

CSE 446/546 Spring 2026 Final Exam

June 10, 2026

Name _____

UW NetID (not the numbers) _____

Please **wait** to open the exam until you are instructed to begin, and please take out your Husky Card and have it accessible when you turn in your exam.

Instructions: This exam consists of a set of short questions (True/False, multiple choice, short answer).

- For multiple choice questions, fill in the circle(s) to indicate your answers. Do not write your answer to the side. Example:



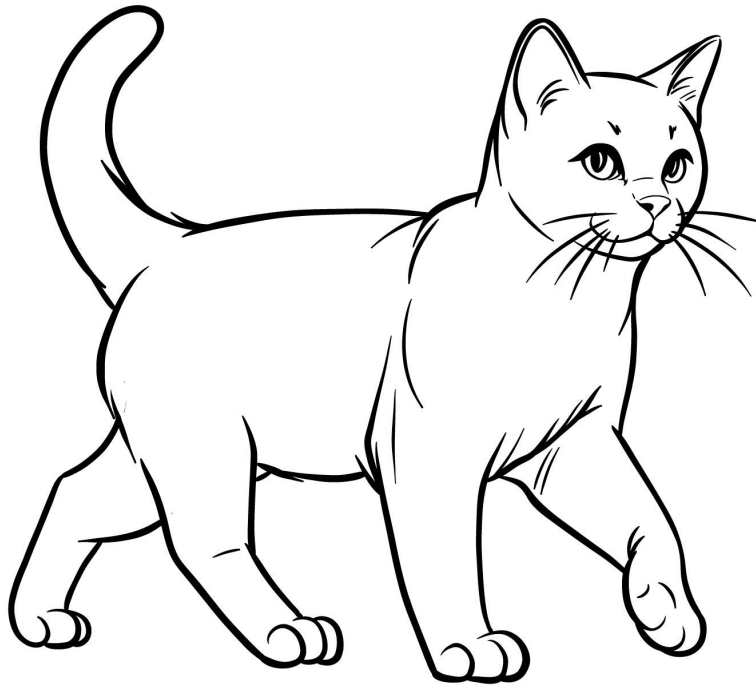
Selected answer.



Unselected answer.

- Multiple choice questions marked with One Answer should only be marked with one answer. All other multiple choice questions are Select All That Apply, in which case any number of answers may be selected (**including none, one, or more**).
- For Select All That Apply questions, we will give partial credit.
- For each short answer question, please write your answer in the provided space.
- If you need to change an answer or run out of space, clearly indicate what you would like graded as your final answer. If we cannot determine your final answer, we cannot give credit.
- If you complete the exam before the last 5 minutes, you may turn in your exam to course staff. Otherwise, please remain in your seats for the last 5 minutes of the exam.
- If you need to use the restroom, please first leave your exam with an instructor at the front of the room. You can then collect and resume your exam when you return.

This image is included only to cover the back of this page. It has no relation to the exam.



1.	1 points
----	----------

Suppose we have a high-dimensional feature map $\phi(\mathbf{x})$ and a valid kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$.

When transitioning to Kernelized Ridge Regression, which statement correctly describes how the kernel function is utilized?

One Answer

- (a) The kernel function replaces the target vector $\mathbf{y}$, allowing the model to switch from regression to classification.
- (b) **The problem is reformulated so that all high-dimensional inner products $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ are replaced by $K(\mathbf{x}_i, \mathbf{x}_j)$, bypassing explicit feature computation.**
- (c) Computing $\phi(\mathbf{x})$ lets us operate in an n -dimensional subspace of the $d \times d$ feature space, making it more computationally efficient.
- (d) The algorithm uses $\phi(\mathbf{x})$ to convert every point first, then fits a standard linear regression model to those d -dimensional features.

Solution: Option B is correct.

The core of the “kernel trick” is that we rewrite the learning algorithm so it only interacts with the data via inner products of the feature maps. By substituting $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$, we can train and predict in an infinite- or high-dimensional space without ever having to explicitly compute or store the massive vector $\phi(\mathbf{x})$.

2. 1 points

Determine whether the following statement is true or false. If you select false, provide a counterexample.

- (a) **Statement:** Any convex function $f : \mathbb{R} \rightarrow \mathbb{R}$ is differentiable everywhere.
- One Answer

- (a) True
- (b) False. Counterexample: $f(x) = \underline{\hspace{2cm}}$

Solution: (A) False. Counter example: $f(x) = |x|$. Convex but not differentiable at $x = 0$.

3. 1.5 points

A radiologist wants to classify chest X-rays where each image may simultaneously show multiple conditions such as pneumonia, pleural effusion, and cardiomegaly.

- (a) A colleague proposes using softmax classification with one output per condition. Is this appropriate? One Answer

☐ a Yes

☒ b No

- (b) Justify your answer in one sentence.

Solution:

- (a) **No.**

- (b) Softmax enforces that the output probabilities sum to 1, implicitly assuming the classes are **mutually exclusive** — but here multiple conditions can be present simultaneously, directly violating this assumption.

4. 2 points

In 1D logistic regression with a bias term, the model is:

$$P(Y = 1 \mid x) = \sigma(w_0 + w_1 x) = \frac{1}{1 + e^{-(w_0 + w_1 x)}} \quad (1)$$

The decision boundary occurs where the model is equally likely to predict either class.

