CIS 5525 Project 1

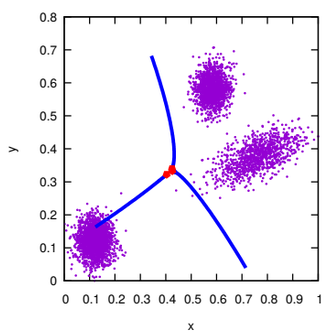
Zhijia Chen

2020-03-01

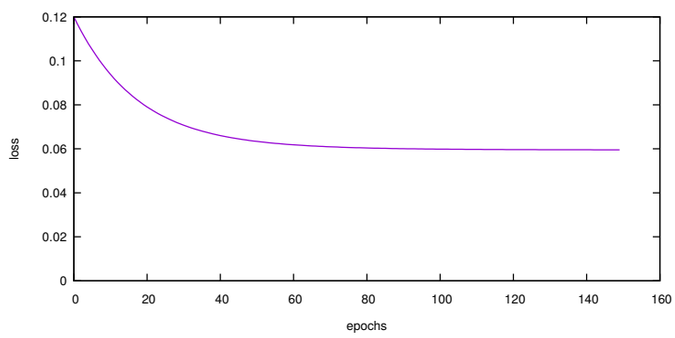
Task 0. In this task, I use the following loss function:

$$loss = \sum_{i=i}^{N} \left(\sum_{k=1}^{K} \|x_i - u_k\|^2 \frac{e^{-\alpha\|x_i - u_k\|^2}}{\sum_{k'=1}^{K} e^{-\alpha\|x_i - u_{k'}\|^2}} \right)$$

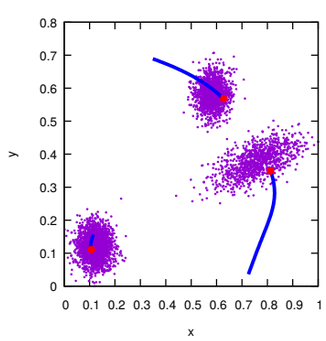
The number of target clusters K is set to 3. The data is normalized so that its mean is 0 and standard deviation is 1. I apply SGD to optimize the loss function. I first try different α values. Figure 1 show the clustering results and the loss curve function over epochs with different α values. The upper two sub-figures shows the results when $\alpha = 1$ and the bottom two show the case for $\alpha = 10$. In the experiments, I find that the loss function is likely to converge faster with a greater α value, but the gradient is also likely to run into 'nan' problem if the scale of α is chosen improperly which is possibly due to gradient overflow.



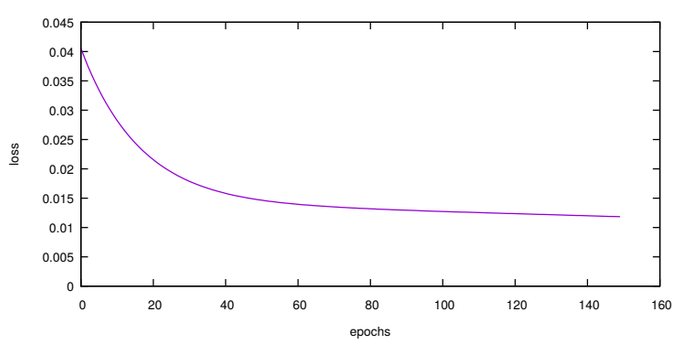
(a) Clustering result for α=1.



(b) Loss function curve for α=1.



(c) Clustering result for α=10.



(d) Loss function curve for α=10.

Figure 1: Clustering result and loss curve function with different α values.

Task 1. In this task, I try to perform unsupervised clustering on 10000 samples of the MNIST dataset in the original feature space. The loss function

$$loss = \sum_{i=1}^{N} \left(\sum_{k=1}^{K} \|x_i - u_k\|^2 w_{ik} \right)$$

Where w_{ik} is the probability that x_i belongs to cluster k. I use two methods to optimize the loss function. The first one is define w_{ik} as a function of the distance between the x_i and the cluster center k. For this method, I will use the same loss function as Task 0, where w_{ik} is computed using the softmax function. For the second method, I will treat w_{ik} as independent variable that being optimized with cluster centers.

