MACHINE LEANRING ENGG6500

Assignment-2

Yaowen Mei

1177855

Yaowen Mei
1177855

Homework 2

ENGG 6500
November 15, 2021

Problem 1– Perceptron

Train a perceptron model to classify the MINIST handwritten digits dataset (we have 7500 data for training, and 1000 data for testing). The model should have 785 input neurons, to represent the 28 by 28 pixels input images, plus one threshold ($28^2 + 1 = 785$), and this model should have 10 output neurons without hidden layers.

Requirements for Training:

- Learning rate should be set to 0.01, 0.1, and 1;
- Input pixels should be normalized to zero to one, while the initial connecting weights should be set to random numbers with norms no greater than 0.05;
- Train for 50 epochs, record the training and testing accuracy for plot.

Requirements for Report:

1. Plot of accuracy vs. epoch number (from 0 to 50, including epoch 0) on both training and testing data.
2. Comments on these plots about whether overfitting and oscillation occurred for each learning rates.
3. Provide confusion matrix on the test data-set for each learning rate
4. Comments on the confusion matrix about which digits are classified most accurately, and which digits tend to be confused with one another.
5. Discuss on the result difference for different learning rates.

SOLUTION-REPORT:

I am actually plotting 4 learning rates (0.0001, 0.01, 0.1, 1), the reason for this is all the connecting weights are randomly initialized between -0.05 to +0.05 (as per the requirement), we need a learning rate which is about two orders smaller than the connecting weights to see the trend of training accuracy improvement in the scale of 100 epochs.

1. **Accuracy Plots:** For the 1st requirement, please see Figure 1 for the training/testing accuracy comparison between different learning rates;
2. **Oscillation and Over-fitting**:

- For learning rate 0.0001, there is almost no oscillation in the accuracy plot; while, there are obviously oscillations for learning rate 0.01, 0.1, and 1. The reason for the oscillation is that the learning rate is too large. On the contrary, too small learning rate might cause the machine learning model be trapped inside of a local optimal position and take too long to converge. Piratically, people will multiply a Gaussian function (a function of epoch number) to the constant learning rate, so that the learning rate is decreasing as epoch increasing, the model will be able to jump out of the local minimum and avoid oscillation.
- For overfitting, generally, large training epoch number and too many hidden layers (and hidden neurons) are the cause for overfitting. They do not really apply in our case (50 epoch is not too large, and we have no hidden layers). To demonstrate overfitting, I have included an accuracy plot for 300 epochs training with learning rate = 0.0001 (See Figure 2). From epoch 0 to 75, both of the training accuracy and the test accuracy are increasing with epoch number increasing; however, after 75 epochs, the training accuracy is increasing and the test accuracy is decreasing. This is where overfitting occurred. The model is trying to follow the training data too much, so that it lost the robustness for real unseen inputs (the test inputs).

3. **Confusion Matrix**: Please see Figure 3 for the confusion matrix after 50 epochs of training.

4. **Discussion of Confusion Matrix**: We use 1000 picture to test the accuracy of the model we just trained. For each digit, there are 100 test pictures. The model for learning rate 0.0001 is not fully trained yet (underfit, small learning rate need more epoch number to converge) at epoch 50, so the performance of this model is not as good as the other three models. As we can see from Figure 3, the digit 0 always have the best accuracy (99% with learning rate 0.01, and 98% with learning rate 0.1 and 1). The other two very accurate digits are digit 1 and digit 6, as both of them have more than 90% accuracy with all different learning rates. It also can be seen from the confusion matrix that

   - digit 2 tend to be confused with digit 1 and 3;
   - digit 4 tend to be confused with digit 6
   - digit 5 tend to be confused with digit 3
   - digit 7 tend to be confused with digit 5
   - digit 8 tend to be confused with digit 3
   - digit 9 is really really tend to be confused with 7 (10% chance with learning rate 0.01)

5. **Short Discussion on Learning Rate**: There are no significant difference between learning rate 0.01, 0.1, and 1 in terms of accuracy within 50 epochs. Both of them are

(a) Learning Rate = 0.0001

(b) Learning Rate = 0.01

(c) Learning Rate = 0.1

(d) Learning Rate = 1

Figure 1: Confusion Matrix Comparison Between 4 Different Learning Rates

too big (so that the accuracy plot is oscillating). While, if we set learning rate really small, like 0.0001 in Figure 1 subplot (a), we can see the model is not fully trained at epoch 50 (test accuracy can still be improved with more epoch). A good model will balance between large learning rate (oscillating) and small learning rate (does not converge until big epoch number, and be trapped in the local optimal position). To obtain a better performance, We can multiply a Gaussian function (a function of epoch number) to the constant learning rate, so that the learning rate is decreasing as epoch increasing, the model will be able to jump out of the local minimum, avoid oscillation and converge very fast.