- (a) For each set of weights below, derive the decision boundary x^* and fill in the table:

	w_0	w_1	x^*
Base Case	-2	1	
Case A	-4	1	
Case B	-2	0.5	

- (b) Comparing Base Case to Case A, what is the effect on the decision boundary of decreasing w_0 ?

-
- (c) Comparing the **Base Case** to **Case B**, decreasing w_1 alters the location of the decision boundary. Which statement best describes the additional effect of decreasing w_1 on the model's geometry? *Hint: $\sigma'(z) = \sigma(z)(1 - \sigma(z))$.* One Answer

- (a) It shifts the entire curve vertically upward by 0.5 units.
- (b) **It flattens the curve, making the transition between classes more gradual and wider around the decision boundary.**
- (c) It sharpens the curve, making the model shift more abruptly between classes at the boundary.
- (d) It has absolutely no effect on the shape or steepness of the curve.

Solution:

- (a) The decision boundary occurs where $w_0 + w_1 x^* = 0$, giving $x^* = -w_0/w_1$:

	w_0	w_1	x^*
Base Case	-2	1	2
Case A	-4	1	4
Case B	-2	0.5	4

- (b) Decreasing w_0 from -2 to -4 shifts the decision boundary to the right (from $x^* = 2$ to $x^* = 4$), without changing the steepness of the sigmoid. The bias w_0 controls the **location** of the boundary.
- (c) **Option B.** Decreasing w_1 from 1 to 0.5 moves the boundary to the right ($x^* = 4$), but additionally **flattens the sigmoid**: the model becomes less confident near the boundary, transitioning more gradually between the two classes. w_1 controls both the location and the **steepness** of the decision.

5. 2 points

Consider a 1D binary classification dataset fit with a logistic regression model with no bias term. Assume that the classifier is able to perfectly fit the training data, i.e. it predicts the correct label y_i for each x_i . The logistic loss on the training set is:

$$\mathcal{L}(w) = \sum_{i=1}^N \log(1 + \exp(-y_i w x_i)) \quad (2)$$

- (a) As $w \rightarrow \infty$, what does $\mathcal{L}(w)$ approach? Does it ever reach this value exactly? Explain why or why not.

- (b) What does this imply about the MLE solution $\hat{w}$ on linearly separable data?

Solution:

- (a) Assuming that the separating w is positive, that is $y_i x_i > 0$ for all i , then as $w \rightarrow \infty$ we have $-y_i w x_i \rightarrow -\infty$, thus $\exp(-y_i w x_i) \rightarrow 0$ for each i , so

$$\mathcal{L}(w) = \sum_{i=1}^N \log(1 + \exp(-y_i w x_i)) \rightarrow \sum_{i=1}^N \log(1 + 0) = 0.$$

The loss approaches 0 but never reaches it exactly, since $e^{-x} > 0$ for all finite x .

Note: In the case that the separating w is negative, then as $w \rightarrow \infty$ we have $-y_i w x_i \rightarrow \infty$ for each i . Hence $\mathcal{L}(w) \rightarrow \infty$ is also an acceptable answer since the question did not specify.

(b) Since $\mathcal{L}(w) > 0$ for all finite w and $\mathcal{L}(w) \rightarrow 0$ only as $w \rightarrow \infty$, there is no finite minimizer. The MLE solution is $\hat{w} \rightarrow \infty$, meaning gradient descent will drive the weights to grow without bound on linearly separable data.

6. 1 points

You have a logistic regression model that classifies a 50x50 image into a single numerical digit (0-9). What function is used to convert the class scores into a probability distribution to achieve this classification? Also provide the general formula for the function. (Name:

0.25 points ; Formula: 0.75 points)

Name of function: _____ Formula: _____

Solution: This is a multi-class logistic regression problem. We can use **softmax** to predict the class. In this scenario it would be: $P(y = k) = \frac{e^{z_k}}{\sum_{j=0}^9 e^{z_j}}$, where z_k is the raw score for class k. The predicted digit is then $\hat{y} = \underset{k}{\operatorname{argmax}} P(y = k)$

7. 1 points

In which scenario would it be appropriate to use logistic regression? Select All That Apply

- ☐ (a) Predicting whether a credit card transaction is a fraud based on the time and location.
- ☐ (b) Predicting the exact cost of a house based on its architecture style, neighborhood, and school district.
- ☐ (c) Predicting if a patient has a disease or not based on the patient's symptoms and medical history.
- ☐ (d) Predicting the exact rainfall depth for a given day in Seattle based on historical data.

Solution: A, C

Logistic regression can be used as a binary classifier, predicting whether something belongs to a class or not. A is correct since it is predicting whether or not a transaction is a fraud (1) or not a fraud (0). C is correct since a patient can have a disease (1) or doesn't have a disease (0). B and D are incorrect as they are more suited to regression

8. 1 points

How does the k-nearest neighbors (KNN) algorithm make predictions on unseen data? What is one disadvantage of this method?

Solution: It learns a nonlinear decision boundary between classes by finding the K nearest data points to a given input and averaging their target value. kNN struggles when there are a large number of features or the data is highly imbalanced.

