Introduction to Machine Learning — *RUJUTA PALANDE : OT30378*

HW2

1. Explain in what case and why we need a regularizer and why l_2 norm regualzrizer is a reasonable one.

   Ans : Regulariser is needed to control the complexity of the model so that it doesn't overfit or underfit depending on the data. L2 norm pushes all weights towards zero. Because of this, it becomes more robust and simplifies the model to some extent. It is smooth under differentiation and easy to optimize.

2. What values of λ in the regularized loss objective will lead to overfitting? What values will lead to underfitting? Note that λ is a multiplier for the regularizer term.
   Ans : Smaller values of λ mean less regularization so this leads to overfitting and larger value leads to underfitting.

3. Explain why the squared loss is not suitable for binary classification problems.
   Ans : If the classification is binary, the labels are also binary in -1,1. The pedicted value will be a scalar quantity.So basically in this case,the squared loss will be enforced if the exact values of the labels is not calculated and predicted.

4. One disadvantage of the squared loss is that it has a tendency to be dominated by outliers – the overall loss $\sum_n (y_n - \hat{y}_n)^2$, is influenced too much by points that have high $|y_n - \hat{y}_n|$. Suggest a modification to the squared loss that remedies this.
   Ans : One modification that can be made to the squared loss function is the introduction of a variable to adjust the points in such a way that the outliers won't be considered.

5. Perceptron algorithm: (Note that you should use exactly the same testing data to be able to compare the results.)

   (a) Implement the perceptron algorithm for binary classification.

   (b) Train and test it for classifying digits "1" and "6" in MNIST dataset. Note that we don't need the data of other digits in this part. Please use randomly sampled 1000 training examples per class and 500 testing examples per class.

   (c) Plot the accuracy on the test set w.r.t. the number of iterations. Here, processing each data-point is considered one iteration, so 2000 iterations means one pass over all training data. You may choose to calculate the accuracy every 5 or 10 iterations if doing it for every iteration is expensive on your hardware.

   (d) Visualize the learned model in the image form to see if it makes sense. Note that the weight vector has both positive and negative values, so you should plot those on two separate planes.

   (e) Visualize the 20 best scoring and 20 worst scoring images from each class all from the testing set.

   (f) Randomly flip the label (add labeling error) for 10% of the training data and repeat (b) and (c).

   (g) Sort the data before training so that all "1"s appear before "6"s and plot the accuracy w.r.t. the number of iterations. Is it faster or slower to train a good model this way? Explain why.
   Ans : It is slower to train a good model this way. The main reason being the W first gets tuned to detect 1's and the model gets trained for that and later on when 6 is introduced,the W starts tuning again according to 6. So to achieve a good accuracy, such a training model will require time.

   (h) Repeat (b), (c), (d), and (e) for classifying "2" and "8". Is it faster or slower that the above problem in training a good model? Explain why.
   Ans : It is slower to train the above model in a good way. This is mainly because of the shape of the figures "2" and "8" whose pixels match upto an extent and so the training model takes a while to achieve the accuracy and correctly distinguish between 2 and 8.

   (i) Repeat (b) and (c) once with 10 training examples only and then with all available training examples (almost 6000 for each class).

