New York University Tandon School of Engineering
Computer Science and Engineering

CS-GY 6923: Final Exam.
Friday, December 19th, 2025, 2:00 - 3:30pm
65 Total Points

Directions

- Write you name at the top of each page.
- Show all of your work to receive full (and partial) credit.
- If more space is required, use extra sheets of paper, marked with your name and the problem number.

Important Note: This exam may be challenging. You do not need to get everything correct to earn a good grade. Partial credit will be awarded generously for clear reasoning and intermediate steps.

Please read each question carefully and manage your time according to the point values indicated. Do your best — clarity and justification matter more than perfect algebra.

Problem 1. Warm-up (14 points)

For each statement below, circle **True** if the statement is correct, **False** if the statement is incorrect.

1. In linear regression, removing features from the data makes the model more likely to overfit.

   True False

2. To validate a model, we first set aside a subset D_{val} of the training data as a validation set. We then train on all the data, including D_{val}. The performance on D_{val} is highly predictive of test performance.

   True False

3. ℓ_1 regularization can be used to perform automatic feature selection in logistic regression.

   True False

4. When optimizing a non-convex loss function, it is guaranteed that SGD will not converge to the global minimum.

   True False

5. A 5-class logistic regression model has the same number of parameters as a linear regression model with the same input features.

   True False

6. An MLP neural network is underfitting. To increase the model's capacity, we can make it wider, deeper, or both.

   True False

7. Consider an MLP where we remove all activation functions, leaving only fully-connected (`torch.nn.Linear`) layers. This network can still perform non-linear operations as long as it has multiple layers.

   True False

8. Skip (residual) connections are only useful for convolutional neural networks, where they enable training of very deep architectures such as ResNets.

   True False

9. Convolutions are a special case of fully-connected (`torch.nn.Linear`) layers. Any convolutional neural network can be converted to a functionally identical fully-connected network (ignore pooling, layernorm, etc).

   True False

10. Vanilla recurrent neural networks must compress all information about past context into a fixed-size hidden state.

True False

11. Transformers can process all tokens in a sequence in parallel, whereas RNNs require sequential processing.

True False

12. Ensembling is a good way to reduce bias in weak models. For example, ensembling is often used for linear regression models where it introduces additional capacity and leads to higher prediction.

True False

13. One of the main advantages of decision trees is that they typically perform well on high-dimensional inputs.

True False

14. The quality of a classifier's uncertainty estimates can be evaluated by measuring accuracy on held-out data.

True False

Problem 2. Rotation angle prediction (15 points)

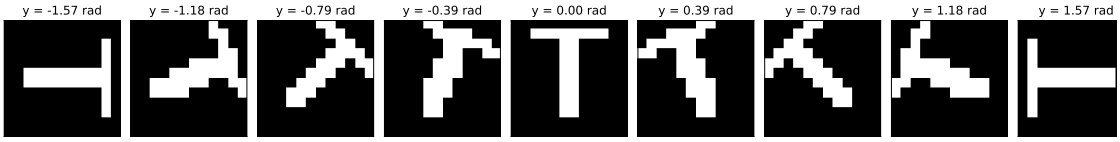


Figure 1: Example datapoints.

We are given a dataset of 12×12 binary (black and white) images, each depicting the letter "T" rotated by some angle θ. All images are generated by rotating the same base letter and rasterizing to a 12×12 grid. The labels y are the true rotation angles used to generate each image. Figure 1 shows representative examples.

Due to the finite resolution of the rasterization process, a range of continuous angles (width ≈ 0.1 rad) maps to the exact same binary image. In other words, rotation angles differing by less than approximately 0.1 radians typically produce visually indistinguishable images. All rotation angles in the dataset lie within the interval $[-\pi/2, \pi/2]$.

(a) [4 points] We wish to design a model that predicts rotation angles strictly within $(-\pi/2, \pi/2)$. Propose a suitable model architecture (linear or neural network). Write down an appropriate loss function and describe a method for training the model.

(b) [4 points] Suppose we train this model on our dataset. On the first 5 training examples, the model predicts:

$$\hat{y} = (-0.4771,\ -1.2654,\ 1.1327,\ 1.4298,\ -1.5224)$$

while the true rotation angles are:

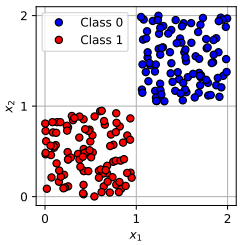
$$y = (-0.4771,\ -1.2654,\ 1.1327,\ 1.4298,\ -1.5224)$$

(values agree to at least 4 decimal places). What can we conclude about this model? Based on this information alone, what can we conclude about the model's performance on a held-out test set?