9. 1 points

An RGB image is passed through a convolutional layer with 10 filters of size 5×5 . How many learnable parameters does this layer have, assuming each filter has one bias term? One Answer

- ☐ (a) $5 \times 5 \times 10 + 10$

(b) $5 \times 5 \times 3 + 10$

(c) $5 \times 5 \times 3 \times 10 + 10$

(d) $5 \times 5 \times 3$

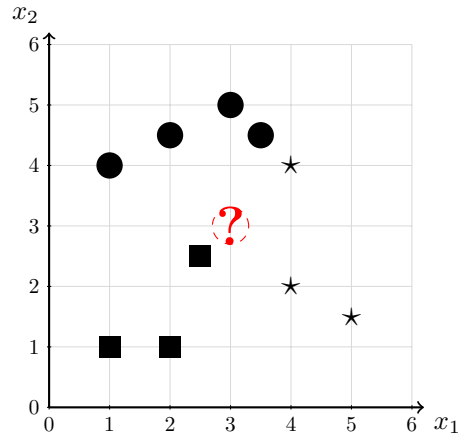
(e) $5 \times 5 \times 3 \times 10$

Solution: C

Each filter spans all 3 input channels, so one filter has $5 \times 5 \times 3$ weights. With 10 filters, the layer has $5 \times 5 \times 3 \times 10$ weights, plus 10 bias terms.

10. 1.5 points

Consider the 2D dataset plotted below containing 10 training instances belonging to three distinct classes: **Square** (■), **Circle** (●), and **Star** (★). A new unclassified test point, denoted by the question mark (?), is located at the coordinates (3,3).



Using the **Standard Euclidean Distance** metric, determine the predicted class for the test point under the following configurations:

(a) What is the predicted class for the test point when using $k = 1$?

Class ($k = 1$): _____

(b) What is the predicted class for the test point when using $k = 3$?

Class ($k = 3$): _____

(c) What is the predicted class for the test point when using $k = 9$?

Class ($k = 9$): _____

Solution: To classify the test point at (3,3), we find the closest training points using squared Euclidean distance: $d^2 = (x_1 - 3)^2 + (x_2 - 3)^2$.

(a) **For** $k = 1$: The single closest neighbor to (3,3) is the **Square** located at (2.5, 2.5).

- Distance $d^2 = (2.5 - 3)^2 + (2.5 - 3)^2 = 0.25 + 0.25 = 0.5 \implies d \approx 0.71$

Answer: Square

(b) **For** $k = 3$: The three closest neighbors to (3,3) are:

(a) **Square** at (2.5, 2.5) $(d = \sqrt{0.5} \approx 0.71)$

(b) **Star** at (4, 4) $(d = \sqrt{(4 - 3)^2 + (4 - 3)^2} = \sqrt{2} \approx 1.41)$

(c) **Star** at $(4, 2)$

$$(d = \sqrt{(4-3)^2 + (2-3)^2} = \sqrt{2} \approx 1.41)$$

Tallying the votes: **Square** (1 vote), **Star** (2 votes).

Answer: Star

- (c) **For** $k = 9$: There are 10 total training points in the entire dataset. A neighborhood of $k = 9$ means we look at almost the entire dataset, excluding only the single furthest point (the Square at $(1, 1)$, which is $d = \sqrt{8} \approx 2.83$ away).

Counting the remaining 9 closest points by class:

- **Circles**: 4 total points available $\implies$ 4 votes
- **Stars**: 3 total points available $\implies$ 3 votes
- **Squares**: 2 points remaining (excluding $(1, 1)$) $\implies$ 2 votes

Answer: Circle

11. 1 points

Which of the following is a reasonable method for finding the number of principal components q to use for PCA? One Answer

- (a) Use cross-validation to find the q that minimizes reconstruction error
- (b) Choose q such that the top q components explain 95% of the total variance
- (c) Choose q to be the maximum number of non-zero eigenvalues in the covariance matrix Σ
- (d) Choose q to be the rank of the data matrix X

Solution: B Option B is correct since it gives the correct tradeoff between dimensionality reduction and information loss. A is incorrect because adding more components always improves reconstruction, so cross-validation would select $q = d$. C is incorrect because the maximum number of non-zero eigenvalues in Σ is an upper bound on q and would produce no dimensionality reduction. D is incorrect for the same reason.

12. 1 points

Match each classifier boundary description below to the correct classifier from the list. Write the corresponding letter (A–E) in the blank provided. Each classifier is used at most once.

Classifiers

- (A) Decision Tree
- (B) Random Forest
- (C) Logistic Regression

- (D) Neural Network
- (E) K-Nearest Neighbors
- (F) Gaussian Mixture Model
- (G) Regularized Kernelized Regression

- _____ (i) The boundary is a single hyperplane dividing the feature space into two half-spaces.
- _____ (ii) The boundary consists of axis-aligned rectangular regions, each assigned a single class label. The boundary can perfectly fit any training set given sufficient model complexity.
- _____ (iii) The boundary is also axis-aligned and blocky, but softer near contested regions where individual models disagree. It is produced by aggregating many independently trained models.
- _____ (iv) The boundary is smooth and nonlinear, produced by composing many linear transformations with nonlinear activations. It does not rely on pairwise similarity between points.

Solution:

- (i) **(C)** Logistic Regression
- (ii) **(A)** Decision Tree
- (iii) **(B)** Random Forest
- (iv) **(D)** Neural Network

13. 1 points

Consider a kernel ridge regression model trained on a 1D dataset of two points, $x_1 = 2$ and $x_2 = 4$. The learned weights are $\alpha = [\mathbf{3}, -\mathbf{1}]^T$ and the bias term is $b = -2$.

