

Midterm Exam II

Exam Data: December 9, 2024

Name: _____

Penn ID number: _____

Section number (4190 or 5190): _____

Instructions

- Write your name, Penn ID number, and section number on this page.
- Write your answer directly on the exam paper. Consider using a pencil and eraser, which will allow you to easily adjust your answers if needed.
- No digital devices are allowed during the exam (e.g., laptops, tablets, phones, watches). Please ensure that all your devices are turned off or set to silent mode.
- This is 75-minute exam, containing 15 questions with 50 points total.
- Each point should take approximately 1.5 minutes; if you find yourself spending too much time on one problem, move on and come back to it.
- At the end of 75 minutes, you will put down your pens and submit your exam.

Good luck!

1. (2 pt) Suppose we have a neural network that has high training accuracy but low validation accuracy. Which of the following can you use to improve the validation accuracy? Select all that apply. _____.
- A. Decrease learning rate.
 - B. Add dropout.
 - C. Increase the number of hidden dimensions of layers.
 - D. Pretrain the neural network on a related task.
2. (2 pt) Which of the following advantages do transformers have over recurrent neural networks (RNNs) for language processing? Select all that apply. _____.
- A. Transformers can better handle long-range dependencies.
 - B. Transformers simplify model training by requiring fewer parameters than RNNs.
 - C. Transformers requires smaller datasets to train than RNNs.
 - D. Transformers enable sequential processing of fixed-length inputs.
3. (2 pt) A machine learning algorithm is said to be *scale invariant* if its performance or results do not change when the scale of any of the input features is altered (the scale is, of course, applied identically to both the training and test data). For instance, doubling the values of a feature should not affect the algorithm's results. Which of the following algorithms is scale invariant? Select all that apply. _____.
- A. ℓ_2 -regularized linear regression.
 - B. Decision Tree.
 - C. Random Forest.
 - D. KNN with Euclidean distance.
 - E. K-means clustering.
4. (2 pt) Bob is training a decision tree on a dataset, but both the training and testing loss are high. Bob comes to you for advice on how to improve his model. Which of the following suggestions might help him? Select all that apply. _____.
- A. Train an ensemble of decision trees using bagging.
 - B. Use boosting to iteratively reduce errors of previous models.
 - C. Prune the decision tree on a hold-out validation set.
 - D. Collect more training data to improve the model's performance.
 - E. Increase the maximum depth of the decision tree.

5. (2pt) Consider two MDPs $M=(S,A,P,R, \gamma)$ and $M' = (S, A, P, R', \gamma)$. M' is identical to M except the reward function, where $R'(s, a, s') = wR(s, a, s') + b$ for all s, a, s' . How is the value function $V'^\pi(s)$ for some arbitrary policy in M' related to the value function $V^\pi(s)$ in M ? Write a concise mathematical expression if possible.

6. (2 pt) After factorizing the user-item utility matrix at Netflix, we get user features $p_{Jack} = [2, 3]$ and $p_{Eve} = [1, 4]$, and movie features $q_{starwars} = [1, 4]$ and $q_{godfather} = [2, 3]$. Who among Jack and Eve is likely to have higher utility for *The Godfather*? Show your work in the space provided.

7. (3 pt) You are designing a CNN with the following block, repeated over and over: 3x5 conv (stride 1, zero pad 1) \rightarrow Batchnorm \rightarrow 3x3 maxpool (stride 1) \rightarrow ReLU

A. (1 pt) After processing through one such block, what is the largest size of the image region that can affect the output feature map activations at any particular pixel location?