Again, I use the SGD to optimize the loss function, and the *alpha* is set to 10 for the method 1. After each epoch, I assgin each sample to the cluster of the maximum probability and then compute the accuracy. Figure 2 shows the accuracy function curve over epochs. I find that it's hard for both the methods to cluster the samples in the original feature space. The clustering accuracy using the first method experiences some fluctuation during the training process but never goes beyond 0.3. On the other hand, the accuracy of the second method remains unchanged at 0.15, and the reason is that all the samples are assigned to the same cluster. Since I do not expect a good clustering performance in the original feature space, I do not try to improve the model, but focuse on the deep clustering method in Task 2.

Task 2. In this task, I first follow [1] to implement the deep clustering which optimizes the reconstrurction loss and the clustering loss jointly. But the performance of my implementation turns out to be very poor. I then use the method proposed in [2] which pre-trains the autoencoder first and then jointly optimized the cluster centers and the parameters of the autoencoder. After the pretaining stage, the cluster centers are initialized using K-Means++, and the model is optimized on an objective function that defined as a KL divergence loss between the soft assignment q_i and the auxiliary distribution p_i:

$$loss = KL(P\|Q) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_{ik} log \frac{p_{ik}}{q_{ik}}.$$

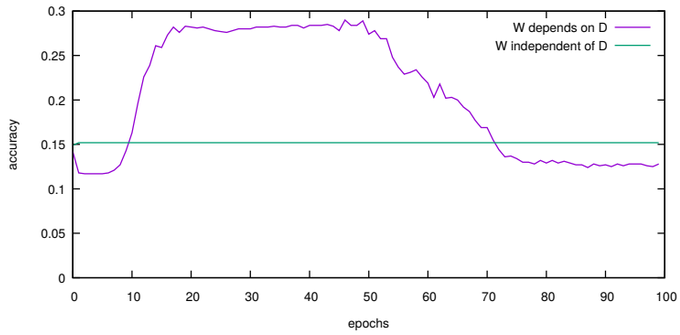


Figure 2: Clustering accuracy in the original feature space with different W.

Where q_{ik} is the similarity between embedded point $z_i = f_\theta(x_i)$ and centroid u_k that measured by t-distribution kernel function, and p_{ik} is computed by first raising q_{ik} to the second power and the normalizing by frequency per cluster:

$$p_{ik} = \frac{q_{ik}^2 / f_k}{\sum_{k'=1}^{K} q_{ik'}^2 / f_{k'}}.$$

I run the pretraining process for 100 epochs and then fine tune the whole model for 200 epochs, all the parameters are set according to [2]. Figure 3 shows the clustering accuracy over epochs and Figure 4 shows the clusters visualized using t-SNE with each point colored according to its ground truth label.

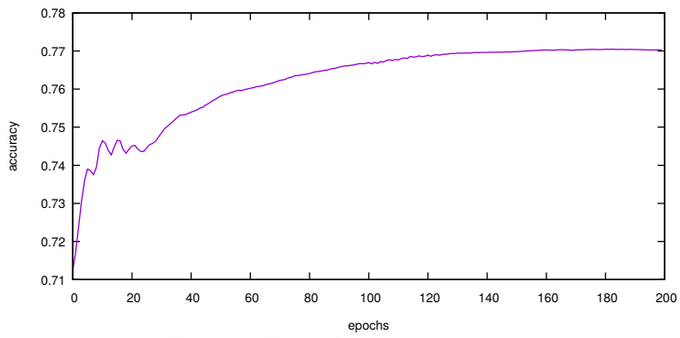


Figure 3: Deep clustering accuracy.

References

[1] M. M. Fard, T. Thonet, and E. Gaussier. Deep k-means: Jointly clustering with k-means and learning representations. *arXiv preprint arXiv:1806.10069*, 2018.