We use a simple linear kernel defined as:

$$K(x, x') = x \cdot x'$$

Compute the predicted value y_{test} for a new test point $x_{\text{test}} = 3$.

$$y_{\text{test}} = \underline{\hspace{2cm}}$$

Solution: The prediction formula for kernelized regression is:

$$y_{\text{test}} = \sum_{i=1}^N \alpha_i K(x_i, x_{\text{test}}) + b$$

Expanding this for our two training points ($N = 2$):

$$y_{\text{test}} = \alpha_1 K(x_1, x_{\text{test}}) + \alpha_2 K(x_2, x_{\text{test}}) + b$$

1. Compute the kernel evaluations using $K(x, x') = x \cdot x'$:

$$K(x_1, x_{\text{test}}) = 2 \cdot 3 = 6$$

$$K(x_2, x_{\text{test}}) = 4 \cdot 3 = 12$$

2. Substitute the weights $\alpha_1 = 3$, $\alpha_2 = -1$, and $b = -2$ into the prediction equation:

$$y_{\text{test}} = 3(6) + (-1)(12) + (-2)$$

$$y_{\text{test}} = 18 - 12 - 2$$

$$y_{\text{test}} = 4$$

14. 1 points

A simple RNN reads a long movie review. The first sentence says, “I’ll be clear from the start – this is a bad movie.” Several paragraphs later, the review concludes, “It was saved at the box office by the world-class actors and the thrilling trailer.” Considering only these two sentences, what do you expect the predicted sentiment to be? One Answer

- (a) Positive
- (b) Negative

Solution: A

The hidden state is repeatedly overwritten as new inputs arrive. Because it’s difficult for RNNs to preserve long-term memory, the prediction may rely more heavily on later tokens.

15. 1.5 points

Suppose you are growing a decision tree for a binary classification task to identify spam emails. The root node (parent node) contains a dataset of **12 emails total** (6 spam, 6 not spam).

You are evaluating three independent **candidate features** to determine the optimal first split for the root node. The distributions resulting from splitting on each feature are detailed below:

Candidate Split	Left Child Node		Right Child Node		Total Size (n)
	Spam	Not Spam	Spam	Not Spam	
Split on Feature A	3	3	3	3	12
Split on Feature B	6	0	0	6	12
Split on Feature C	4	0	2	6	12

Recall Formulas & Reference values:

$$H(\text{node}) = - \sum_{c \in C} p(c) \log_2 p(c) \quad \Delta H = H(\text{parent}) - \sum_{i \in \text{children}} \frac{|R(i)|}{n} H(R(i))$$

For your convenience: $\log_2(1) = 0$, $\log_2(2) = 1$, and the entropy of a distribution with a (1 : 3) ratio is $H(1/4, 3/4) \approx 0.81$.

- (a) **Geometric and Qualitative Intuition:** Without performing any calculations using the formulas above, inspect the table and complete the following statements:
- Split _____ achieves the absolute lowest possible uncertainty across its children because it creates perfectly pure leaf nodes.
 - Split _____ provides absolutely zero new information about the target classes, meaning its Information Gain (ΔH) must equal 0.
- (b) **Evaluating Asymmetry:** Look closely at **Feature C**. It successfully isolates 4 spam emails into a perfectly pure Left Child, but leaves a mixed Right Child.

Compute the total weighted child entropy for **Feature C**. Then, explain why the perfect purity ($H = 0$) of the left child isn't enough to make this an optimal split.

Solution:

- (a) • Split **Feature B** achieves the lowest possible impurity because both resulting leaves have single-class uniform targets ($H = 0$).

- Split **Feature A** provides zero information gain because the class probabilities in both children ($3/6 = 0.5$) match the parent node distribution exactly, reducing no uncertainty.

(b) The parent entropy is $H(\text{parent}) = 1.0$. For Feature C:

- Left Child: $|R(\text{Left})| = 4$. Probability distribution is $(4/4, 0/4) \implies H(\text{Left}) = 0$.
- Right Child: $|R(\text{Right})| = 8$. Probability distribution is $(2/8, 6/8) = (1/4, 3/4) \implies H(\text{Right}) \approx 0.81$.

The weighted child entropy is:

$$\sum_i \frac{|R(i)|}{n} H(R(i)) = \frac{4}{12}(0) + \frac{8}{12}(0.81) = 0 + 0.54 = 0.54.$$

$\Delta H_C = 1.0 - 0.54 = \mathbf{0.46}$. This shows how the larger size of the unpure right child drags down the total value of the split.

16. 2 points

Suppose you want to train a Random Forest model. Let N be the number of training examples, d be the total number of features in the training data, m be the number of features randomly selected at each split, and $n_{\min}$ be the minimum node size.

Consider the following algorithmic skeleton for training a single decision tree T_b within the ensemble:

Algorithm: Train Tree T_b

Step 1: _____

_____ from the training data.

Step 2: Set up the root node containing this data.

Step 3: For each node in the tree, if the node size is greater than $n_{\min}$:

(a) _____ [**B**]

(b) Find the best split-point among these selected variables.

(c) Split the node into two daughter nodes.

Step 4: Recursively repeat Step 3 until all terminal nodes have $n_{\min}$ or fewer examples.