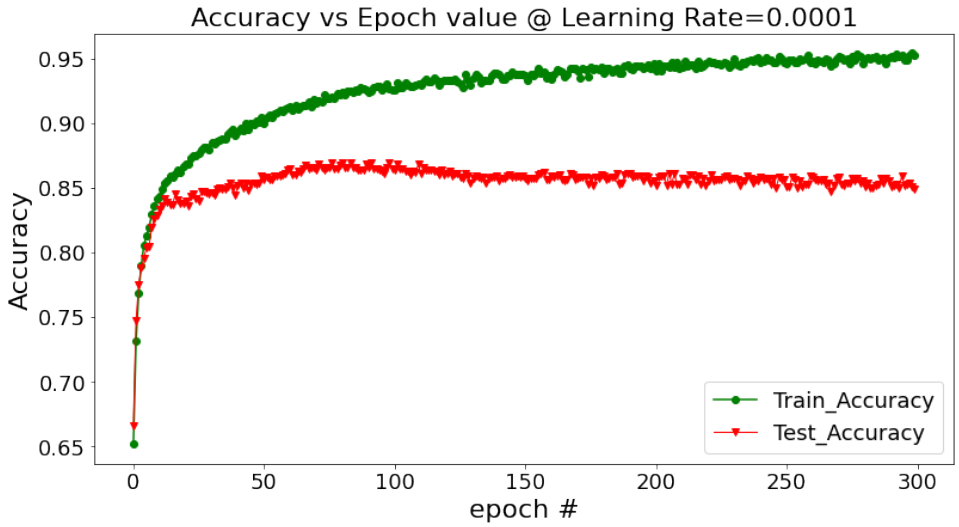


Figure 2: Accuracy Plot for Learning rate = 0.0001 over 300 epochs

Problem 2-Regression

Write a Python program to obtain the regression equations that represent the parking lot occupancy.

Requirements for Training:

- Parking fees for all the parking lots are equal;

Requirements for Report:

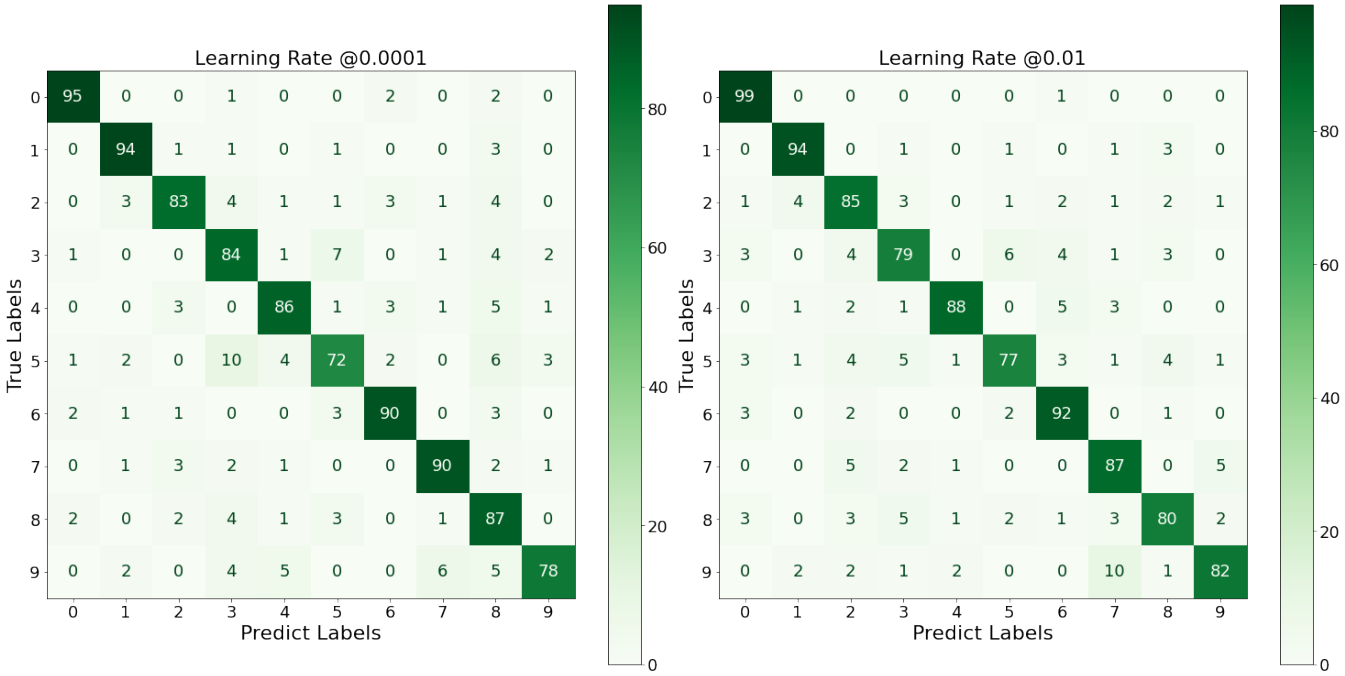
- Advise the City based on the regression about which parking lot should be enlarged;
- Rank the parking lots necessity for enlargement from the most to least urgent;

