Arielle Moore

CS 370: Current/Emerging Trends in CS

February 9th, 2026

6-2 Submit Assignment: Cartpole Revisited

A classic benchmark in reinforcement learning, the Cartpole problem involves balancing a pole on a moving cart by applying discrete forces to the left or right (cart position, cart velocity, pole angle, and pole tip velocity) (Yoon, 2018). One method that addresses this task is the REINFORCE algorithm, a policy gradient approach that directly optimizes the probability distribution over actions (Yoon, 2018). The way that the agent interacts with the environment is by observing states, sampling actions from its policy, and receiving awards, all of which are used to compute the total return for a single episode. The policy's parameters are then updated to increase the probability of actions that ultimately led to higher cumulative rewards, as well as decrease the probability of less successful actions (Yoon, 2018). In short, the policy is updated in the direction of actions that led to higher returns. The pseudocode below demonstrates how the policy parameters are adjusted after each episode, thus reinforcing actions that will result in higher cumulative rewards (Yoon, 2018).

REINFORCE Pseudocode:

```
for episode in range(max_episodes):
    states, actions, rewards = [], [], []
    state = env.reset()
    done = False
    while not done:
        action = sample(policy(state))
        next_state, reward, done = env.step(action)
        store(states, actions, rewards)
        state = next_state
    update_policy(states, actions, rewards)  # Scale gradients by total discounted reward
```

The Advantage Actor-Critic (A2C) algorithm represents a more sophisticated approach by combining the power of both policy-based and value-based methods (Wang, 2021). Two networks perform in concert in the A2C approach: the actor maps states to action probabilities as the critic estimates the value of each state (Wang, 2021). The critic evaluates each state using the *temporal difference* (TD) error (meaning the difference between predicted and observed awards), which represents the advantage of any given action (Wang, 2021). The actor's policy updates are guided by this advantage; actions with positive advantage are reinforced, and in turn, actions with negative advantage are discouraged (Wang, 2021). In contrast to REINFORCE, which only updates the policy at the end of an episode, A2C provides a more stable, incremental learning path by updating both actor and critic networks with every step (Wang, 2021; Vellanki, 2026). In turn, this continuous update reduces variance while allowing the agent to respond quickly to new state information (Wang, 2021). The pseudocode below describes a single episode of an actor-critic reinforcement learning loop in which the agent selects actions using the actor network, evaluates them using the critic to compute the advantage, then updates both the actor and critic networks with each step until the episode ends.

A2C Pseudocode:

```
for episode in range(max_episodes):
    state = env.reset()
    done = False
    while not done:
        action = sample(actor(state))
        next_state, reward, done = env.step(action)
        advantage = reward + gamma *
critic(next_state) - critic(state)
        update_actor(state, action, advantage)
        update_critic(state, reward, next_state)
        state = next_state
```

Differing from value-based methods like Q-learning, policy gradient methods directly optimize the action-selection policy rather than estimating an expected future reward for every state-action pair (Yoon, 2018; Wang, 2021). Policy gradients produce a probability distribution over actions, as opposed to value-based methods that typically produce deterministic results by selecting the action with the highest-predicted value; the former encourages more flexible exploration and stochastic behavior (Yoon, 2018; Wang, 2021).

Actor-critic approaches such as A2C are set apart from both pure policy and value-based methods because they are able to learn a stochastic policy and a value function simultaneously (Wang, 2021). While the actor selects actions according to the learnt policy, the critic evaluates resulting states in order to provide feedback on action quality (Wang, 2021). This robust combination results in stabilized learning, allows incremental updates with each step, and reduces the high variance of pure policy gradient methods, all while maintaining the stochastic exploration advantage over value-based methods (Wang, 2021; Vellanki, 2026).

In summary, REINFORCE solves the CartPole problem by updating a policy based on cumulative episodic rewards, while A2C enhances learning efficiency by using actor-critic collaboration and continuous TD-error-based updates. Both methods represent different strategies when balancing exploration and exploitation in reinforcement learning tasks, demonstrating the balance and trade-offs between stability, simplicity, and learning speed.

References

Vellanki, Divya. (2026). *6-2 Assignment: Cartpole Revisited.* Codio Virtual Lab.

Wang, M. (2021, January 22). *Advantage Actor Critic Tutorial: minA2C.* Towards Data Science. https://towardsdatascience.com/advantage-actor-critic-tutorial-mina2c-7a3249962fc8/

Yoon, C. (2018, December 29). *Deriving Policy Gradients and Implementing REINFORCE.* Medium. https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63