(a) Fill in the blank for step 1.

(b) Which of the following correctly identifies the operations that should fill placeholder [B]?

One Answer

- (a) Select all d variables to find the globally optimal split.
- (b) Draw a bootstrap sample of size N with replacement.
- (c) **Select a random subset of m variables out of the d total features.**
- (d) Select the top m features with the highest correlation to the target variable.

Solution:

- (a) **Step 1 Answer: bootstrap sample of size N (with replacement)**
Random Forests utilize bagging (Bootstrap Aggregating). Each individual tree is trained on a distinct dataset generated by sampling N points uniformly at random with replacement from the original collection.
- (b) **Option C: Select a random subset of m variables out of the d total features.**
To decorrelate individual trees, Random Forests intentionally restrict each splitting choice to a randomly drawn candidate subset of m features rather than scanning all d available features.

17. 1 points

In the theoretical motivation for Random Forests, we analyzed the variance of the average of B individual tree predictors, given by the formula:

$$\text{Var} \left(\frac{1}{B} \sum_{i=1}^B Y_i \right) = \frac{\sigma^2}{B} + \frac{B-1}{B} \rho \sigma^2$$

where σ^2 is the variance of an individual tree and ρ is the correlation coefficient between predictors.

Based on this framework and ensemble theory, which of the following best explains why the Random Forest algorithm achieves a significant improvement over a single decision tree or standard bagging? One Answer

- (a) By increasing the number of trees B to infinity when trees are perfectly correlated ($\rho = 1$), the total variance converges to 0.
- (b) **By introducing feature randomness at each node split, the algorithm decreases the correlation ρ between the trees, allowing the second term of the variance formula to be reduced as B increases.**
- (c) The algorithm dramatically reduces the variance of the ensemble by decreasing the variance σ^2 of individual trees below that of standard pruned trees.
- (d) According to the Condorcet Jury Theorem, ensemble performance improves only if individual trees are perfectly independent ($\rho = 0$) and have an accuracy of 100%.

Solution: Option B.

The core breakthrough of Random Forests over standard bagging is feature subspace sampling (choosing m out of p variables at each split). This intentional randomization reduces the correlation ρ between the trees in the ensemble. As the variance formula shows, if $\rho = 1$, increasing B does not reduce the variance at all; but when $\rho < 1$, increasing the number of trees B allows us to diminish the impact of the variance components, yielding a more stable and accurate model.

18. 1 points

Which of the following statements about neural networks are true? Select All That Apply

- (a) Fully connected layers are preferred over convolutional layers for image processing because they preserve spatial structures like edges and textures.
- (b) **Overparameterized neural networks can sometimes be more effective for generalization than neural networks with less parameters.**

- c) The Universal Approximation Theorem tells us that a two-layer neural network with some finite number of neurons and suitable nonlinear activation (e.g. ReLU) can approximate any continuous function on a compact subset of $\mathbb{R}^n$ to arbitrary precision.
- d) Activation functions are primarily used in neural networks to improve training stability.
- e) A neural network without nonlinear activation functions is always equivalent to a single linear transformation, regardless of the depth and width of the network.

Solution:

- **Dense vs. Conv Layers (False):** Fully connected layers flatten images, destroying spatial structures. Convolutional layers are preferred for image processing because their localized receptive fields explicitly preserve spatial relationships like edges and textures.
- **Overparameterization (True):** In modern deep learning (related to the "double descent" phenomenon), overparameterized models can generalize exceptionally well because the implicit bias of optimization algorithms guides them toward smooth, robust solutions.
- **Universal Approximation Theorem (True):** A single hidden layer with a non-linear activation function possesses sufficient capacity to approximate any bounded continuous function arbitrarily closely, provided it has enough hidden units.
- **Activation Functions (False):** The primary purpose of activation functions is to inject **non-linearity** into the model, allowing it to learn complex mappings. They often make initialization and training stability **harder** to manage, not easier.
- **Linear Composition (True):** The composition of consecutive linear functions is mathematically equivalent to a single linear transformation ($W_2(W_1x) = (W_2W_1)x$). Without non-linear activations, increasing depth adds zero representational capacity.

19. 3 points

Suppose you are building a 3-layer neural network without a bias term for classifying 2D coordinates into two classes ($y \in \{0, 1\}$). The network architecture is as follows:

- **Layer 1:** $z_1 = \text{ReLU}(W_1 x)$
- **Layer 2:** $z_2 = \text{ReLU}(W_2 z_1)$
- **Layer 3:** $\hat{y} = \sigma(W_3 z_2)$

Where ReLU and Sigmoid (σ) are applied elementwise. Consider the following input x and weight matrices W_1 , W_2 , and W_3 :

- $x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- $W_1 = \begin{bmatrix} 2 & -2 \\ 1 & 1 \end{bmatrix}$
- $W_2 = \begin{bmatrix} 3 & -2 \\ -1 & 2 \end{bmatrix}$
- $W_3 = \begin{bmatrix} -2 & 1 \end{bmatrix}$

To evaluate the performance of this network, calculate the following values

- (a) Compute the first hidden layer vector z_1 .
- (b) Compute the second hidden layer vector z_2 .
- (c) Compute the network's final predicted classification probability $\hat{y}$. Which class does it predict? *Hint: $e^{-1} \approx 0.37$, $e^{-3} \approx 0.05$, $e^{-5} \approx 0.0067$, $e^{-7} \approx 0.0009$. You don't need an exact value for $\hat{y}$ here, just an approximation to determine the predicted class.*

- (d) Suppose we introduce an additional task where we want to minimize the squared difference between the network's raw logits $s = W_3 z_2$ and a target value of $t = 5$. Let the loss function be:

$$\ell = \frac{1}{2}(s - t)^2$$

Calculate the gradient of the loss with respect to the weights W_3 (i.e. $\frac{\partial \ell}{\partial W_3}$).

