here,best validation accuracy is given by network with hidden layer of 500 nodes.

Q1.3) Train and validation cost curves for different learning rates.

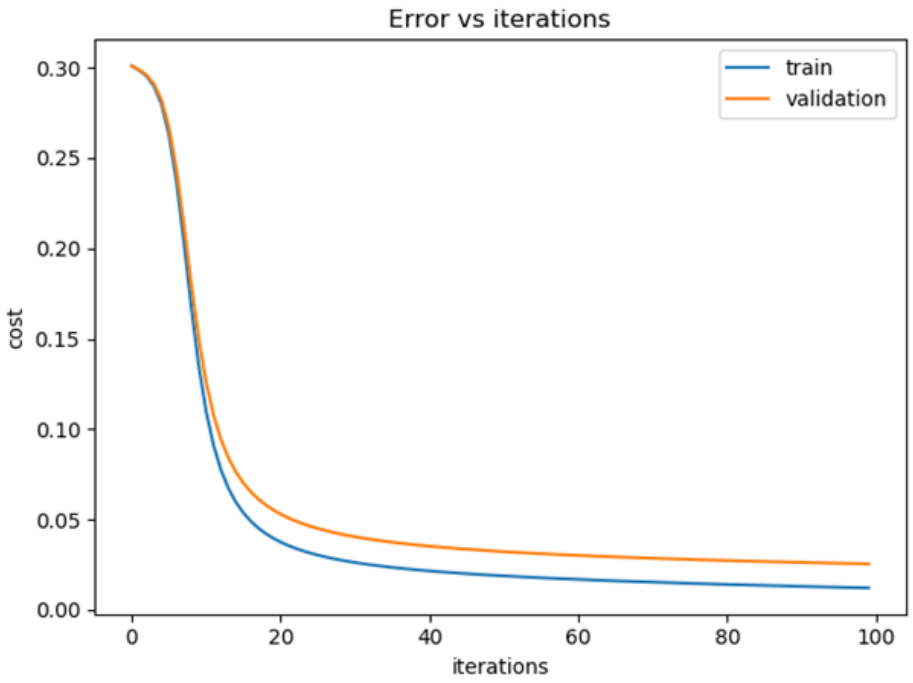
These have been plotted above for n_h=100:


Error vs iterations
train
validation
cost
iterations
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
100

N_h:500


Error vs iterations
train
validation
cost
iterations
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
20
40
60
80
100

N_h=200



Q1.4) Test accuracy using the model with the best validation cost

Test accuracy is 98.700 % for architecture with 500 nodes in hidden layer which is best architecture according to validation cost curve .

As you have seen above, validation curve is at lower height when n_h=500 compared to n_h=200 or 100 hence it has best validation cost.

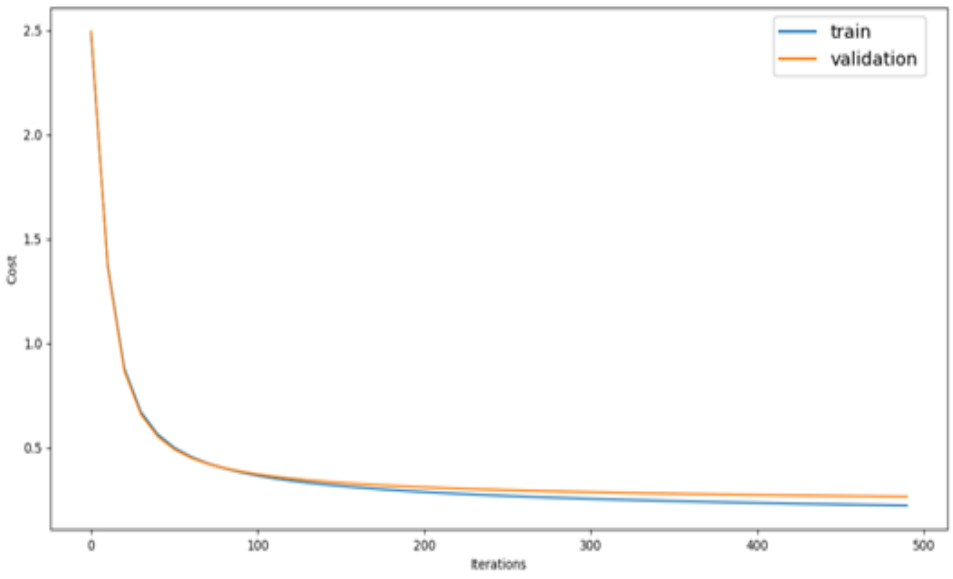
Q1. 5) A 100 – 300 word report on findings.