Figure 1: Q.5C : Accuracy vs iteration graph

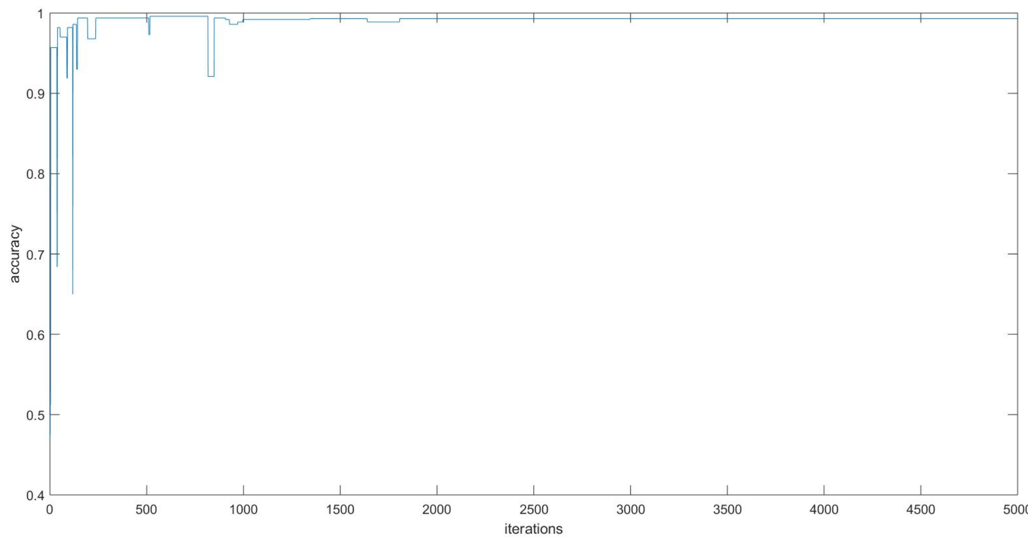

Figure 2: Q.5D : Visualized learning model

Figure 3: Q.5E : 20 best scoring 1s

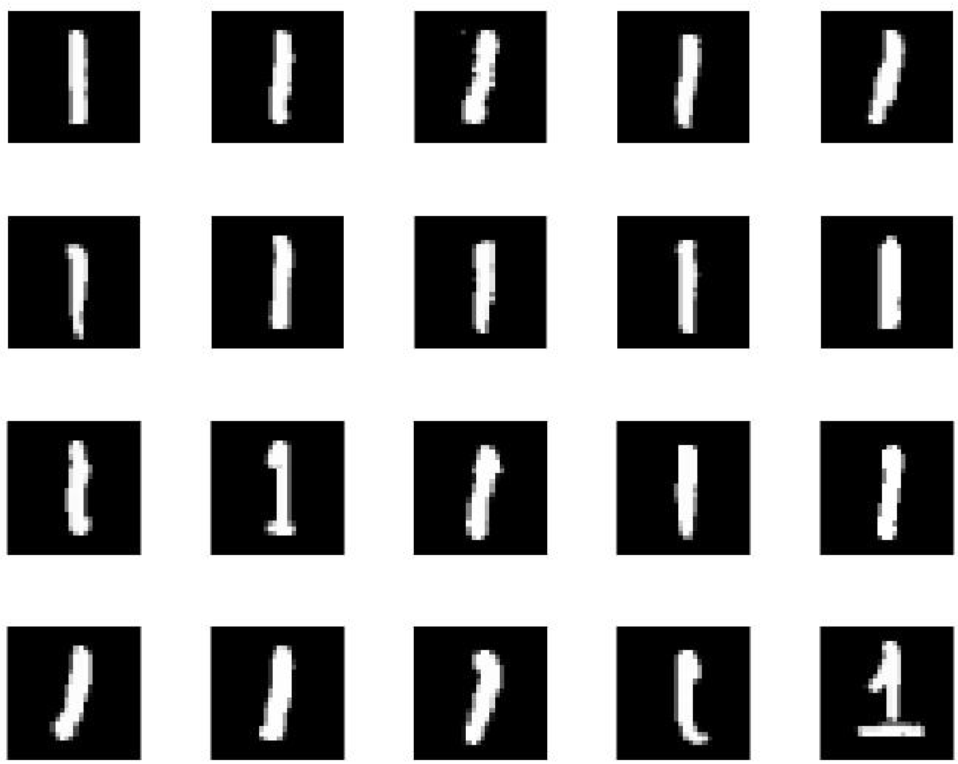

