CSE-569 Homework-3 Solution

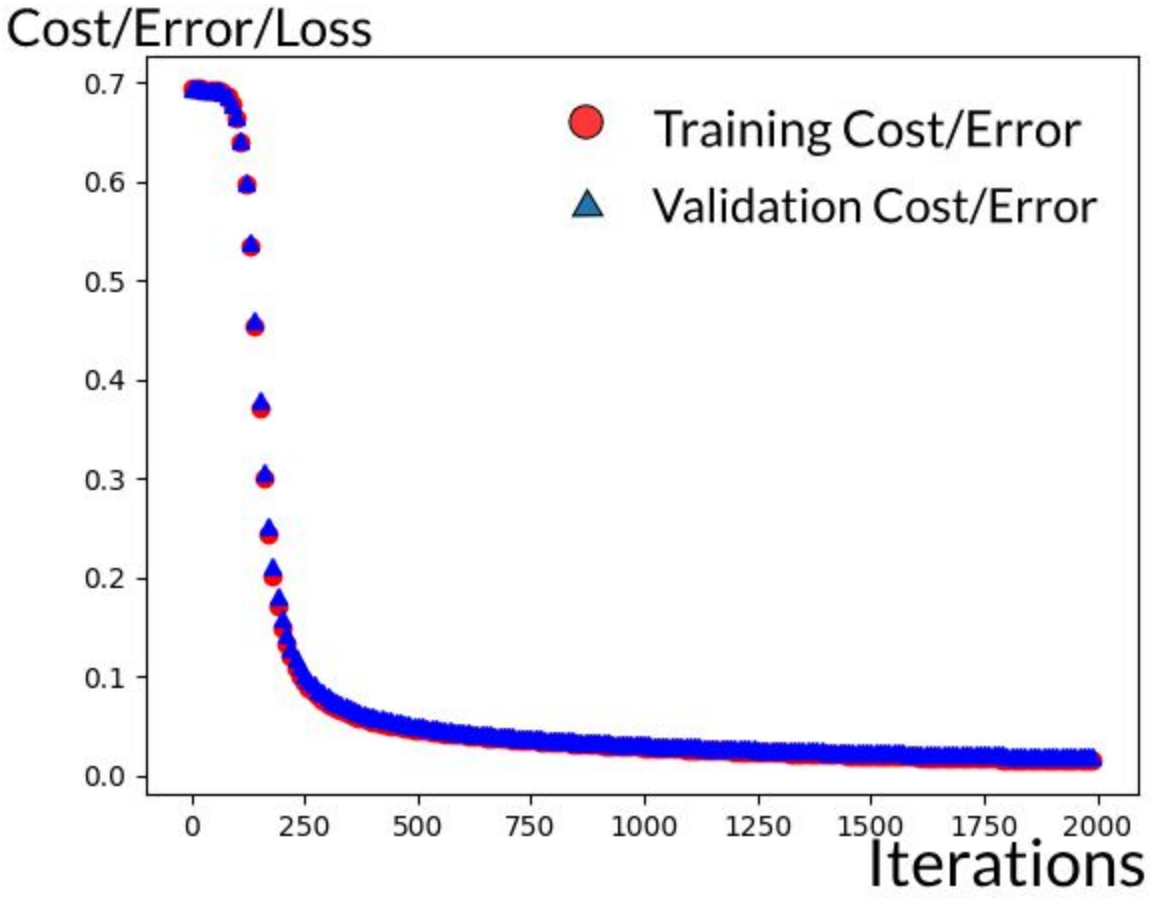
Name: Kunal Vinay Kumar Suthar
ASURite ID: 1215112535

Q1.) Train 1-hidden Layer Binary Classification Network.

