MLSP HW1

Non-negative Matrix Factorization & Auto Encoder

Due date: 2024/03/26 23:55

Requirements:

- The code should be written in Python.
- Zip code (.py/.ipynb) and reports (PDF) and name the zipped file as "HW1_YOURSTUDENTID.zip" and submit it to the E3 system.
- We will deduct a late penalty of 20% per additional late day.
- Note that you cannot use any well-developed library. e.g., `sklearn.decomposition.NMF` in Python.

In this assignment, we will investigate:

1. The concept of Non-negative Matrix Factorization (NMF).
2. The relation between NMF and Auto Encoder (AE).

The following 2 problems should be answered with codes (2 separate code files) and explanation (1 unified PDF file):

1. By using the LFW dataset in `sklearn.dataset`, implement NMF on your own (calling NMF packages is not allowed). Plot the matrix and describe the concept of NMF.

2. AE is a dimension reduction method in deep learning that can be directly related to NMF. Mathematically, minimizing the reconstruction error in a constraint AE is the same as conducting NMF. If we use AE to approximate NMF, it needs to satisfy constraints. The latent outputs by the encoder and the weights of the decoder should be non-negative. Implement AE architecture (Several linear layers for encoder and decoder with activate function, e.g., Relu) following the constraints to design the loss function, and train the model with data same as problem 1. Show the reconstruction results and compare them with NMF results. Describe the concept of AE.

The report should contain two parts:

1. Explain each step of your implementation in detail.
2. Result of problems 1 and 2. (contain original input and reconstruction result)

References:

- Lee, Daniel, and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." Advances in neural information processing systems 13 (2000). [Paper Link]
- Squires, Steven, Adam Prügel Bennett, and Mahesan Niranjan. "A variational autoencoder for probabilistic non-negative matrix factorization." arXiv preprint arXiv:1906.05912 (2019). [Paper Link]