While training 1-hidden Layer Binary Classification Network, we observe that with each iteration the cost of training reduces ,ultimately at the last iterations becomes minimum where rate of change in cost of training becomes negligible ,ie, cost reaches a minimum threshold. If neural network overfits the data,validation curve will ascend .So, best validation cost curve is which observes similar changes in cost when compared to training cost curve .The plots given above show that our validation cost is reducing in similar manner as training cost, showing that neural network is neither overfitting nor underfitting.Here, we are using mnist dataset which has images of digits and their corresponding labels. For no of nodes in hidden layer =500, we get minimum validation cost ND Maximum validation accuracy.When we change no of nodes in hidden layer, we get different costs.The learning rate here is constant.

Q2.

a)Learning rate: 0.1

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure



```
C:\Users\Admin\PycharmProjects\p1>python deepMultiClassNetwork_starter.py "[784,500,100]"
Network dimensions are:[784, 500, 100, 10]
Cost at iteration 0 is: 2.48439, learning rate: 0.10000
Cost at iteration 10 is: 1.37475, learning rate: 0.09091
Cost at iteration 20 is: 0.88039, learning rate: 0.08333
Cost at iteration 30 is: 0.67088, learning rate: 0.07692
Cost at iteration 40 is: 0.56409, learning rate: 0.07143
Cost at iteration 50 is: 0.49971, learning rate: 0.06667
Cost at iteration 60 is: 0.45641, learning rate: 0.06250
Cost at iteration 70 is: 0.42506, learning rate: 0.05882
Cost at iteration 80 is: 0.40115, learning rate: 0.05556
Cost at iteration 90 is: 0.38209, learning rate: 0.05263
Cost at iteration 100 is: 0.36650, learning rate: 0.05000
Cost at iteration 110 is: 0.35341, learning rate: 0.04762
Cost at iteration 120 is: 0.34222, learning rate: 0.04545
Cost at iteration 130 is: 0.33251, learning rate: 0.04348
Cost at iteration 140 is: 0.32396, learning rate: 0.04167
Cost at iteration 150 is: 0.31637, learning rate: 0.04000
Cost at iteration 160 is: 0.30959, learning rate: 0.03846
Cost at iteration 170 is: 0.30345, learning rate: 0.03704
Cost at iteration 180 is: 0.29786, learning rate: 0.03571
Cost at iteration 190 is: 0.29276, learning rate: 0.03448
Cost at iteration 200 is: 0.28806, learning rate: 0.03333
Cost at iteration 210 is: 0.28371, learning rate: 0.03226
Cost at iteration 220 is: 0.27968, learning rate: 0.03125
```

Cost at iteration 230 is: 0.27592, learning rate: 0.03030

Cost at iteration 240 is: 0.27241, learning rate: 0.02941

Cost at iteration 250 is: 0.26911, learning rate: 0.02857

Cost at iteration 260 is: 0.26600, learning rate: 0.02778

Cost at iteration 270 is: 0.26307, learning rate: 0.02703

Cost at iteration 280 is: 0.26030, learning rate: 0.02632

Cost at iteration 290 is: 0.25766, learning rate: 0.02564

Cost at iteration 300 is: 0.25517, learning rate: 0.02500

Cost at iteration 310 is: 0.25279, learning rate: 0.02439

Cost at iteration 320 is: 0.25053, learning rate: 0.02381

Cost at iteration 330 is: 0.24836, learning rate: 0.02326

Cost at iteration 340 is: 0.24629, learning rate: 0.02273

Cost at iteration 350 is: 0.24430, learning rate: 0.02222

Cost at iteration 360 is: 0.24239, learning rate: 0.02174

Cost at iteration 370 is: 0.24056, learning rate: 0.02128

Cost at iteration 380 is: 0.23880, learning rate: 0.02083

Cost at iteration 390 is: 0.23710, learning rate: 0.02041

Cost at iteration 400 is: 0.23546, learning rate: 0.02000

Cost at iteration 410 is: 0.23388, learning rate: 0.01961

Cost at iteration 420 is: 0.23235, learning rate: 0.01923

Cost at iteration 430 is: 0.23088, learning rate: 0.01887

Cost at iteration 440 is: 0.22945, learning rate: 0.01852

Cost at iteration 450 is: 0.22807, learning rate: 0.01818

Cost at iteration 460 is: 0.22673, learning rate: 0.01786

Cost at iteration 470 is: 0.22543, learning rate: 0.01754

Cost at iteration 480 is: 0.22417, learning rate: 0.01724

Cost at iteration 490 is: 0.22295, learning rate: 0.01695

Accuracy for training set is 94.200 %

Accuracy for testing set is 89.400 %

Accuracy for validation set is 92.300 %

2) Test accuracy: 89.400 %

b)Learning rate: 0.5

output:

(venv) C:\Users\Admin\PycharmProjects\p1>python deepMultiClassNetwork_starter.py "[784,500,100]"

Network dimensions are:[784, 500, 100, 10]

Cost at iteration 0 is: 2.48439, learning rate: 0.50000

Cost at iteration 10 is: 1.07576, learning rate: 0.45455

Cost at iteration 20 is: 0.82362, learning rate: 0.41667

Cost at iteration 30 is: 0.30819, learning rate: 0.38462

Cost at iteration 40 is: 0.26337, learning rate: 0.35714

Cost at iteration 50 is: 0.23848, learning rate: 0.33333

Cost at iteration 60 is: 0.20592, learning rate: 0.31250

Cost at iteration 70 is: 0.18803, learning rate: 0.29412

Cost at iteration 80 is: 0.17378, learning rate: 0.27778

Cost at iteration 90 is: 0.16185, learning rate: 0.26316

Cost at iteration 100 is: 0.15164, learning rate: 0.25000

Cost at iteration 110 is: 0.14275, learning rate: 0.23810

Cost at iteration 120 is: 0.13493, learning rate: 0.22727

Cost at iteration 130 is: 0.12802, learning rate: 0.21739

Cost at iteration 140 is: 0.12185, learning rate: 0.20833

Cost at iteration 150 is: 0.11627, learning rate: 0.20000

Cost at iteration 160 is: 0.11121, learning rate: 0.19231

Cost at iteration 170 is: 0.10659, learning rate: 0.18519

Cost at iteration 180 is: 0.10236, learning rate: 0.17857

Cost at iteration 190 is: 0.09847, learning rate: 0.17241

Cost at iteration 200 is: 0.09488, learning rate: 0.16667

Cost at iteration 210 is: 0.09155, learning rate: 0.16129

Cost at iteration 220 is: 0.08845, learning rate: 0.15625

Cost at iteration 230 is: 0.08557, learning rate: 0.15152

Cost at iteration 240 is: 0.08286, learning rate: 0.14706

Cost at iteration 250 is: 0.08033, learning rate: 0.14286

Cost at iteration 260 is: 0.07795, learning rate: 0.13889

Cost at iteration 270 is: 0.07571, learning rate: 0.13514