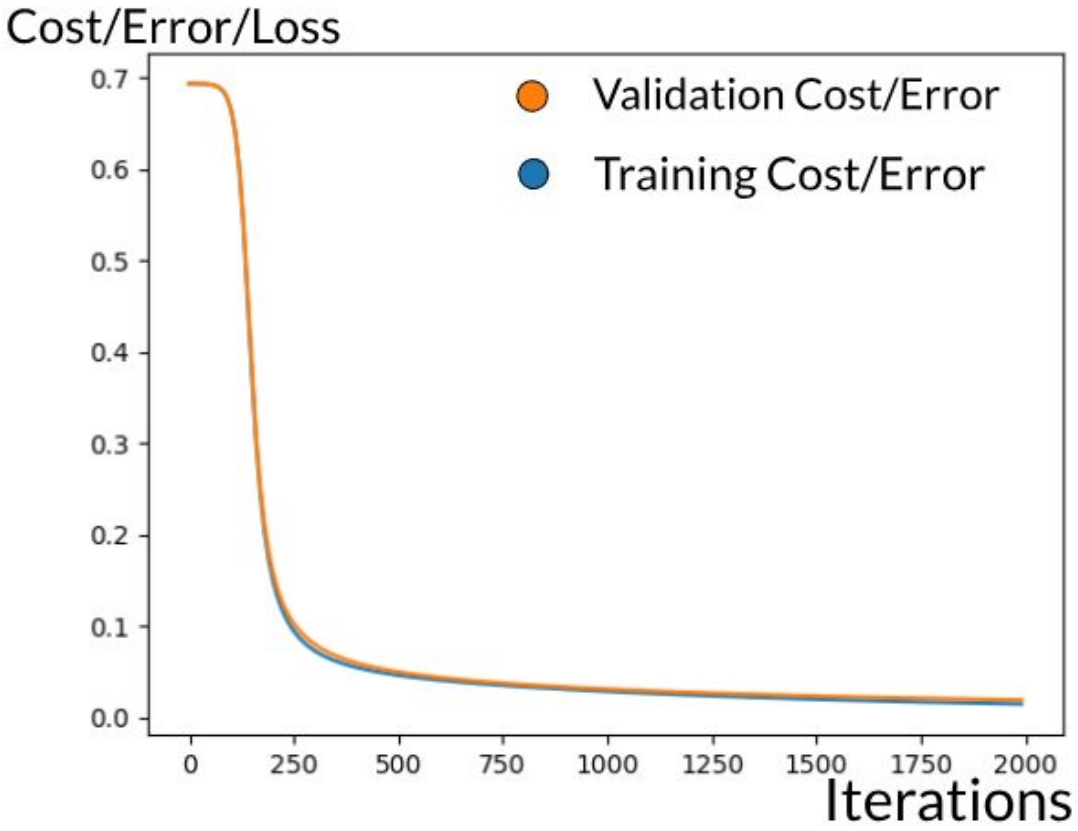
1) Plot of Train cost (average loss) vs iterations and validation error vs iterations in the same figure.

Solution 1)

Training and Validation cost plot with [784,200,1] architecture



Continuous Curve for the above plot:



Training Cost at iteration 2000 is: 0.015317
Validation Cost at iteration 2000 is: 0.019757

2) Test Accuracy.

Solution 2)

Assuming **Test error = 1 - test accuracy,** we get test error as **0.013** as the **test accuracy** is 0.987 i.e. **98.7%**.

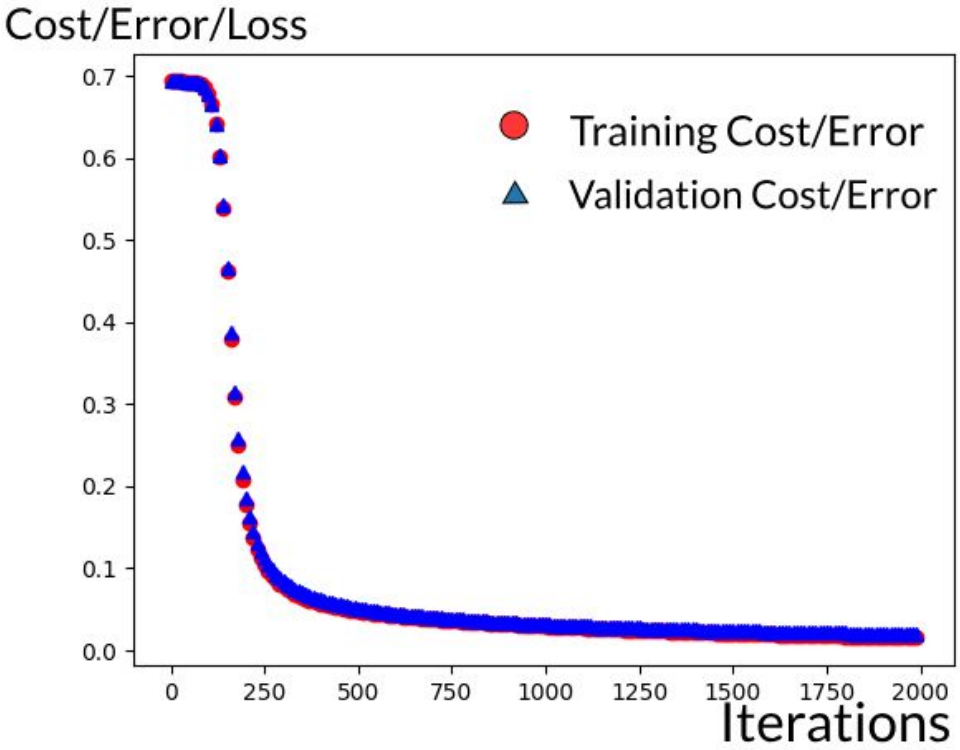
3) Train and validation cost curves for different architectures.

