Hong Kong University of Science and Technology
COMP 4211: Machine Learning
Spring 2024

Problem Set
Due: 10 April 2024, Wednesday, 11:59pm

Important Instructions:

- If an answer requires calculation or derivation, you are expected to show the steps as well in addition to the final answer.
- You may use LaTeX (`http://www.latex-project.org/`), Word or other mathematical typesetting software to typeset your answers. If you choose to submit handwritten answers, they must be written clearly or else marks will be deducted.
- Your submission must be done electronically in one PDF file via the Canvas course site.
- Late submission will be accepted but with penalty. The late penalty is deduction of one point (out of a maximum of 100 points) for every hour late after 11:59pm with no more than two days (48 hours). Being late for a fraction of an hour is considered a full hour. For example, two points will be deducted if the submission time is 01:23:34.
- While you may discuss with your classmates on general ideas about solving the problems, your submission should be based on your own independent effort.
- In case you seek help from any person or reference source, you should state it clearly in your submission. Failure to do so is considered plagiarism which will lead to appropriate disciplinary actions. Please refer to the regulations for student conduct and academic integrity on this webpage: `https://registry.hkust.edu.hk/resource-library/academic-standards`.

1. **Linear Regression** (13 points)

   (a) (8 points) Using the method illustrated in slides #6 and #7 of the Linear Regression notes (instead of that in slide #9 using multivariable calculus), derive the least squares estimate in slide #8 for the general case with $d \geq 1$.

   (b) (2 points) Suppose we want to use gradient descent to estimate the solution iteratively instead of the closed-form solution above. Derive the weight update rule for each of the weights.

   (c) (3 points) Prove that minimizing the squared loss with respect to each of the weights is equivalent to maximizing the R^2 score on the training set.

2. **Logistic Regression** (10 points)

Consider a logistic regression model with $K \geq 2$ outputs. To perform multiclass classification, the softmax function is applied and the loss function as defined in slide #17 of the Logistic Regression notes is used for model training. The implicit assumption of this model is that each input belongs to one and only one of K classes corresponding to the K outputs.

Suppose we now use the same model to solve a different problem by defining a different loss function. Each input may belong to any $(0, 1, \ldots, K)$ number of categories (or called labels) corresponding to the K outputs.

(a) (6 points) Give a suitable loss function for the alternative model formulation.

(b) (4 points) If we consider the alternative formulation as a multiclass classification problem in the sense that each input belongs to one and only one of a certain number of disjoint classes, how many classes are there in total? Explain your answer.

3. **Binary Classification** (10 points)

Let P and R denote the precision and recall metrics for binary classification. We define the following score for combining P and R:

$$F_\beta = \frac{(1+\beta^2)\,PR}{\beta^2 P + R},$$

where $\beta > 0$ is a positive real number.

(a) (2 points) Show that F_β is a generalization of the F1 score.

(b) (4 points) Explain the effects of $F_{0.5}$ and F_2 respectively.

(c) (4 points) Explain the effects of F_β as β tends to 0 and $+\infty$ respectively.

4. **Feedforward Neural Networks** (15 points)

We consider the feedforward neural network discussed in class with its weight update rules summarized in slides #16 and #17 of the notes. Suppose all units in the two hidden layers use the hyberbolic tangent function as their activation functions, i.e.,

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

(a) (3 points) Describe one advantage of using the hyperbolic tangent function as activation function as compared to using the sigmoid function.

(b) (3 points) Derive and express the tanh activation function in terms of the sigmoid function σ.

(c) (3 points) Derive and express the first derivative of the tanh activation function in terms of the sigmoid function σ.

(d) (3 points) By referring to the weight update rules, explain why it is undesirable to initialize the weights to large magnitudes.

(e) (3 points) By referring to the weight update rules, explain why it is undesirable to initialize all the weights to zero.

5. **Deep Neural Networks** (11 points)

   (a) (5 points) Why is the rectifier activation function computationally more efficient than the sigmoid activation function?

   (b) (6 points) Although batch normalization may be seen as extending the idea of normalizing the raw input to the upper layers of a network, there exist major differences between batch normalization and ordinary data normalization. Describe two such differences.

6. **Feedforward Neural Networks and Convolutional Neural Networks** (15 points)

(a) (8 points) Consider a feedforward neural network with the following network architecture:

Layer (type)	Output Shape
input_2 (InputLayer)	[(None, 784)]
dense_3 (Dense)	(None, 64)
dense_4 (Dense)	(None, 64)
dense_5 (Dense)	(None, 10)

i. Calculate the number of learnable parameters in each of the four layers.

ii. If L_2 regularization is applied to the model, calculate the number of learnable parameters that are regularized in each of the four layers.

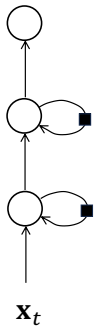
(b) (7 points) Consider a convolutional neural network with the following network architecture where the input tensor has shape $(32, 32, 3)$ and the local receptive fields of the two convolutional and two pooling layers have sizes 3×3 and 2×2, respectively:

Layer (type)	Output Shape
conv2d (Conv2D)	(None, 30, 30, 32)
max_pooling2d (MaxPooling2D)	(None, 15, 15, 32)
conv2d_1 (Conv2D)	(None, 13, 13, 64)
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)
flatten (Flatten)	(None, 2304)
dropout (Dropout)	(None, 2304)
dense (Dense)	(None, 10)

Calculate the number of learnable parameters in each of the seven layers.

7. **Recurrent Neural Networks** (10 points)

Shown below is a recurrent neural network with two hidden layers of recurrent units. Only the hidden layers and output layer consist of processing units. Each of these three processing layers is represented as a circle.



(a) (5 points) Suppose an input sequence $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ formed by vectors of three time steps is presented to the network one step at a time. Draw the unfolded representation of the recurrent neural network showing all three time steps.

(b) (5 points) Mark the longest feedforward path in the unfolded representation in part (a). How many layers of processing units are there along this path?

8. **Principal Component Analysis** (16 points)

We are given a two-dimensional dataset $\mathcal{S}$ which consists of six data points:

$$\mathcal{S} = \left\{\mathbf{x}^{(\ell)}\right\}_{\ell=1}^{6} = \left\{\begin{pmatrix}2\\1\end{pmatrix}, \begin{pmatrix}3\\5\end{pmatrix}, \begin{pmatrix}4\\3\end{pmatrix}, \begin{pmatrix}6\\6\end{pmatrix}, \begin{pmatrix}7\\5\end{pmatrix}, \begin{pmatrix}8\\10\end{pmatrix}\right\}.$$

(a) (2 points) Calculate the mean vector $\boldsymbol{\mu}$ of $\mathcal{S}$.

(b) (2 points) Subtract the mean vector $\boldsymbol{\mu}$ from the given feature vectors $\mathbf{x}^{(\ell)}$ representing the six data points.

(c) (4 points) Using $\frac{1}{N}\sum_{\ell=1}^{N}(\mathbf{x}^{(\ell)} - \boldsymbol{\mu})(\mathbf{x}^{(\ell)} - \boldsymbol{\mu})^{\top}$ as the (biased) estimator of the sample covariance matrix of a data set of N points, calculate the covariance matrix $\boldsymbol{\Sigma}$ of $\mathcal{S}$.

(d) (5 points) Calculate the eigenvalues of the covariance matrix $\boldsymbol{\Sigma}$. You may calculate it mathematically or using a Python-based software tool. You should show the steps to obtain the answers mathematically or write down the Python code to obtain them.

(e) (3 points) If we apply principal component analysis to project $\mathcal{S}$ to one dimension, calculate the resulting proportion of variance (PoV).