Figure 4: Q.5E : 20 worst scoring 1s

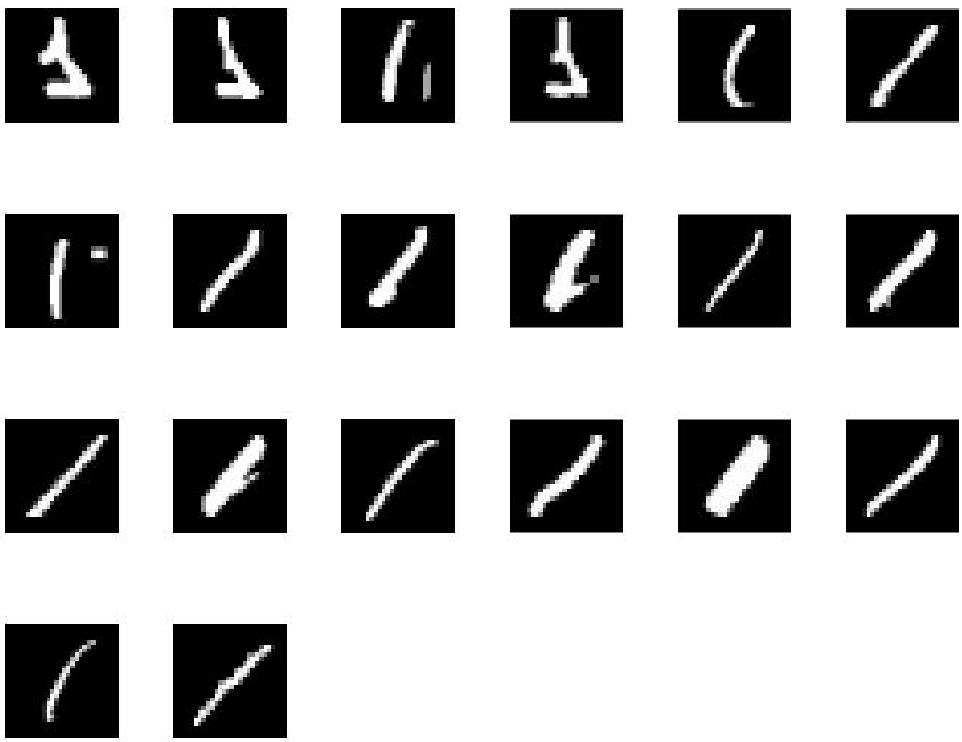

Figure 5: Q.5E : 20 best scoring 6s

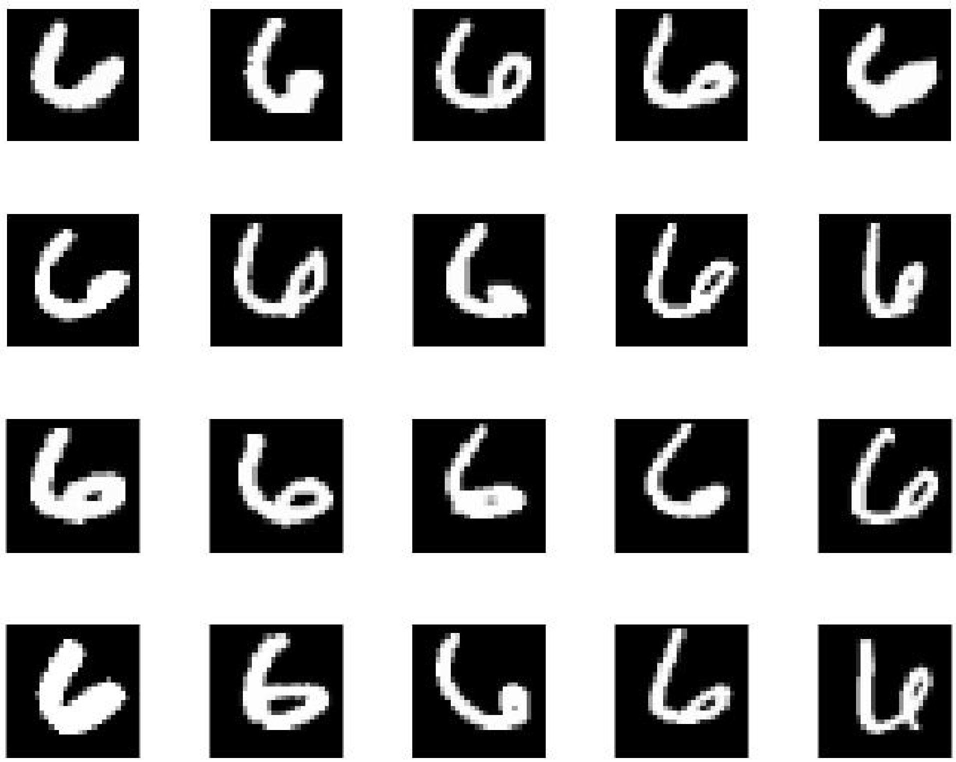

Figure 6: Q.5E : 20 worst scoring 6s

