Machine learning homework 7 report

Q1: Try to modify the code a little bit and make it back to symmetric SNE (therefore you need to first understand how the t-SNE is implemented and find out the specific code piece to modify)

A1: The only different between symmetric SNE and t-SNE is that the distribution of latent space. The symmetric SNE adopts gaussian distribution as the measure of lower dimensional space. On the contrary, the t-SNE uses student-t distribution. We can just revise this part and the result can be gained. You can check my code for more detail.

Q2: Try to visualize the embedding of both t-SNE and SNE and discuss their differences.

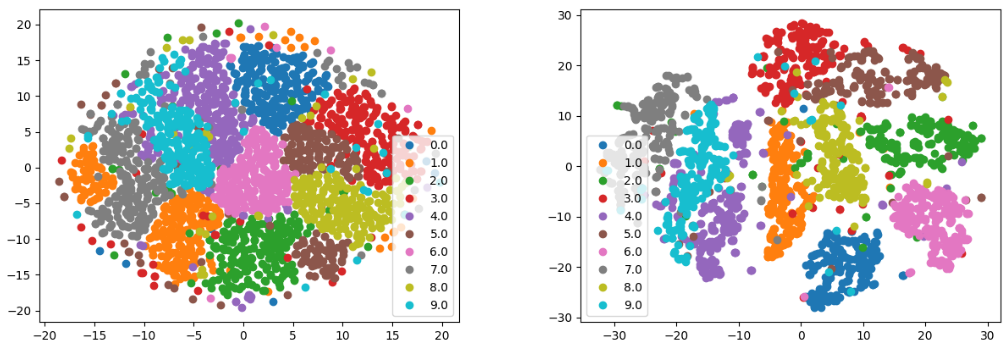

Figure 1. The visualization of the embedding

A2: The Figure 1 shows the visualization result of embedding. The left part is the result of SNE, and the right part is t-SNE's. The both experiments are under 200 iterations. As you can see, the distribution of left side is more crowded. However, the result of t-SNE can be separated more clearly. This phenomenon shows the advantage of t-SNE visualization that it can shrink the data distribution which are highly correlated.

On the other hand, I also compare the convergence phenomenon in my experiments. The Figure 2 shows the cost curve of both cases. The left one is the SNE result, and the right one is t-SNE's. In both experiments, we record the loss value for each 10 iterations. As you can see, the process of t-SNE can converge more rapidly than SNE, and both methods can get the low cost at the end.

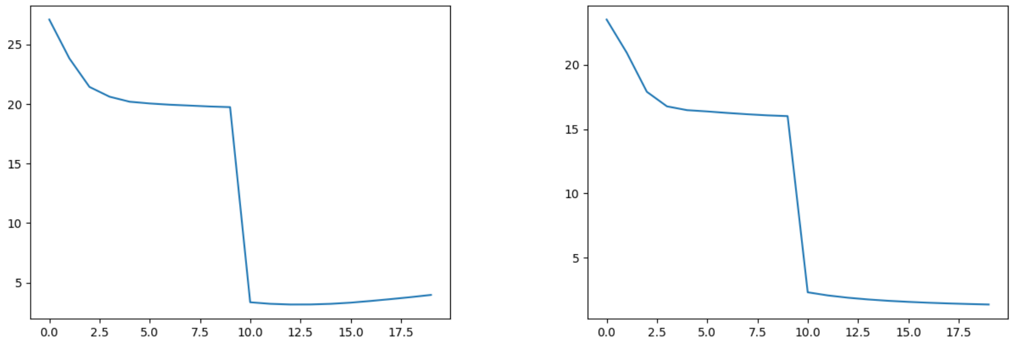

Figure 2. The cost curve of both experiments

Q3: Try to visualize the distribution of pairwise similarities in both high dimensional space and low-dimensional space, based on both t-SNE and symmetric SNE

A3: In this bonus question, I try to use confusion matrix to present the relationship of pairwise similarities. However, It's not very clear.