Solution:

$$(a) \quad W_1 x = \begin{bmatrix} 2 & -2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix} \rightarrow z_1 = \text{ReLU}\left(\begin{bmatrix} -2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

$$(b) \quad W_2 z_1 = \begin{bmatrix} 3 & -2 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 3 \end{bmatrix} = \begin{bmatrix} -6 \\ 6 \end{bmatrix} \rightarrow z_2 = \text{ReLU}\left(\begin{bmatrix} -6 \\ 6 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 6 \end{bmatrix}$$

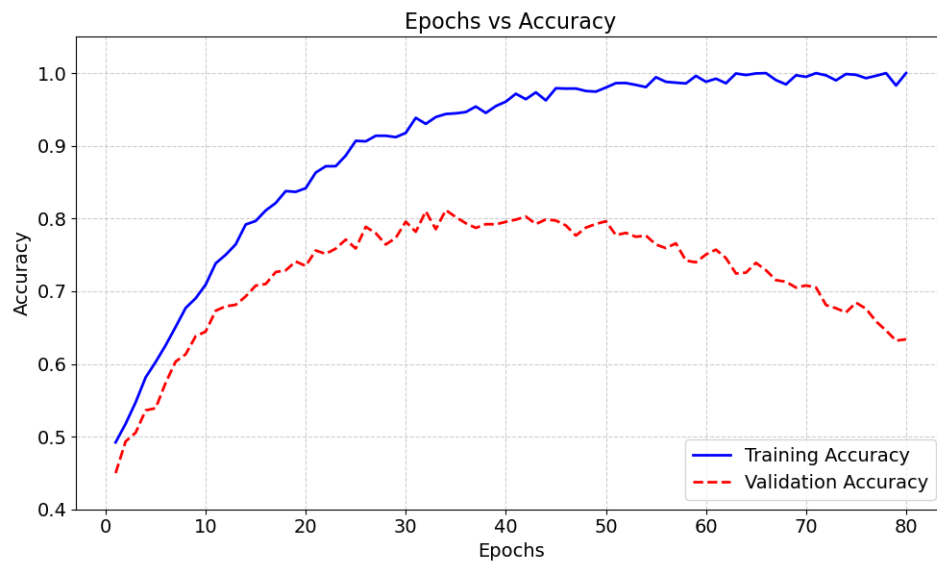
$$(c) \quad W_3 z_2 = \begin{bmatrix} -2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 6 \end{bmatrix} = 6 \rightarrow \hat{y} = \sigma(6) = \frac{1}{1+e^{-6}}$$

Since e^{-6} lies between $e^{-5} \approx 0.0067$ and $e^{-7} \approx 0.0009$, it's a small value, so $\hat{y} = \frac{1}{1+\text{small value}} > 0.5 \rightarrow$ predicted class is 1

$$(d) \quad \text{By the chain rule, we have } \frac{\partial \ell}{\partial W_3} = \frac{\partial \ell}{\partial s} \cdot \frac{\partial s}{\partial W_3}. \text{ Solving for each term, we get } \frac{\partial \ell}{\partial s} = s - 5 \text{ and } \frac{\partial s}{\partial W_3} = z_2. \text{ Putting it together } \rightarrow \frac{\partial \ell}{\partial s} \cdot \frac{\partial s}{\partial W_3} = (s - 5)z_2$$

20. 2.5 points

You are training a neural network for a classification task, and get the following plot when you plot accuracy against the number of epochs.



- (a) What training phenomenon is displayed in this plot, and (roughly) at which epoch does it become apparent? Briefly justify your answer.

- (b) You decide to modify your model and/or training algorithm to fix the problem. State two approaches you could take that would likely address the issue. (Note that since you're not given the model type, state generic approaches that work across various models).

Approach 1: _____

Approach 2: _____

- (c) For one of your suggested approaches from part (b), state a potential issue with the approach.

Solution:

- (a) Overfitting; around epoch 40. The gap between the training curve and validation curve starts increasing roughly around epoch 40. The increasing gap indicates overfitting, as it means that the model is still improving on the training data but getting worse on the validation data.
- (b) Valid approaches: simplify the model, collect more data, add regularization, early stopping
- (c) Simplify the model: if we oversimplify, we may underfit; collect more data: collecting new data is often expensive; add regularization: if we add too much regularization, we may underfit; early stopping: we may stop training before the model has fully converged.

21. 1 points

Which of the following statements about convolution operations in CNNs are true?

Select All That Apply

- ☐ (a) The same filter weights are applied at every spatial location of the input.

- (b) Each output neuron is connected to every input neuron.
- (c) Convolutional layers can detect local patterns such as edges and textures.
- (d) Adding more filters in a layer increases the number of output channels.