Solution 3)

i) Cost Plot for architecture with dimensions[784,100,1]:

Training Cost at iteration 2000 is: 0.015428

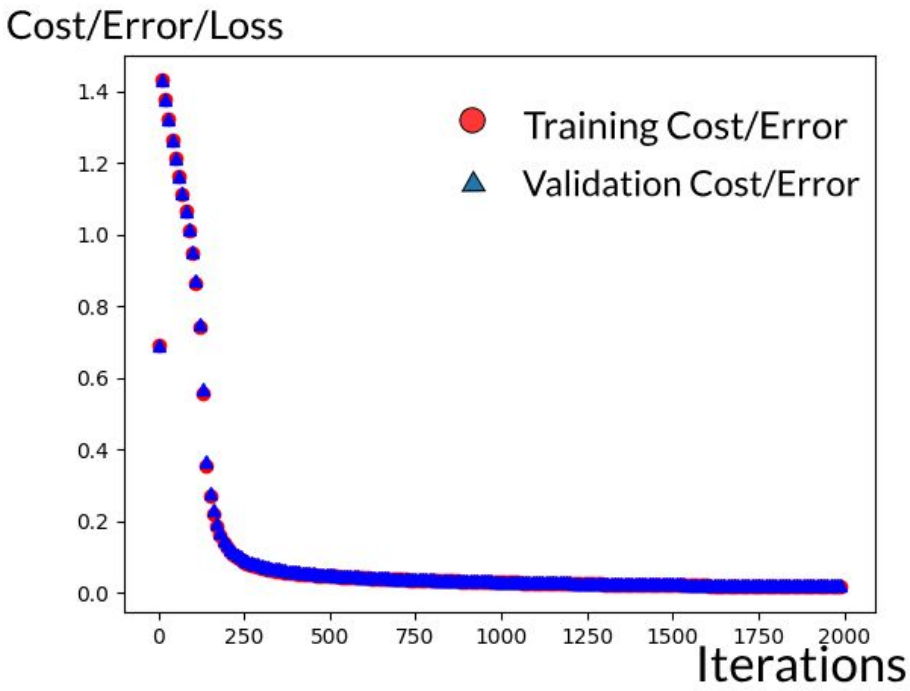
Validation Cost at iteration 2000 is: 0.019882



ii) Cost Plot for architecture with dimensions[784,500,1]:

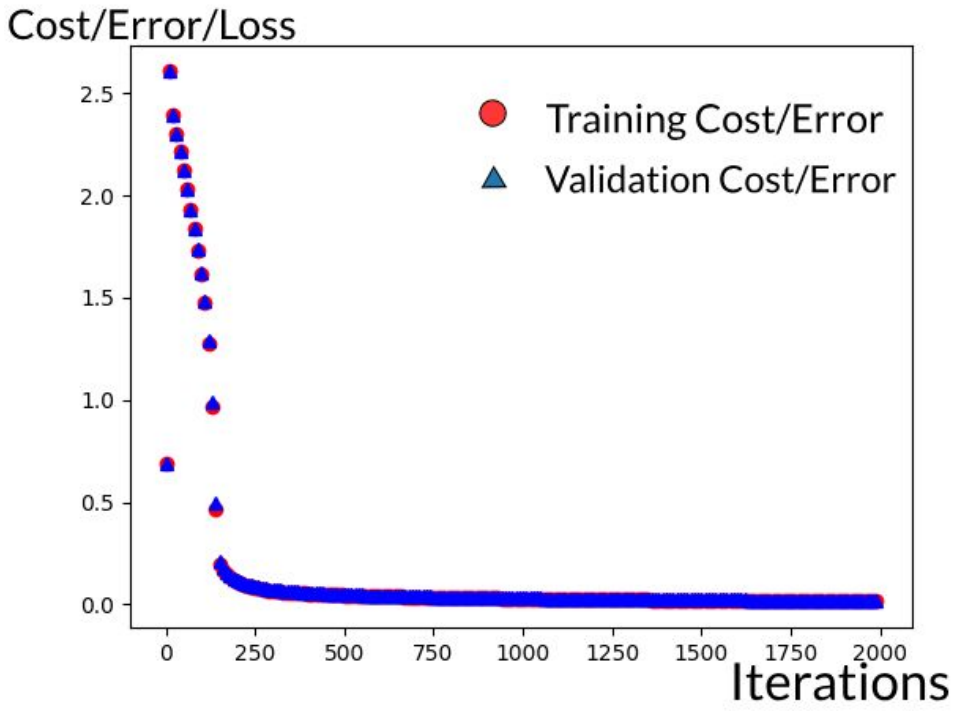
Training Cost at iteration 2000 is: 0.015380

Validation Cost at iteration 2000 is: 0.019604



iii) Cost Plot for architecture with dimensions[784,800,1]:

Training Cost at iteration 2000 is: 0.015634
Validation Cost at iteration 2000 is: 0.019731



4) Test accuracy using the architecture with the best validation error.

