Instructions and Policy: Each student should write up their own solutions independently, no copying of any form is allowed. You MUST to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.
YOU MUST INCLUDE YOUR NAME IN THE HOMEWORK
You need to submit your answer in PDF. LaTeX is typesetting is encouraged but not required. Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.

The code of Homework Y, question QX, item i should be named HWY_QX_i.py. Your code is REQUIRED to run on Python 3 at scholar.rcac.purdue.edu. The TAs will help you with the use of the scholar cluster. If the name of the executable is incorrect, it wont be graded. Please make sure you didnt use any library/source explicitly forbidden to use. If such library/source code is used, you will get 0 pt for the coding part of the assignment. If your code doesnt run on scholar.rcac.purdue.edu, then even if it compiles in another computer, your code will still be considered not-running and the respective part of the assignment will receive 0 pt.

Q0 (0pts correct answer, -1,000pts incorrect answer: (0,-1,000) pts): A correct answer to the following questions is worth 0pts. An incorrect answer is worth -1,000pts, which carries over to other homeworks and exams, and can result in an F grade in the course.

(1) Student interaction with other students / individuals:
  (a) I have copied part of my homework from another student or another person (plagiarism).
  (b) Yes, I discussed the homework with another person but came up with my own answers. Their name(s) is (are) ____
  (c) No, I did not discuss the homework with anyone

(2) On using online resources:
  (a) I have copied one of my answers directly from a website (plagiarism).
  (b) I have used online resources to help me answer this question, but I came up with my own answers (you are allowed to use online resources as long as the answer is your own). Here is a list of the websites I have used in this homework:
  ____
  (c) I have not used any online resources except the ones provided in the course website.

Q1 (5 pts): PCA Implementation

In lecture 26, we have learned that PCA can be viewed as projecting the data onto a lower-dimensional subspace. Let's denote $\mathbf{X}$ as the original $N \times D$ data and $\mathbf{Z}$ as the transformed $N \times M$ data, then we have the following properties for $\mathbf{Z}$:

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{I},$$

if we have infinite number of data points.

One way to get the transformed data $\mathbf{Z}$ to perform SVD of the covariance matrix $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$$

We further divide the $\mathbf{\Sigma}$ into two parts $\mathbf{\Sigma} = \mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}$.

$$\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\mathbf{U}^T$$

Then,

$$\mathbf{\Sigma}^{-1/2}\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U}\mathbf{\Sigma}^{-1/2} = \mathbf{I}.$$

Hence, we can obtain $\mathbf{Z}$ as

$$\mathbf{Z} = \mathbf{X}\mathbf{U}\mathbf{\Sigma}^{-1/2}.$$

Recall in lectures 22 and 24, we have shown how to perform SVD with a neural network. We can use the neural network model here to perform SVD of the covariance matrix $\mathbf{X}^T\mathbf{X}$ and obtain the transformed data $\mathbf{Z}$. We can write our objective function as:

$$J = ||\mathbf{X}^T\mathbf{X} - \mathbf{Y}||^2,$$

Where $\mathbf{Y}$ is the output of the neural network.

Note that in the example shown in class, we were interested in finding $\mathbf{U}$, $\mathbf{V}^T$ and $\mathbf{\Sigma}$ such that $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. However, in our case, with the covariance matrix $\mathbf{X}^T\mathbf{X}$, we will have $\mathbf{U}$, $\mathbf{U}^T$ and $\mathbf{\Sigma}$. Hence, you will need to copy the weight matrix $\mathbf{U}$ to replace $\mathbf{V}$.

In this part of the homework, using PyTorch libraries, you need to implement the neural network described above that performs PCA to the input data. You can download the data from `https://www.cs.purdue.edu/homes/ribeirob/courses/Spring2018/data/PCAdata.txt` . The data is organized as a matrix: each line represents a row and columns are separated by a comma.

(1) (1pt) (PDF) Draw the neural network using the similar drawing symbols used in the lectures. Describe the input and output. You need to clarify the number of hidden layers, the dimension of the hidden layers, input layer and output layer, and the loss function.

(2) (1pt) (PDF) Show the equations of the forward pass of your neural network.

(3) (2pts) (CODE) Let $M = 10$, $N = 1000$ and $D = 100$. Implement the neural network described above using PyTorch and train your neural network with the data provided. Upload your python code with turnin.

(4) (1pt) (PDF) Now, extract the transformed data from the learned neural network, call it $\mathbf{Z}_{nn}$. Next, perform PCA using sklearn package with the data, call the transformed data as $\mathbf{Z}_{sk}$. Report the Frobenius Norm of $(\mathbf{Z}_{nn} - \mathbf{Z}_{sk})$.

Q2 (5 pts): Multi-armed Bandits

A retailer wants to sell a video game by randomly contacting people from a contact list. Assume that only individuals under 18 years old will buy the game at 1% conversion rate (that is, 1% of the under-18 contacted, will buy the game). The revenue per game sold is \$10. The company can advertise to a single random individual at a cost of \$0.01, in two different states (say, a geo-located Facebook ad). In Case 1, the ads are bought in Indiana and California. In Case 2, the ads are bought in Indiana and Illinois. Consider the following population statistics given by the U.S. Census Bureau:

	State	Adult pop.	Under-18 pop.
Case 1:	Indiana	4,875,504	1,608,298
	District of Columbia	500,908	100,815

	State	Adult pop.	Under-18 pop.
Case 2:	Indiana	4,875,504	1,608,298
	Illinois	9,701,453	3,129,179

Using the above data, simulate UCB1 applied to the action of deciding in which state we should run each advertisement. Assume that as soon as the ad runs, the customer decides to buy or not buy the game and we get immediate feedback. In Case 1, the bandit arms are "running the ad in Indiana" and "running the ad in Illinois". In Case 2, the bandit arms are "running the ad in Indiana" and "running the ad in the District of Columbia".

The cumulative reward is the sum of the revenue minus the sum of the cost. It costs \$0.01 to show the ad to a particular person in a state. Each copy of game generates revenue of \$30. Assume that a person below 18 years of age buys the game with probability 0.3 independent of each other, while no one above 18 years is interested in buying the game. Due to time constraints we can display a maximum of 10,000 ads.

(a) (2pts) (PDF) Consider dividing the first 1,000 ads into two parts: 500 ads go to one state, 500 ads go to another state. Choose the state with the largest cumulative reward to run the remaining 9,000 ads. What is the total cumulative reward you were able to obtain with the 10,000 ads.

(b) (2pts) (PDF) Find the cumulative reward that the company will collect by using the UCB1 algorithm both in Case 1 and Case2. Describe in details how you found your results (with the UCB1 equations). Is the result expected? Why?

(c) (1pt) (PDF) For Case 1, give a plot showing the cumulative reward of each arm over time (both arms should be on the same plot): i.e., the vertical axis shows the cumulative reward and the horizontal axis shows the current number of purchased ads. Do the same for Case 2 and explain why you see a difference in behavior.