|  |  |
| --- | --- |
| Lab 4: Clustering and Dimensionality Reduction | |
| Student ID: | Student Name: |

Table 1. Clustering accuracies on FashionMNIST dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Initialization | Mean | Std |
| Image | Random initialization |  |  |
| Image | Kmean++ initialization |  |  |
| Latent representation | Kmean++ initialization |  |  |

**1. Kmeans (3pt)**

1. Implement kmeans function in src/kmeans.py (Hint, use implemented building blocks in src/kmeans.py
2. Run 01.kmeans.py 5 times and report the mean and standard deviation of accuacy in Table 1
3. Implement init\_cluster\_advanced function in in src/kmeans.py. This function should initialize cluster centroids based on kmeans++ algorithms
4. Replace init\_clusters function in kmeans function with init\_cluster\_advanced function. Then, run again 01.kmeans.py 5 times and report mean / std statistics in Table 1
5. Table 1 compares kmeans clustering results depending on initialization algorithm. Which one gives more stable results? What's the reason?
6. clustering\_accuracy in src/metrics.py is slightly different from classification accuracy. Explain how they are different and why they should be different

**2. Autoencoder (2pt)**

1. You should implement 02\_autoencoder\_train.py to train a denosing autoencoder in src/autoencoders.py. The detailed procedure is available in 02\_autoencoder\_train.py. (This starts from empty code. If you ignore your team project, this may be the longest code that you should implement in this class. But you have already seen most building blocks to write this code in previous labs !)
2. After the pretraining stage, we want to finetune the baseline model for test tasks. Run "transfer\_finetune.py". To perform fine-tuning, you should implement the code of loading pretrained weights. Then report the accuracy of baseline (finetune) model on Table 1
3. Compare two baseline models. Which one gives better performance on test tasks? What do you think is driving the performance improvement?

**3. t-SNE** **(1pt)**

1. Visualization of image space and latent space of FashionMNIST using sklearn.manifold.TSNE

(https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html)

1. Compare image space and latent space of FashionMNIST. Which space represent inputs' structure better in tSNE visualization?