Solution 4)

Architecture **[784,500,1]** has the best validation error amongst the other architectures, because it has the lowest validation cost value. The test accuracy for the architecture is **99.1%**.

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

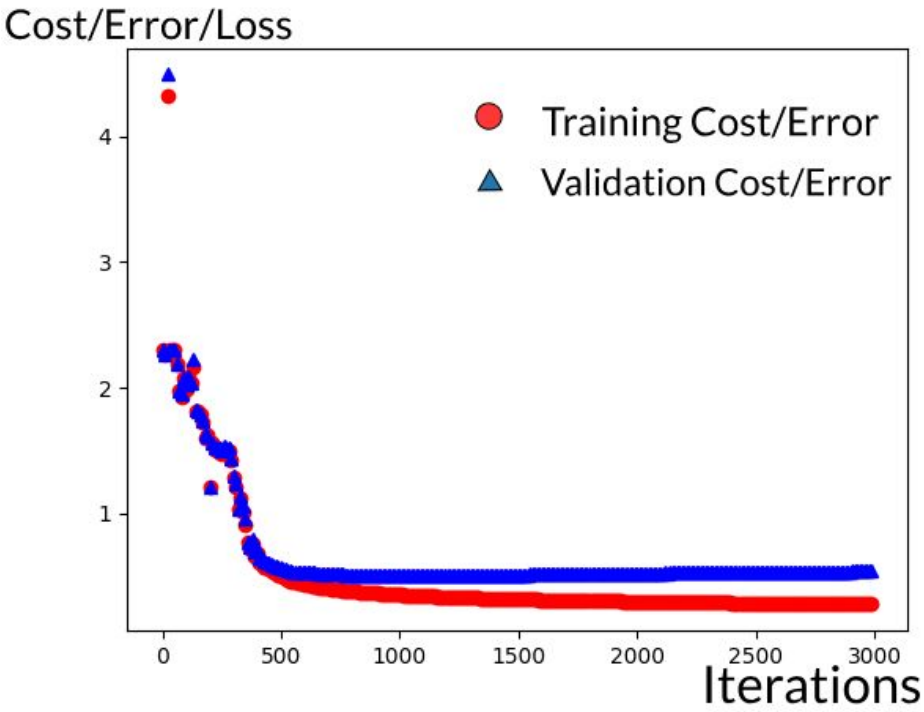
Solution 5)

The **Train** and **Validation Error** are almost similar and overlap at most iterations(with Validation Error being slightly more than the Training Error at some iterations) and their variations with respect to the number of iterations is also similar for all architectures. For architectures **[784,500,1]** and **[784,800,1]**(with higher hidden layers) during the initial iterations, the cost/error for the training and validation goes up and then the error gradually reduces and converges. This initial spike in the cost happens because of the initially high random values of W1,b1,W2,b2. After a few iterations, the network gets better and the cost gradually reduces and converges in the end. For architectures **[784,100,1]** and **[784,200,1]**, the cost improves slowly during the initial iterations and then the cost reduces at a better pace . The training and the validation cost curve improve and vary together. This means that we are **avoiding overfitting**. The train and validation error converge at **0.015** and **0.019** for almost all architectures and generates the best training and testing accuracy of **99.5%** and **99.1%** respectively on the **[784,500,1]** architecture, after **2000** iterations.

Q2.) Train Fully-connected Multilayer Neural Network.