Cost at iteration 280 is: 0.07361, learning rate: 0.13158

Cost at iteration 290 is: 0.07163, learning rate: 0.12821

Cost at iteration 300 is: 0.06976, learning rate: 0.12500

Cost at iteration 310 is: 0.06799, learning rate: 0.12195

Cost at iteration 320 is: 0.06632, learning rate: 0.11905

Cost at iteration 330 is: 0.06473, learning rate: 0.11628

Cost at iteration 340 is: 0.06323, learning rate: 0.11364

Cost at iteration 350 is: 0.06180, learning rate: 0.11111

Cost at iteration 360 is: 0.06044, learning rate: 0.10870

Cost at iteration 370 is: 0.05914, learning rate: 0.10638

Cost at iteration 380 is: 0.05789, learning rate: 0.10417

Cost at iteration 390 is: 0.05671, learning rate: 0.10204

Cost at iteration 400 is: 0.05558, learning rate: 0.10000

Cost at iteration 410 is: 0.05449, learning rate: 0.09804

Cost at iteration 420 is: 0.05345, learning rate: 0.09615

Cost at iteration 430 is: 0.05246, learning rate: 0.09434

Cost at iteration 440 is: 0.05150, learning rate: 0.09259

Cost at iteration 450 is: 0.05059, learning rate: 0.09091

Cost at iteration 460 is: 0.04971, learning rate: 0.08929

Cost at iteration 470 is: 0.04886, learning rate: 0.08772

Cost at iteration 480 is: 0.04804, learning rate: 0.08621

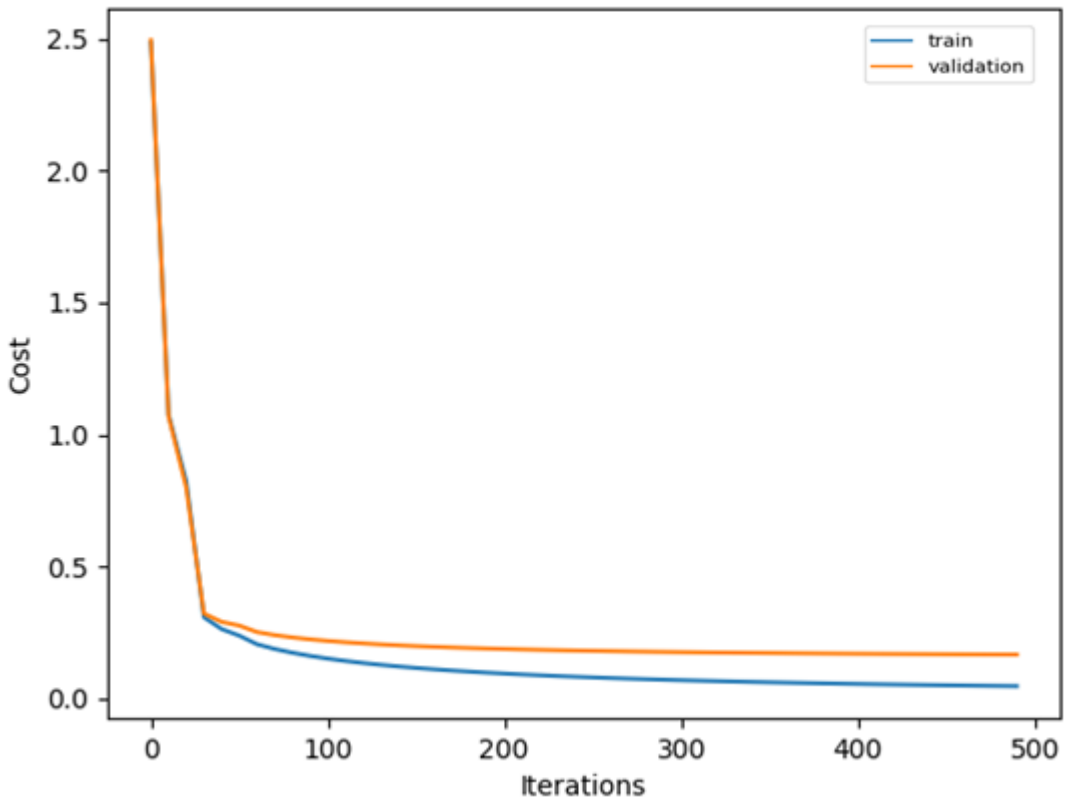
Cost at iteration 490 is: 0.04726, learning rate: 0.08475

Accuracy for training set is 99.260 %

Accuracy for testing set is 92.200 %

Accuracy for validation set is 95.200 %

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure



2)test accuracy is: 92.200 % according to output for given learning rate

c)learning rate:1.0

output:

(venv) C:\Users\Admin\PycharmProjects\p1>python deepMultiClassNetwork_starter.py "[784,500,100]"

Network dimensions are:[784, 500, 100, 10]

