Exercise 4

Hou, Jue

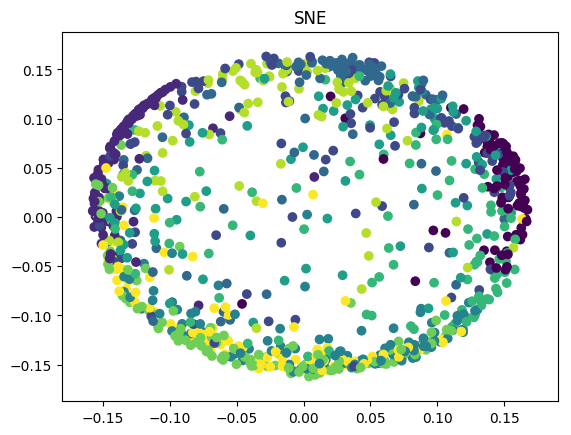
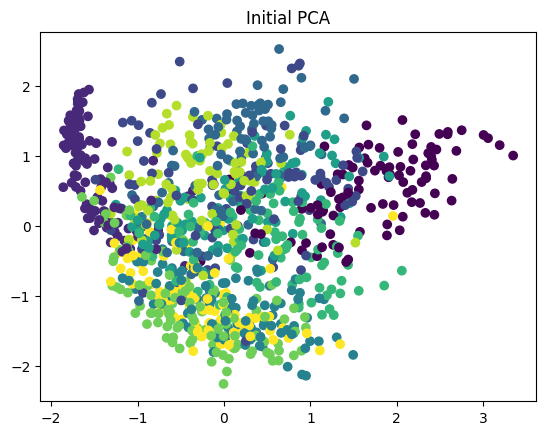
014695647

1. Regularized linear regression

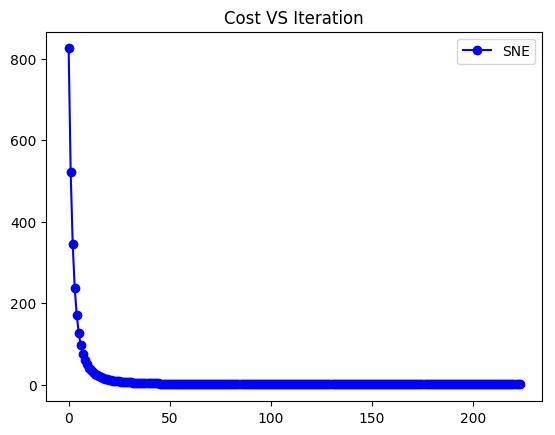
With increasing, the solution will decrease to 0, which leads to that estimate will be pulled to 0

1. Stochastic neighbor embedding

The following two plot visualize initial PCA and the final SNE representations:



The following plot shows training loss as a function of the algorithm iteration.



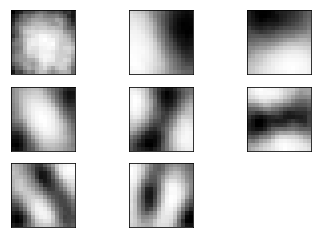
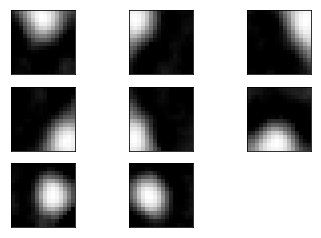
The of and will control the probability of that data point i is a neighbor of data point j in the original space and in the low-dimensional space respectively. and will match for the neighborhood of each point from original space to the project space. The reason why we should use large value for former one is that in this dataset the Euclidean distance between two points is extremely large. By using large value, it can decrease the effect of large .

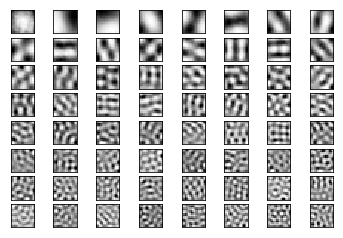
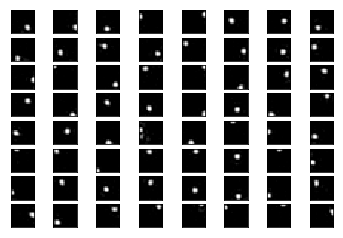
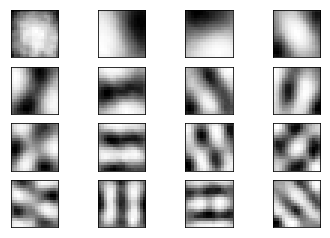
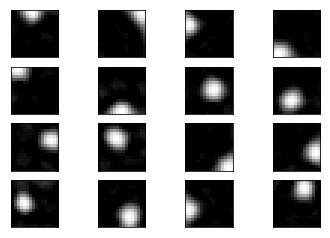
1. Non-negative matrix factorization

|  |  |  |
| --- | --- | --- |
| K | Final Training Loss(NMF) | Loss(PCA) |
| 8 | 225129.56 | 221364.13 |
| 16 | 168906.89 | 162254.36 |
| 64 | 62369.67 | 54352.83 |

With the increasing value of K, both the loss of NMF and PCA decreased. It is noticeable that, in all choices of K, PCA performs better than NMF. The reason for that, I think, is because that NMF is constrained to have non-negative numerical result in every dimension, while PCA is not constrained in this way. Therefore, PCA may fit the data better, though there might be negative value. Another reason is that what NMF find might only be a local optimal solution. What PCA find are actually the global optima. That’s why PCA performs better than NMF.

The following pictures are the visualization of NMF and PCA for K=8, K=16 and K=64:





I think K=16 is the most useful choice. We can interpret from its visualization. Meanwhile, a relative small loss can be obtained.

Estimated complete time: 7.5h