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

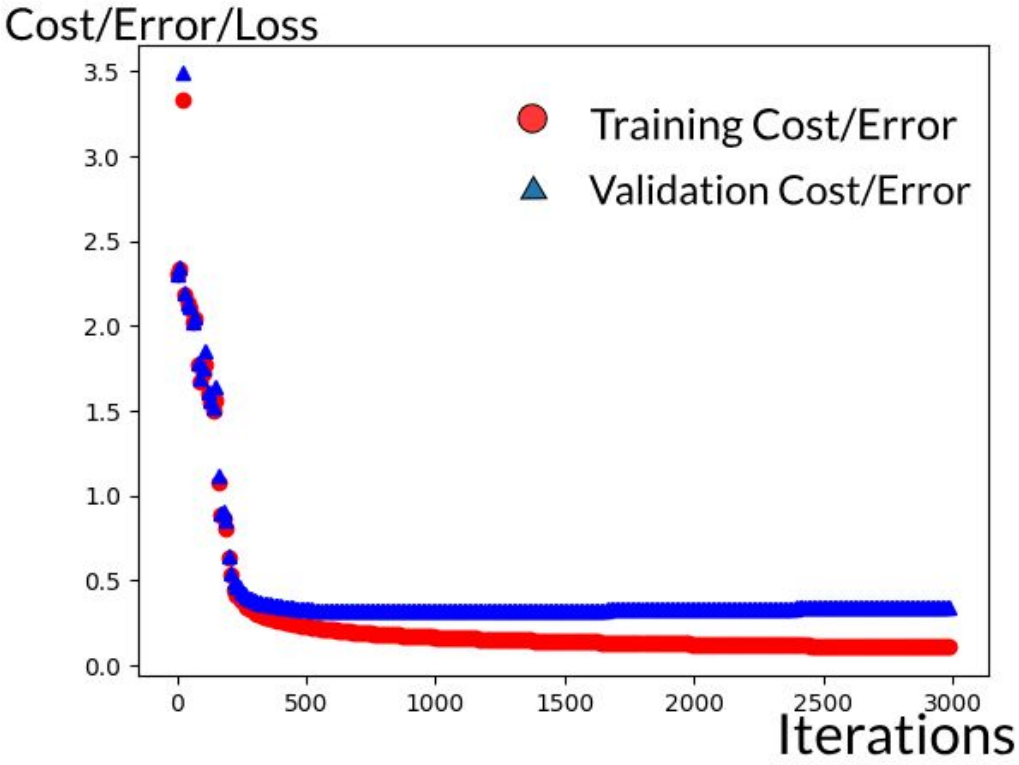
For learning rate: 0.2:
Accuracy for the training set is ***92.7%****.*



2) Test Accuracy.
Solution 2)
Accuracy for the testing set is ***80.7%****.*

3) Train and validation cost curves for different learning rates and/or decay rate.
Solution 3)
The decay rate used is **0.01**.

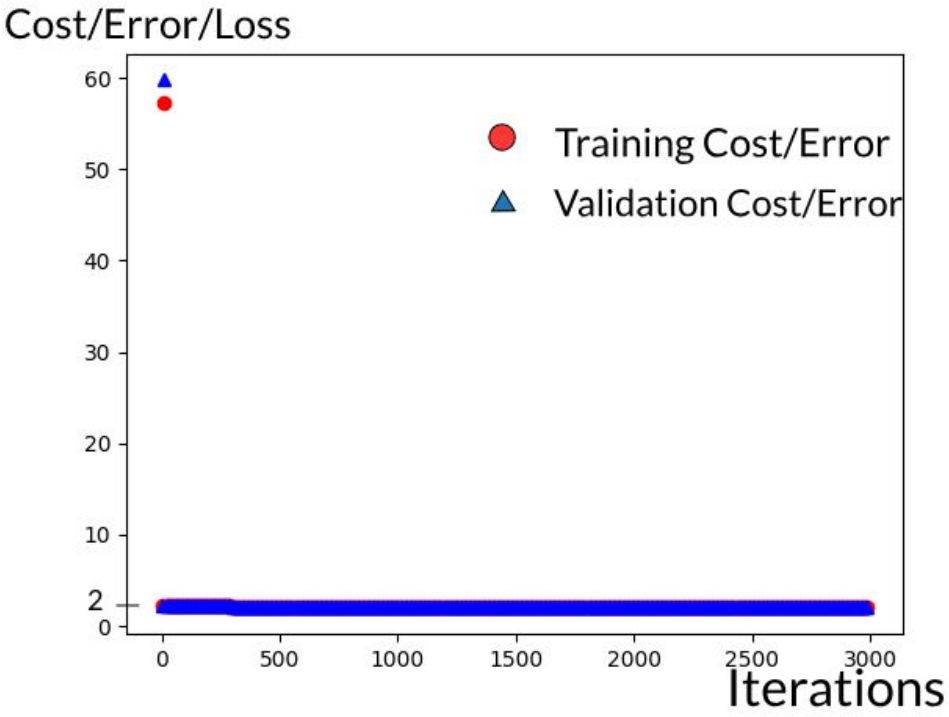
i) For learning rate of 0.1:
Accuracy for training set is ***97.5 %***
Accuracy for testing set is ***86.2 %***



