Shawn Millin

CS-370

Cartpole revised

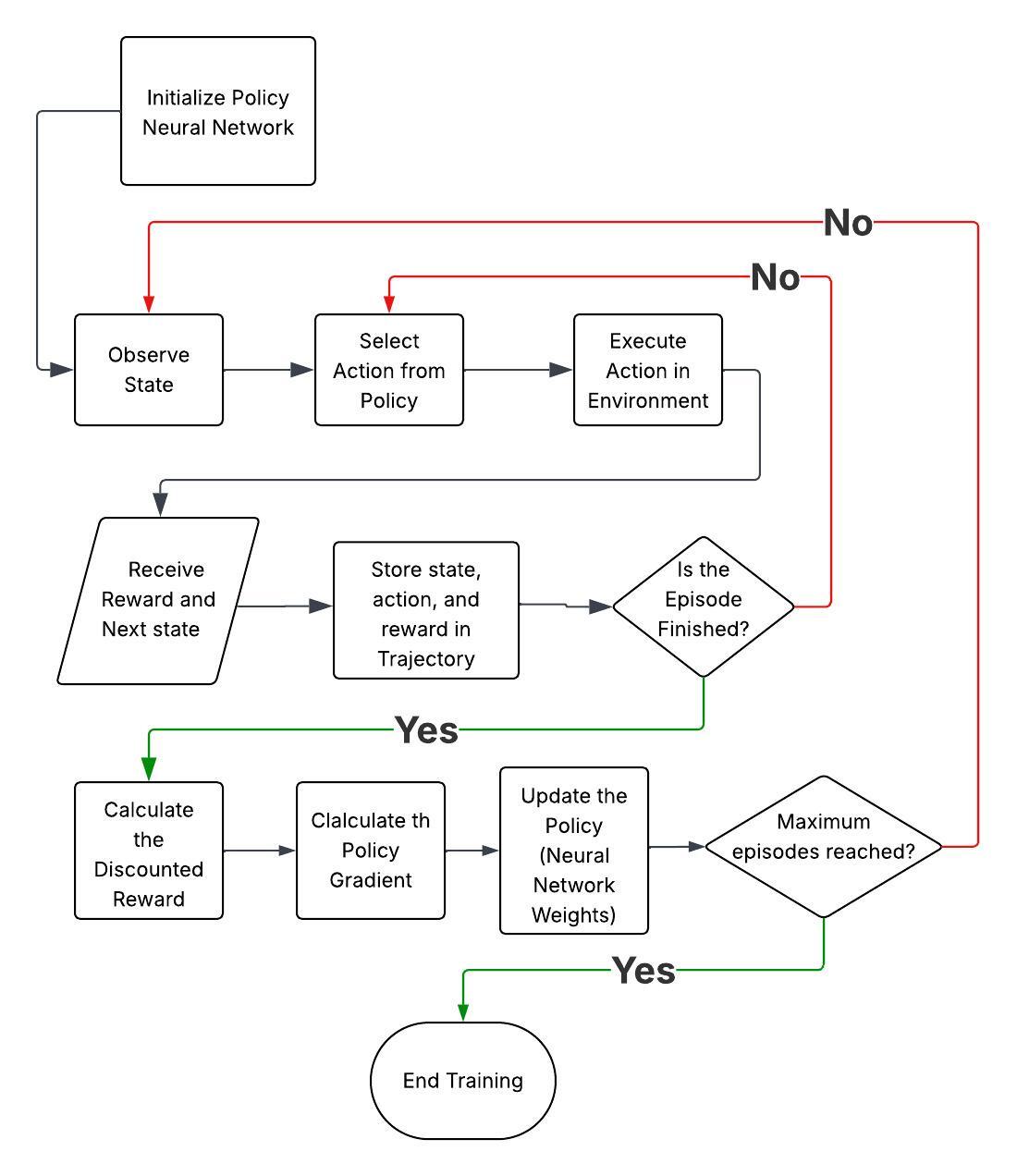
October 12th 2025

Cartpole Revised

The cartpole problem can be solved using the policy gradient REINFORCE Algorithm. This algorithm outputs a probability distribution with inputs of the cart position, cart velocity, pole angle, and pole angular velocity. The neural network used is policy-gradient based where in each episode the agent interacts with the environment by following the current policy and each step in the episode is sampled form the policy’s probability distribution. The results of the state, reward, and action probability are then recorded continuously until the end of the episode.

After the episode finishes there is a calculation of the discounted return, cumulative future rewards and the policy is updated, this being the neural network’s weights, to maximize the expected return. The policy is updated by calculating the policy gradient, which “indicates the direction of parameter adjustment that leads to higher expected rewards.” Joshi(2022). The algorithm iterates through many episodes calculating discounted reward and updating policy which leads to the policy learning more beneficial actions to maximize rewards and solving the cartpole problem.