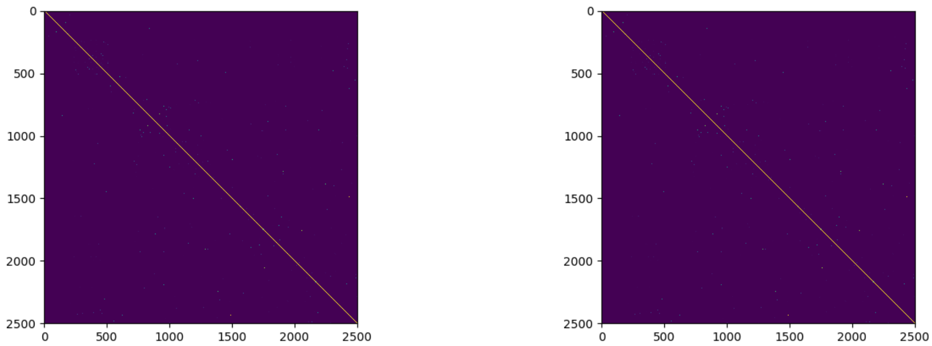

Figure 3. The confusion matrix of pairwise similarities in high dimensional space

The reason why the visualization isn't clear is that the program will try to reduce the Shannon entropy in the construction process of high dimensional distribution. If we try to make this reducing process stop at first step, the result is shown in Figure 3. There're more correlation square can been observed in the Figure 3 rather than deleting reducing the Shannon entropy. You can check the original image for more detail.

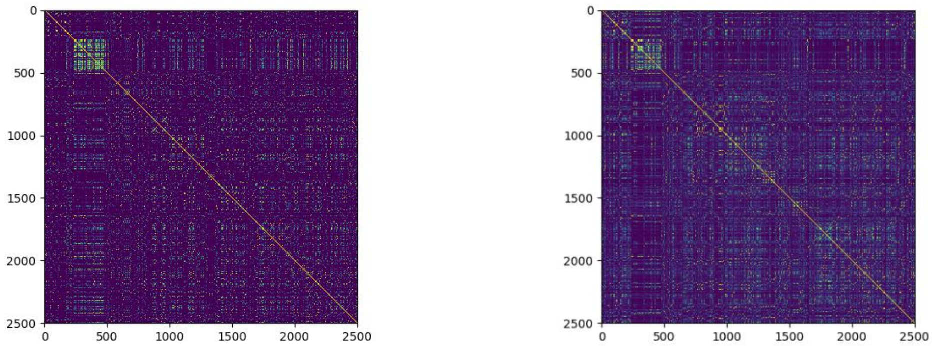

Figure 4. The confusion matrix of pairwise similarities in low dimensional space

For this situation, we try to make the low dimensional distribution conduct converge process for only 30 iterations. The corresponding result is shown in Figure 4. In the result of lower dimensional space, the correlation between each digit can be more clearly seen. By this experiment, the converge process can be imaged as that we make the result of Figure 4 converge and gain the result of Figure 3. Unlike the common sense of usual confusion matrix representation, we should make the correlation block be more unclear.

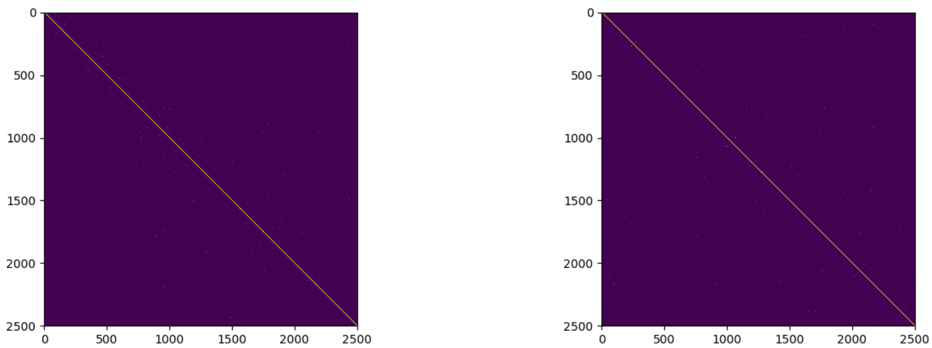

Figure 5. The confusion matrix of similarities with 50 reduction of Shannon entropy

For the formal experiment, we set the Shannon entropy reduction process iteration as 50. Next, we do the same experiment again, and the result is shown in Figure 5. As we can see, the points distributions are more blur than Figure 3.