SOLUTION-REPORT:

I choose to use linear regression for this question, the regression expression is $y = kx + b$. Where k is the slope of the regression line, and b is the intercept of the regression line, x is the day number (from day 1 to the last day of the record for each parking lot), and y is the percentage occupancy of the parking lot.

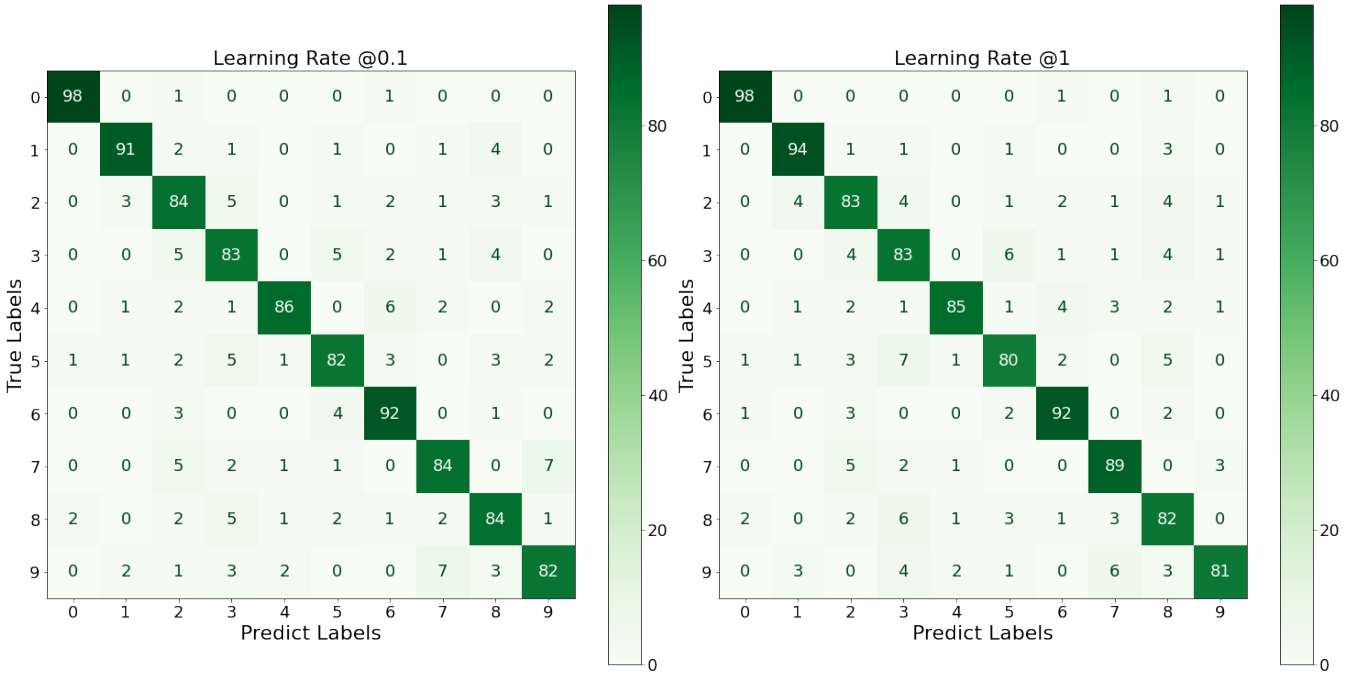
The percentage occupancy, y, is a percentage number between 0 and 1, which is defined as

$$y = \frac{\text{Num of Occupied Parking Spots}}{\text{Capacity of the Parking Lot}}$$



(a) Learning Rate = 0.0001

(b) Learning Rate = 0.01



(c) Learning Rate = 0.1

(d) Learning Rate = 1

Figure 3: Accuracy Comparison Between 4 Different Learning Rates

Please see Figure 4 for the linear regression plots, and the detailed linear regression result is presented in the Table 1.