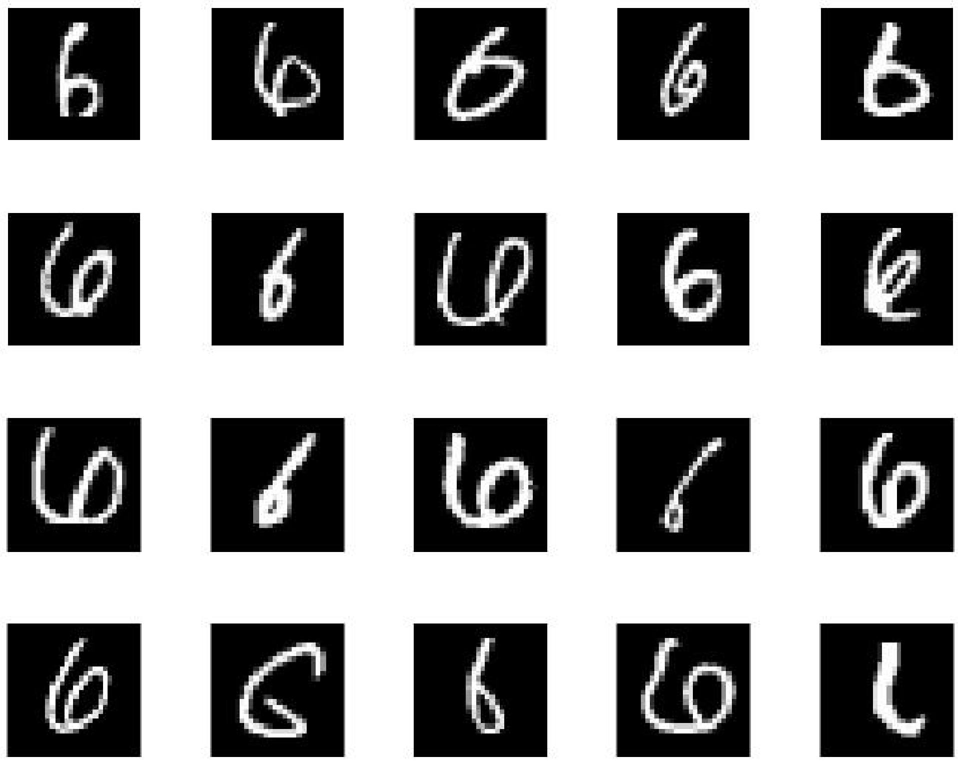

Figure 7: Q.5F : Accuracy vs Iteration for 10 percent error introducedl

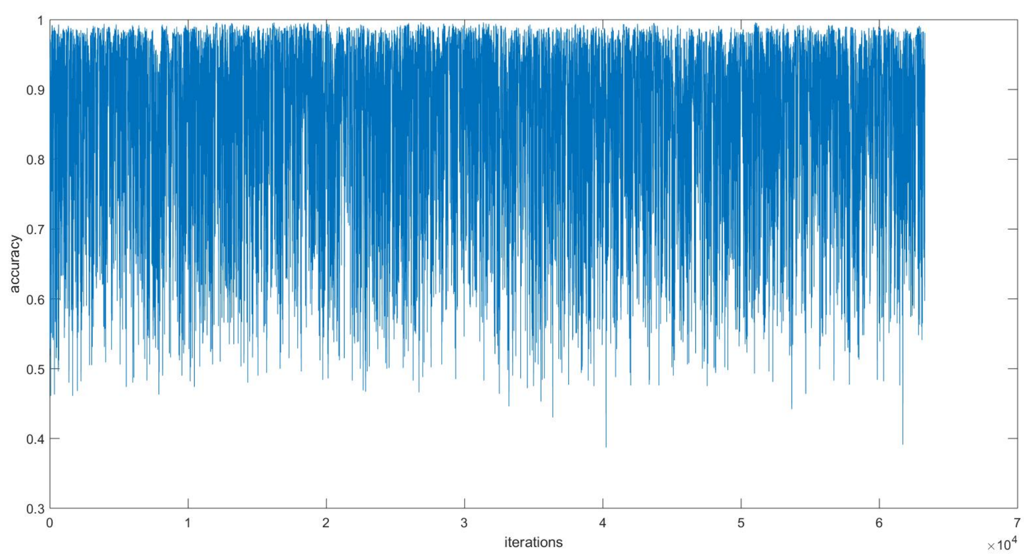

Figure 8: Q.5G : Accuracy vs Iterations for 1s before 6s

