ECE 461/661 Spring 2025 Mini-Exam 2

Introduction to Machine Learning for Engineers

Prof. Gauri Joshi and Prof. Carlee Joe-Wong

Wed, March 26, 2025, 9:00am - 9:55am PT/12:00pm - 12:55pm ET/6:00pm - 6:55pm CAT

Instructions

a. If a problem asks you which of its choices is TRUE, you should treat choices that may be either true or false as FALSE.

b. **Unless otherwise stated, only one option is correct in each multiple-choice question.** No partial credit will be given for multiple-choice or true/false questions.

c. For descriptive questions, make sure to explain your answers and reasoning. We will give partial credit for wrong answers if portions of your reasoning are correct. Conversely, correct answers accompanied by incomplete or incorrect explanations may not receive full credit.

d. You are allowed one **physical, single-sided** US-letter or A4 sized cheat sheet. No other notes or material (aside from blank pieces of scratch paper) may be used.

e. You may only use a pen/pencil, eraser, and scratch paper. The backside of each sheet in the exam can also be used as scratch paper. If you do not wish for us to grade your scratch work, please clearly indicate which parts of your work we should ignore.

f. Calculators are not permitted and not necessary. No other electronic devices such as phones, tablets or laptops can be used during the exam.

g. If you would like to ask a clarification question during the exam, raise your hand and an instructor or TA will come over. Note that we will not help you answer the questions but can give clarifications.

Problem	**Type**	**Points**
1-4	True/False	4
5-6	Multiple Choice	4
7	Descriptive	6
8	Descriptive	6
9	Honor Pledge	0
Total		20

1 True or False (4 points)

Problem 1: *[1 points]* Bagging is a method used to reduce the bias of classical decision trees.

- ◯ True
- ◯ False

Solution: FALSE. Bagging aims to reduce the variance of decision trees whereas boosting is used to reduce the bias of simple classifiers like decision stumps.

Problem 2: *[1 points]* Random forests increase the correlation between the trees in the ensemble by reducing the number of candidate features available at each split of the decision tree.

- ◯ True
- ◯ False

Solution: FALSE. Random forests decrease correlation between trees in the ensemble.

Problem 3: *[1 points]* The gradient of the ReLU activation function is

$$\frac{\partial ReLU(x)}{\partial x} = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$$

- ◯ True
- ◯ False

Solution: TRUE.

Problem 4: *[1 points]* Recurrent Neural Networks (RNNs) can process sequences of varying lengths.

- ◯ True
- ◯ False

Solution: RNNs process sequences step by step while maintaining a hidden state, allowing them to handle sequences of different lengths naturally. This makes them well-suited for tasks involving sequential data. Thus, the correct answer is **True.**

2 Multiple Choice (4 points)

Problem 5: *[2 points]* Which of the following adjustments is recommended when incorporating dropout into a neural network to address its effects on network capacity and convergence? **Only one option is correct.**

- ◯ Decrease the number of network layers, since dropout randomly deactivates neurons and thereby increases effective capacity.
- ◯ Reduce the learning rate to avoid overshooting during training.
- ◯ Increase the network size by a factor of n/p (where n is the original number of hidden units and p is the dropout probability).
- ◯ Avoid using a momentum-based optimizer, to counteract the slower error convergence caused by dropout.

Solution: C

Problem 6: *[2 points]* The AdaGrad optimizer proposes the following adaptive learning rate schedule for each element i of the parameter vector $\mathbf{w}$:

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}_i^{(t)} - \frac{\eta}{\sqrt{\sum_{k=1}^{t}(\nabla E(\mathbf{w}^{(k)})_i)^2 + \epsilon}} \nabla E(\mathbf{w}_i)$$

where $E(\cdot)$ denotes the error function we wish to minimize, $\epsilon > 0$ is a small constant, η is the nominal learning rate, and $\mathbf{w}_i^{(t)}$ denotes the ith element of the parameter vector $\mathbf{w}$ at time t. Which of the two problems below is avoided by the addition of ϵ? **Only one option is correct.**

- ◯ Exploding effective update when the sum of squares of gradients is very small.
- ◯ Vanishing effective update when the sum of squares of gradients is very large.