Solution: A, C, D

CNNs use **parameter sharing** (A): the same filter slides across the input, unlike fully connected layers where each connection has a unique weight (ruling out B). Local filters are well-suited to detect low-level patterns like edges (C). Each filter produces one output channel, so more filters mean more channels (D).

22. 1 points

Which of the following correctly describes $P(X; \theta)$ for an autoregressive language model?

- (a) $P(X; \theta) = \prod_{t=1}^L P(X_t | X_{t-1}; \theta)$
- (b) $P(X; \theta) = \prod_{t=1}^L P(X_t | X_{i < t}; \theta)$
- (c) $P(X; \theta) = \sum_{t=1}^L P(X_t | X_{i < t}; \theta)$
- (d) $P(X; \theta) = \sum_{t=1}^L P(X_t, X_{t-1} | X_{i < t-2}; \theta)$

Solution: b

An autoregressive language model is conditioned on all previous inputs. A is wrong because it is only conditioned on the previous input. C & D are wrong because it is not a sum.

23. 2 points

- (a) Listed below are the six main steps of the backpropagation process, in randomized order. Order them (by letter) in the correct way:
- A. Compute the gradients ($\frac{\partial L}{\partial W}$, $\frac{\partial L}{\partial b}$) for a hidden layer.
 - B. Apply the gradient updates to the weight and bias matrices.
 - C. Propagate the error signals (δ) backward from later layers to earlier hidden layers.
 - D. Compute the activations at each neuron based on the input.
 - E. Compute the initial loss and error signal at the output layer nodes.
 - F. Compute the gradients ($\frac{\partial L}{\partial W}$, $\frac{\partial L}{\partial b}$) for the output layer.

Ordering: _____

(b) Below is a list of observations/actions you may observe or take during the backpropagation process. For each item, write the corresponding stage from part A. You may just write the letter (i.e. if an item belongs to the “Compute the error at the output node” stage, just write ”E”). Letters may be used once, more than once, or not at all.

- Subtract learning rate $\times$ gradient from a weight matrix: _____
- Compare the prediction to the true value/label: _____
- Multiply the output error by the activations coming from the previous hidden layer: _____
- Each hidden node receives a weighted sum of the errors from the nodes it fed into: _____
- Where the vanishing gradient problem originates in deep networks using Sigmoid activation: _____
- Multiply each hidden node’s error by the activation feeding into it from the previous layer: _____
- If the learning rate is too large here, training may diverge: _____

Solution:

(a) **Correct Ordering:** **D $\rightarrow$ E $\rightarrow$ F $\rightarrow$ C $\rightarrow$ A $\rightarrow$ B**

(b) (in order of the listed observations/actions) B, E, F, C, A, A, B; B, E, F, C, C, A, B was also accepted

24. 1 points

Max pooling is applied to a feature map. Which of the following best captures its primary purpose? One Answer

- (a) To increase the spatial resolution of the feature map
- (b) To introduce learnable parameters that compress the representation
- (c) To reduce the dimension of the feature map while retaining the most prominent activations
- (d) To normalize activations across the spatial dimensions

Solution: C

Max pooling reduces spatial dimensions by selecting the maximum activation within each pooling window, preserving the strongest signal while discarding positional detail. It has *no learnable parameters* (ruling out B), reduces rather than increases resolution (ruling

out A), and does not normalize values (ruling out D). A side effect is a limited degree of translation invariance.

25. 1 points

Which of the following tasks are more appropriate for the type of attention mechanism proposed in the original Vaswani et al. (2017) paper, as opposed to attention with a causal decoding mask? ☐ Select All That Apply

- ☐ (a) Autocomplete
- ☐ (b) Machine translation
- ☐ (c) Sentiment analysis
- ☐ (d) Live translation
- ☐ (e) Autonomous driving

Solution: B, C

Bidirectional RNNs require sequence information from both directions - forward and backward. For A, D, and E, information from the future is not available because the tasks are done real-time.

26. 1 points

How does the original attention mechanism proposed in “Jointly Learning to Align and Translate” (Bahdanau et al., 2014) fundamentally address the vanishing gradient problem encountered in vanilla RNN architectures?

Solution: While LSTMs do not provide any theoretical guarantees on vanishing gradients, they have an additive cell state (c_t) that has a linear pathway for gradient flow during backpropagation. This ensures that compared to vanilla RNN, components from further back are able to affect the current prediction.

27. 1 points

- (a) What are the gates of the LSTM architecture and what role do they play?

Solution: Forget gate: Determines what to forget or remember from the cell state. Input gate: Determines what to add to the cell state. Output gate: Determines what to reveal from the newly updated cell state to the next layer through the hidden state.

- (b) Which gate does the GRU architecture drop?

Solution: Output gate. GRU merges both input and forget into the update gate, and adds a reset gate.

28. 2 points

Suppose we sufficiently train a transformer language model with causally masked attention on a dataset only containing several instances of the phrase “hello how are you i am good thank you” (tokenized by splitting on white space). However, while training, we forgot to use positional encodings.

- (a) What is the expected output sequence of our model if we pass in the input sequence “how hello you are”?
- (a) The output sequence will be gibberish
 - (b) The model will not output any tokens
 - (c) “i am good thank you”
 - (d) “good i am you thank”

Solution: Solution: C

We would still expect the model to output the completion “i am good thank you”.

(b) Explain your previous answer.