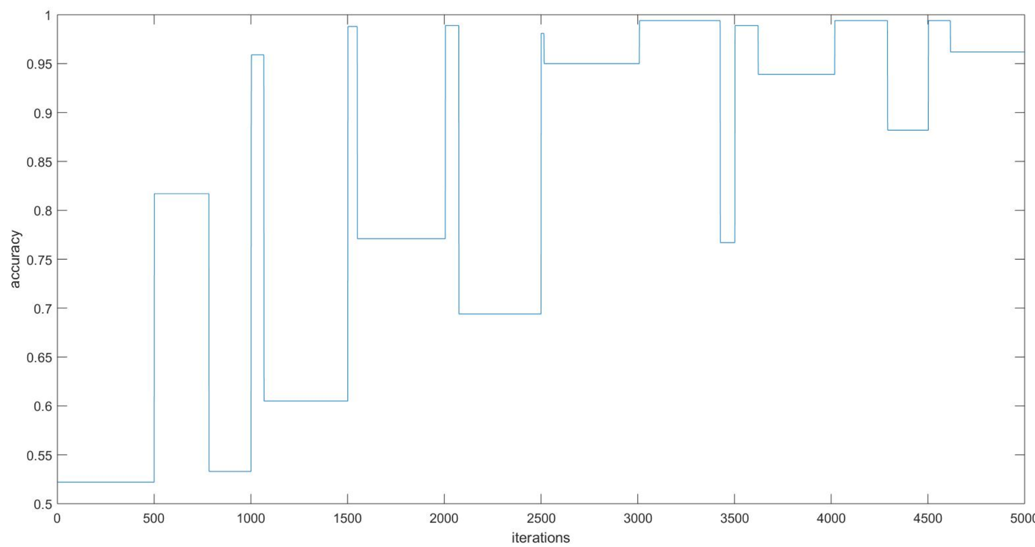

Figure 9: Q.5H : Repeating B and C : Accuracy vs Iteration graph

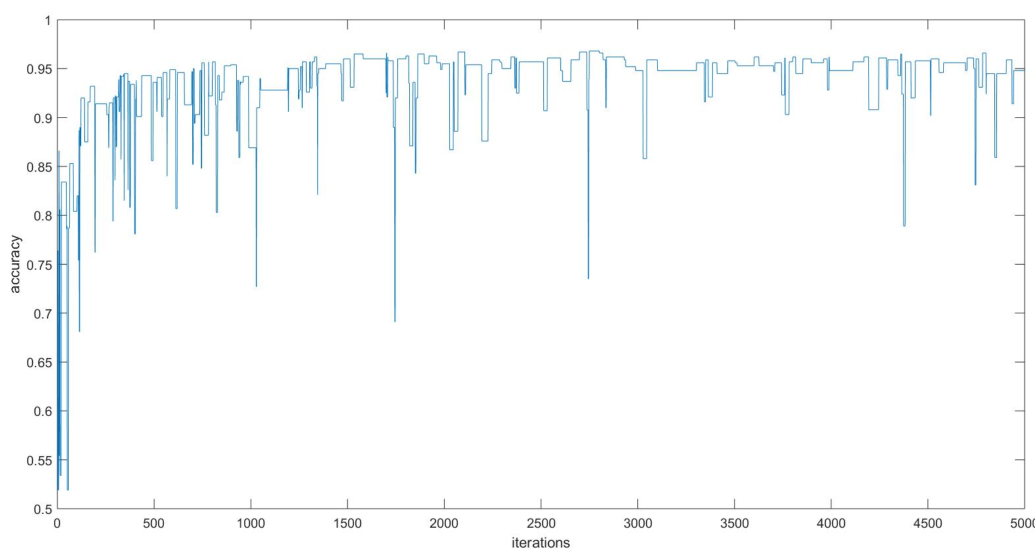

