CS-370-Current/Emerging Trends in CS

Module 6-2 Assignment: Cartpole Revisited

Stephen Owusu-Agyekum

Southern New Hampshire University

November 29, 2023

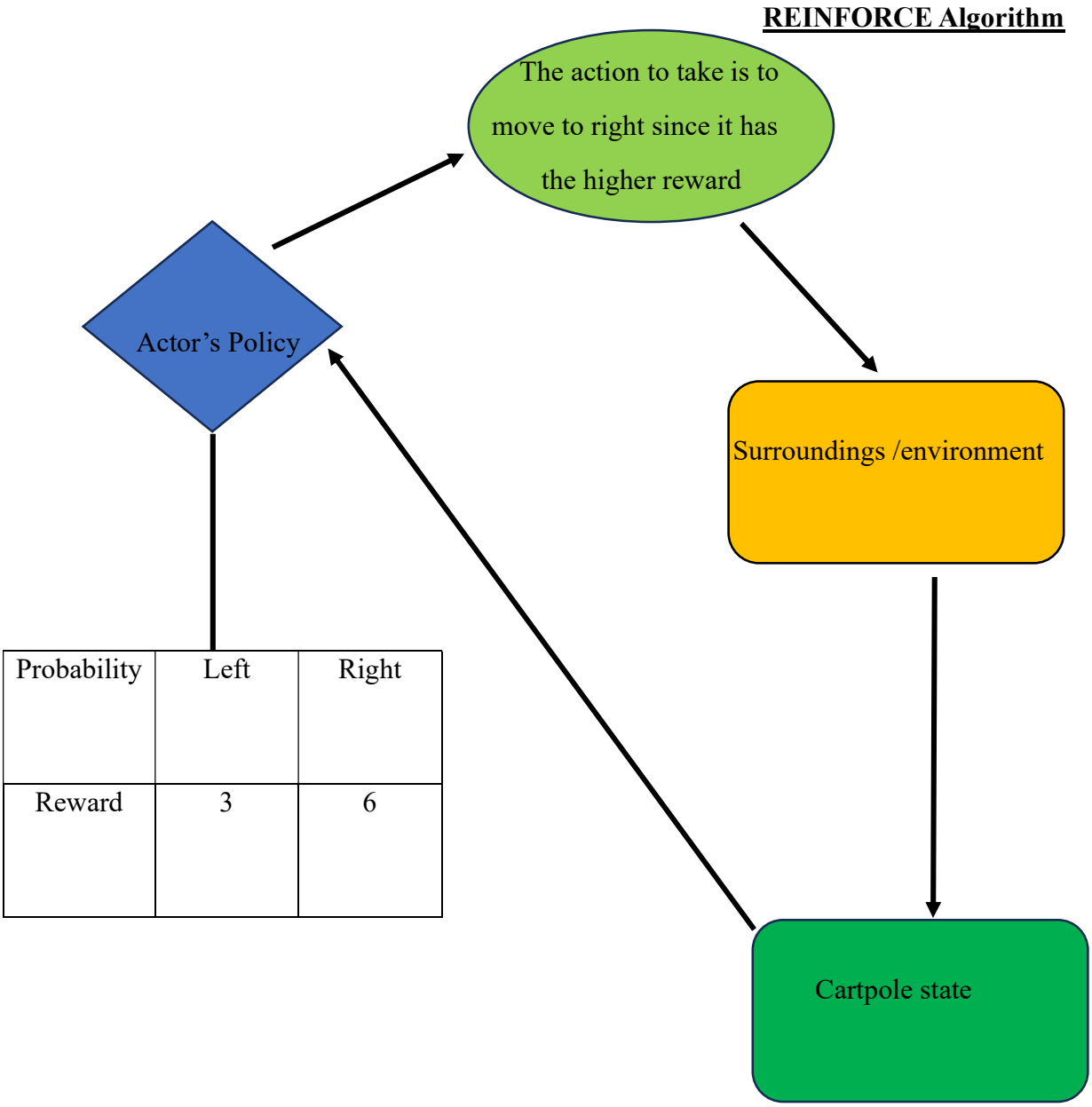
Explain how the cartpole problem can be solved using the REINFORCE algorithm. Consider using pseudocode, UML, diagrams, or flowcharts to help illustrate your solution.

