Final Exam Review Assignment (5% Weight)

Question No 1: Reinforcement Learning for Solving the Game of NIM

NIM is a two-player, turn-based, deterministic game played as follows:

- There is **a single pile of stones**.
- At the start of the game, the pile contains **N stones**.
- Players take turns.
- On each turn, a player must **remove either 1, 2, or 3 stones** from the pile.
- The player who **removes the last stone wins**.
- The game ends immediately when no stones remain.

Although a mathematical solution to this puzzle is well known but to test our understanding of RL a teacher requires us to design a **Reinforcement Learning (RL) agent** in order to learn an optimal strategy for this game using a self-play

(a) Define the **state space**, **action space**, and **reward function** suitable for an RL formulation of this game.

(b) Explain the reward update equation used by Q**-learning** that can be used to learn the optimal policy for NIM. Clearly write the **Q-learning update equation** and explain the meaning of each parameter

(c) Give the updated Q-table entry for a game starting with **N = 5 stones**, assume that the following each of the following episodes/sequence of moves are used by the gents:

Episode 1: {2, 1, 2}, Episode 2: {2, 1, 1, 1}, Episode 3: {1, 1, 3}.

Take

- Learning rate ($\alpha = 0.5$)
- Discount factor ($\gamma = 0.9$)
- All Q-values initially set to 0

For the update.

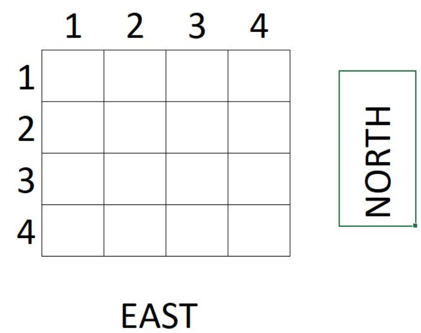
Show your calculation for each episode

(d) Why do applying Q-Table learning to the game of connect-4 might not be a good idea? Explain in terms of state-space and number of episodes required for q-table learning. Also suggest a solution for using RL in such a case

Question No 2: Histogram filtering

Prediction in a 4 x 4 grid with known initial state

You have a mobile robot in a 2D grid world of size 4 × 4 with coordinates ((x, y)), where (x, y take one of the values {1, 2, 3, 4}). The robot's orientation is one of {North, East, South, West}. At time (t = 0), the robot's state is known exactly: position ((2, 2)) and orientation East.



The robot receives an action command at (t = 1): "move forward by one cell."

- **Motion model:**
  - **Forward success:** With probability 0.7, the robot moves one cell forward along its current orientation.
  - **Overshoot:** With probability 0.1, the robot moves two cells forward.
  - **Veer:** With probability 0.1, the robot moves one cell perpendicular to the left of its orientation (i.e., if facing East, it moves North by one cell).
  - **Slip/stay:** With probability 0.1, the robot stays in its original cell due to slippage.
  - **Orientation update:** Orientation remains unchanged after the action in all cases.
  - **Collision handling:** Any movement that would take the robot outside the grid results in the robot staying in the cell adjacent to the boundary.
- **Grid boundaries and orientation-relative moves:**
  - Facing East at ((x, y)): forward is ((x+1, y)), overshoot tries ((x+2, y)), left-perpendicular is ((x, y + 1)).
  - The grid has no obstacles; only boundary limits apply.

Compute the updated belief i.e. **bel_1(x , y), θ))** after the single move command, using the motion model above. Provide the probability for each reachable state and ensure the distribution is normalized.

Question No 3: Particle filtering

Repeat question 2 using particle filtering using three particles and assuming that the original position of the robot is totally unknown but it is oriented/facing towards North. Use the same motion model as in question 2

Question No 4: Neural Networks

a) A neuron has inputs ([2, -1]), weights ([0.5, 1.0]), and bias (0.2).
Compute the weighted sum (z) and the final output using sigmoid activation. Also compute the gradient of the loss function for this network and update the weights and bias using learning rate of 0.1

b) Repeat the above question if L^2 regularization is used with penalty of 0.1.

c) Assume that the True labels are ([1, 0, 0]) and the predictions are ([0.8, 0.3, 0.4])
Compare the values of Mean Squared Error and Cross Entropy Loss.

d) A fully connected layer has 8 inputs 5 neurons in a single hidden layer and 3 output neurons. How many weights are there?

e) A 5 x 5 conv-filter is applied with stride 1 and no padding to an input image of 32 x 32 size. Compute output feature map size for this filter.

f) An input image has 3 channels and 10 filters of size (3 x 3) are used to compute the output volume. Compute total number of weights in this layer.

g) Why does weight sharing reduce the number of parameters?
Answer numerically using a (3 x 3) filter vs a fully connected layer for a 28 x 28 input.

h) For an RNN hidden state size is 10, input size is 5 and output size is 3. How many weights are in the input-to-hidden, hidden-to-hidden, and hidden to output matrices?

i) Why does vanishing gradient occur more frequently in RNN than in feed-forward networks.

Question No 5: Linear Classifiers, Decision Trees and KNN

a) In 2D, plot the decision boundary for a linear classifier f(x) = sign (w . x + b) if w = [2, -1], and the bias b = -1. On your plot, show a point in 2D that will be classified as +ve by this classifier and a point that will be classified as -ve

b) Compute the value of functional margin of a linear classifier for data point x = [2, 1]) with true label y = +1 if the classifier is represented using weights w = [1, -1] and bias b = 0.

c) A logistic regression model is a linear model that computes P(y=1|x) = σ(w . x + b) Compute the model output given: w = [1, 1], b = -2, and x = [2,1]. Also compute the value of loss if binary cross-entropy is used as the loss function and actual output value is 1

d) Differentiate between **classification** and **regression** problems. Give **one example** of each

e) Why would you prefer short-decision trees over larger ones? Name a principle that prefers shorter trees over larger trees.

f) You are given the following dataset for predicting whether a student **Passes** an exam.

Study Hours	**Attendance**	**Pass**
High	High	Yes
High	Low	Yes
Low	High	No
Low	Low	No

Which feature would you choose as the **root node**? Justify your answer briefly. Also draw the resulting **decision tree**.

g) A KNN classifier is used with **k = 3** to classify a new data point. The distance of new data point from all the points are given below

Data Point	**Distance**	**Class**
A	1.2	Red
B	0.8	Blue
C	1.0	Blue
D	2.5	Red

What class will be assigned to the new point?