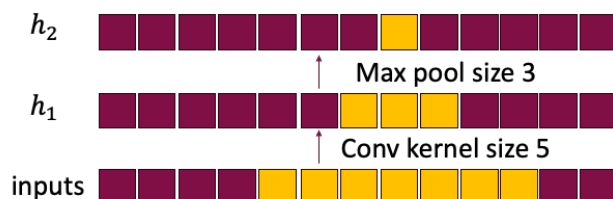


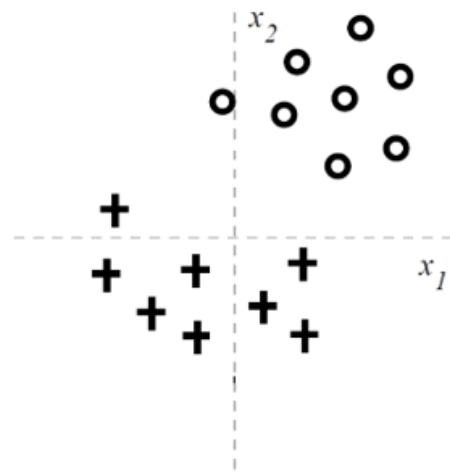
Figure 1: The yellow pixel in the output is influenced by the yellow regions indicated in each layer.

B. (2 pt) A particular image pattern requires observing a spatial extent of 8x6 pixels in the input image in order to recognize it. After how many blocks can one of the feature maps within the CNN correctly identify and locate this pattern? Show your work in 1-2 lines.

8. (4 pt) Consider a logistic regression classifier $P(y = 1|x, \beta) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$, trained to maximize the following objective:

$$\sum_{i=1}^n \log P(y_i|x; \beta) - \lambda_1 \beta_1^2 - \lambda_2 \beta_2^2$$

For the training dataset in the figure, what can you say about the decision boundary (you can sketch it approximately if you



like) and training loss if:

- A. (2 pt) λ_2 becomes very large (approaching ∞) and λ_1 is small and positive?
 - B. (2 pt) λ_1 becomes very large (approaching ∞) and λ_2 is small and positive?
9. (5 pt) You are considering two alternative designs of a neural network to map from 100x100x500 (height x width x channels) inputs x to 100x100x500 outputs y .
- A. (2 pt) Your first option is a single 3x3 convolution layer that maps directly from x to y : $x \rightarrow 3 \times 3 \text{ conv} \rightarrow y$. How many learnable parameters are in this layer? (you may skip biases to approximate)
 - B. (3 pt) Your second option is the following block: $x \rightarrow 1 \times 1 \text{ conv} \rightarrow h_1 \rightarrow 3 \times 3 \text{ conv} \rightarrow h_2 \rightarrow 1 \times 1 \text{ conv} \rightarrow y$. h_1 and h_2 each have exactly 10 channels. How many learnable parameters are in this block? (once again, you may skip biases to approximate)
10. (5 pt) In this problem, we consider a self-attention layer using dot-product attention. Suppose the input to this layer is three word embeddings. After applying the query (q), key (k), and value (v) projections, we obtain the following vectors for the input embeddings:

$$q_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad k_1 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad v_1 = \begin{bmatrix} 2 & 1 \end{bmatrix}$$

$$q_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad k_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 & 3 \end{bmatrix}$$

$$q_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad k_3 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

- A. (1 pts) Compute the attention score matrix S , where each element $s_{i,j}$ is the dot-product attention score between q_i and k_j .
- B. (2 pts) Compute the attention weights based on the attention score matrix S . To simplify computation, assume this self-attention layer uses a modified softmax function with base 2, i.e.,

$$\text{softmax}(\mathbf{z})_i = \frac{2^{z_i}}{\sum_j 2^{z_j}}.$$

- C. (2 pts) Using the attention weights and the value vectors, compute the output representation y_i for each word embedding.

11. (5 pt) Gradient boosting builds an ensemble of base models iteratively. At each step, the pseudo-residuals (negative gradients of the loss with respect to current predictions) are

computed to train the next base model. Suppose the input to a gradient boosting model is a dataset $\{(x_i, y_i)\}_{i=1}^n$, where x_i is the input feature, y_i is the target label, and $F_t(x_i)$ is the current model's prediction. The pseudo-residuals z_i are used to construct the dataset $\{(x_i, z_i)\}_{i=1}^n$ for training the next base model. Write the expressions for z_i for the following loss functions.

