CSCI3230 / ESTR3108: Fundamentals of Artificial Intelligence 2021-2022 Term 1
The Chinese University of Hong Kong

Assignment 3

Due date: 17 November 2021 (Wed) 23:59 Full mark: 100
Expected time spent: 4-6 hours

Aims: 1. Understand the knowledge about PCA and have hands-on practice about optimizations.
2. Understand basic knowledge about neural networks and hands-on programming of simple neural networks, including training and testing.

Description:

In Assignment 3, you will conduct some calculation and proofs related to PCA using elementary linear algebra and calculus knowledge. Furthermore, you will practice on essentials of neural network, including neural network construction, backpropagation, model training and evaluation, PyTorch programming, etc.

Questions:

1. Recall that the geometric formulation of PCA for dimensionality reduction is the optimization problem as shown below:

$$U^* = \arg\min_{U \in \mathbb{R}^{D \times d}, U^T U = I} \| \mathbf{X} - UU^T \mathbf{X} \|_F$$

where $\mathbf{X} \in \mathbb{R}^{D \times m}$ is a zero-mean data matrix with D-dimensional features and m samples, and $d \ll D$ is the target lower dimension. When $d = 1$, we say U^* is the first principal axis and $U^{*T}\mathbf{X}$ is the first principal component.

Suppose we have data matrix: $\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}$, which contains 4 samples with 3-dimensional features. We can observe there are some redundant information in the data matrix. In this case, we would like to perform PCA on the data matrix to reduce dimensionality of the original data.

(a) Find all eigenvalues of $\mathbf{X}\mathbf{X}^T$. Show all of your calculation steps. (9%)

(b) Find the first principal axis and the first principal component of $\mathbf{X}$. (8%)

(c) If we have another data matrix $\mathbf{X}'$ as:

$$\mathbf{X}' = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

which is obtained by exchanging the last two columns of $\mathbf{X}$. Do you think the first principal axis of $\mathbf{X}'$ will differ from the first principal axis of $\mathbf{X}$? If you think they are different, find the first principal axis of $\mathbf{X}'$. Otherwise, prove they are identical, and state a benefit of this property. (8%)

2. The PCA problem for non-zero-mean data matrix $\mathbf{X} = \left[X^{(1)}, \ldots, X^{(m)}\right]$ can be formulated as:

$$\min_{U \in \mathbb{R}^{D \times d}, U^T U = I_d, \mu \in \mathbb{R}^D} \sum_{i=1}^{m} \| \left(X^{(i)} - \mu\right) - UU^T\left(X^{(i)} - \mu\right) \|_2^2$$

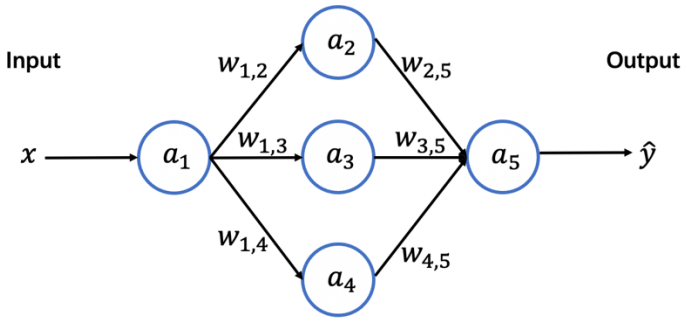
where D is the original dimension of feature space, and $d \ll D$ is the targeting lower dimension. Fixing U as any $D \times d$ matrix such that $U^T U = I_d$ (we denote I_d as a $d \times d$ identity matrix), we denote $f(\mu) = \sum_{i=1}^{m} \| \left(X^{(i)} - \mu\right) - UU^T\left(X^{(i)} - \mu\right) \|_2^2$.

(a) Find the closed-form expression of $\frac{\partial f(\mu)}{\partial \mu}$ by chain rule. (8%)

(b) Find $\frac{\partial^2 f(\mu)}{\partial \mu^2}$ and prove it is semi-positive definite. (12%)

(c) Using the results of (a) and (b), further show that $\mu^* = \frac{1}{m}\sum_{i=1}^{m} X^{(i)}$ yields the global minimal value of $f(\mu)$. (5%)

3. Consider a simple neural network for binary classification as the following:



In this neural network, the input is x and the output is $\hat{y}$. There is 1 input neuron (a_1), 3 hidden neurons (a_2, a_3, a_4) and 1 output neuron (a_5) in this network. The weights of connections between a_i and a_j is defined as $w_{i,j}$. For the input neuron, no activation function is used. For every hidden neuron, we use *ReLU* activation function. For the output neuron, we use *Sigmoid* activation function. To simplify the problem, we can set $W_1 = \left[w_{1,2}, w_{1,3}, w_{1,4}\right], W_2 = \left[w_{2,5}, w_{3,5}, w_{4,5}\right]^T$.