Cost at iteration 0 is: 2.48439, learning rate: 1.00000

Cost at iteration 10 is: 2.03190, learning rate: 0.90909

Cost at iteration 20 is: 2.14968, learning rate: 0.83333

Cost at iteration 30 is: 1.19612, learning rate: 0.76923

Cost at iteration 40 is: 0.67191, learning rate: 0.71429

Cost at iteration 50 is: 0.31881, learning rate: 0.66667

Cost at iteration 60 is: 0.26585, learning rate: 0.62500

Cost at iteration 70 is: 0.21670, learning rate: 0.58824

Cost at iteration 80 is: 0.18934, learning rate: 0.55556

Cost at iteration 90 is: 0.16870, learning rate: 0.52632

Cost at iteration 100 is: 0.15228, learning rate: 0.50000

Cost at iteration 110 is: 0.13881, learning rate: 0.47619

Cost at iteration 120 is: 0.12751, learning rate: 0.45455

Cost at iteration 130 is: 0.11788, learning rate: 0.43478

Cost at iteration 140 is: 0.10951, learning rate: 0.41667

Cost at iteration 150 is: 0.10214, learning rate: 0.40000

Cost at iteration 160 is: 0.09552, learning rate: 0.38462

Cost at iteration 170 is: 0.08960, learning rate: 0.37037

Cost at iteration 180 is: 0.08426, learning rate: 0.35714

Cost at iteration 190 is: 0.07943, learning rate: 0.34483

Cost at iteration 200 is: 0.07507, learning rate: 0.33333

Cost at iteration 210 is: 0.07113, learning rate: 0.32258

Cost at iteration 220 is: 0.06757, learning rate: 0.31250

Cost at iteration 230 is: 0.06431, learning rate: 0.30303

Cost at iteration 240 is: 0.06135, learning rate: 0.29412

Cost at iteration 250 is: 0.05862, learning rate: 0.28571

Cost at iteration 260 is: 0.05611, learning rate: 0.27778

Cost at iteration 270 is: 0.05378, learning rate: 0.27027

Cost at iteration 280 is: 0.05163, learning rate: 0.26316

Cost at iteration 290 is: 0.04964, learning rate: 0.25641

Cost at iteration 300 is: 0.04780, learning rate: 0.25000

Cost at iteration 310 is: 0.04608, learning rate: 0.24390

Cost at iteration 320 is: 0.04447, learning rate: 0.23810

Cost at iteration 330 is: 0.04297, learning rate: 0.23256

Cost at iteration 340 is: 0.04156, learning rate: 0.22727

Cost at iteration 350 is: 0.04023, learning rate: 0.22222

Cost at iteration 360 is: 0.03899, learning rate: 0.21739

Cost at iteration 370 is: 0.03781, learning rate: 0.21277

Cost at iteration 380 is: 0.03671, learning rate: 0.20833

Cost at iteration 390 is: 0.03567, learning rate: 0.20408

Cost at iteration 400 is: 0.03469, learning rate: 0.20000

Cost at iteration 410 is: 0.03376, learning rate: 0.19608

Cost at iteration 420 is: 0.03287, learning rate: 0.19231

Cost at iteration 430 is: 0.03204, learning rate: 0.18868

Cost at iteration 440 is: 0.03124, learning rate: 0.18519

Cost at iteration 450 is: 0.03049, learning rate: 0.18182

Cost at iteration 460 is: 0.02976, learning rate: 0.17857

Cost at iteration 470 is: 0.02907, learning rate: 0.17544

Cost at iteration 480 is: 0.02840, learning rate: 0.17241

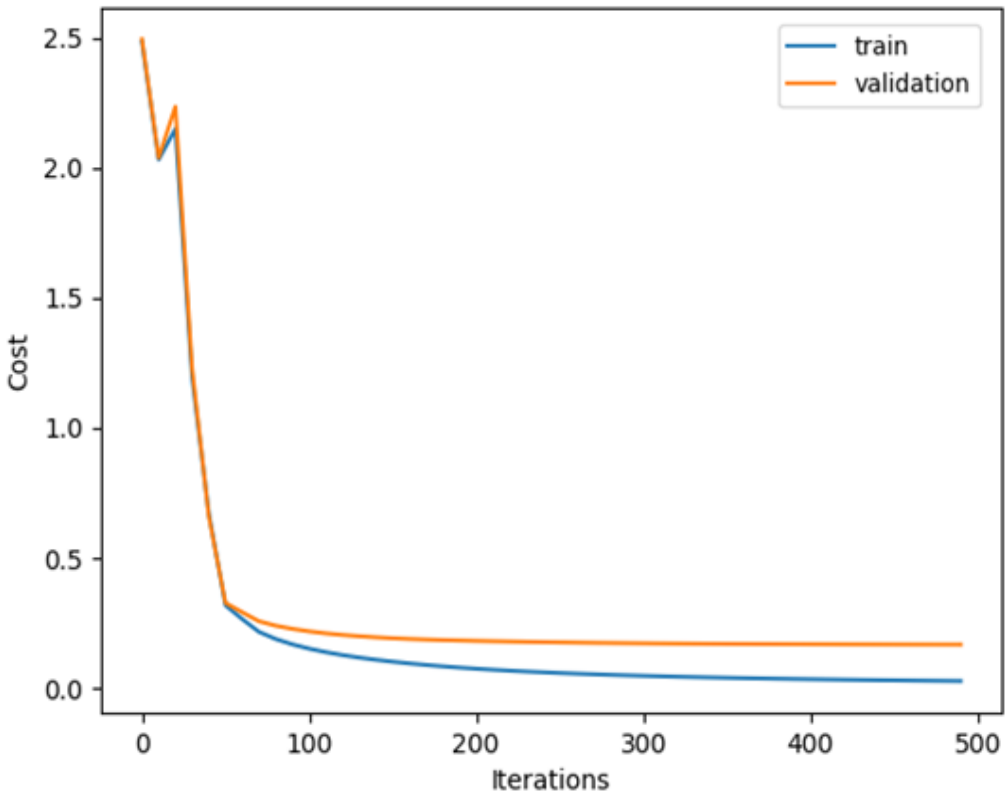
Cost at iteration 490 is: 0.02777, learning rate: 0.16949