- A. (2 pt) Gradient boosting with the Mean Squared Error (MSE) loss:

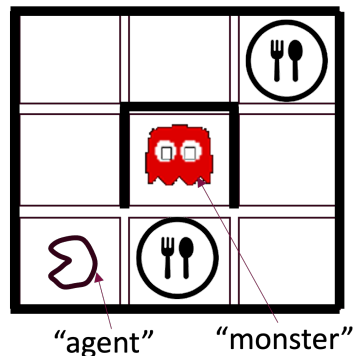
$$\mathcal{L}_{\text{MSE}} = \frac{1}{2} \sum_{i=1}^n (F_t(x_i) - y_i)^2.$$

- B. (3 pt) Gradient boosting with the Negative Log-Likelihood (NLL) loss:

$$\mathcal{L}_{\text{NLL}} = - \sum_{i=1}^n [y_i \log \sigma(F_t(x_i)) + (1 - y_i) \log(1 - \sigma(F_t(x_i)))],$$

where $\sigma(z)$ is the sigmoid function, with $\sigma'(z) = \sigma(z) \cdot (1 - \sigma(z))$.

12. (5 pt) Imagine a simplified version of Pacman, played over a 3x3 grid, with the agent, one ghost, and food particles scattered over states. The agent's objective is to consume as much food as possible while avoiding the monster – once the agent reaches a cell with food, the food is consumed and vanishes. The agent can only move horizontally or vertically one cell at a time. The monster too has the same movement capabilities as the agent, and moves uniformly at random at each step. Food spawns randomly in any state at any step with a small probability. The thick lines in the grid denote fixed walls that cannot be crossed by the agent or the monster. The game ends if the monster and the agent are in the same cell.

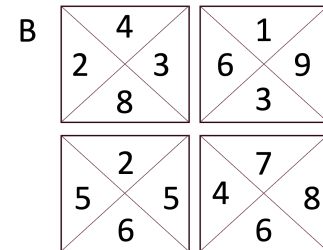
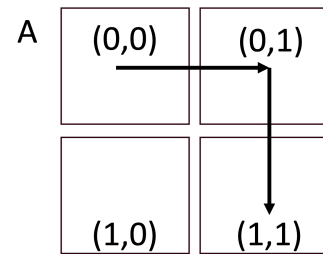


- A. (2 pt) Describe the state space. Can you count the total number of states? Show your work, okay to approximate.
- B. (1 pt) Describe the agent's action space. Can you count the total number of actions?
- C. (1 pt) Describe the transition function. Is it deterministic or stochastic?
- D. (1 pt) For the reward function, is it sufficient to reward the agent positively each time it gets food? Do we need any other reward terms for the reward-optimal agent to behave correctly?

13. (5 pt) An agent is dropped into an unknown 2x2 grid world. The agent can select four actions at each step: move up, down, left, or right. The agent makes the following two moves, depicted in Figure A:

- it starts at $(0,0)$, chooses “right” at the first step, moves to $(0,1)$ and gets a reward -1.
- it then chooses “down” at the next step, moves to $(1,1)$, and gets a reward +10.

As it follows this trajectory, the agent performs a Q-Learning update once after step 1, and then a second time after step 2. Its initial Q-values before any learning (initialized at random) are shown in Figure B. Compute the updated Q values after each step. Show your work. Assume that the discount factor is 0.9, and the learning rate is 0.1.



14. (6 pt) A diagonal co-variance d-dimensional gaussian is defined by the probability density function:

$$P(x) = \frac{1}{(\sigma\sqrt{2\pi})^d} \exp\left(-\sum_{j=1,\dots,d} \frac{(x_j - \mu_j)^2}{\sigma_j^2}\right)$$

- A. (2pt) What is the log-likelihood of a sample x under $P(x)$? It is okay to leave hard-to-simplify logarithms in your answer.
- B. (1 pt) How does this log-likelihood expression change if $\sigma_j = 1$ for all j , corresponding to a unit-spherical gaussian?
- C. (1 pt) Now consider a mixture of K unit-spherical gaussians. What is the log-likelihood of x under this mixture?
- D. (2 pt) Recall that k-means clustering optimizes the sum of squared distances loss function: $\min_S \sum_{k=1}^K \sum_{x \in S_k} \|x - \mu_k\|_2^2$, where each S_k corresponds to a cluster. Can this be interpreted as maximizing the log-likelihood from part C? Why or why not?

Extra space

Extra space

Extra space