The AdaDelta optimizer modifies AdaGrad by changing the sum over squared gradients from $k = 1$ to $k = t$ to a sliding window from $k = t - \Delta$ to $k = t$. Which of the two problems below is avoided with this modification? **Only one option is correct.**

- ◯ Exploding effective update when the sum of squares of gradients is very small.
- ◯ Vanishing effective update when the sum of squares of gradients is very large.

Solution: A: Avoids exploding effective update because ϵ stops the denominator from becoming zero
B: Avoids vanishing effective update when the sum is over a large number of time steps.

3 Descriptive (12 pts)

Problem 7: *[6 points]* Suppose that you wish to learn a decision tree to solve a binary classification problem on the dataset $\left\{\left(\mathbf{x}^{(j)}, y^{(j)}\right), j = 1, 2, \ldots, 6\right\}$. We suppose that $\mathbf{x}^{(j)}$ is a 2-dimensional feature vector and use x_1, x_2 to respectively denote the first and second features. We use "circle" and "square" to denote our two classes, i.e., $y^{(j)} \in \{\text{circle}, \text{square}\}$.
Suppose that your first partition thresholds the feature x_1 by whether or not $x_1 \leq 2$. Table 1 shows the 3 data points that fall into each branch of this partition.

$x_1 \leq 2$			$x_1 > 2$		
Feature 1 (x_1)	Feature 2 (x_2)	Label (y)	Feature 1 (x_1)	Feature 2 (x_2)	Label (y)
1	-1	circle	3	2	square
0	2	circle	4	0	square
-1	1	square	5	-2	square

Table 1: Data points falling into the partitions $x_1 \leq 2$ and $x_1 > 2$ for Problem 7.

a. *[1.5 points]* Suppose that you decide to train only a decision stump, i.e., you stop training the tree after the first partition given above. What are the labels of the $x_1 \leq 2$ branch and the $x_1 > 2$ branch, given the data from Table 1? Briefly (in 1-2 sentences) explain your answer.

Solution: The $x_1 \leq 2$ branch should have label "circle" and the $x_1 > 2$ branch should have label "square," by majority voting of the data points in each branch.

b. *[1.5 points]* Now suppose that you wish to train a two-level decision tree. Find the next split of the $x_1 \leq 2$ branch, or explain why no additional split is needed. Remember to show your work.

Solution: Examining Feature 1, we see that a split of $x_1 \leq \theta$, where $\theta \in (-1, 0)$, will perfectly separate the circle and square data points. Examining Feature 2, we see that there is no threshold that perfectly separates our data points. Thus, we choose to split on $x_1 \leq \theta$. Full credit should be given for any feasible value of θ.

c. *[1.5 points]* Find the next split of the $x_1 > 2$ branch, or explain why no additional split is needed. Remember to show your work.

Solution: No additional split is needed, since all data points in this branch have the label "square."

d. *[1.5 points]* Your answers to parts (b) and (c) should give a two-level decision tree. Use the axes in Figure 1 to draw the resulting partition of the two-dimensional feature space (x_1, x_2). Include the label ("circle" or "square") for each region of the partition.

Solution: Vertical lines should be drawn at $x_1 = 2$ and $x_1 = \theta$, where $\theta \in (-1, 0)$. The middle region should be labeled "circle" and the other two "square."