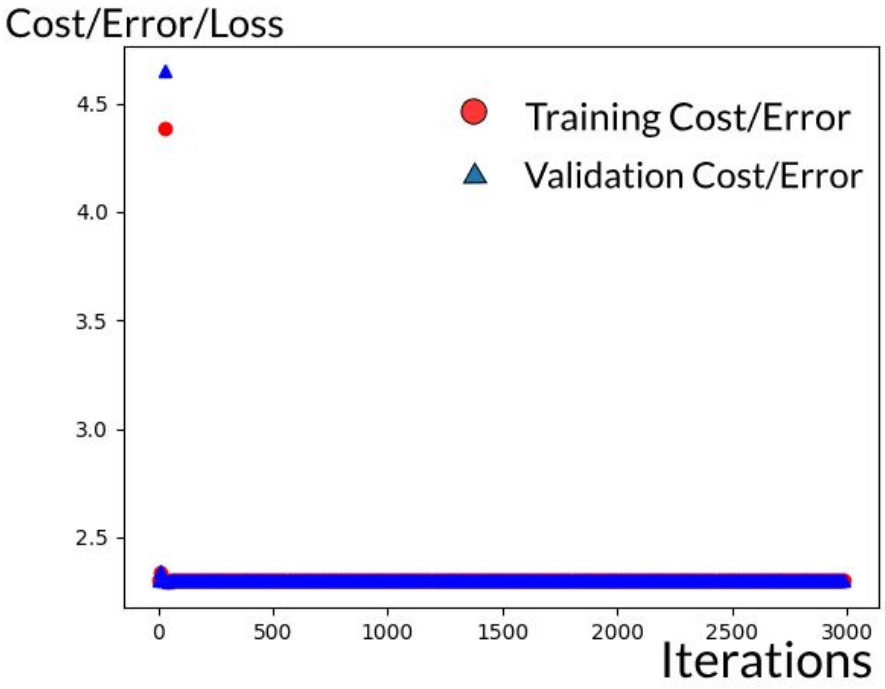
ii) For learning rate of 0.5:

Accuracy for training set is ***19.4 %***

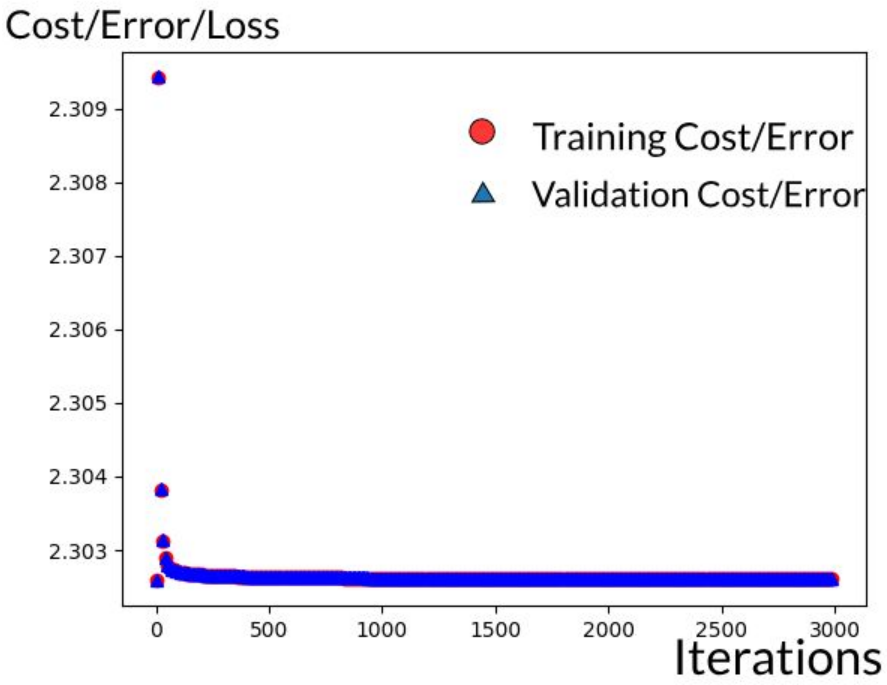
Accuracy for testing set is ***18.6 %***



iii) For learning rate of 1.0:
Accuracy for training set is ***10.0 %***
Accuracy for testing set is ***10.1 %***



iv) For learning rate of 10.0:
Accuracy for training set is ***10.0 %***
Accuracy for testing set is ***9.9 %***



4) Test Accuracy using the model with the best validation cost.

Solution 4)

The model with the learning rate of 0.1 has the best validation cost. **Test Accuracy** for that model is **86.2 %** and the **training accuracy** for that model is **97.5%**.

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

Solution 5)

Using the architecture **[784,500,100,10]**, with learning rates of 0.2, 0.1, 0.5, 1.0 and 10.0 , we observe that the train and validation cost goes very high during the initial iteration, and then it gradually improves/stays constant. This spike can be again explained due to the initialization of W's and b's with random values. For models with learning rates **0.1** and **0.2**, we observe that after the initial spike in the training and validation costs, the costs gradually improve and converge to a very low value after 3000 iterations. This results in good accuracies for the training and testing data sets. For models with learning rates **0.5, 1.0, 10.0**, we observe that after the initial spike in the costs, it drops to a constant value(around 2)which is high error. Thus these models perform badly. This is because higher learning rate implies that we are taking bigger steps towards the local minima and taking bigger steps towards local minima hints at a possibility of missing the local minima. That is why the models with higher learning rates are performing badly, than the models with lesser learning rates. Therefore the model with the learning rate of **0.1** is the best model to test the dataset.

