

Alfredo Romero

CS 370 Current/Emerging Trends

SNHU

10/08/2023

6-2 Assignment: Cartpole Revisited

In the world of computers, there are smart algorithms that can help machines learn to perform tasks on their own. Today, we're going to explore two different methods to teach a computer to balance a pole on a moving cart, a problem known as the cartpole challenge. We'll look at two ways: one is like teaching the computer dance moves, and the other is like having a coach give it tips. We'll also talk about how these methods are different from each other and from some other ways we teach computers.

1. Dancing with the REINFORCE Algorithm:

Imagine you're teaching a robot to dance. You want it to learn a dance routine that gets it the most applause from the audience. This is what the REINFORCE algorithm does for the cartpole problem. Here's how it works in simpler terms:

- We start with a robot and a dance routine (a policy) that it follows.
- The robot tries out different moves while dancing (taking actions).
- After each dance, we give it a score (a reward) based on how long it balanced the pole.
- The robot learns to adjust its dance moves to get better scores.
- We keep repeating this process until the robot becomes a great dancer.

2. Getting Coaching with the A2C Algorithm:

Now, think of the A2C algorithm as a robot getting coaching while it's dancing. It has a dance coach (actor network) and a critic (someone who tells the robot how well it's doing). Here's how A2C works:

- The robot and its coach start with some dance moves (a policy) and a way to tell how good the dance is (a value function).
- The robot dances and gets feedback from the coach on how well it's doing.
- The coach helps the robot adjust its moves to improve its dancing.
- Both the robot and the coach get better over time by working together.
- With time, the robot becomes an awesome dancer with the coach's help.

3. Different Ways of Teaching Computers:

Now, let's talk about why these methods are different from each other and from other ways we teach computers:

- Policy-based methods like REINFORCE focus on teaching computers to make decisions directly. It's like telling them, "Just do whatever works best." These methods are great when the computer has to make lots of choices, like in continuous or complicated tasks.
- Value-based methods, on the other hand, teach computers to figure out the value of different actions and then decide which one is best. It's like telling them, "Think about it and then choose." These methods are handy when the computer faces simpler decisions in discrete tasks.

4. Actor-Critic Methods:

Actor-critic methods, such as A2C, combine the best of both worlds. It's like having a dancing robot with a coach who tells it how good its moves are. The robot learns to dance better, and the coach gets better at giving feedback. These methods are super helpful because they balance exploring new dance moves (policy-based) with learning from feedback (value-based).

In conclusion, we've seen how computers can learn to balance a pole on a cart using two different methods: REINFORCE, where it learns to make decisions directly like a dancer, and A2C, where it gets coaching from a critic like a dancer with a coach. These methods are cool because they teach computers to learn from their actions and feedback. We've also learned how these methods are different from each other and from other ways we teach computers. So, next time you see a dancing robot, you'll know it might have used one of these methods to become a great dancer!

Source

- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning (ICML) (pp. 1928-1937).
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.