LUMS
A Not-for-Profit University

Department of Electrical Engineering
School of Science and Engineering

EE514/CS535 Machine Learning

ASSIGNMENT 3

Due Date: 5:00 pm, Thursday, May 03, 2023.
Format: 7 problems, for a total of 100 marks
Instructions:

- You are allowed to collaborate with your peers but copying your colleague's solution is strictly prohibited. This is not a group assignment. Each student must submit his/her own assignment.
- Solve the assignment on blank A4 sheets and staple them before submitting.
- Submit in the dropbox labeled EE-514 outside the instructor's office.
- Write your name and roll no. on the first page.
- Feel free to contact the instructor or the teaching assistants if you have any concerns.

- You represent the most competent individuals in the country, do not let plagiarism come in between your learning. In case any instance of plagiarism is detected, the disciplinary case will be dealt with according to the university's rules and regulations.

Problem 1 (10 marks)

Consider a 3-layer neural network with linear activation functions. The first layer has two input nodes, the second layer has three hidden nodes, and the third layer has one output node.

Let the activation functions in the hidden layer and the output layer be linear i.e., $f(x) = x$.

(a) (i) Write down the mathematical expression for the output of the network given input vector $\boldsymbol{x} = [x_1, x_2]$. Represent the weights and biases of the network using matrices $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and vectors $\boldsymbol{b}_1$, $\boldsymbol{b}_2$.

(ii) Calculate the output of the network for a given input vector $\boldsymbol{x} = [2, 3]$, assuming the following weights and biases:

$$\boldsymbol{W}_1 = \begin{bmatrix} 1 & 0 \\ 0 & -4 \\ 5 & 1 \end{bmatrix}, \boldsymbol{b}_1 = \begin{bmatrix} 3 \\ -5 \\ 10 \end{bmatrix}, \boldsymbol{W}_2 = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix}, \boldsymbol{b}_2 = \begin{bmatrix} -1 \end{bmatrix}.$$

(b) (i) Show that the 3-layer neural network can be reduced to a single-layer linear network by combining the weight matrices and bias vectors. Derive the resulting weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$ for the single-layer linear network.

(ii) Calculate the output of the reduced single-layer linear network for the same input vector $\boldsymbol{x} = [2, 3]$, using the derived weight matrix $\boldsymbol{W}$ and bias vector $\boldsymbol{b}$. Verify that the output is the same as the output obtained in Part a (ii). Briefly explain why this is so.

Problem 2 (10 marks)

(a) If you apply a filter of size $k \times k$ and stride s to an input of size $n \times n$ with padding p, what will be the dimensions of the resulting feature map?

(b) Consider a 3×3 matrix representing a patch of an image I, where each entry corresponds to the gray scale color of a pixel. Additionally, you are given a 2×2 convolutional kernel K as follows:

$$I = \begin{bmatrix} 3 & 1 & 1 \\ 3 & 0 & 2 \\ 4 & 4 & 0 \end{bmatrix}$$

and

$$K = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Assuming a stride of 1 and no padding, what is the output of applying the filter to the input?

Problem 3 (15 marks)

Suppose you have a simple neural network with one input layer, one hidden layer, and one output layer. The input layer has 2 neurons, the hidden layer has 3 neurons, and the output layer has 1 neuron. The activation function for all neurons is the sigmoid function. The network has already been initialized with the following weights and biases:

$$W_{\text{hidden}} \begin{bmatrix} 0.3 & 0.5 & 0.9 \\ 0.8 & 0.1 & 0.7 \end{bmatrix}, \quad b_{\text{hidden}} = \begin{bmatrix} 0.2 \\ 0.4 \\ 0.5 \end{bmatrix}, \quad W_{\text{output}} = \begin{bmatrix} 0.3 & 0.5 & 0.9 \end{bmatrix}, \quad b_{\text{output}} = \begin{bmatrix} 0.2 \end{bmatrix}$$