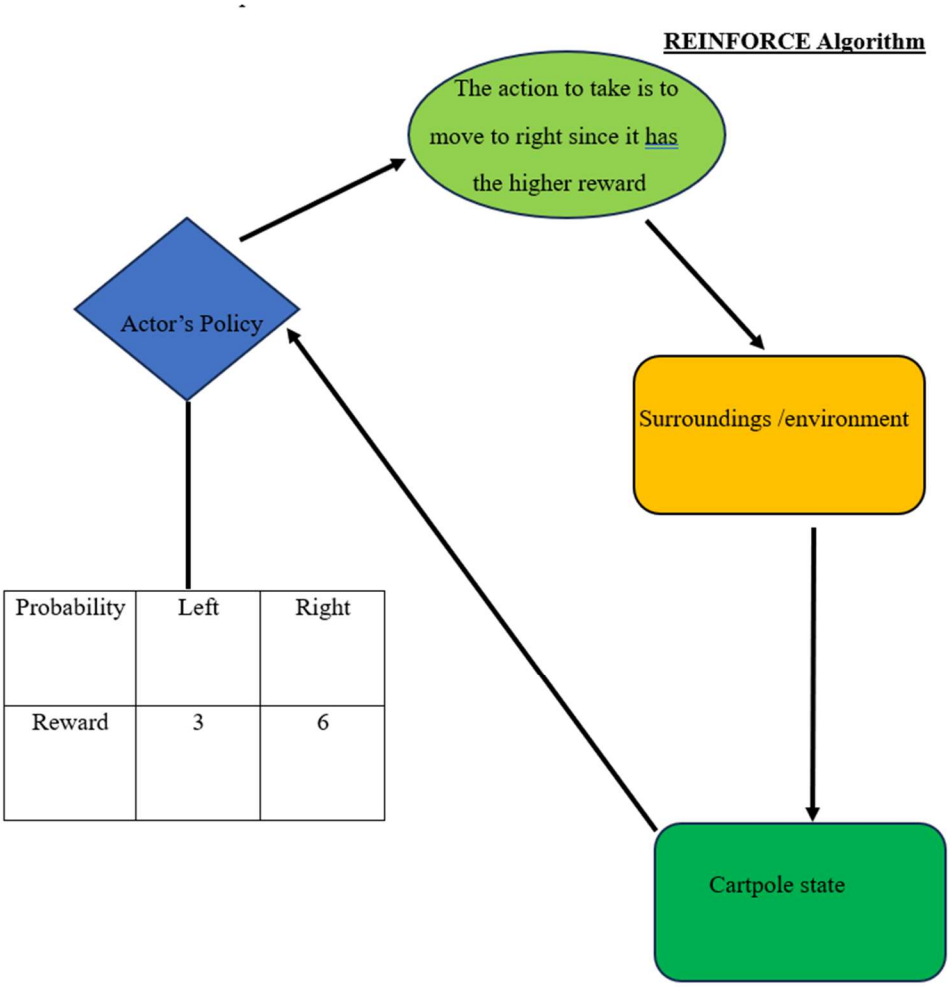
1. The REINFORCE algorithm is a policy-based method that aims to an actor to directly learn the policy by optimizing the expected cumulative reward. It can also refer to a computational approach or strategy used in reinforcement learning that helps an agent learn from and make decisions in its surroundings through interaction. The goal of the cartpole issue is to balance a pole by shifting the cart left or right while ensuring that the pole has to remain balanced as long as feasible. These algorithms play a crucial role in directing the agent's behavior in order to maximize cumulative rewards over time.