Accuracy for training set is 99.660 %

Accuracy for testing set is 92.300 %

Accuracy for validation set is 95.000%

1) Plot of Train cost (average loss) vs iterations and validation cost vs iterations in the same figure



2) **Test accuracy is 92.300 % according to the output above for learning rate=1.0**

d) learning rate=10

output:

(venv) C:\Users\Admin\PycharmProjects\p1>python deepMultiClassNetwork_starter.py "[784,500,100]"

Network dimensions are:[784, 500, 100, 10]

Cost at iteration 0 is: 2.48439, learning rate: 10.00000

Cost at iteration 10 is: 6.92374, learning rate: 9.09091

Cost at iteration 20 is: 5.74604, learning rate: 8.33333

Cost at iteration 30 is: 3.98665, learning rate: 7.69231

Cost at iteration 40 is: 2.68375, learning rate: 7.14286

Cost at iteration 50 is: 2.30260, learning rate: 6.66667

Cost at iteration 60 is: 2.30258, learning rate: 6.25000

Cost at iteration 70 is: 2.30258, learning rate: 5.88235

Cost at iteration 80 is: 2.30258, learning rate: 5.55556

Cost at iteration 90 is: 2.30258, learning rate: 5.26316

Cost at iteration 100 is: 2.30258, learning rate: 5.00000

Cost at iteration 110 is: 2.30258, learning rate: 4.76190

Cost at iteration 120 is: 2.30258, learning rate: 4.54545

Cost at iteration 130 is: 2.30258, learning rate: 4.34783

Cost at iteration 140 is: 2.30258, learning rate: 4.16667

Cost at iteration 150 is: 2.30258, learning rate: 4.00000

Cost at iteration 160 is: 2.30258, learning rate: 3.84615

Cost at iteration 170 is: 2.30258, learning rate: 3.70370

Cost at iteration 180 is: 2.30258, learning rate: 3.57143

Cost at iteration 190 is: 2.30258, learning rate: 3.44828

Cost at iteration 200 is: 2.30258, learning rate: 3.33333

Cost at iteration 210 is: 2.30258, learning rate: 3.22581

Cost at iteration 220 is: 2.30258, learning rate: 3.12500

Cost at iteration 230 is: 2.30258, learning rate: 3.03030

Cost at iteration 240 is: 2.30258, learning rate: 2.94118

Cost at iteration 250 is: 2.30258, learning rate: 2.85714

Cost at iteration 260 is: 2.30258, learning rate: 2.77778

Cost at iteration 270 is: 2.30258, learning rate: 2.70270

Cost at iteration 280 is: 2.30258, learning rate: 2.63158

Cost at iteration 290 is: 2.30258, learning rate: 2.56410

Cost at iteration 300 is: 2.30258, learning rate: 2.50000

Cost at iteration 310 is: 2.30258, learning rate: 2.43902