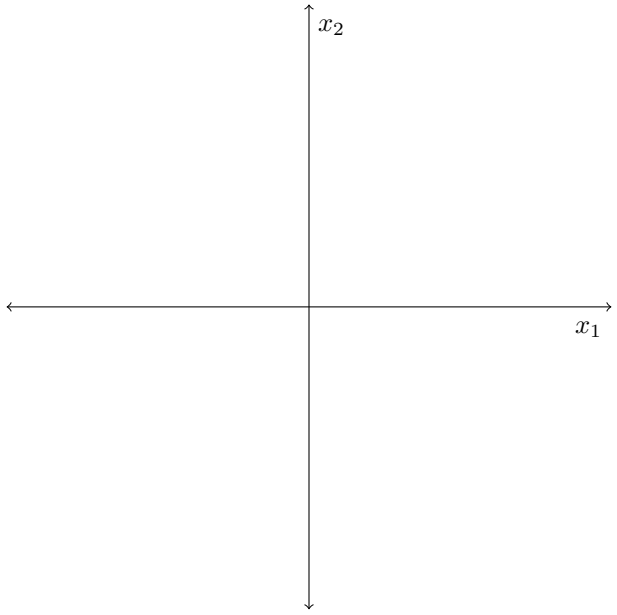

Figure 1: Draw your answer to problem 7d here.

WRITE YOUR ANSWER TO PROBLEM 7 BELOW

CONTINUE YOUR ANSWER TO PROBLEM 7, IF NEEDED

Problem 8: *[6 points]* Consider that you have a 10×10 input grayscale image where each element of the input takes an integer value in the range $[0, 255]$. The desired output of a neural network model is the probability $y \in (0, 1)$ indicating the confidence of the model that the image contains a car.

a. *[2 points]* Consider a fully connected neural network that flattens the 10×10 size input into 100 input neurons, and connects them to one hidden layer that has 64 neurons. The output layer maps these 64 neurons to the probability that the image contains a car.

How many trainable parameters are there in the network? Include bias terms for the input and hidden layer. Write down the correct expression for the number of parameters and simplify it as much as possible without a calculator. Make sure to explain your reasoning.

Solution: The number of trainable parameters is $(100 + 1) \times 64 + (64 + 1) * 1 = 6529$. The addition of 1s corresponds to the bias terms

b. *[2 points]* Suppose we instead use a convolutional layer to map the input to the hidden layer. The convolutional layer has 1 filter of size 3×3 with a stride of 1 and no padding, so that its output has dimension 8×8 (same as the number of hidden layer neurons in part (a)). The output layer remains the same as in part (a) and maps the $8 \times 8 = 64$ hidden layer neurons to the probability that the image contains a car.

How many trainable parameters are there in this network? Include a bias term for each filter. Write down the correct expression for the number of parameters and simplify it as much as possible without a calculator. Make sure to explain your reasoning.

Solution: The number of trainable parameters is $(9 + 1) + (64 + 1) * 1 = 75$. The addition of 1s corresponds to the bias terms.

c. *[1 points]* For both the neural networks described in parts (a) and (b), we use the sigmoid activation for the final layer, which contains a single neuron. Write down the expression of the final probability of the image containing a car in terms of the outputs $y_1, \ldots, y_{64}$ of the 64 hidden layer neurons, and the weights and biases mapping it to z, the input to the final layer.

Solution: The probability is $\sigma(z) = \sigma(\sum_{i=1}^{64} w_i y_i + b)$, where $\sigma(z) = 1/(1 + e^{-z})$.

d. *[1 points]* What is a good choice of the loss function when you train the parameters of the neural network? Write your answer in terms of the values $z^{(n)}$, the input to the final layer for the n-th training example in a dataset containing N samples. The label of the n-th sample is denoted by target $t^{(n)}$, which is 1 if the image contains a car and 0 otherwise. Make sure to explain (in 1-3 sentences) your reasoning for the choice of loss function.

Solution: The cross-entropy loss function:

$$L = \sum_{n=1}^{N} \left(t^{(n)} \log \sigma(z^{(n)}) + (1 - t^{(n)}) \log(1 - \sigma(z^{(n)})) \right) \tag{1}$$

is a good choice. It maps $z^{(n)}$ to a probability between 0 and 1 by applying the sigmoid function to it.

WRITE YOUR ANSWER TO PROBLEM 8 BELOW

CONTINUE YOUR ANSWER TO PROBLEM 8, IF NEEDED

4 Honor Pledge

Problem 9: *[0 points]* To affirm that you did not cheat on the exam, please write out the below statement. Sign your name beneath it. **Failure to do so will be taken as a sign that you have cheated on the exam.**

I pledge my honor that I neither gave nor received unauthorized assistance on this examination.