The REINFORCE algorithm also stimulates an actor to engage with its immediate surroundings by employing rules and policies that specify what actions it can do at any given time. “REINFORCE is a member of a unique class of Reinforcement Learning algorithms known as Policy Gradient algorithms” (Samsudin, 2020). REINFORCE algorithm emerges as a potent tool. At its core, REINFORCE revolves around policy gradients, seeking to optimize the policy that governs decision-making in an environment. Here are some of the ways REINFORCE algorithms can be used to solve cartpole problems.

The position, velocity, angle of the cart, and angular velocity of the cart represent the state in the cartpole problem. The actions include pushing the cart to the left or right with force. As the agent interacts with the environment, it collects crucial data: the state at each time step, the chosen action, and the resulting rewards. These rewards signify how well the agent is performing its balancing act. The algorithm calculates returns, typically cumulative rewards, from the point of action till the episode's conclusion. Ultimately, the REINFORCE algorithm navigates the cartpole landscape by iteratively learning from experiences, shaping a policy that maximizes long-term rewards. After assessing the cartpole's condition, the actor would follow the policy and make the optimum adjustment to balance the pole.

Let’s consider the example in the Flowchart below.





The flowchart above illustrates the fundamental steps of REINFORCE, which involve iteratively interacting with the environment, collecting data, and adjusting the policy to increase the likelihood of actions that lead to higher rewards. It started by initializing the policy's weights randomly. The agent interacts with the environment, selecting actions according to the policy and observing resulting states and rewards. After each episode, the agent updates its policy by computing the gradients of the policy with respect to the returns obtained. These gradients are used to update the policy's weights to encourage actions that lead to higher rewards.

2. Explain how the cartpole problem can be solved using the A2C algorithm. Consider using pseudocode, UML, diagrams, or flowcharts to help illustrate your solution.

The A2C algorithm, known as Advantage Actor Critic, is a policy gradient method made up of three fundamental components. This algorithm combines the advantages of both actor-critic methods and synchronous training, utilizing a policy network, value network, and an algorithm that updates both networks simultaneously, enhancing its efficiency in reinforcement learning tasks. The actor network, the critic network, and the advantage work together.

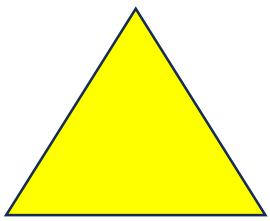
During this process, the actor selects the course of action, the critic determines the likelihood that the course of action will be adopted, and the advantage, in turn, forecasts whether the outcome will be better or worse than expected and values the course of action accordingly. “The concept is that the Advantage function determines how much better an action is taken in a state in comparison to the state's average value” (Simonini, 2022). For instance, let’s say you (actor) are performing some actions based on certain policies you have set, and your friend is observing to make sure that you are doing it right by providing you with feedback and learning from it.

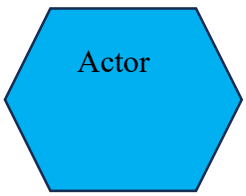
Through this learning process, the critic (your friend) provides feedback based on the action of the actor (you), and you learn from the feedback. “If the critic (your friend) provides negative feedback, you (actor) update your parameters to improve it and be better next time (Simonini, 2022) and if the feedback is positive, the actor is motivated to continue employing its parameter. However, your friend (Critic) will also make changes to their feedback-giving process so that it can be improved moving forward. The cartpole problem is a classic control problem in which the goal is to balance a pole on a moving cart.

