CSCE 636: Deep Learning (Spring 2024)
Assignment #4

Due 11:59PM on 04/19/2024

---

1. You need to submit (1) a report in PDF and (2) your code files, both to Canvas.
2. Your PDF report should include (1) answers to the non-programming part, and (2) results and analysis of the programming part. For the programming part, your PDF report should at least include the results you obtained, for example the accuracy, training curves, parameters, etc. You should also analyze your results as needed.
3. Please name your PDF report "HW#_FirstName_LastName.pdf". Please put all code files into a compressed file named "HW#_FirstName_LastName.zip". Please submit two files (.pdf and .zip) to Canvas (i.e., do not include the PDF file into the ZIP file).
4. Only write your code between the following lines. Do not modify other parts.

   ### YOUR CODE HERE

   ### END YOUR CODE
5. All students are highly encouraged to typeset their reports using Word or LaTeX. In case you decide to hand-write, please make sure your answers are clearly readable in scanned PDF.
6. Unlimited number of submissions are allowed and the latest one will be timed and graded.
7. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
8. Please start your submission to Canvas at least 15-30 minutes before the deadline, as there might be latency. We do NOT accept E-mail submissions.

---

1. (15 points) This question refers to the textbook "Deep Learning : Foundations and Concepts". Show that a graph attention network in which the graph is fully connected, so that there is an edge between every pair of nodes, is equivalent to a standard transformer architecture.
2. (20 points) This question refers to the textbook "Deep Learning : Foundations and Concepts". In this exercise we write the update equations (13.16) as graph-level equations using matrices. To keep the notation uncluttered, we omit the layer index l. First, gather the node-embedding vectors $\{\mathbf{h}_n\}$ into an $N \times D$ matrix $\mathbf{H}$ in which row n is given by $\mathbf{h}_n^{\mathrm{T}}$. Then show that the neighbourhood-aggregated vectors $\mathbf{z}_n$ given by

$$\mathbf{z}_n = \sum_{m \in \mathcal{N}(n)} \mathbf{h}_m$$

   can be written in matrix form as $\mathbf{Z} = \mathbf{AH}$ where $\mathbf{Z}$ is the $N \times D$ matrix in which row n is given by $\mathbf{z}_n^{\mathrm{T}}$, and $\mathbf{A}$ is the adjacency matrix. Finally, show that the argument to the nonlinear activation function in (13.16) can be written in matrix form as

$$\mathbf{AHW}_{\text{neigh}} + \mathbf{HW}_{\text{self}} + \mathbf{1}_D \mathbf{b}^{\mathrm{T}}$$

   where 1_D is the D-dimensional column vector in which all elements are 1 .

3. (15 points) This question refers to the textbook "Deep Learning : Foundations and Concepts". By making use of the equivariance property (13.19) for layer l of a deep graph convolutional network along with the permutation property (13.4) for the node variables, show that a complete deep graph convolutional network defined by (13.18) is also equivariant.

4. (10 points) Given a symmetric matrix $A \in \mathbb{R}^{3\times 3}$, suppose its eigen-decomposition can be written as

$$A = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} u_{11} & u_{21} & u_{31} \\ u_{12} & u_{22} & u_{32} \\ u_{13} & u_{23} & u_{33} \end{pmatrix}. \tag{1}$$

What is the singular value decomposition of this matrix?

5. (20 points) Provide a complete proof of the Ky Fan Theorem given on page 4 of the notes "Principal Component Analysis and Autoencoders". The Theorem is also given below:

Theorem. (*Ky Fan*) Let $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and the corresponding eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$. Then

$$\lambda_1 + \cdots \lambda_k = \max_{\boldsymbol{A}\in\mathbb{R}^{n\times k}:\boldsymbol{A}^T\boldsymbol{A}=\boldsymbol{I}_k} \text{trace}\left(\boldsymbol{A}^T\boldsymbol{H}\boldsymbol{A}\right).$$

And the optimal $\boldsymbol{A}^*$ is given by $\boldsymbol{A}^* = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]\boldsymbol{Q}$ with $\boldsymbol{Q}$ an arbitrary orthogonal matrix.■

6. (70 points) (Coding Task) **PCA vs Autoencoder:** In this assignment, you will apply the PCA and the autoencoder (AE) to a collection of handwritten digit images from the USPS dataset. The data file is stored in the "PCA/data" folder as "USPS.mat". Please check the starting code in folder "PCA/code" and follow the instructions. The whole dataset is already loaded and stored in the matrix A with shape 3000×256. Each row of matrix A represents a 16×16 handwritten digit image (between 0 and 9), which is flattened to a 256-dimensional vector. Note that you will need to use PyTorch in this assignment. **Please read the "Readme" file carefully before getting started.** You are expected to implement the solutions based on the starting code. The files you need to modify are "solution.py" and "main.py". You will test your solution by modifying and running the "main.py" file.

   (a) (10 points) In the ***class PCA()***, complete the ***_do_pca()*** function.

   (b) (5 points) In the ***class PCA()***, complete the ***reconstruction()*** function to perform data reconstruction. Please evaluate your code by testing different numbers of the principal component that $p = 32$, 64, 128.

   (c) (10 points) In the ***class AE()***, complete the ***_network()*** and ***_forward()*** function. Please follow the note (`http://people.tamu.edu/~sji/classes/PCA.pdf`) to implement your network. Note that for problems (c), (e), and (f), the weights need to be shared between the encoder and the decoder with weight matrices transposed to each other.

   (d) (5 points) In the ***class AE()***, complete the ***reconstruction()*** function to perform data reconstruction. Please test your function using three different dimensions for the hidden representation d that $d = 32$, 64, 128.

(e) (10 points) Compare the reconstruction errors from PCA and AE. Note that you need to set $p = d$ for comparisons. Please evaluate the errors using $p = d = 32, 64, 128$. Report the reconstruction errors and provide a brief analysis.

(f) (10 points) Experimentally justify the relations between the projection matrix $\boldsymbol{G}$ in PCA and the optimized weight matrix $\boldsymbol{W}$ in AE. Note that you need to set $p = d$ for valid comparisons. Please explore three different cases that $p = d = 32, 64, 128$. We recommend to first use ***frobeniu_norm_error()*** to verify if $\boldsymbol{W}$ and $\boldsymbol{G}$ are the same. If not, please follow the note (`http://people.tamu.edu/~sji/classes/PCA.pdf`) to implement necessary transformations for two matrices $\boldsymbol{G}$ and $\boldsymbol{W}$ and explore the relations. You need to modify the code in "main.py".

(g) (10 points) Please modify the ***_network()*** and ***_forward()*** function so that the weights are **not** shared between the encoder and the decoder. Report the reconstructions errors for $d = 32, 64, 128$. Please compare with the sharing weights case and briefly analyze you results.

(h) (10 points) Please modify the ***_network()*** and ***_forward()*** function to include more network layers and nonlinear functions. Please set $d = 64$ and explore different hyperparameters. Report the hyperparameters of the best model and its reconstruction error. Please analyze and report your conclusions.