The third column in the Table is the Person Correlation Coefficient (R):

The Correlation Coefficient (R) for Parking Lot 1 and Parking Lot 2 are -0.008, -0.005, which means there is no relationship or week indirect relationship between the number of days and parking lot occupancy rate. The Correlation Coefficient (R) for Parking Lot 5 is 0.75, means there is a week direct relationship between days and parking lot occupancy rate. The Correlation Coefficient (R) for Parking Lot 3 and 4 are 0.386 and 0.385. This means there is a intermediate directr relationship between the day number of the parking lot occupancy rate, for these two parking lots, $R^2 \approx 0.38^2 = 0.15$, which means, day number can explain about 15% of the parking lot occupancy. Overall, day number does not explain the parking lot occupancy rate well enough, so we need a more complex model to reduce the bias of the prediction.

The fourth column in the Table is the expected parking lot Occupancy Rate:

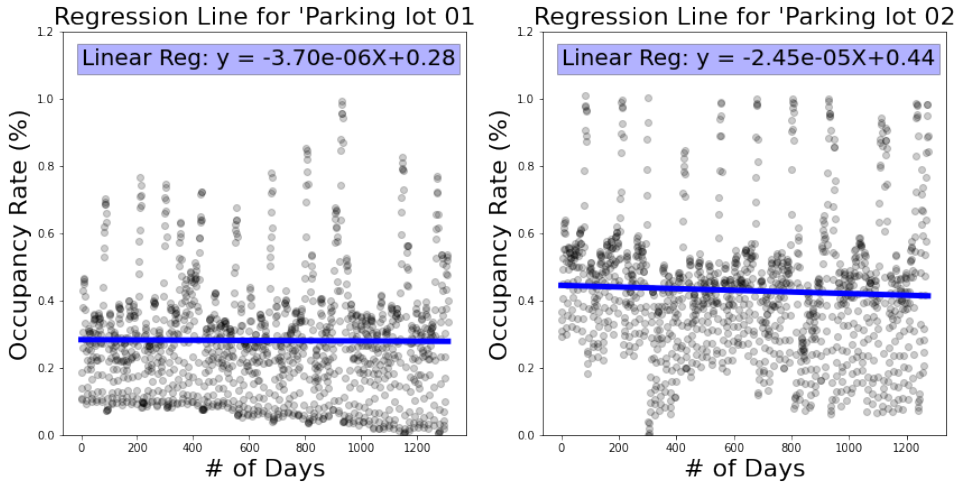
These numbers are calculated from each of the the regression line equation, based on the **last day number in the input historical data**. If we get expected occupancy rate equals to 1, then it means the parking lot is totally fulled and need to be enlarged right away, while, the 0 occupancy rate means the parking lot is empty and no need to be enlarged. Based on the value of this Predicted Occupancy Rate, I rank the priority rank for enlargement to be:

1. Parking Lot 4 (89% fulled already, need to be enlarged right away)
2. Parking Lot 3 (82% fulled already, need to be enlarged right away)
3. Parking Lot 5 (61% fulled, might need to be enlarged in the next few years)
4. Parking Lot 4 (41 % fulled, might need to be enlarged in the next few years)
5. Parking Lot 1 (28% fulled, does not need to be enlarged)

K-Mean Clustering with MINIST Dataset

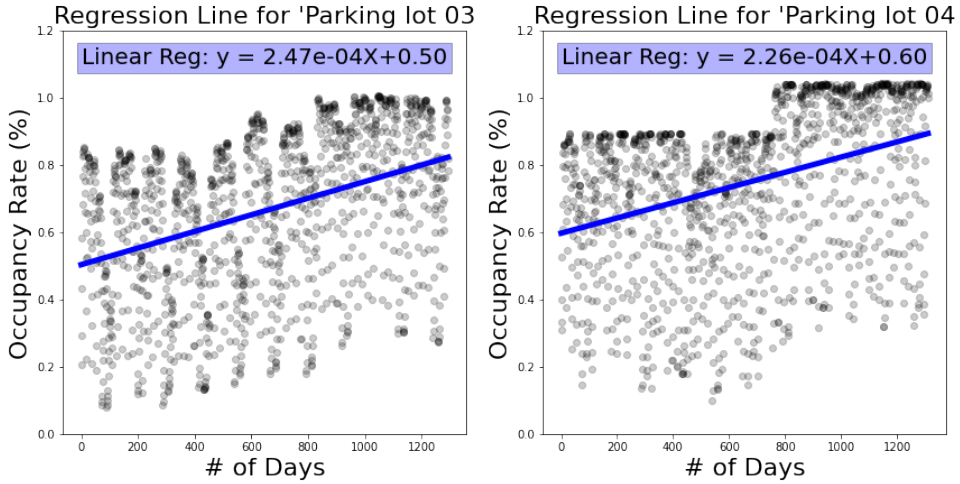
Requirements for Training:

- Write a function that takes two inputs, a list of N pictures to be clustered, and a number of clusters K in the function signature, and return:
    1. a list of N group assignment indices indicating the association of each picture to a specific group $(c_1, c_2, ..., c_N)$;



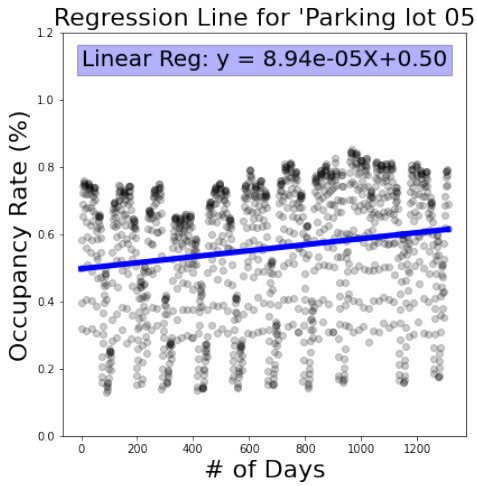
(a) Parking Lot 01

(b) Parking Lot 02



(c) Parking Lot 03

(d) Parking Lot 04



(e) Parking Lot 05

Figure 4: Linear Regression on Parking Lot Occupancy Based on Historical Data

	Slope	Intercept	Corr.Coef.	Occupancy Rate	Rank
Lot-1	-4E-6	0.28	-0.008	0.28	5
Lot-2	-2.5E-5	0.44	-0.005	0.41	4
Lot-3	2.5e-4	0.50	0.386	0.82	2
Lot-4	2.3E-4	0.60	0.359	0.89	1
Lot-5	8.9E-5	0.50	0.175	0.61	3

Table 1: The Slope, Intercept, Pearson correlation coefficient, Predicted Occupancy Rate, and Priority Rank for enlargement based on linear regression model for all 5 parking lots

2. a list of K groups of pictures ($\mu_1, \mu_2, ...\mu_N$);
3. a list of $J^{Cluster}$ value for each cluster fro each epoch until convergence (shape: 1 x epoch).

Initialize K-mean by randomly generate K data points as the K group representatives;

Let K = 20, Epoch Limit, P = 30, Comments on two runs with the highest and lowest $J^{Cluster}$ value, and :

1. Plot the value of $J^{Cluster}$ vs. Epoch Number;
2. Visualize the K groups as Images. Interpret the group representatives and compare them for these two runs.
3. Find the 10 nearest data points in each of the K group representatives. Manually list the digits (by eye estimation) in two tables of 20 (cluster numbers) by 10 (sample size), and record the percentage of classification.
4. Plot a few of the correctly classified as well as classified data points.

Repeat the same thing for K=10 and Epoch Limit, P = 20.

Repeat the same thing for K=5 and Epoch Limit, P = 10.

SOLUTION-REPORT:

First of all, I want to declare that:

- For the initialization, I randomly assigned K data points as the K center points.
- For the termination, I have tested that with 10 clusters, and the default tolerance rate, 0.0001, the MINIST Dataset takes roughly 68 epochs to fully converge. As the maximum epoch number we are asked to plot is 30, so the model will never terminate before the epoch number that we need to plot. I have added a default variable equals to 30 in my function signature to represent the termination epoch number.

For 10 Clusters, train 20 epochs

(a) As can be seen from Figure 5, the 1st run has the highest $J^{Cluster}$ result and the last one has the lowest $J^{Cluster}$ result.