Solution: Because we forgot to use positional encodings, the transformer is position invariant, so passing in “hello how are you” is equivalent to passing in “how hello you are”. Because of the causal mask, the model is still able to learn NTP over the phrase “hello how are you” due to its frequency in the dataset.

29. 1.5 points

A Gaussian Mixture Model (GMM) with K clusters is used to model a dataset. To fully define this model, three distinct types of parameters must be learned from the data.

Name these three types of parameters. If you don’t remember the exact name, you can try to describe what aspect of the mixture model they control.

A. **Parameter Type:** _____

B. **Parameter Type:** _____

C. **Parameter Type:** _____

Solution: The three types of parameters learned in a GMM are:

- (a) **Mixture Weights / Coefficients (π_k):** They determine the prior probability of a data point belonging to cluster k (i.e., the relative size or importance of each cluster).
- (b) **Means (μ_k):** They specify the center or location of each individual Gaussian component in the feature space.
- (c) **Covariances / Variances (Σ_k or σ_k^2):** They define the shape, spread, and orientation of each individual Gaussian component.

30. 2 points

Recall that the reconstruction of a data point $\mathbf{x}_i \in \mathbb{R}^d$ in PCA is:

$$\hat{x}_i = \bar{x} + V_q V_q^\top (x_i - \bar{x})$$

where $V_q \in \mathbb{R}^{d \times q}$ is the matrix whose columns are the top q eigenvectors of the covariance matrix.

(a) Explain briefly what multiplying by $V_q^\top$ does geometrically to $(\mathbf{x}_i - \bar{\mathbf{x}})$.

(b) Explain briefly what multiplying the result in part (a) by V_q does (geometrically).

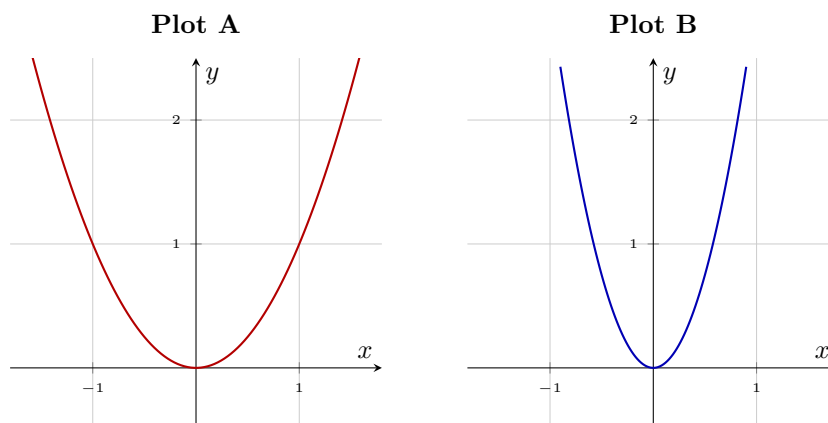
Solution: Students should say something along the lines of:

a) Multiplying by $V_q^\top$ projects the centered point $(\mathbf{x}_i - \bar{\mathbf{x}})$ from $\mathbb{R}^d$ down into $\mathbb{R}^q$. The result is a lower-dimensional representation of the point in the principal component subspace. Alternatively, they can say that this would equal z_i , which is the embedding of the point into the PCA dimensions.

b) Multiplying z_i by V_q maps the q -dimensional representation back up into $\mathbb{R}^d$, giving the closest point to $(\mathbf{x}_i - \bar{\mathbf{x}})$ that lies in the subspace spanned by the top q principal components. Adding $\bar{\mathbf{x}}$ then shifts the result back to the uncentered space.

31. 1 points

Consider the two functions plotted below. Which function has a **smaller** Lipschitz constant L over the displayed interval? One Answer



(a) Plot A

(b) Plot B

Solution: Plot A changes more slowly, so it has a smaller Lipschitz constant.

32. 1 points

Explain what it means when we say that the MLE estimate for σ^2 , for datapoints $x_1, \dots, x_n$ drawn i.i.d from $\mathcal{N}(\mu, \sigma^2)$, is “biased.”

Solution: When we say the MLE estimate $\hat{\sigma}_{\text{MLE}}^2$ is **biased**, we mean that its expected value over repeated sampling is **not equal to the true parameter** σ^2 . Formally:

$$\mathbb{E}[\hat{\sigma}_{\text{MLE}}^2] \neq \sigma^2$$

For reference, the two estimators are:

$$\hat{\sigma}_{\text{MLE}}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2 \quad (\text{biased})$$

$$\hat{\sigma}_{\text{unbiased}}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{\mu})^2 \quad (\text{unbiased})$$

33. 1 points

Suppose we are fitting a linear regression model using L1 penalty term (LASSO regression). To do this, we decide to use minibatch stochastic gradient descent (SGD) until convergence. In other words, for all k until convergence,

$$w_{k+1} = w_k - \eta \left(\frac{1}{m} \sum_{i=1}^m \nabla \ell_{I_i}(w_k) \right)$$

- (a) How does the variance of the gradient estimate change as we increase m from 1 to n , where n is the size of the dataset?

Solution: The variance decreases as you increase m

- (b) How does the bias of the gradient estimate change as we increase m from 1 to n ?

Solution: Regardless of the size of the batch, SGD is an unbiased estimator of gradient descent

END OF EXAM