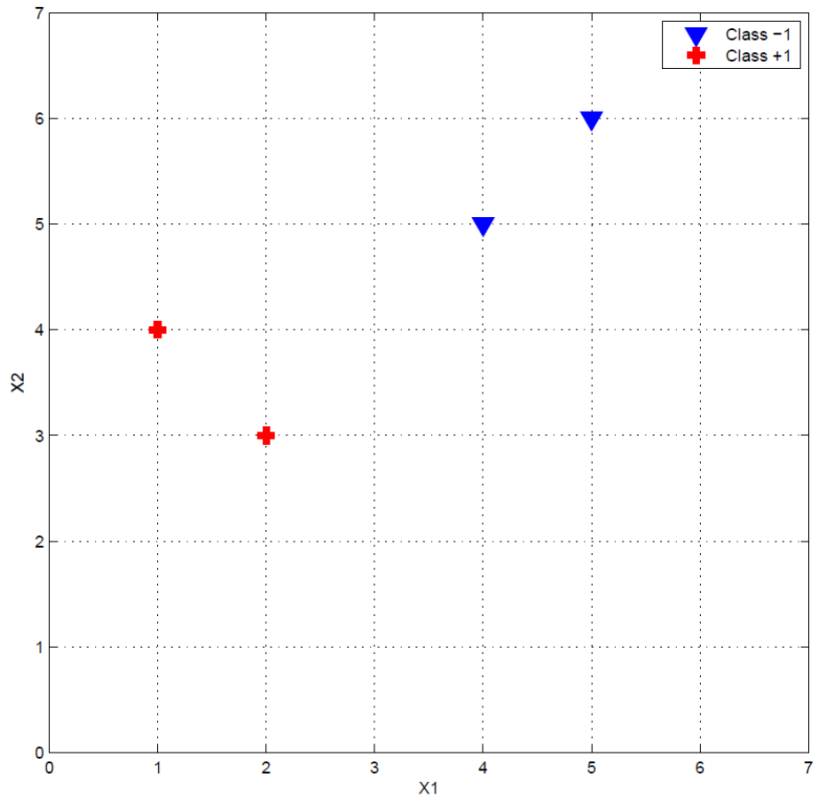
(a) Given the input vector (0.5, 0.8), perform a forward pass through the network to compute the output.

(b) Suppose the true output for the given input is 0.6. Compute the error between the true output and the computed output.

(c) Perform a backward pass through the network using backpropagation to update the weights and biases. Use a learning rate of 0.1, to determine the updated weights and biases after one iteration of backpropagation.

Problem 4 (15 marks)

For the training data plotted below, find the weight vector and bias for the decision boundary $\mathbf{w}^T\mathbf{x} - \theta = 0$ maximizing the classification margin. Also, indicate the support vectors and compute the classification margin.



Problem 5 (15 marks)

Given a dataset,

#	Data Point
x_1	$(2, 2)$
x_2	$(2, 3)$
x_3	$(3, 2)$
x_4	$(8, 7)$
x_5	$(7, 8)$
x_6	$(5, 8)$
x_7	$(4, 7)$
x_8	$(5, 3)$
x_9	$(11, 2)$
x_{10}	$(11, 3)$
x_{11}	$(10, 3)$

we wish to partition this dataset into 3 clusters using the K-Means algorithm.

(a) Show that the K-Means algorithm always converges to a local minimum in a finite number of steps. (Hint: Use the fact that the algorithm decreases the objective function in each iteration.)

(b) Even though it does converge every time, it is highly sensitive to the initialization of centroids. Poor initialization can result in poor clustering.
To overcome this we decide to use K-means++. K-Means++ is a variant of the K-Means algorithm that uses an improved initialization scheme. The algorithm first selects one centroid uniformly at random from the data points and then selects subsequent centroids from the remaining data points with probability proportional to the square of their distance from the nearest already chosen centroid.
Run the K-Means++ Algorithm on this dataset until convergence.

(c) Although K-Means++ requires more computations than K-Means for initialization, it often results in quicker convergence to a better clustering solution. Why is this the case?

Problem 6 (15 marks)

Consider a binary classification problem with two inputs and the following labeled data-set for training.

Label y	Data Point $(x^{(1)}, x^{(2)})$
1	$(-3, -3)$
1	$(-3, 3)$
1	$(3, 3)$
-1	$(2, 2)$
-1	$(2, -2)$
-1	$(-2, 2)$

(a) Plot the points on a 2D plane. Can we use hard SVM for this problem? Provide a brief justification to support your answer.

(b) Since the data is not linearly separable, we map the 2D feature space to 3D feature space using the mapping function $\phi(x)$ to make it linearly separable. Determine the mapping function that can enable us to use hard SVM in 3D feature space.

(c) We have a linear decision boundary (hard SVM) in 3D space to separate the transformed data in 3D (new feature space). Indicate this boundary as a (non-linear) decision boundary on the plot obtained in part (a).

(d) Instead of mapping the data into 3D space and using hard SVM to learn the decision boundary in 3D, we can use the kernel trick to learn a non-linear boundary you have plotted in part (c) in the original 2D feature space. Formulate a kernel function associated with the mapping function you used in part (b).

Problem 7 (20 marks)

You are given the following dataset of emails with 4 features, that is, the presence of the key word in the email: account, money, links, and password. The output variable is a binary label indicating whether the email is spam (Yes).

account	**money**	**links**	**password**	**spam**
No	No	Yes	No	No
No	No	No	No	No
Yes	Yes	Yes	Yes	No
Yes	No	Yes	Yes	Yes
No	Yes	No	No	Yes
No	No	No	Yes	Yes
No	No	Yes	Yes	Yes
Yes	Yes	Yes	No	Yes

(a) What is the entropy of the target value 'spam' in the data?

(b) Which attribute would the decision tree algorithm that minimizes entropy choose to use for the root of the tree?

(c) Determine the information gain due to the split you chose in the previous question?

(d) Draw the full decision tree that would be learned for this data.

— End of Assignment —