Figure 10: Q.5H : Repeating D : Visualizing the learned model

Figure 11: Q.5H : Repeating E : Visualizing 20 best scoring 2s

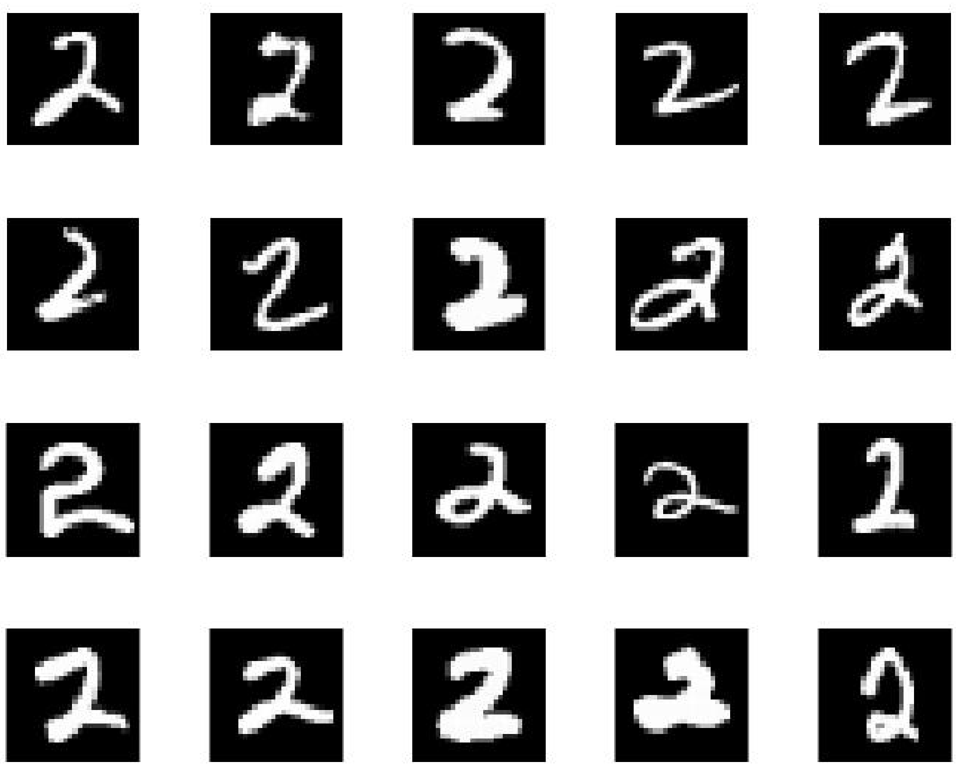

Figure 12: Q.5H : Repeating E : Visualizing 20 worst scoring 2s