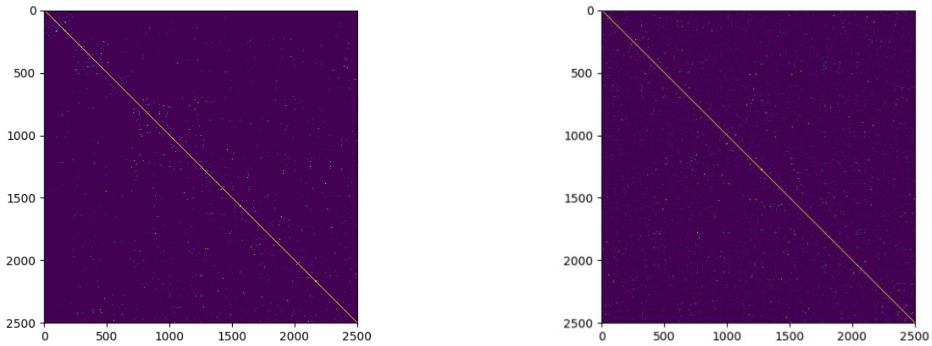

Figure 6. The result in low dimensional space with 50 reduction of Shannon entropy

Then, we set the loopy iteration as 200 to let the low dimensional space converge. The corresponding distribution result is shown in Figure 1, and the confusion matrix of pairwise similarity in low dimensional space is shown in Figure 6. As we can see, the distribution of points in confusion matrix is more blurred than Figure 4, and it's closer to the result of confusion matrix in high dimensional space (Figure 5). By this experiment, the evidence can be proven that the confusion matrix can be clarify from noise situation to clear result by iteration process. Moreover, the more separable result can be gained.

Q4: Try to play with different settings of perplexity, and see if there is any change in visualization.

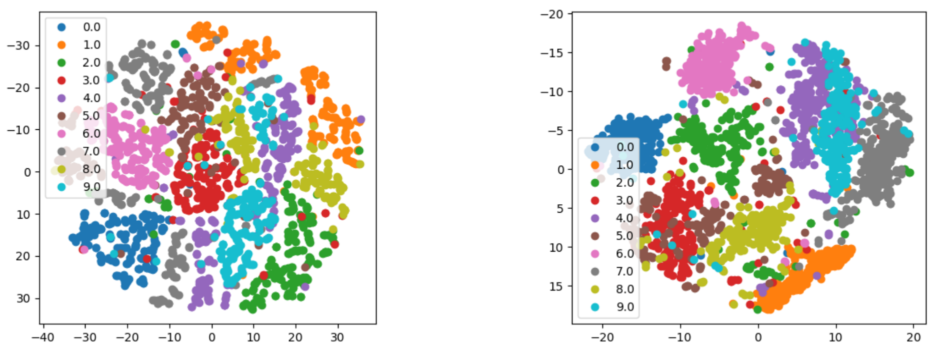

Figure 7. The result with different perplexity setting

In the original setting, the perplexity is set as 30. On the other hand, I try to set with others value. The Figure 7 shows the results of different perplexity setting. The perplexity is set as 5 in left experiment while it’s set as 100 in the right. As you can see, the more the perplexity it is, the more gather the distribution does. On the other words, we can get larger margin between different classes.

Q5: submit a report: what you have done (code, testing performance), what you have visualized, what you have learned.

In this homework, I learn the whole process of t-SNE. By knowing the background knowledge and code tracing, I can realize the function of the t-SNE. Moreover, the difference between traditional symmetric SNE and t-SNE can be visualized by this work. By setting iteration as 200, we can easily see the result distribution with different methods.

For the conclusion, t-SNE is a very popular data visualization approach in recent years. By this homework, it’s a great chance to implement the data visualization.