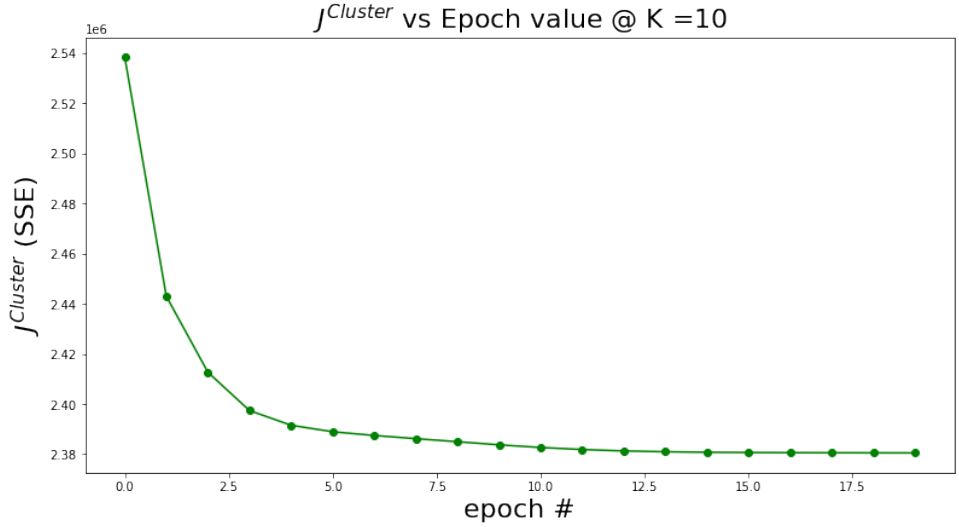


Figure 5: $J^{Cluster}$ vs. Epoch Number for 10 clusters and 20 epochs, J Cluster equals to 2359942

(b) Randomly plot 8 images from each cluster, the result is shown in Figure 6. As can be seen from the plots, this model is not fully trained yet. Cluster 0 to Cluster 9 are the clustering for the similarities between the handwritten digits. For example, Cluster-0 are mostly purely 0, but cluster 8 is kind of a mix of digit 9, 4, and 7, as they all looks similar.

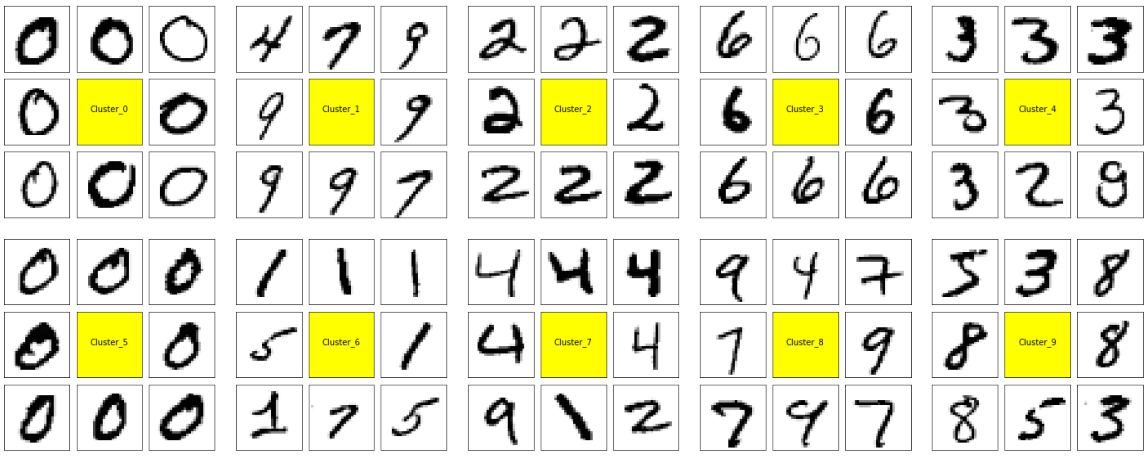


Figure 6: Clustering result for 10 clusters and 20 epochs (part B)

Label	*# Correct*	*# Wrong*	*Confused With*	*Real Name*
Cluster-0	9	0	NA	0
Cluster-1	9	0	NA	7
Cluster-2	9	0	NA	2
Cluster-3	9	0	NA	6
Cluster-4	9	0	NA	3
Cluster-5	9	0	NA	0
Cluster-6	9	0	NA	1
Cluster-7	7	2	9	4
Cluster-8	5	4	9	7
Cluster-9	9	0	NA	8

Table 2: The table view for Figure7

(c) Plot 9 closest images for each clusters. And manually record the miss-classification rate for each clusters:
I do not have space to print 10 of each clusters, so I will do 9 instead: the 9 most close to center images are presented in Figure 7. The statistical data is reported in Table 1. As can be seen from the table, there are no cluster for 5, and 9, but there are two cluster for 0, also there are high miss-classification rate between 4, 9, 7. The reason behind this is these digits are similar to each other in the vector space. The model is under-trained at epoch 30. We need train it with more epoch for it to converge.

