A policy-based quiz

Quiz, 7 questions

1 point

1.

In broad strokes, how do policy-based methods work?

- [ ] Define a policy as an arg-max of Q-values learned by value-based methods.
- [ ] Define exploration policy (e.g. epsilon-greedy). Then train Q-values in a way that accounts for current exploration policy.
- [ ] Learn the optimal reward function given a fixed policy of a rational agent.
- [x] Parameterize the action-picking policy. Find such policy parameters that maximize expected returns.

---

1 point

2.

Policy gradient -- it's a gradient of what function and with respect to what inputs?

- [x] A gradient of expected reward w.r.t. action probabilities
- [ ] A gradient of policy w.r.t. action probabilities
- [ ] A gradient of policy w.r.t. actions
- [ ] A gradient of policy w.r.t. states

---

1 point

3.

Which of those methods can learn from partial trajectories?

- [ ] Value Iteration
- [ ] REINFORCE

A policy-based quiz

Quiz, 7 questions

- [x] Q-learning
- [x] Advantage Actor-Critic
- [x] SARSA
- [ ] Crossentropy method

---

1 point

4.

What are valid reasons to use Q-learning and not REINFORCE

- [x] Unlike reinforce, Q-learning can be trained much more efficiently with experience replay
- [x] Unlike REINFORCE, Q-learning can be trained on partial experience (e.g. s,a,r,s')
- [ ] Unlike REINFORCE, Q-learning can work with discounted rewards.
- [ ] Unlike REINFORCE, Q-learning does not require exploration.
- [ ] Unlike REINFORCE, Q-learning directly optimizes expected sum of rewards over session

---

1 point

5.

Which of the following is a valid expression for policy gradient J?

Legend:

- G(s,a) - discounted reward
- r(s,a) - immediate reward
- gamma - discount factor for discounted reward
- d(s) - a probability of being in this state at a random moment along random trajectory sampled with current policy
- pi(a|s) - agent's policy

- (●) \nabla J = \underset{E}{s \sim d(s), a \sim \pi} \nabla log pi (a | s) * G(s, a)
- ( ) \nabla J(s) = \underset{E}{s \sim d(s), a \sim \pi, s' \sim P(s' | s,a)} r(s,a) + gamma * \nabla J(s')
- ( )

\nabla J = \underset{E}{s \sim d(s), a \sim \pi, s' \sim P(s' | s,a)} \nabla \pi (a | s) * G(s, a)

A policy-based quiz

Quiz, 7 questions

\nabla J = \underset{E}{s \sim d(s), a \sim \pi, s' \sim P(s' | s,a)} \nabla \pi(a | s) * \nabla G(s,a)

---

1 point

6.

How does advantage actor critic works?

- It trains a network to predict advantage A(s,a) = Q(s,a) - V(s) and picks action with highest predicted advantage
- Actor is trained by the gradients propagated through the critic.
- It uses learned state values(critic) as a baseline for policy gradient(actor)
- It trains an ensemble of two models - Q-learning(critic) and REINFORCE(actor) - and picks actions by voting.
- It trains an agent (actor) with a help of human critic

---

1 point

7.

How do you train critic in Advantage Actor Critic?

- A critic predicts V(s), we minimize [r + gamma * const(V(s')) - V(s)] ^2
- A critic predicts Q(s, a), we minimize [r + gamma * max(Q(s',a')) - Q(s,a)] ^2
- \nabla J = \underset{E}{s \sim d(s), a \sim \pi, s' \sim P(s' | s,a)} \nabla \pi (a | s) * G(s, a)
- In advantage actor-critic there's no need to train critic
- With policy gradient \nabla J(s) = \underset{E}{s \sim d(s), a \sim \pi, s' \sim P(s' | s,a)} r(s,a) + gamma * \nabla J(s')

I, **Jiadai Zhao**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

A policy-based quiz

Submit Quiz

Quiz, 7 questions