Reinforcement Learning
Assignement - Stochastic Policy Gradient

In this assignment we will implement Stochastic Policy Gradient (SPG), first on simle MDPs, and then on a more challenging one.

1. **River Swim**
   (a) Implement the REINFORCE algorithm with and without baseline. You can use a tabular function approximator, discount factor $\gamma = 0.99$, learning rates $\alpha_\theta = 10^{-4}$, $\alpha_v = 10^{-3}$, episodes truncated after 100 time steps, with initial state $s = 1$. Provide an initial guess having the same probability of picking each action, e.g., $\theta = 1$, $v = 0$.
   (b) Consider episodes of 100 time steps, each initialized at state $s = 1$ with discount factor $\gamma = 0.99$. Implement a stochastic actor-critic algorithm based on the TD-error δ with V learned by standard TD(0). You can use learning rates $\alpha_\theta = 10^{-4}$ and $\alpha_v = 10^{-2}$. Provide an initial guess having the same probability of picking each action, e.g., $\theta = 1$, $v = 0$.

   Hint: Pseudocode for this algorithm is provided in the slides. For the value function you can use a tabular approximation. For the stochastic policy you can use a softmax FA with tabular linear features.
   (c) What happens if you provide an initial guess favoring the selection of $a = 1$?
   (d) What happens if you provide an initial guess having the same probability of picking each action, but with $n_s = 5$ states, i.e., a shorter river?
   (e) **Optional** implement SPG in combination with LSTDA using 1 episode per batch, i.e., batches of 100 time steps

2. **Frozen Lake**
   (a) Implement REINFORCE with and without baseline. You can use $\gamma = 0.9$, a tabular function approximator and learning rate $\alpha_\theta = 0.1$ over 2000 training episodes. For the value function baseline, you can use a tabular approximator and a learning rate $\alpha_v = 0.5$.
   (b) Try to use an NN with 2 ReLU neurons followed by a linear layer with as many outputs as actions for the policy and a tabular approximator for the value function.
   (c) **Optional** Implement an Actor-Critic SPG algorithm based on the TD-error δ. To this end, similarly to the previous question, you will have to learn both a stochastic policy and a value function. You can use $\gamma = 0.9$, a tabular function approximator and learning rate $\alpha_\theta = 0.1$ over 2000 training episodes. For the value function baseline, you cna use a tabular approximator and a learning rate $\alpha_v = 0.5$. You can try to use an NN with 2 ReLU neurons followed by a linear layer with as many outputs as actions for the policy and a tabular approximator for the value function.

3. **Cart-Pole**
   (a) Implement REINFORCE with and without baseline . You can use $\gamma = 0.99$, a tabular function approximator and learning rate $\alpha_\theta = 10^{-4}$ over 2000 training episodes. For the value function baseline, you cna use a tabular approximator and a learning rate $\alpha_v = 10^{-3}$. You can try to use an NN with 8 ReLU neurons followed by a linear layer with as many outputs as actions for the policy and an NN with 8 ReLU neurons followed by a linear layer with one output for the value function.
   (b) Implement an Actor-Critic SPG algorithm based on teh TD-error δ. To this end, similarly to the previous question, you will have to learn both a stochastic policy and a value function. You can use $\gamma = 0.99$, a tabular function approximator and learning rate $\alpha_\theta = 10^{-4}$ over 10000 training episodes. For the value function baseline, you cna use a tabular approximator and a learning rate $\alpha_v = 10^{-4}$. You can try to use an NN with 8 ReLU neurons followed by a linear layer with as many outputs as actions for the policy and an NN with 3 layers of 8 ReLU neurons followed by a linear layer with one output for the value function.