Problems

Please select **ONE problem** arbitrarily to answer. We will evaluate your problem solving skills according to your answer.

Problem 1: Second-order sufficient optimality conditions

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable at $\mathbf{x}$. Show that $\mathbf{x}$ is a strict local minimum if $\nabla f(\mathbf{x}) = 0$ and the Hessian matrix $\mathbf{H}(\mathbf{x})$ is positive definite.

Problem 2: Low-rank approximation

Please solve the problem as follows

$$\min_{X \in \mathbb{R}^{m \times n}} \{\|A - X\|_F : \mathbf{rank}(X) \leq K\}.$$

Problem 3: Random walk on $\mathbb{Z}$

Consider the random walk $X = \{X_n\}_{n \geq 0}$ on $\mathbb{Z}$ that starts at $X_0 = 0$. The particle moves with probability p one unit to the right and with probability $q = 1 - p$ one unit to the left at each transition. Prove that the state 0 is recurrent (i.e., $\mathbb{P}\{X_n = 0 \text{ i.o.} | X_0 = 0\} = 1$) if and only if $p = q = 1/2$.

Problem 4: Computing the Number of Groups in a Knowledge Graph

Knowledge graphs (KGs) are usually collections of factual triples—(head entity, relation, tail entity), representing human knowledge in a structured way. In a knowledge graph, we use nodes to denote entities and edges to denote relations, respectively.
Consider a knowledge graph with n entities and 1 relation called $DirectlyConnectedWith$. Suppose some of them are connected, while some are not. If entity e_1 is connected directly with entity e_2 and entity e_2 is connected directly with entity e_3, then we think that entity e_1 is connected indirectly with entity e_3. A set of entities in this knowledge graph is called a group, if any two entities in this set are connected directly or indirectly and no other entities outside this set are connected with the entities in this set.
Given an $n \times n$ adjacency matrix A, where $A_{ij} = 1$ if there exists a triplet (the i-th entity, DirectlyConnectedWith, the j-th entity) and $A_{ij} = 0$ otherwise. Please compute the total number of groups in this KG. Notice:

- to simplify the problem, we assume this knowledge graph is undirected, i.e., A is a symmetric matrix;
- given an input $A = [[1, 1, 0], [1, 1, 0], [0, 0, 1]]$, the correct output is 2;
- you only need to submit the code as an attachment;
- the python template is given as follows.

```
def compute_groups(A: List[List[int]]) -> int:
    # Fill in your code here.
```

Problem 5: Recognizing the Handwritten Digit

You are expected to build a binary classifier to predict whether the handwritten digit of a given image is 0 or 1. Notice:

- you can download the dataset from MNIST;
- you only need to use the data with label 0 or 1;
- please unzip the data files and put the code file in the same directory;
- you can use any model and libraries you want;
- the accuracy on the test set is supposed to reach at least 90%;
- you only need to submit the code as an attachment.

Problem 6: Solve the Pommerman Game by Reinforcement Learning

Consider an interesting Pommerman game described here. In the default free-for-all (FFA) mode, all agents play against each other. The game ends when only one agent remains, and the last survivor agent will be the winner. Our goal is to control an agent to win the game. We use the reinforcement learning (RL) algorithm to train this agent.

1. A Markov decision process (MDP) is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$. How to define the reward function is a core issue in most RL tasks. Please define the reward function according to your understanding of this task and explain the reasons.
2. Now we need to choose an algorithm to solve this task. Please write down which RL algorithm you would like to choose and explain why you choose it.
3. We find that the agent tends to learn policies that do not lay any bombs during the early training process. Intuitively, what may be the potential reason for this phenomenon and what can we do to avoid that?
4. Compared with the 1v1 Go game, what are the additional challenges in the Pommerman Game?