[2] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

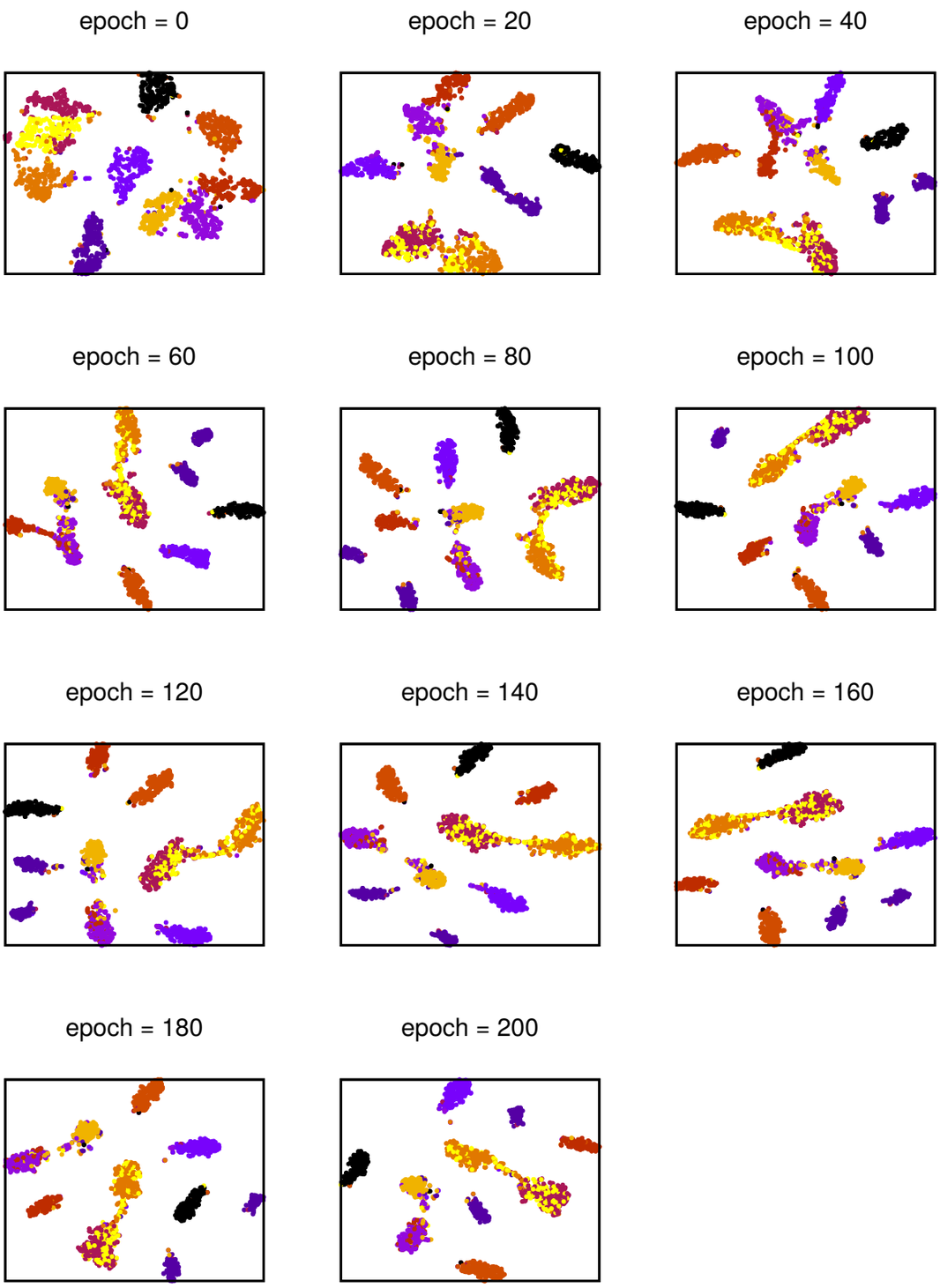


Figure 4: Clusters over epochs.