Figure 13: Q.5H : Repeating E : Visualizing 20 best scoring 8s

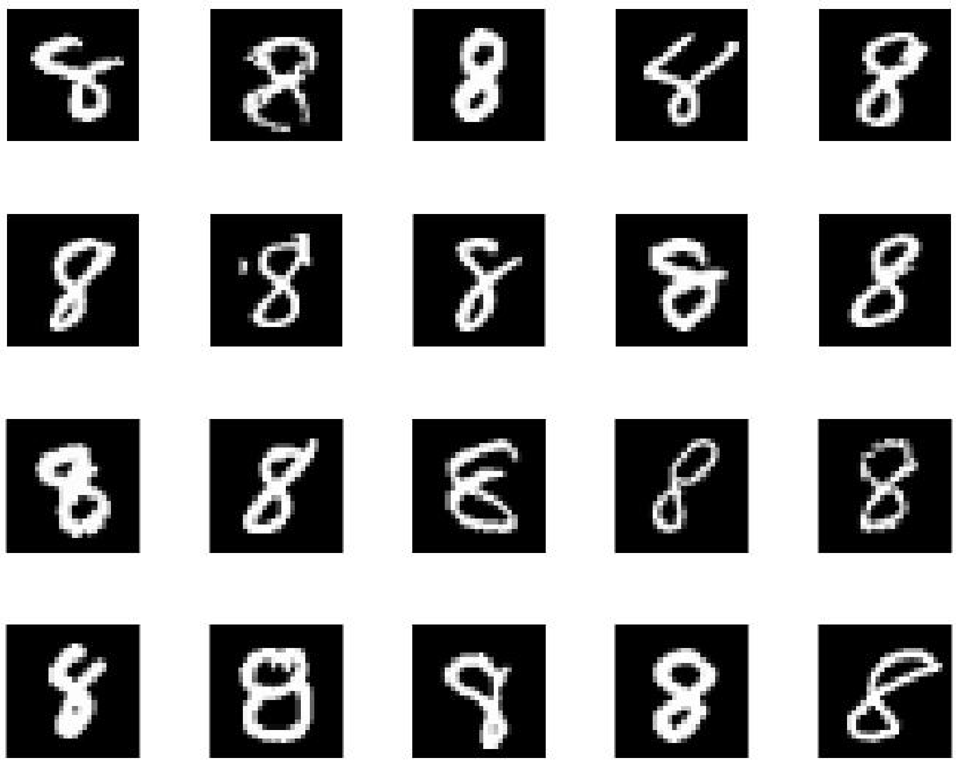

Figure 14: Q.5H : Repeating E : Visualizing 20 worst scoring 8s