Q3.) Kernel Soft-margin SVM.

1) Validation accuracies for different values of SVM parameters C and the kernel parameters (γ) for the polynomial kernel and RBF kernel.

Solution 1)

Index	C	γ (Gamma)	Validation Accuracy for the POLYNOMIAL Kernel(Out of 1)	Validation Accuracy for the RBF Kernel(Out of 1)
1.	0.001	0.001	0.56	0.56
2.	0.001	0.01	0.56	0.56
3.	0.001	0.1	0.56	0.56
4.	0.1	0.001	0.56	0.56
5.	0.1	0.01	0.56	0.56
6.	0.1	0.1	0.56	0.78
7.	10	0.001	0.56	0.66
8.	10	0.01	0.56	0.7
9.	10	0.1	0.72	0.86

10.	1000	0.001	0.56	0.74
11.	1000	0.01	0.56	0.74
12.	1000	0.1	0.78	1.0

2) Select the best values for the parameters (C and γ) based on the validation accuracies and report your accuracies on the test dataset.

Solution 2)

Best values for parameters, C and γ, based on the best validation accuracy obtained is **C=1000** and **γ=0.1**.

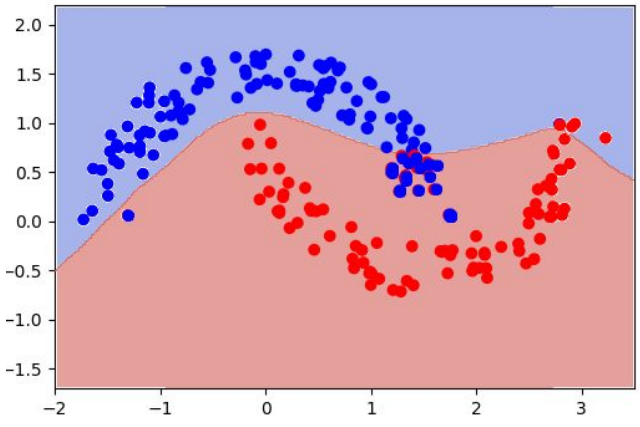
Test dataset accuracy on the **POLYNOMIAL** Kernel with parameters C=1000 and γ=0.1: **90.5%**
Test dataset accuracy on the **RBF** Kernel with parameters C=1000 and γ=0.1: **100.0%**

3) Plot the SVM boundaries along with the test data points for at least 3 different configurations of C and γ. (HINT: create a grid of the entire plot and classify every point on the grid. The boundary lies on the grid points that have equal probability of belonging to both the classes.).

Solution 3)

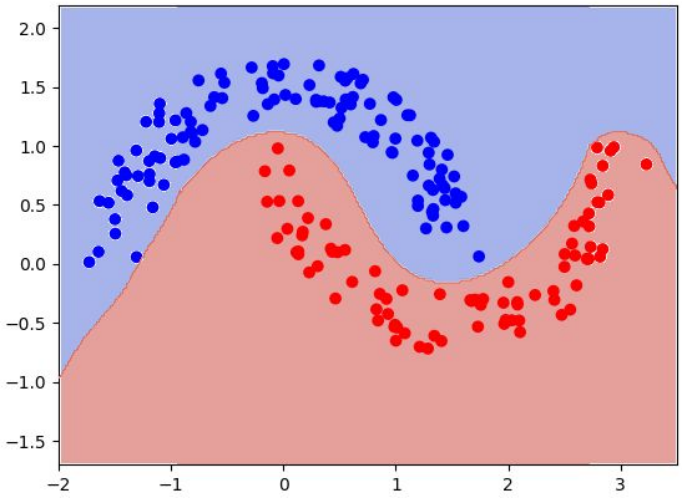
Plot for C=1000 and γ=0.1(Polynomial and RBF):

