ECE/SIOC 228 Machine Learning for Physical Applications: Assignment 2 (Spring 2022)

Instructor: Yuanyuan Shi
Teaching Assistants:
Srinivas Rao Daru, sdaru@ucsd.edu
Tawaana Homavazir, thomavaz@ucsd.edu
Rohin Garg, rgarg@ucsd.edu
Rishabh Jangir, rjangir@ucsd.edu

*** Deadline is Monday May 16th 11:59PM**

Submission format:

- For the Written Questions you can upload a hand-written or Latex format(preferable but not mandatory) solutions on the Gradescope.(If you are submitting a hand-written solutions please make sure your writing is clear)

PART I: Written Questions (50 Points)

1. Question 1: Physics Informed Neural Networks (PINNs). [15 Points]
In the lecture, we saw how to construct a PINN for solving PDEs. This problem guides you through the steps involved in finding Data-Driven solutions for PDEs with complex-valued solutions. Consider the nonlinear Schrodinger equation along with the given initial and boundary conditions,

$$ih_t + 0.5h_{xx} + |h|^2h = 0, x \in [-5,5], t \in [0,\pi/2], \tag{1a}$$
$$h(0,x) = 2\,\mathrm{sech}(x), \tag{1b}$$
$$h(t,-5) = h(t,5), \tag{1c}$$
$$h_x(t,-5) = h_x(t,5), \tag{1d}$$

where $h(t,x)$ is the complex-valued solution of the above PDE. Let $u(t,x)$ denote the real part of h and $v(t,x)$ denote the imaginary part, i.e., solution will be computed as $u(t,x) + iv(t,x)$. $\mathrm{sech}(x) = \frac{2}{e^{-x}+e^{x}}$ is the hyperbolic secant function. Then, we can place a prior on the multi-output solution neural network $h(t,x) = [u(t,x), v(t,x)]$. This will result in a complex-valued (multi-output) PINN f(t,x).

Data available to you, $\{t_0^i, x_0^i, h_0^i\}_{i=1}^{N_0}$ denotes the initial condition data with $t_0^i = 0$, $\{t_b^i\}_{i=1}^{N_b}$ corresponds to the boundary training data, and $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ represents the sample points on to enforce the structured imposed by the Schrodinger equation on $f(t,x)$

(a) Define the physics informed neural network f(t, x) you might use for getting the solutions of the above PDE. [3 Points]

(b) How many neural network do you need to solve the problem? If you need more than one network, do these networks share learn-able parameters between them? Write down the pseudocode for defining the neural network(s). (Note: follow the pseudocode style on Lecture 11, Example for Burgers' Equation) [8 Points]

(c) Define the loss function in detail for learning the parameters in $h(t, x)$ and $f(t, x)$ (Hint: The loss function has 3 terms, $LOSS_0$, corresponding to the loss on the initial data, $LOSS_b$ for enforcing the boundary conditions, and $LOSS_f$ for penalizing the Schrodinger equation not being satisfied on the collocation points.) [4 Points]

2. Question 2: Markov Decision Process. [20 Points]
Consider an environment in which our agent wants to find a policy that will lead to the shortest path to coffee. Once the agent reaches the coffee shop, it will stick around and enjoy the coffee there.

We can model this scenario as an MDP. Recall that an MDP is defined as tuple (S, A, T, R, γ), where:

S: The (finite) set of all possible states.

A: The (finite) set of all possible actions.

T : The transition function $T : S \times S \times A \to R$, which maps (s', s, a) to $P(s'|s, a)$, i.e., the probability of transitioning to state $s' \in S$ when taking action $a \in A$ in state $s \in S$. Note that $\sum_{s' \in S} P(s'|s, a) = 1$ for all $s \in S, a \in A$.

R: The reward function $R : S \times A \times S \to R$, which maps (s, a, s') to $R(s, a, s')$, i.e., the reward obtained when taking action $a \in A$ in state $s \in S$ and arriving at state $s' \in S$.

γ: The discount factor, which controls how important are rewards in the future. We have $\gamma \in (0, 1)$, where smaller values mean more discounting for future rewards.

