Kennedy Uzoho

Southern New Hampshire University

CS 370 AI/ML

6-2 Assignment: Cartpole Revisited

Solving the CartPole Problem using REINFORCE Algorithm

The REINFORCE algorithm is a policy gradient method used to train a neural network to output actions based on state observations. Here's how the CartPole problem can be solved using the REINFORCE algorithm:

Let us define the problem, the CartPole problem involves balancing a pole on a cart by moving the cart left or right. The goal is to keep the pole upright for as long as possible. To solve this problem, we will define the policy: The policy is a function that maps a state observation to an action. In this case, the policy is a neural network that takes in the state of the cart and pole and outputs the action to take. Next, we define the reward function. The reward function gives a numerical score for each action taken by the agent. In this case, the reward is 1 for each time step the pole remains upright and 0 when it falls. Next, we define the loss function. The loss function is used to update the weights of the policy network to maximize the expected reward. In the case of REINFORCE, the loss function is the negative log probability of the chosen action multiplied by the total reward received after taking that action. Then, we can update the policy network using the gradient of the loss function with respect to the network weights.

Pseudocode:

1. Initialize policy network parameters

2. For each episode:
a. Reset the environment to its initial state
b. Collect state-action-reward sequences using the current policy
c. Calculate the total reward for each sequence.
d. Calculate the loss for each sequence using the REINFORCE formula.
e. Update the policy network weights using the gradients of the loss with respect to the network parameters.

3. Repeat steps 2-3 until convergence

Solving the CartPole Problem using A2C Algorithm

The A2C algorithm (Advantage Actor-Critic) is a policy gradient method that combines the actor and critic networks to improve the learning process. Here's how the CartPole problem can be solved using the A2C algorithm: We have defined our problem which is CartPole problem that involves balancing a pole on a cart by moving the cart left or right. The goal is to keep the pole upright for as long as possible. Next, we define the policy and value networks. The policy network is a function that maps a state observation to an action, and the value network estimates the expected total reward for a given state. Next, we define the reward function. The reward function gives a numerical score for each action taken by the agent. In this case, the reward is 1 for each time step the pole remains upright and 0 when it falls. Next, we define the loss function. The policy loss is the negative log probability of the chosen action multiplied by the advantage function, which is the difference between the expected total reward and the estimated value for a given state. The value loss is the mean squared error between the expected total reward and the estimated value. Then, we can update the weights and the value network using the gradients of the loss functions with respect to the network weights.

Pseudocode

1. Initialize policy and value network parameters
2. For each episode:
  a. Reset the environment to its initial state.
  b. Collect state-action-reward sequences using the current policy.
  c. Calculate the total reward and estimated value for each sequence.
  d. Calculate the advantage for each state-action pair.
  e. Calculate the policy and value loss using the A2C formula.

f. Update the policy and value network weights using the gradients of the loss with respect to the network parameters.

3. Repeat steps 2-3 until convergence

Policy Gradient vs Value-Based Approaches

Policy gradient methods, such as REINFORCE and A2C, directly optimize the policy function by computing gradients of the expected reward with respect to the policy parameters. In contrast, value-based methods, such as Q-learning, estimate the value function of a state-action pair and choose actions that maximize the expected value. Policy gradient methods have the advantage of being able to handle continuous action spaces and optimize stochastic policies directly, while value-based methods require discretization of the action space or the use of a separate policy network. However, policy gradient methods may suffer from high variance in the estimates and slower convergence compared to value-based methods.

Actor-Critic vs Policy Gradient and Value-Based Approaches

Actor-critic methods, such as A2C, combine both the policy and value functions and estimate both at the same time. The actor (policy) updates are driven by the advantage function, while the critic (value) updates are driven by the mean squared error between the predicted value and the actual total reward.

Actor-critic methods have the advantage of combining the strengths of both policy gradient and value-based methods, resulting in faster convergence and more stable learning. However, they require additional networks and computations compared to the other approaches.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (1970, January 1). [PDF] *language models are few-shot learners: Semantic scholar*. ArXiv. Retrieved February 18, 2023, from https://www.semanticscholar.org/paper/Language-Models-are-Few-Shot-Learners-Brown-Mann/6b85b63579a916f705a8e10a49bd8d849d91b1fc

Part 1: Key Concepts in RL - Spinning Up documentation. (n.d.). Retrieved February 20, 2023, from https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

Juliani, A. (2017, May 26). *Simple reinforcement learning with tensorflow part 0: Q-learning with tables and Neural Networks.* Medium. Retrieved February 20, 2023, from https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-0-q-learning-with-tables-and-neural-networks-d195264329d0

Sergios Karagiannakos. (2018, November 17). *The idea behind actor-critics and how A2c and A3C improve them.* AI Summer. Retrieved February 20, 2023, from https://theaisummer.com/Actor_critics/

Yoon, C. (2019, May 23). *Deriving policy gradients and implementing reinforce*. Medium. Retrieved February 20, 2023, from https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63