POLYNOMIAL



Accuracy on the Test set: 90.5 %

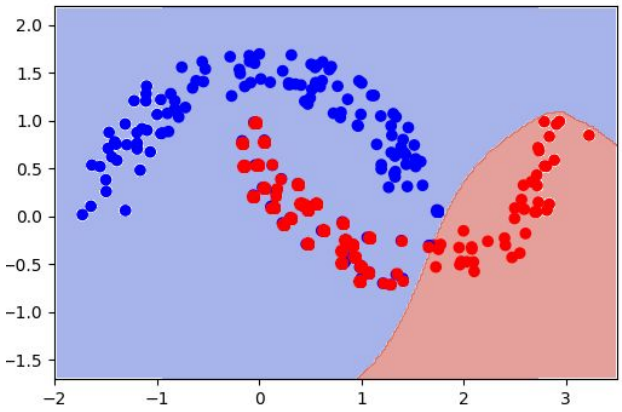
RBF



Accuracy on the Test set: 100%

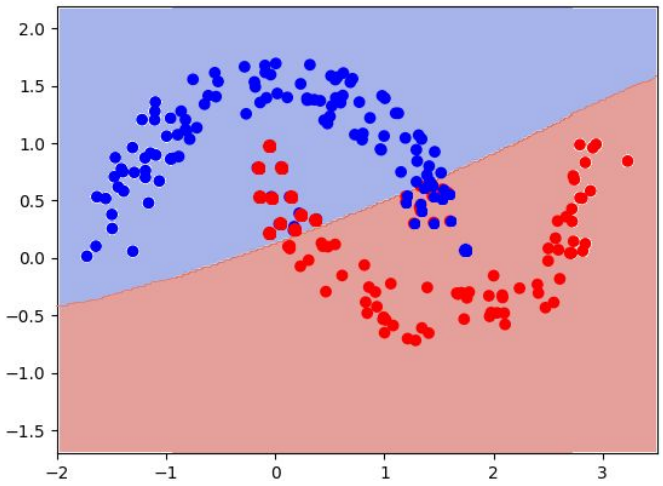
Plot for C=1000 and γ=0.01(Polynomial and RBF):

POLYNOMIAL



Accuracy on the Test set: 57.5 %

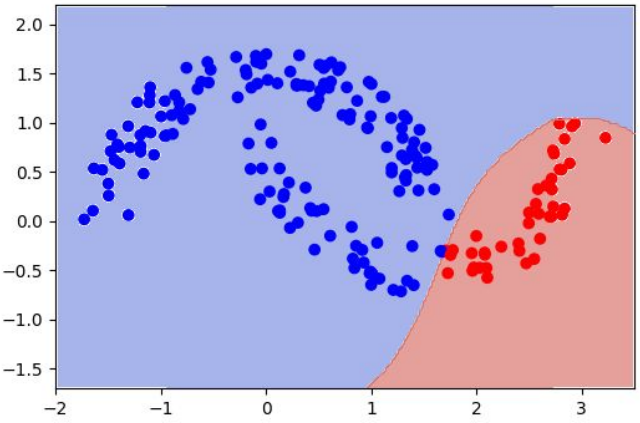
RBF



Accuracy on the Test set: 95%

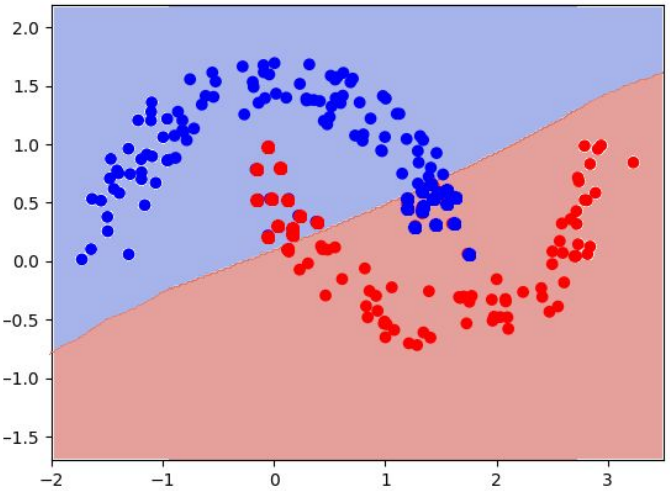
Plot for C=10 and γ=0.1(Polynomial and RBF):

POLYNOMIAL



Accuracy on the Test set: 79.0 %

RBF



Accuracy on the Test set: 95%

4) A 100 – 300 word report on your findings.

Solution 4)

From the validation accuracies of Q1, we observe that increasing both C and gamma improves the validation accuracy and thus improves the test accuracies over the two moon dataset. C is the cost parameter of the soft-margin cost function. It controls the effect of every support vector and the cost of misclassification. Increasing C(Hard Margin) forces the input data to explain the data in a stricter manner. This might cause overfitting. Hence in our case, the validation accuracy improves and we

get sharp boundaries for the test data which would have less violations of misclassification. Gamma is the kernel parameter. A small Gamma implies that the class of a support vector Y will have influence on deciding the class of vector X, even if they have large distance between them. A high value of Gamma implies otherwise, that the class of a support vector Y will not have high influence on deciding the class of vector X when they both have large distance between them. That is why increasing the value of Gamma on our model drastically improves the validation and test accuracies. In terms of kernels, the RBF kernel performs better on the two Moon dataset than the Polynomial kernel. This is because the RBF kernel can entail and explain higher complexity in the data than the polynomial kernel, as it has an exponential function. On the other hand, the Polynomial kernel would get saturated and providing it with more data, would not help it to explain the more complex relationships.