Consider the instance shown in Figure 1. In the figure, the agent is at (1, 1), but it can start at any of the grid cells. The goal, displayed as a coffee cup, is located at (6, 6). The agent is able to move one square up, down, left and right (deterministically). Walls are represented by thick black lines. The agent cannot move through walls. All actions are available in all states. If the agent attempts to move through a wall, it will

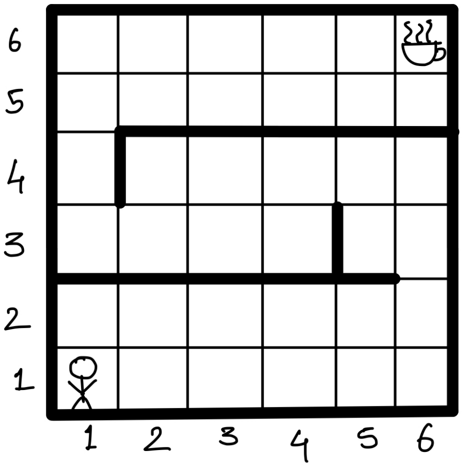

Figure 1: A particular instance of the shortest path problem. The goal is for the agent currently located in state (1, 1) to have a policy that always leads it on the shortest path to the coffee in state (6, 6).

remain in the same state. When the agent reaches the coffee cup, the episode ends. In other words, every action in the coffee cup state keeps the agent in the same state.

(a) Deterministic MDP [14 Points]
Assume we are modelling the MDP as an infinite horizon MDP. The coffee cup is still an absorbing state. Using the above problem description answer the following questions:

a) How many states are in the MDP? (i.e. what is $|S|$). [1 Points]

b) How many actions are in this MDP? (i.e. what is $|A|$). [1 Points]

c) What is the dimensionality of the transition function T? [1 Points]

d) Fill in the probabilities for the transition function T. [3 Points]

		s'				
s	**a**	(1, 2)	(2, 2)	(1, 3)	(2, 5)	(6, 6)
(1, 2)	up					
(1, 2)	down					
(2, 4)	up					
(5, 6)	left					

e) Describe a reward function $R : S \times A \times S$ and a value of γ that will lead to an optimal policy giving the shortest path to the coffee cup from all states. [2 Points]

f) Does $\gamma \in (0, 1)$ affect the optimal policy in this case? Explain why. [2 Points]

g) How many possible deterministic policies are there? (All policies, not just optimal policies). [2 Points]

h) What is the optimal policy? Draw the grid and label each cell with arrows in the direction of the optimal action. If multiple arrows, include the probability of each arrow. There may be multiple optimal policies, pick one and show it. [3 Points]

(b) Stochastic MDP. [6 Points]
Now consider that our agent often goes the wrong direction because of how tired it is. Now each action has a 10% chance of going perpendicular to the left of the direction chosen and 10% chance of going perpendicular to the right of the direction chosen. Given this change answer the following questions:

a) Fill in the values for the transition function T [2 Points].

		s'		
s	**a**	(2, 3)	(3, 4)	(4, 3)
(3, 3)	up			

b) Does the optimal policy change compared to deterministic case? Justify your answer [2 Points].

c) Will the value of the optimal policy change? Explain [2 Points].

3. Question 3: Dynamic Programming [15 Points]

Consider an MDP with a finite set of states and a finite set of actions. Denote the reward function $R(s, a)$, which is the immediate reward at state s upon taking action a. Let $Pr(s'|s, a)$ be the transition probability of moving from state s to s' upon taking action a. Let $0 \leq \gamma < 1$ be the discount factor.

Let V denote a value function, which associates a value $V(s)$ for each state s. Let $Bell(V)$ be the Bellman update operator when applied to V. Specially, it is defined as follows: $\tilde{V} = Bell(V)$ where,

$$\tilde{V}(s) = \max_a \left(R(s, a) + \gamma \sum_{s'} Pr(s'|s, a)V(s') \right),$$

Let us now prove that this update rule converges to the optimal values.

(a) Show that the Bellman operator is a contraction mapping, Specially, show that for any two value functions,

$$\|Bell(V_1) - Bell(V_2)\|_\infty \leq \gamma \|V_1 - V_2\|_\infty \,,$$

where the $\|Z\|_\infty$ denotes the infinity norm of a vector Z, i.e., $\|Z\|_\infty = \max_x |Z(x)|$. [8 Points]

(b) Suppose that upon repeated updating, we reach a fixed point, i.e., we find a V such that $Bell(V) = V$. Show that this V is unique. [3 Points]

(c) Suppose we have found the unique V^* such that $Bell(V^*) = V^*$. V^* represents the value of the optimal policy. Please specify the optimal policy $\pi^*(s)$, which defines the action to be taken in state s, in terms of V^* and other relevant quantities. [4 Points]