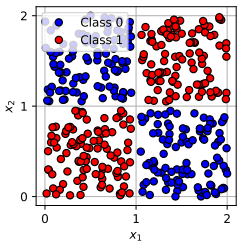
(c) ~~[3 points] Now suppose we generate a new training set and retrain the model. Our new training set consists of images with rotation angles sampled on a uniform grid over $[-\pi/2, \pi/2]$ with step size 0.01 radians. The test set consists of images with rotation angles sampled uniformly at random from the same interval. If our model achieves very low loss on the training set, should we be concerned about overfitting? Justify your answer.~~

(d) [4 points] Instead of treating this as a regression problem, we now wish to frame it as a classification problem using exactly 10 classes. Propose a model architecture and loss function suitable for this formulation (be explicit, add detail). Explain how you would define the classes.

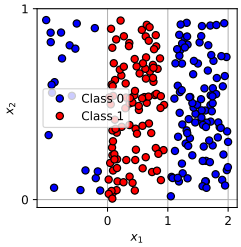
Problem 3. Tress on Chessboards (16 Points Total)



Dataset 1 for parts (a, b)



Dataset 2 for part (c)



Dataset 3 for part (d)

Figure 2: Two-class classification dataset. Blue circles represent Class 0; red circles represent Class 1.

Consider the dataset shown in Figure 2, consisting of two classes in $\mathbb{R}^2$. Throughout this problem, assume that when constructing decision trees:

- We use axis-aligned splits of the form $x_i \leq t$ for some feature $i \in \{1, 2\}$ and threshold $t \in \mathbb{R}$.
- When multiple (feature, threshold) pairs yield the same optimal value of the splitting criterion, we select uniformly at random among them.
- The depth of a tree is defined as the maximum number of splits along any path from the root to a leaf (so a tree with just a root node has depth 0).

(a) [4 points] What is the minimum depth of a decision tree that achieves perfect (100%) training accuracy on this dataset? For a tree of this minimum depth, sketch the decision regions on the x_1-x_2 plane, labeling each region with the predicted probability $P(\text{Class} = 1 \mid x)$. If there are multiple trees that achieve perfect accuracy, describe all of them.

(b) [4 points] Now consider a random forest consisting of a large number of trees (e.g. 10^3), trained on the same dataset with the following specifications:

- No bootstrap sampling is performed; each tree is trained on the full dataset.
- Each tree is grown to the minimum depth needed for perfect training accuracy.
- The forest outputs the average predicted probability across all trees.

Sketch the predicted probability $P(\text{Class} = 1 \mid x)$ as a function of position in the x_1-x_2 plane. Clearly indicate the probability in each distinct region.

In what region(s) does the random forest exhibit high uncertainty (i.e., $P(\text{Class} = 1 \mid x) \approx 0.5$)? Explain why this occurs and contrast this behavior with that of a single decision tree.

(c) [4 points] What is the minimum depth of a decision tree that achieves perfect training accuracy on Dataset 2 in Figure 2?

Consider the standard greedy algorithm for constructing decision trees discussed in class, which selects the split at each node that maximizes the reduction in impurity (e.g., Gini impurity). If we allow this algorithm to grow a tree up to the depth you identified above, will it achieve perfect training accuracy? Why or why not?

(d) [4 points] Suppose we want to train an ensemble of depth-1 decision trees (i.e., decision stumps) on Dataset 3 in Figure 2. Which ensemble method is more likely to achieve good accuracy on this dataset: *bagging* or *boosting*? Explain your reasoning.

Problem 4. The Mixer (20 points)

Consider a single-layer transformer for language modeling, consisting of a self-attention block followed by a feed-forward (MLP) block. Let the input be a matrix $X \in \mathbb{R}^{T \times D}$, where T is the sequence length and D is the model dimension (number of channels).

(a) [4 points] A transformer can be viewed as operating along two axes:

- The *channel axis* (dimension D): each position has a D-dimensional representation.
- The *sequence axis* (dimension T): the sequence consists of T token positions.

For each of self-attention and the MLP block, identify which axis it *mixes* information along. That is, which operation allows different token positions to communicate with each other, and which operation allows different channels within a single position to interact?

(b) [16 points] Self-attention can be complex to analyze. Suppose we want to replace it with a simpler MLP-based operation while still allowing both tokens and channels to communicate.

Propose an architecture that uses MLPs to mix information along *both* axes. Your design should be *symmetric*: the operations on the channel axis and the sequence axis should both be MLPs, differing only in which dimension they operate across.

(i) [6 points] Write pseudocode for a single layer of your architecture. Clearly specify the input and output shapes of each operation. Assume the input is $X \in \mathbb{R}^{T \times D}$. Ignore the layernorms, but include skip connections.

Constraints & Assumptions:

- For the MLP mixing along the sequence axis, assume the hidden layer size is also T (i.e., it maps $\mathbb{R}^T \to \mathbb{R}^T \to \mathbb{R}^T$).
- Assume **weight sharing**: the MLP mixing along the sequence axis uses the *same* weights for every channel, and the MLP mixing along the channel axis uses the *same* weights for every token position.

(ii) [6 points] In language modeling, we require *causal masking*: when predicting the token at position t, the model may only use information from positions $1, \ldots, t$. Given the constraints in (i), what condition must you impose on the weight matrices of the sequence-mixing MLP to enforce causal masking?

(iii) [4 points] Discuss at least one other limitation of your proposed architecture compared to standard self-attention.