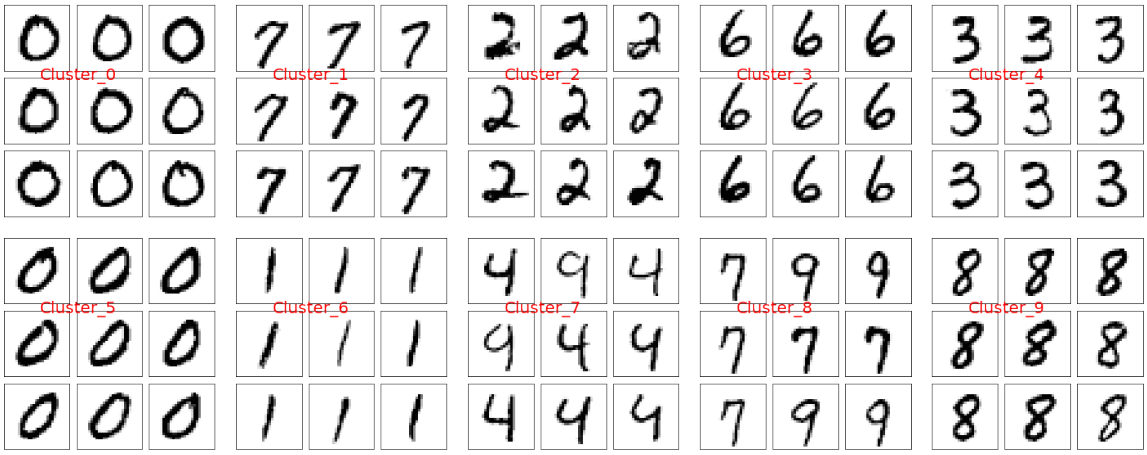


Figure 7: Clustering result for 10 clusters and 20 epochs (part c)

For 10 Clusters, train 20 epochs

(a) As can be seen from Figure 8, the 1st run has the highest $J^{Cluster}$ result and the last one has the lowest $J^{Cluster}$ result.

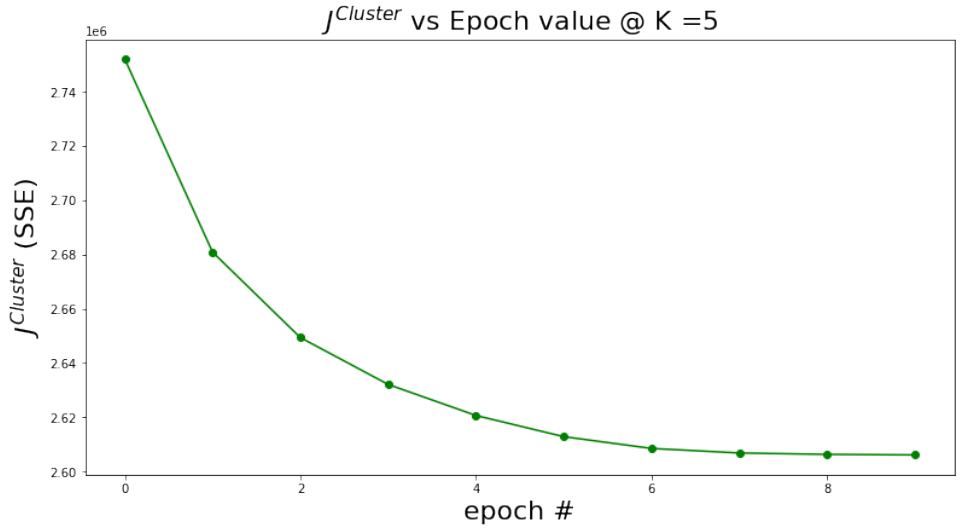


Figure 8: $J^{Cluster}$ vs. Epoch Number for 5 clusters and 10 epochs, J Cluster equals to 2606122.670077964

(b) Randomly plot 8 images from each cluster, the result is shown in Figure 9. As can be seen from the plots, this model is not fully trained yet. Cluster 0 to Cluster 4 are the clustering for the similarities between the handwritten digits. For example, Cluster-0 are mostly purely 6, but cluster 4 has a mix of 5 and 3

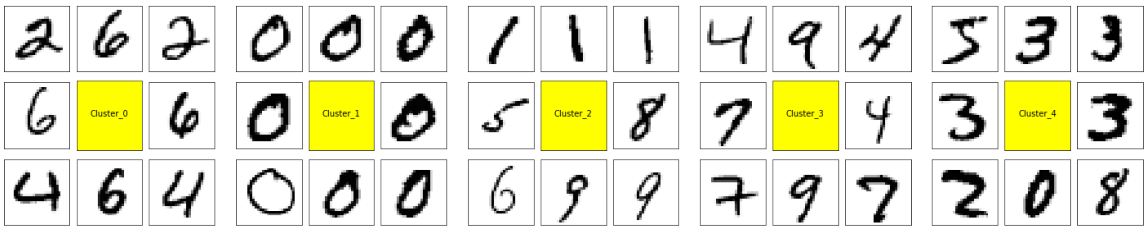


Figure 9: Clustering result for 5 clusters and 10 epochs (part B)

(c) Plot 9 closest images for each clusters. And manually record the miss-classification rate for each clusters:
I do not have space to print 10 of each clusters, so I will do 9 instead: the 9 most close to center images are presented in Figure 10. These images looks nice and clean, the only miss-classification is in cluster-4, there is a digit 5 mixed into 3.