(a) Write the specific formula of $\hat{y}$, which is composed of $x, w_{1,2}, w_{1,3}, w_{1,4}, w_{2,5}, w_{3,5}, w_{4,5}$ and *ReLU*. (Hint: a_2 can be represented by $a_2 = ReLU(w_{1,2}a_1)$) (5%)

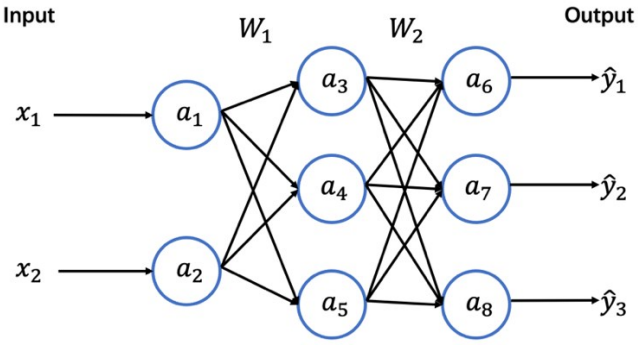
(b) Suppose we use the loss function of Binary Cross Entropy (BCE) to optimize this network, with input x, its corresponding ground-truth label y, and its corresponding network output $\hat{y}$, please write down the specific loss function based on the formula in (a), and calculate the gradient for $w_{1,2}$ *and* $w_{2,5}$. (10%)

(c) Given a training set and test set listed in *train_q3.csv* and *test_q3.csv*, write PyTorch code to learn the above network, and present the followings: 1) plot the curve of training loss, 2) give the final optimized weights for (W_1, W_2), 3) test the network on the given test set and report the testing accuracy.

Please set the initial weights as $W_1 = [0.12,\ 0.26, -0.15], W_2 = [0.11, 0.13, 0.07]^T$, the initial learning rate as 0.5, training iteration as 50. The learning rate is decayed with a factor of 0.3 every 20 iterations. For binary classification, we typically set the threshold as 0.5, i.e., if the output probability is larger than 0.5, then the output class is 1. Otherwise, the prediction is 0. (10%)

4. In this question, we consider a multi-class classification problem, i.e., to discriminate C $(C \geq 3)$ classes. We use softmax function in the output layer to produce probability predictions for each class. In specific, consider the following neural network with 2 input neurons, 3 neurons in hidden layer and 3 output neurons. For each hidden neuron, we use *ReLU* activation function. The input and output of this neural network is represented by $\boldsymbol{x} \in R^2$ and $\hat{\boldsymbol{y}} \in R^3$. The weight matrix from the input layer to hidden layer is denoted by $W_1 \in R^{2\times3}$ and the weight matrix from the hidden layer to output layer is denoted by $W_2 \in R^{3\times3}$.

Here, we use one-hot encoding to convert ground truth labels into "one-hot" vectors, where only an entry is 1 and other entries are 0. For example, suppose there are C classes in total, the one-hot encoding of a label c (an integer) is a n-dimensional vector where only the c-th entry is equal to 1 while other entries are 0. We use loss function of Negative Log Likelihood (NLL) to optimize the network. Given the ground truth label in the one-hot encoding format $y = [y_1, y_2, y_3]$ and the probability prediction of the network $p = softmax([\hat{y}_1, \hat{y}_2, \hat{y}_3]) = [p_1, p_2, p_3]$, the NLL loss between the ground truth label and probability prediction is defined as $\ell = -\sum_{i=1}^{3} y_i log p_i$.



(a) Given a training set and test set listed in *train_q4.csv* and *test_q4.csv*, write PyTorch code to learn the above network, and present the followings: 1) plot the curve of training loss, 2) give the final optimized weights for (W_1, W_2), 3) test the network on the given test set and report the testing accuracy.

Please set the initial weights as following:$W_1 = [[0.74, 0.1, 0.98],\ [-2.04, -1.4, -0.31]]$, $W_2 = \left[[1.37, -0.9, -0.8], [-0.08, 0.94, 0.47], [-0.3, 0.57, 0.93]\right]^T$, the initial learning rate as 0.1, training iterations as 50. The learning rate is decayed with a factor of 0.3 every 20 iterations. (10%)

(b) The learning rate is an important hyperparameter when training a neural network. Let's study this hyperparameter here. Please re-train the network with learning rates in the set {1.0, 0.1, 0.01}, respectively. Meanwhile, observe the training loss curve and testing accuracy for each learning rate.

Questions: 1) plot the training loss curve and report the testing accuracy for each learning rate; 2) what is your finding from these training loss curves and testing accuracy? (10%)

(c) Let's further study the impact of weight initialization. If we randomly initialize the weights $W_1 \in R^{2\times3}$ and $W_2 \in R^{3\times3}$ using torch.rand, please try your best to tune the hyperparameter learning rate to obtain the highest possible testing accuracy. After that, please present the followings: 1) plot the curve of training loss, 2) test the network on the given test set and report the testing accuracy. (5%)

Submission:

Submit a single file named <ID>_asmt3.pdf, where <ID> is your student ID.
Your file should contain the following header. Contact Professor Dou before submitting the assignment if you have anything unclear about the guidelines on academic honesty.

```
CSCI3230 / ESTR3108 2021-22 First Term Assignment 3

I declare that the assignment here submitted is original except for source
material explicitly acknowledged, and that the same or closely related material
has not been previously submitted for another course. I also acknowledge that I
am aware of University policy and regulations on honesty in academic work, and
of the disciplinary guidelines and procedures applicable to breaches of such
policy and regulations, as contained in the following websites.

University Guideline on Academic Honesty:
http://www.cuhk.edu.hk/policy/academichonesty/
Faculty of Engineering Guidelines to Academic Honesty:
http://www.erg.cuhk.edu.hk/erg-intra/upload/documents/ENGG_Discipline.pdf

Student Name: <fill in your name>
Student ID  : <fill in your ID>
```

Submit your files using the Blackboard online system.

Notes:

1. Remember to submit your assignment by 23:59pm of the due date. We may not accept late submissions.
2. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be considered.

University Guideline for Plagiarism

Please pay attention to the university policy and regulations on honesty in academic work, and the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details can be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students will be required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.