Task 1

In the context of the *k*-arm bandit problem, what is the primary challenge faced by the decision-making algorithm?

1. Determining the optimal sequence of actions before the first time-frame begins.
2. Balancing exploration of unknown actions with exploitation of known high-reward actions.
3. Predicting how the reward for each action will change over time.
4. Minimizing the total number of actions taken within the time period *T*.

[Total 3 marks]

?

Task 2

Which of the following statements is **TRUE**?

1. The kernel or mask of a Convolutional layer is the distance between adjacent receptive fields in a specific direction (horizontal or vertical).
2. Average pooling is a Convolution operation where all neurons have trainable weights.
3. A Convolutional layer can have more than one feature maps.
4. Two neurons *A*, *B* within a Convolutional feature map have different weights $\boldsymbol{w}_A \neq \boldsymbol{w}_B$.

[Total 3 marks]

?

Task 3

During the development of a Large Language Model (LLM), a team implements "Differential Privacy" (DP) as an ethical design constraint to prevent the leakage of Personally Identifiable Information (PII). This process involves a privacy budget parameter ε.

Which of the following best describes the resulting impact of this constraint on the system's performance?

1. The model's utility remains constant, but the training time increases exponentially to handle the privacy layers.
2. There is an inherent trade-off where increasing the privacy guarantee (decreasing ε) necessarily degrades the model's accuracy and convergence rate.
3. The constraint acts as a regularization term, improving generalization by preventing the model from over-fitting on specific data points.
4. Ethical constraints like Differential Privacy are purely architectural and do not affect the loss function or the final weights of the model.

[Total 3 marks]

?

Task 4

Let X be a finite set of some finite size k. Let A be a learner that on a sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \{0, 1\})^m$ outputs a function $A(S)$ such that:

> If $x = x_i$ for some $i \in \{1, \ldots, m\}$ then $A(S)(x) = y_i$,
>
> If $x \notin \{x_1, \ldots, x_m\}$, then $A(S)(x)$ is picked randomly with probability $1/2$ for each label in $\{0, 1\}$.

Given a function $f : X \to \{0, 1\}$, define a probability distribution (U, f) over $X \times \{0, 1\}$ by assigning probability $1/k$ to any pair $(x, f(x))$ and probability 0 to any $(x, y) \notin \{(x, f(x)) : x \in X\}$.

Show that for every $f : X \to \{0, 1\}$ and $S = ((x_1, f(x_1)), \ldots, (x_m, f(x_m)))$,

$$L_{(U,f)}(A(S)) - L_S(A(S)) \geq \frac{(k - m)}{2k}.$$

[Total 7 marks]

?

Task 5

In sequence models like the Transformer, the core self-attention mechanism processes all tokens simultaneously and natively lacks any concept of sequence order. To address this, a positional encoding vector is added to each token's embedding to inject positional information.

For a token at an integer sequence position pos, its positional encoding is a vector of dimension d. The values of this vector are constructed using sine and cosine functions. Specifically, for a given dimension index i (where $0 \leq i < d/2$), the encoding values for the $2i$-th and $(2i+1)$-th dimensions are given by:

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10000^{\frac{2i}{d}}}\right)$$
$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10000^{\frac{2i}{d}}}\right)$$

Which one of the following mathematical properties holds true for this specific encoding formulation? Select one correct option and justify your answer.

1. Orthogonality: The encodings for any two different positions are statistically orthogonal, ensuring that position information does not interfere with the semantic meaning of the word embeddings.
2. Linear Translation: For any fixed offset k, the encoding at position $\mathrm{pos}+k$ can be represented as a linear function (rotation) of the encoding at position pos.
3. Decay: The magnitude of the encoding vector decays exponentially as the position index increases, naturally biasing the model towards recent context.
4. Symmetry: The function is perfectly symmetric around the midpoint of the sequence, allowing the model to process bidirectional context with shared weights.

[Total 7 marks]

?

Task 6

Let us consider a multi-label classification problem. For each class c, TP_c represents the instances that are in class c correctly identified as belonging to c; TN_c represents the instances that are not in class c correctly identified as not belonging to c; FP_c represents the instances that are in not class c incorrectly labeled as belonging to c; FN_c represents the instances that are in class c incorrectly labeled as not belonging to c.

We have the following metrics:

- per-class **recall**:

$$\mathrm{Recall}_c = \frac{TP_c}{TP_c + FN_c} = P(\mathrm{pred} = c \mid \mathrm{true} = c);$$

- per-class **precision**:

$$\mathrm{Precision}_c = \frac{TP_c}{TP_c + FP_c} = P(\mathrm{true} = c \mid \mathrm{pred} = c);$$

- per-class **specificity**:

$$\mathrm{Spec}_c = \frac{TN_c}{TN_c + FP_c} = P(\mathrm{pred} \neq c \mid \mathrm{true} \neq c);$$

- for each class-dependent error metric M_c, the **macro average error metric** is the arithmetic mean of the class specific scores

$$\mathrm{Macro} - M = \frac{\sum_c M_c}{\text{number of classes}};$$

- for a distribution $\pi = (\pi_c)$ of the labels, the **accuracy**:

$$\mathrm{acc}(\pi) = \sum_c \pi_c \mathrm{Recall}_c;$$

- if $C = (C_{ij})$ is a cost matrix where $C_{ij} = \mathrm{Cost}(\mathrm{pred} = j \mid \mathrm{true} = i)$, the **expected cost** is

$$\mathbb{E}_\pi(C) = \sum_{i,j} \pi_i P(\mathrm{pred} = j \mid \mathrm{true} = i) C_{ij}.$$

You can assume that every class has nonzero support and the classifier makes both correct and incorrect predictions for every class (non-degeneracy).