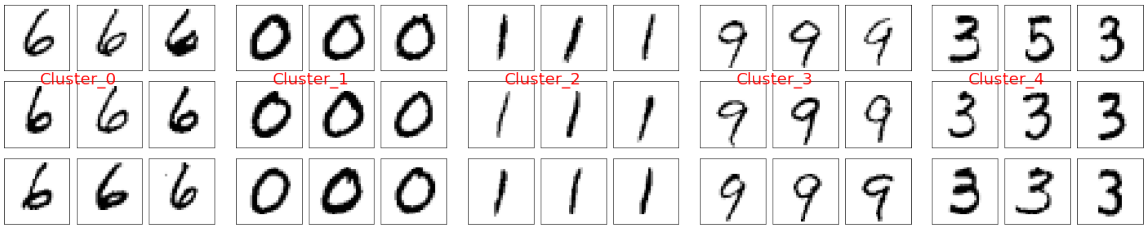


Figure 10: Clustering result for 5 clusters and 10 epochs (part c)

For 20 Clusters, train 30 epochs

(a) As can be seen from Figure 11, the 1st run has the highest $J^{Cluster}$ result and the last one has the lowest $J^{Cluster}$ result.

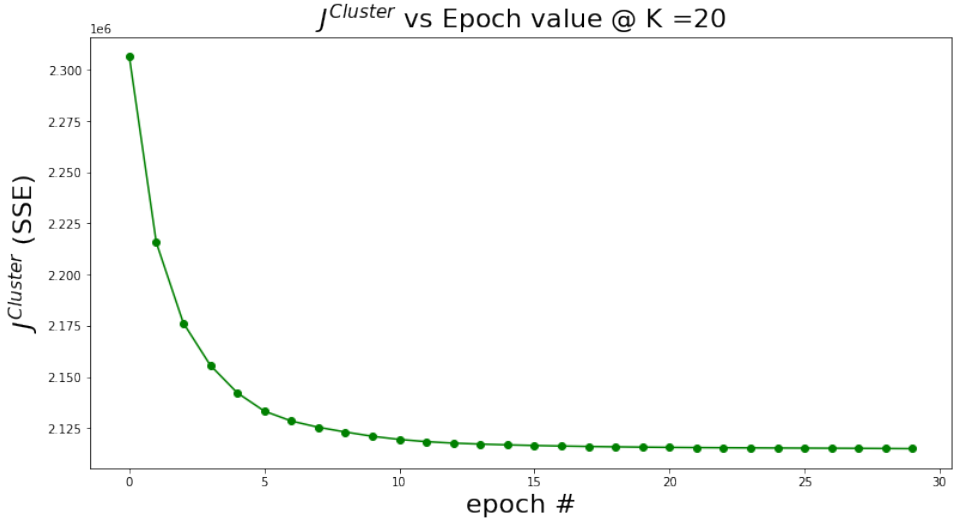


Figure 11: $J^{Cluster}$ vs. Epoch Number for 20 clusters and 30 epochs, J Cluster equals to 2114948.3991600866

(b) Randomly plot 8 images from each cluster, the result is shown in Figure 12. As can be seen from the plots, this model showing more details than the previous two models. This model has 20 clustrers, so it is presenting" in cluster-3, some digit 4 looks like digit 1, in cluster-10, some digit 4 looks like digit 5. Noticed that cluster 5 and cluster 14 are both clusters for digit 1. This is because this model is not trained yet. Or maybe because I have a bad initial value to start the move of the cluster centers.

Figure 12: Clustering result for 20 clusters and 30 epochs (part B)

(c) Plot 9 closest images for each clusters. And manually record the miss-classification rate for each clusters:
I do not have space to print 10 of each clusters, so I will do 9 instead: the 9 most close to center images are presented in Figure 13. These images looks nice and clean, the only miss-classification are:

- In cluster 4, there are half 4 and half 9
- In cluster 6, there are half 7 and half 9
- In cluster 9. there is a 0 mixed into 8
- Cluster 10, 13 are a bad mixtures
- Cluster 11 we have a 9 and 0 mixed into 3
- Cluster 12 has a 9 mixed into 7

- Cluster 14 has a 2 mixed into 1
- Cluster 15 has a 7 and 9 mixed into 4
- Cluster 18 has a 5 mixed into 3

in cluster-4, there is a digit 5 mixed into 3.

Figure 13: Clustering result for 5 clusters and 10 epochs (part c)