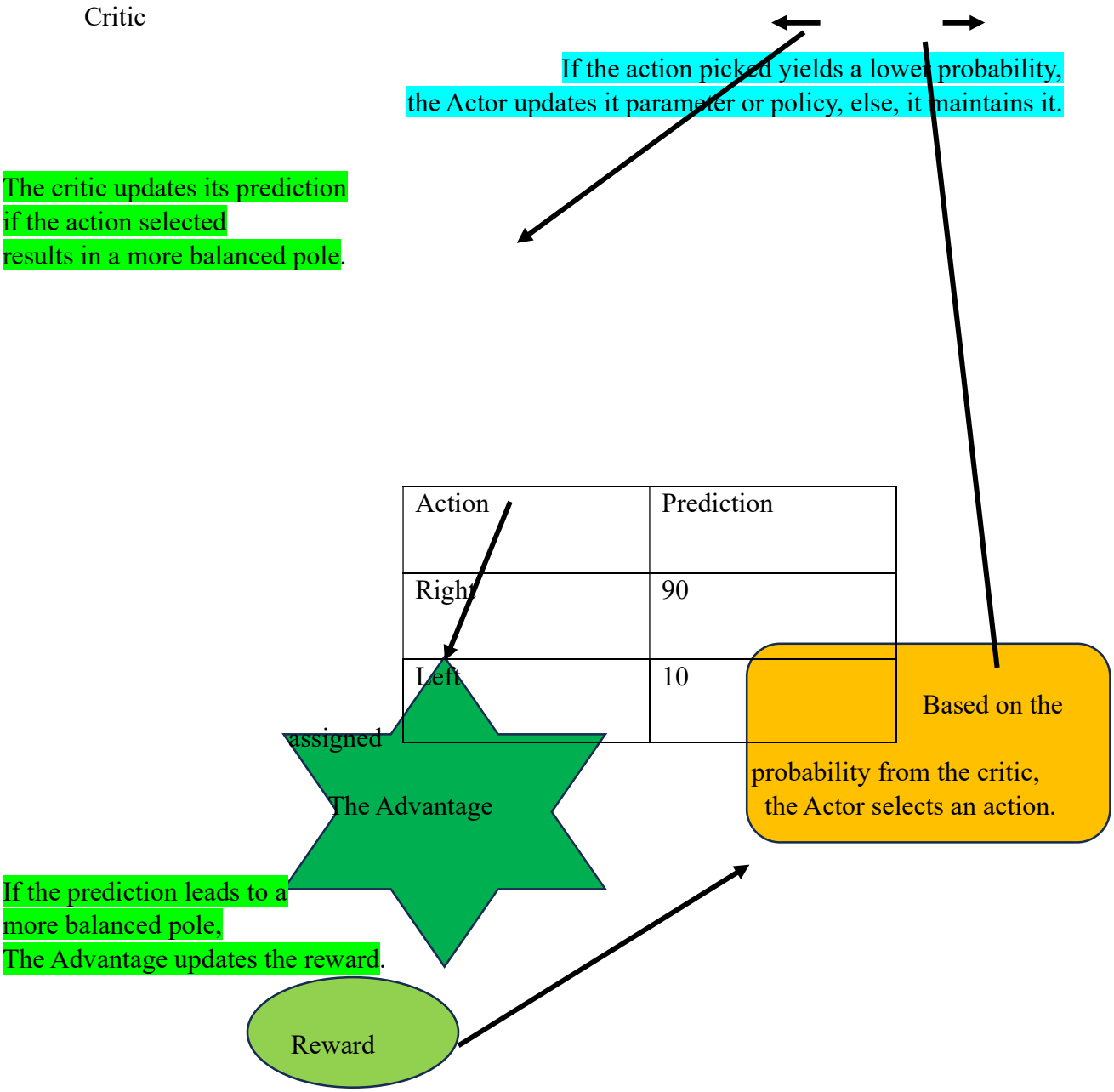
By Utilizing the 2AC method, the actor could select an action based on a forecast provided by the critic in order to solve the cart pole problem. If the action led to a better-balanced pole after it was taken, the critic updated the prediction. The advantage helps in updating the policy (the actor's strategy) by guiding it towards actions that result in higher advantages. Actions with a positive advantage are more favorable and likely to be chosen.

Flowchart (2AC Algorithm)

Flowchart (2AC Algorithm)






Critic
If the action picked yields a lower probability,
the Actor updates it parameter or policy, else, it maintains it.
The critic updates its prediction
if the action selected
results in a more balanced pole.
Action
Prediction
Right
90
Left
10
Based on the
probability from the critic,
the Actor selects an action.
assigned
The Advantage
If the prediction leads to a
more balanced pole,
The Advantage updates the reward.
Reward

Flowchart (2AC Algorithm)