1. Let us assume that you are in a binary classifier, with classes 0 and 1. Prove that, if

$$\mathrm{accuracy} = \mathrm{precision}_1 = \mathrm{recall}_1,$$

?

the support of the two classes must be the same.

[*5 marks*]

For the remaining parts, you can assume the following context.

Context

A public-health triage system classifies incoming messages into one of three classes:

$$U = \text{Urgent}, \quad S = \text{Semi-urgent}, \quad N = \text{Non-urgent}.$$

The training dataset was intentionally enriched with urgent and semi-urgent cases.
On the *training set*, the classifier produced the following confusion matrix
(rows correspond to the true label, columns to the predicted label):

True \ Pred	U	S	N	Total
U	200	30	20	250
S	30	270	50	350
N	10	30	360	400

The corresponding training label distribution is

$$\pi_{\text{train}} = \big(P(U) = 0.25,\ P(S) = 0.35,\ P(N) = 0.40\big).$$

At deployment time, the real-world label distribution is different:

$$\pi_{\text{dep}} = \big(P(U) = 0.05,\ P(S) = 0.15,\ P(N) = 0.80\big).$$

You may assume that the classifier's conditional behavior is **stable**:

$$P_{\text{train}}(\text{pred} = j \mid \text{true} = i) = P_{\text{dep}}(\text{pred} = j \mid \text{true} = i) \quad \text{for all } i, j \in \{U, S, N\}.$$

We are given the following cost matrix
(cost of predicting the column label when the true label is the row label):

$$C(\text{pred} = j \mid \text{true} = i) = \begin{array}{c|ccc} & U & S & N \\ \hline U & 0 & 20 & 100 \\ S & 5 & 0 & 10 \\ N & 1 & 1 & 0 \end{array}$$

2. Compute Precision_U under both π_{train} and π_{dep}.
   Explain why precision changes under distribution shift while recall does not.

[*5 marks*]

3. Consider two strategies for choosing the deployed decision rule:
   1. maximizing training accuracy;
   2. minimizing expected deployment cost.

   Explain why strategy (1) can lead to a higher expected cost at deployment than strategy (2).

[*3 marks*]

[Total 13 marks]

Task 7

1. Express the distance of the image of a point $\mathbf{x}$ from the centre of mass of a set of examples

$$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$$

in a feature space defined by kernel $\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ in terms of kernel evaluations.

[*2 Marks*]

2. Using the result from item 1. give pseudocode for the novelty detection algorithm that labels a test point novel if it lies outside the smallest sphere centred at the centre of mass that contains all of the training data.

[*3 Marks*]

3. Show that if we remove one point $\mathbf{x}_j$ from S, the centre of mass moves away from $\phi(\mathbf{x}_j)$ on the line from $\phi(\mathbf{x}_j)$ through the centre of mass by $\frac{1}{m-1}$ times the distance between them.

[*3 Marks*]

4. Hence or otherwise show that the novelty detection method using the same centre as in item 2. above but increasing the radius of the sphere by $\frac{m}{m-2}$ ensures that the leave one out error (an error occurs if the left out point is not inside the sphere) on the training set is at most 1.

[*5 Marks*]

[Total 13 marks]

?

Task 8

Context:
In Statistical Machine Translation, a common sub-problem is *Word Ordering*. You are given a ``bag of words'' (a set of scrambled words) and must reconstruct the original sentence order to maximize the probability of the sequence.

Let $W = \{w_1, w_2, \ldots, w_N\}$ be a set of N unique words.
You are given a **Bigram Language Model**, $P(w_j|w_i)$, which provides the probability of word w_j following word w_i.
There are two special tokens: $\langle \mathrm{S} \rangle$ (Start) and $\langle \mathrm{E} \rangle$ (End). The sentence must begin with $\langle \mathrm{S} \rangle$ and end with $\langle \mathrm{E} \rangle$.

Objective:
Find a permutation π of the words in W that maximizes the sentence probability:

$$P(\pi) = P(w_{\pi_1}|\langle \mathrm{S} \rangle) \times \left(\prod_{i=1}^{N-1} P(w_{\pi_{i+1}}|w_{\pi_i}) \right) \times P(\langle \mathrm{E} \rangle|w_{\pi_N})$$

To apply the A* algorithm, we transform this maximization problem into a minimization problem by defining the cost of a transition between word u and word v as the negative log-probability:

$$C(u, v) = -\ln(P(v|u))$$

Part A: State Space Formalization
Formalize this problem as a State Space Graph search suitable for the A* algorithm.
Specifically, define

1. The structure of a **State** n.
2. The **Start State** and **Goal State**.
3. The **Successor Function** (how transitions are generated) and the accumulated cost $g(n)$.

[*3 Marks*]

Part B: Heuristic Design
The core difficulty lies in the factorial structure of the search space ($N!$), making a trivial heuristic (e.g., $h = 0$) computationally intractable for even modest input sizes (e.g., $N > 15$).
Design a non-trivial heuristic $h(n)$ that estimates the remaining cost to complete the sentence.
Your heuristic must meet the following criteria:

1. It must be **admissible**.
2. It must be **monotonic** (consistent).

Define your heuristic mathematically. You do not need to prove the properties yet, but you must explain the logic behind your design.

?

Hint: Consider the set of words that have not yet been placed. In a valid completed sentence, every one of these words must have exactly one incoming edge.

[*5 Marks*]

Part C: Theoretical Proofs

Using the heuristic $h(n)$ you defined in Part B, provide formal proofs for the following properties:

1. **Admissibility**: Prove that $h(n) \leq h^*(n)$, where $h^*(n)$ is the true minimum cost to the goal.
2. **Monotonicity**: Prove that $h(n) \leq cost(n, n') + h(n')$, where n' is a successor of n.

[*5 Marks*]

[Total 13 Marks]