ECE 411 Introduction to Machine Learning
Fall 2024 Exam 1
Instructor: Dr. Chau-Wai Wong

This is a closed-book exam. You may use a scientific calculator with cleared memory, but not a smartphone or computer. One-sided letter-sized handwritten cheatsheet is allowed. You should answer ***all four*** problems.

Problem 1 [Linear Statistical Model] (25 pts) An ECE student named Tom plans to test the fuel economy of his car in terms of how many gallons are needed for driving one mile. He will do four test drives of x_i miles each, $i = 1, \ldots, 4$, and will measure the corresponding gas consumption Y_i gallons, $i = 1, \ldots, 4$ using a meter connected to his car's microcontroller. Denote the ground-truth fuel economy as k gallon/mile.

(a) Tom believes that the readings of the gas consumption Y_i are inaccurate but unbiased, so he set up a linear model $Y_i = kx_i + e_i$, $i = 1, \ldots, 4$, where e_i are measurement noise with zero-mean and variance σ^2. Express this model in the matrix–vector form. Explicitly define $\mathbf{y}$, $\mathbf{X}$, $\boldsymbol{\beta}$, and $\mathbf{e}$.

(b) Use the normal equation $\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y}$ to directly obtain the analytic form of the least-squares estimator $\hat{k}$ for the fuel economy, and simplify $\hat{\boldsymbol{\beta}}$ up to a point that it cannot be further simplified.

(c) Mathematically show that $\hat{k}$ is unbiased. (Hint: x_i's are constants whereas Y_i's are random variables.)

(d) Tom's friend proposed another way to estimate the fuel economy: $\tilde{k} = \frac{1}{4}\sum_{i=1}^{4} \frac{Y_i}{x_i}$. Is $\tilde{k}$ unbiased? If yes, prove it. If not, explain why.

(e) Tom's friend went back and figured out yet a third way to estimate the fuel economy: $\check{k} = \left(\sum_{i=1}^{4} Y_i\right) \Big/ \left(\sum_{i=1}^{4} x_i\right)$. Is $\check{k}$ unbiased? If yes, prove it. If not, explain why.

Problem 2 [Linear Independence, Basis, and Vector Space] (20 pts)

(a) Are vectors $[3 \;\; 2], [4 \;\; 5]$, and $[6 \;\; 7]$ linearly independent? Justify your answers.

(b) You are given a vector space $V = \text{span}\,\{[-1 \;\; 0 \;\; 1], [0.5 \;\; 0 \;\; 0.5]\}$.

(i) Express V in a set representation.

(ii) Can you find a basis for V?

(iii) Are $[5 \;\; 8 \;\; 0]$, $[8 \;\; 0 \;\; 5]$, and $[0 \;\; 5 \;\; 8]$ in vector space V? If yes, what is the coefficient for each vector of the basis you found in (ii)?

(c) Let

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

What is the column rank of matrix $\mathbf{A}$? Show detailed calculation/simplification steps. Do not use intermediate steps to show the row rank.

Problem 3 [Deep Learning Concepts] (30 pts)

(a) (16 pts) True or False. If a sentence is true, write "True" on the answer sheet (2 pts). Otherwise, write "False" (1 pt) and copy a few keywords (at most 3 words) responsible for making it a false statement (1 pt).

