AMCS 215
Mathematical Foundations of Machine Learning
Assignment 3, Due September 25, 2023

Notes:

- This assignment is due on Monday Sep 25 since Sunday Sep 24 is National Day.
- Reminder of the weekly recitation session on Tuesdays at 5pm, where we go over solutions of previous homework and give further explanations, hints, and tips on the current one.
- Submit to Blackboard a single pdf file showing your work. The file my contain scanned versions of any work you and explanations your wrote on paper.

1. Eigenfaces with truncated SVD.

The truncated singular value decomposition is a very effective matrix decomposition used to generate low dimensional representations of large scale data sets. In this exercise, we consider its use with a data set of facial images, where we seek to find an orthonormal basis for a subspace containing all the facial images in a data set. These are known as eigenfaces.

- your will find on Blackboard a starter code for loading a set of 400 facial grayscale images of size 64×64 each. The data matrix is of size 4096×400, with each column corresponding to a facial image. Compute the mean of every row (to obtain a vector a) and subtract it from the data matrix.
- compute the SVD of the de-meaned data matrix and plot the resulting singular values on a semilogy (log on the y-axis) plot.
- display the first few (say, 6) singular vectors. These are the first elements of the basis (eigenfaces). Comment on their significance.
- the singular values decay rapidly in your plot above. It is therefore reasonable to *truncate* the SVD to only use the first k singular values and their corresponding singular vectors (say, $k = 50$ or so in this exercise) to approximate the data as $A \approx U_{:k}\Sigma_{:k}V_{:k}^T$. Write an expression for the root mean square error introduced by this approximation.
- in this k-dimensional space, a face f (vector of 4096 entries) can be expressed in terms of a vector y (vector of only k entries) by projection: $U_{:k}y = (f - a)$, i.e., $y = U_{:k}^T(f - a)$. Choose a few faces in the data set, project them on this basis, and then reconstruct them from the computed coordinates. Comment on the quality of the reconstructions. A good reconstruction implies that the *effective dimension* of the data containing the facial images is k—much smaller than its nominal 4096 dimension.

2. Embedding via SVD.

In this exercise, we would like to use the SVD to build a 50-dimensional latent space representation of MNIST data set, i.e. reduce its dimensionality from 784 (28×28) to 50.

- generate the truncated SVD of the (transposed) data matrix
- test the quality of the representation by reconstructing a few randomly chosen examples
  - you may do this qualitatively first by taking a few images, projecting them in the latent space, and then reconstructing the image from this latent space representation, and assessing the visual quality of the reconstruction
  - what quantitative metric could you use to assess the quality of the latent representation? what does the SVD guarantee about the sum of all these errors for the whole dataset?
- suppose we wanted to build a classifier for the MNIST data. Based on your observations above, is it possible to use the latent space representation as input to the classifier? (i.e., would you expect the confusion matrix to be similar to the one you generated in the last assignment?). Answer briefly.

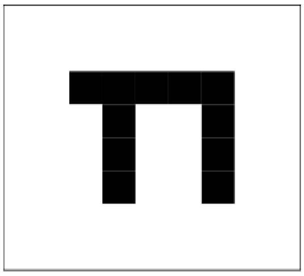
3. Spectral clustering.
Consider the following data set of N samples that we wish to cluster into two $K = 2$ clusters. The "correct" clustering will presumably identify the two spiral patterns of the data set. A starter file that generates the data is posted on Blackboard.

- Perform the clustering with the K-means algorithm you developed in Assignment 1. Does the algorithm correctly cluster the spirals? Explain briefly its results. (why does it or does it not work?)
- In spectral clustering we consider a graph whose edges connect every point to all other points, with positive edge weights inversely correlated with distance, e.g., $W_{ij} = \exp(\frac{-1}{2\sigma^2}\|x_i - x_j\|^2)$ where σ is a tunable scaling parameter. W is essentially a similarity matrix whose entries represent the pairwise closeness/affinity of the data points. (but diagonal entries of W are zero.)

  We build the matrix $D - W$, where D is a diagonal matrix whose entries are the row sums of W, and normalize it as $L = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2}$. The K eigenvectors corresponding to the K smallest eigenvalues of L are then used to generate a clustering.

  Your task is to work out the details of this procedure in Python to perform spectral clustering on the data above. (Use `np.linalg.eig` to compute the eigenvalues and eigenvectors.)
- Does the spectral clustering identify the two spirals? Explain briefly its results. (why does it or does it not work?)

4. 2D convolutions.
Consider the image below, which we represent as an 8×9 matrix of grayscale values. A starter file withe the data is posted on Blackboard.



For each of the convolution kernels below, compute $Y = X * B$. You may use the `scipy.signal.convolve2d` routine.

$$B = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}, \quad B = \begin{bmatrix} -1.0 & 1.0 \end{bmatrix}, \quad B = \begin{bmatrix} -1.0 \\ 1.0 \end{bmatrix}, \quad B = \begin{bmatrix} -1.0 & 2.0 & -1.0 \\ -1.0 & 2.0 & -1.0 \end{bmatrix}, \quad B = \begin{bmatrix} -1.0 & -1.0 \\ 2.0 & 2.0 \\ -1.0 & -1.0 \end{bmatrix}$$

- where do you observe large values in the resulting images Y, i.e, what features in the image X is each of the kernels capturing? Describe, briefly, in English.