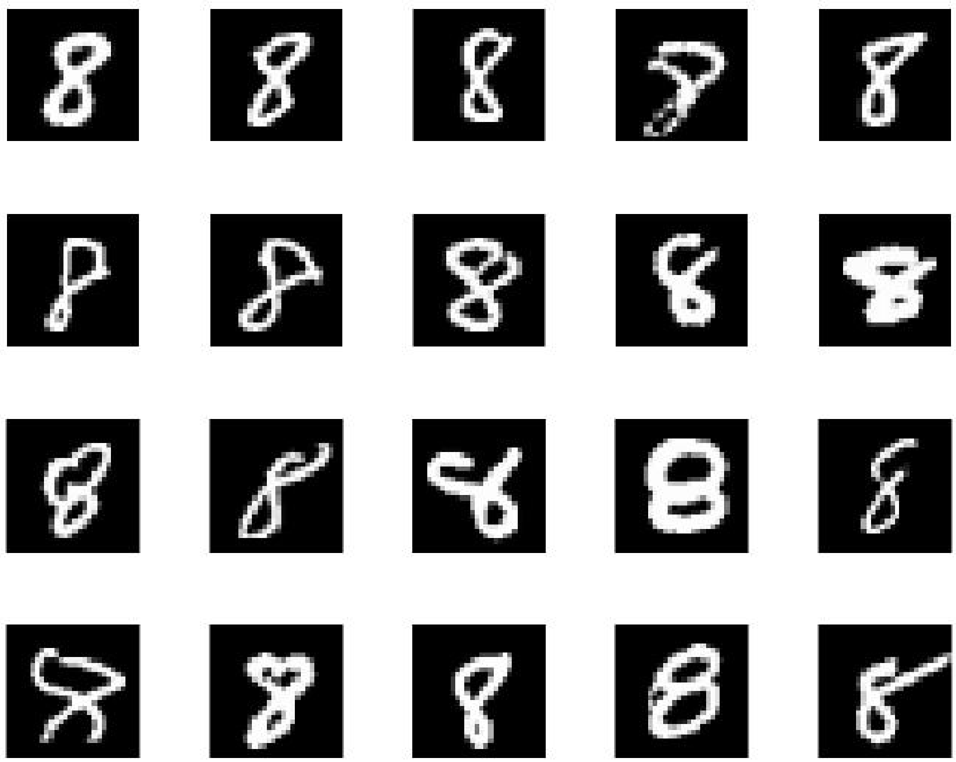

Figure 15: Q.5I :Accuracy vs Iterations for 10 training samples

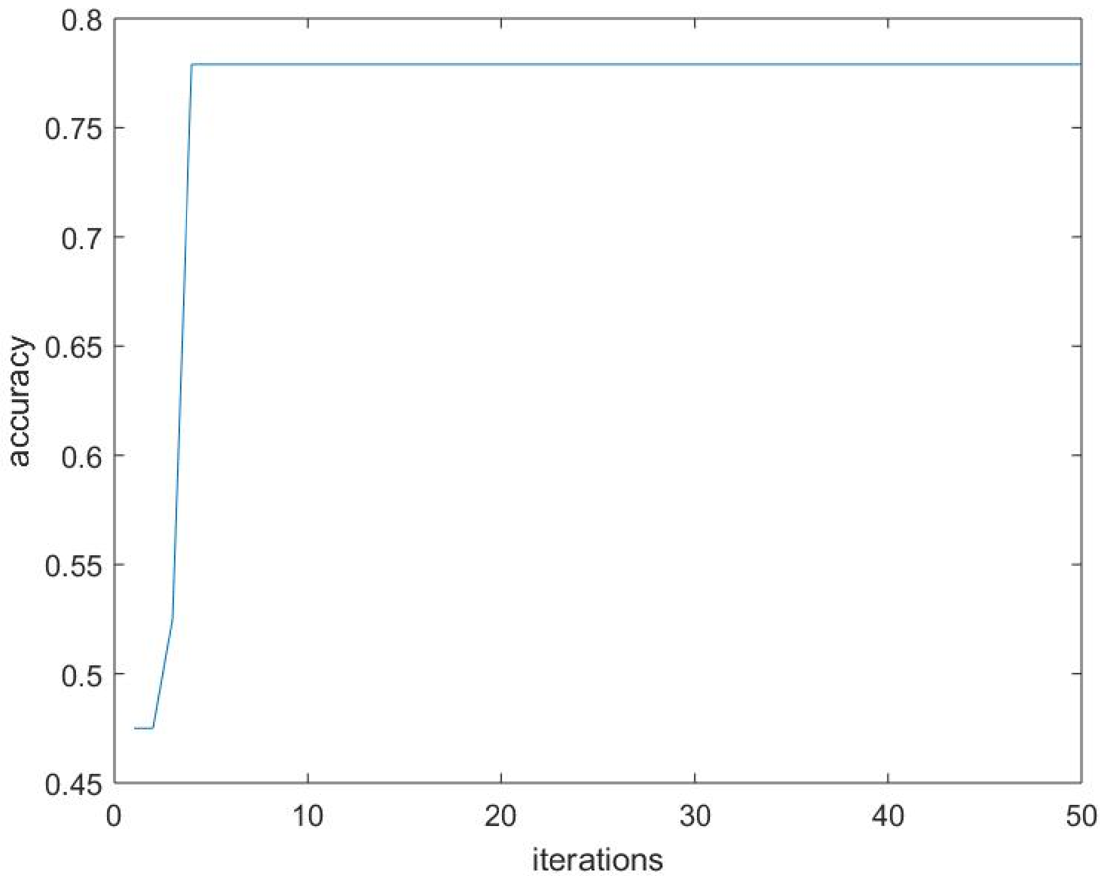

Figure 16: Q.5I : Accuracy vs Iterations for all training samples