1. A convolutional layer with a kernel size of $5 \times 5 \times 27$ and an output size of $140 \times 140 \times 100$ suggests that the input data dimension is $144 \times 144 \times 27$, with the layer containing 100 individual convolutional filters.
2. In a convolutional neural network well-trained for image classification, layers near the output tend to focus on more low-level image structures, such as background noise and edge directions, to enhance classification accuracy.
3. Simple mathematical operations like addition and multiplication have straightforward derivatives, allowing gradient information to propagate back to target parameters without the risk of vanishing or exploding.
4. A recurrent neural network's internal state vector is designed to retain all information about the input tokens it has encountered, ensuring that none of the tokens are completely forgotten.
5. Transformers contextualize the embedding of each input sentence token by computing a weighted sum of the projected embeddings of all tokens, where less-related tokens are assigned more negative weights.
6. Self-attention weights are derived from pairwise similarities between the tokens of the input sentence and are normalized into a probability mass function using the softmax operation.
7. Both BERT and GPT are generative learners capable of learning the joint distribution of all tokens they have seen through the next-sentence prediction objective. The connection between "generative" and the aforementioned objective is connected through the iterative decomposition of the joint distribution of tokens using Bayes' rule.

8. A diffusion model can generate a photorealistic image through the iterative denoising of a Gaussian image. When additional guidance is provided as a parallel input, the generated image can appear significantly different.

(b) (14 pts) Suppose we have a cost function/loss defined as $J(\boldsymbol{\beta}) = 2\beta_1 + \beta_2^2 + 3\beta_1\beta_2$, where $\boldsymbol{\beta} = [\beta_1, \beta_2]^T$. Our goal is to use the gradient descent algorithm to find the best guess for $\boldsymbol{\beta}$. We use an initial guess $\boldsymbol{\beta}^{(0)} = [1, -1]^T$ and update $\boldsymbol{\beta}^{(n)}$ via the update rule $\boldsymbol{\beta}^{(n+1)} = \boldsymbol{\beta}^{(n)} - \eta \nabla J(\boldsymbol{\beta}^{(n)})$, where $\eta = 2$ is the learning rate/step size.

(i) Derive the analytic form of $\nabla_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$.

(ii) Manually perform the first two steps of the gradient descent using the provided update rule. Provide the numerical values of the intermediate results $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(2)}$.

(iii) Draw the trajectory of gradient descent on a 2-D plane of (β_1, β_2). With the help of axis tick labels, clearly indicate the precise locations of $\left\{\boldsymbol{\beta}^{(n)}\right\}_{n=0}^{2}$ and the two corresponding update vectors. [You do not need to draw contours for $J(\boldsymbol{\beta})$.]

(iv) Is the gradient descent is successful? Why?

Problem 4 [Softmax in Neural Network Training] (25 pts) A softmax function $f : \mathbb{R}^K \to \mathbb{R}^K$, $K \geq 2$ is defined as follows:

$$f_i(\mathbf{x}) = \frac{\exp(\beta x_i)}{\sum_{k=1}^{K} \exp(\beta x_k)}, \quad i = 1, \ldots, K,$$

where x_i is the ith component of vector $\mathbf{x} \in \mathbb{R}^K$.

(a) Explain in your own words what the softmax function can do in machine learning. Please limit your explanation to at most two complete sentences.

(b) Derive the following two expressions:

$$\frac{\partial f_2(\mathbf{x})}{\partial x_2} = \beta f_2(\mathbf{x}) \left[1 - f_2(\mathbf{x})\right], \tag{1a}$$

$$\frac{\partial f_2(\mathbf{x})}{\partial x_j} = -\beta f_2(\mathbf{x}) f_j(\mathbf{x}), \text{ for } j \neq 2. \tag{1b}$$

[Recall that $(\frac{u}{v})' = \frac{u'v - uv'}{v^2}$.]

(c) When training a neural network for a classification task, the gradient update term will always contain at least one partial derivative of softmax as a multiplicative term. Based on the theoretical result in (b), could you explain what will happen to the training process if β is set to a number that is either too small or too large?

[This Page Intentionally Left Blank]

Answer sheet for Problem 1 Name: Obtained Points:

Answer sheet for Problem 1

Answer sheet for Problem 2 Name: Obtained Points:

Answer sheet for Problem 2

Answer sheet for Problem 3 Name: Obtained Points:

Answer sheet for Problem 3

Answer sheet for Problem 4 Name: Obtained Points:

Answer sheet for Problem 4