Cost at iteration 320 is: 2.30258, learning rate: 2.38095

Cost at iteration 330 is: 2.30258, learning rate: 2.32558

Cost at iteration 340 is: 2.30258, learning rate: 2.27273

Cost at iteration 350 is: 2.30258, learning rate: 2.22222

Cost at iteration 360 is: 2.30258, learning rate: 2.17391

Cost at iteration 370 is: 2.30258, learning rate: 2.12766

Cost at iteration 380 is: 2.30258, learning rate: 2.08333

Cost at iteration 390 is: 2.30258, learning rate: 2.04082

Cost at iteration 400 is: 2.30258, learning rate: 2.00000

Cost at iteration 410 is: 2.30258, learning rate: 1.96078

Cost at iteration 420 is: 2.30258, learning rate: 1.92308

Cost at iteration 430 is: 2.30258, learning rate: 1.88679

Cost at iteration 440 is: 2.30258, learning rate: 1.85185

Cost at iteration 450 is: 2.30258, learning rate: 1.81818

Cost at iteration 460 is: 2.30258, learning rate: 1.78571

Cost at iteration 470 is: 2.30258, learning rate: 1.75439

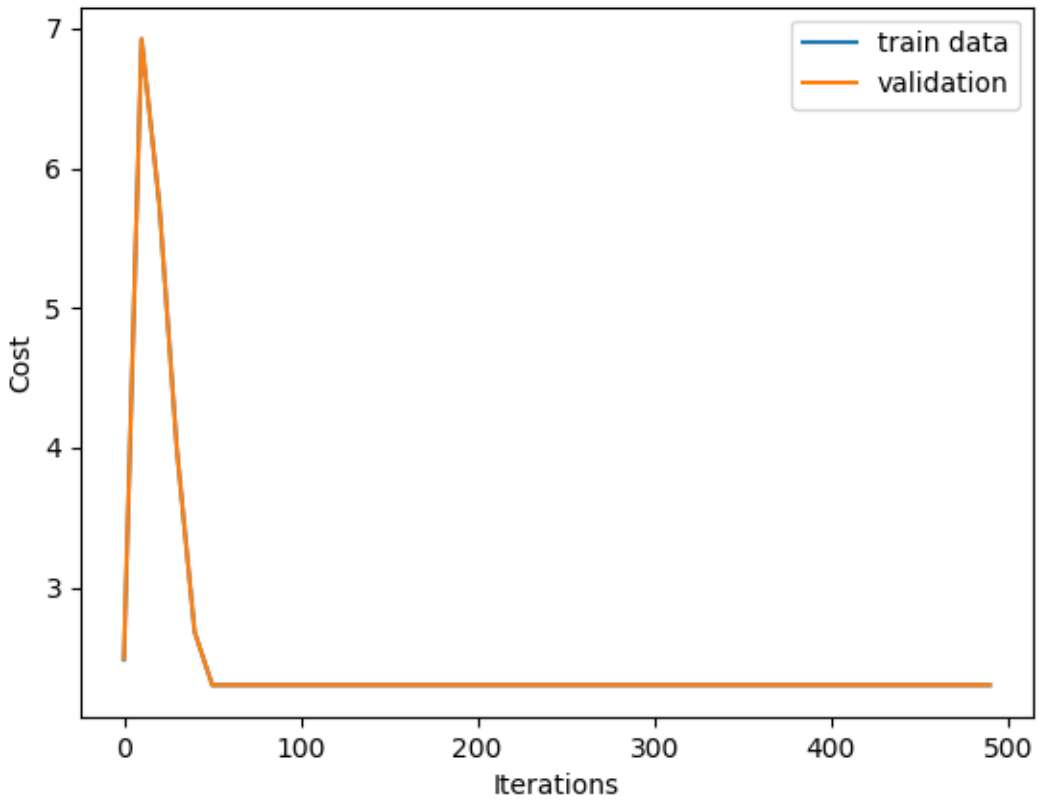
Cost at iteration 480 is: 2.30258, learning rate: 1.72414

Cost at iteration 490 is: 2.30258, learning rate: 1.69492

Accuracy for training set is 10.000 %

Accuracy for testing set is 10.000 %

Accuracy for validation set is 10.000 %



Q2.3) Train and validation cost curves for different learning rates.

The curves for each learning rate have been plotted above

Q2.4)Test accuracy using the model with the best validation cost.

Learning rate =0.5 gives best validation accuracy(Accuracy for validation set is 95.200 %) and minimum validation cost

where

Accuracy for testing set is 92.200 %

Q2.5) A 100 – 300 word report on your findings.

In this question, when learning rate is very less ie, 0.1 we find very smooth curve for train and validation cost, but it takes forever for costs to reduce and become minimum threshold. When learning rats are

1.0 and 10, we observe sudden and sharp declines in cost, but as learning rates are too high, the changes made in parameter are significant , hence the accuracy of each validation and testing decreases. Specially for learning rate=10, we find that accuracy is very low. The perfect balance of learning rate is at 0.5 where changes are not sharp and also not too slow.So, even though learning rate ie alpha is decreasing, it still performs better than other learning rates.