The Cartpole problem is a classic reinforcement learning (RL) environment where the agent must learn and understand to balance a pole on a moving cart. During module five, the Deep Q-Network (DQN) approach was used to solve the problem by estimating the value of actions. I explored two alternative methods:

* REINFORCE
* Advantage Actor- Critic (ARC)

Then explained how they are different from value -based approaches like Q-learning. ***REINFORCE*** is a policy gradient algorithm that directly optimizes the agent’s policy by adjusting weights based on the rewards received. Now instead of estimating action values, REINFORCE samples actions from a policy network and updates using the gradient of the expected return.

Process involves:  
1. Running an episode and storing probabilities of actions

2. Calculating discounted rewards

3. Updating the policy network using the product of log probabilities and rewards

Pseudocode: (Patel, 2025) (Microsoft Corporation, 2022)

A computer screen shot of a program code

AI-generated content may be incorrect.This method works well for Cartpole because it learns directly from episodic feedback, though it can be unstable due to high variance in gradient estimates. This collects trajectory though self-play, computing discounted rewards, and use of those rewards to update the policy network vis gradient ascent.

A2C (Advantage Actor- Critic) combines the strengths of policy- based and value- based methods. Usage of two networks:

* **ACTOR**: suggests actions based on the current policy
* **CRITIC**: estimates the value of the current state

Pseudocode: (GeeksforGeeks, 2025) (Microsoft Corporation, 2022)  
A computer screen shot of text

AI-generated content may be incorrect.This approach is more sample- efficient and stable rather than REINFORCE, making it ideal for environments like Cartpole. Reminder that Actor improves policy, critic improves value estimates.

Policy gradient methods optimize the policy directly, making them suitable for continuous or high-dimensional action spaces. In contrast, value-based methods like Q-Learning estimate the value of actions and select the best one. Policy gradients can learn stochastic policies and adapt better to so easy environments, while value-based methods are more efficient in discrete settings.

Actor- critic methods like A2C blend both paradigms. The actor learns the policy, while the critic provides feedback on its performance. This dual structure reduces variance and improves convergence speed.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... & Hassabis, D. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676), 354–359<https://www.nature.com/articles/nature24270>

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press. <https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf>

Patel, R. (2025, September 24). Implementing policy gradient in Python. AIGreeks. <https://aigreeks.com/implementing-policy-gradient-in-python/>

GeeksforGeeks. (2025, July 23). Policy gradient methods in reinforcement learning. <https://www.geeksforgeeks.org/machine-learning/policy-gradient-methods-in-reinforcement-learning/>

Mitchell, A. (2024, August 14). An introduction to policy gradients with Cartpole and Doom. ExpertBeacon. <https://expertbeacon.com/an-introduction-to-policy-gradients-with-cartpole-and-doom/>

DeepMind. (2017, October 18). *AlphaGo Zero: Starting from scratch*. <https://deepmind.google/discover/blog/alphago-zero-starting-from-scratch/>

OpenAI. (2023). Cartpole environmental documentation. <https://gymnasium.farama.org/>

Microsoft Corporation. (Year). Visual Studio (Version 17.14.16) [Computer software]. <https://visualstudio.microsoft.com/>