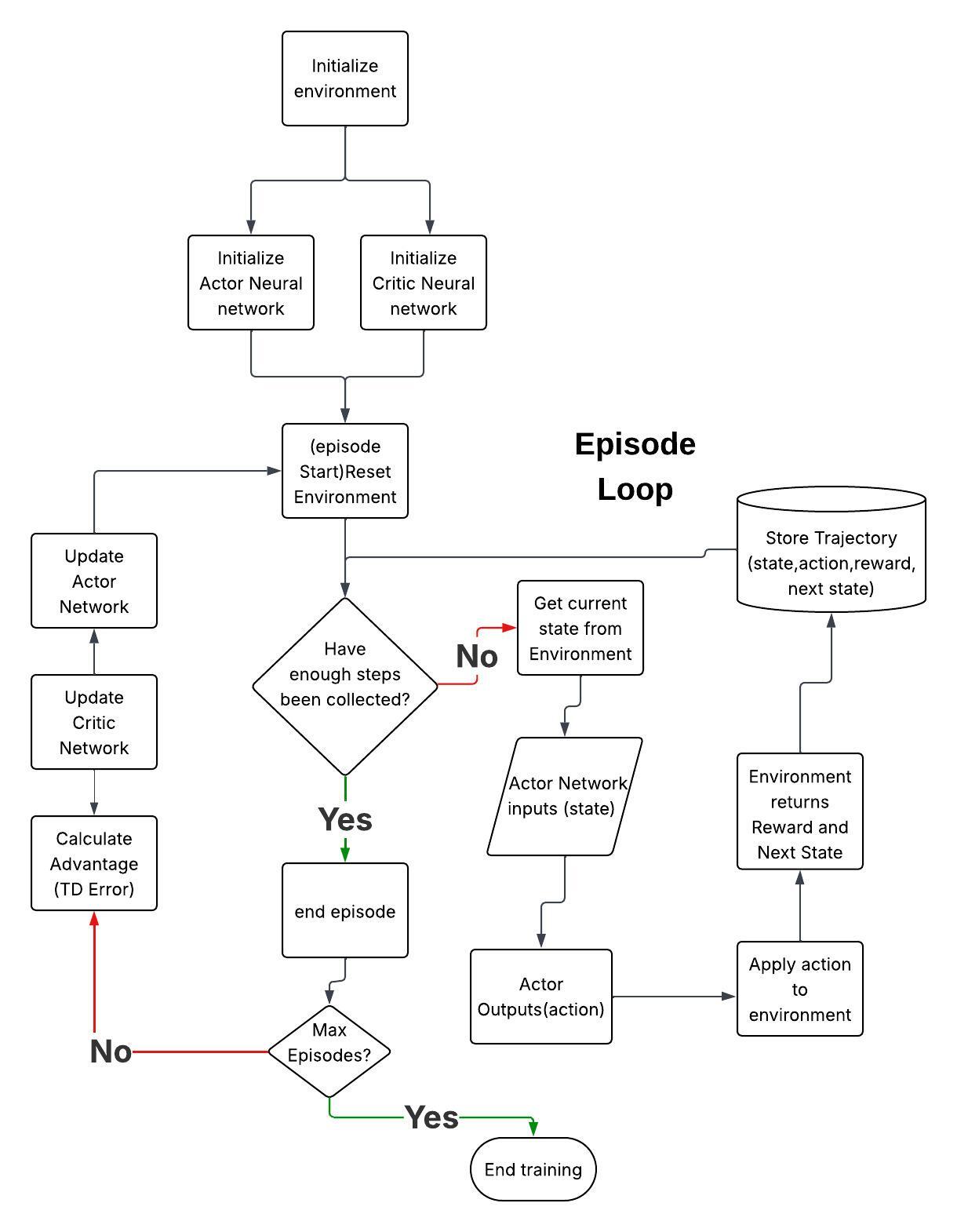
Flow chart of the policy-gradient REINFORCE algorithm:



The cartpole problem is also able to be solved with the Advantage Actor Critic Algorithm. This algorithm uses the two neural networks of Actor, policy-based, and Critic, value-based, which take the inputs of state(cart position, cart velocity, pole angle, and pole angular velocity), action (movement), and reward. The Actor neural network works similar to the REINFORCE algorithm by outputting a probability distribution from a state input and updated with policy-gradient..

The Critic neural network takes an input of state and “maps each state to its corresponding Q-value.” Joshi(2022), which is an estimated value of that state, and outputs the Temporal Difference Target (TD Target), which is the prediction of future rewards from the current state. This network is updated with the use of the Mean Square Error of the Temporal Difference Error (TD Error), in which the TD Error is the loss between the Temporal Difference Target and the predicted value of the current state (also called the Advantage), the updated output is called the Prediction Error. Getting the TD Error closer to zero will improve predicting the outcome from the current state.

Flowchart for Actor-Critic approach:



The approach of policy-gradient differs from value-based approaches in that value based (Q-Learning) does not have a predefined model of the environment upon starting each episode. Q-Learning uses a table format and uses the Q-Value Function, inputs of state and action, to calculate a Q-value for predicting future rewards and optimizing the policy to select actions for maximum rewards. The value-based approach is suited for discrete action spaces such as a grid environment of a game.

The policy-gradient approach starts with initialized parameters for the environment which are “optimized over subsequent time steps or episodes.” Walkerastro(2024). The optimization come from a probability distribution as the output from the sigmoid function and policy updates after every episode.

The policy-based approach can handle continuous action spaces such as the cartpole problem.

The approach of actor-critic differ from value-based and policy-based approaches in that the actor-critic approach combines the value-based method and the policy-based method. This combination allows the ability to benefit from handling a continuous action space (policy-based) and benefit from the improved stability (value-based). The actor -critic method uses the policy-base (actor) to select actions and value-base (critic) to estimate the value function resulting in the actor’s policy update being guided by the critic’s evaluation.

Resources:

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

Joshi, NandaKishore. (2022). *Part 2: Policy Based Reinforcement Learning – OpenAI’s Cartpole with REINFORCE algorithm.* <https://nandakishorej8.medium.com/part-2-policy-based-reinforcement-learning-openais-cartpole-with-reinforce-algorithm-18de8cb5efa4>

Walkerastro. (2024). *Policy Gradient methods vs Q-Learning.* Medium.

<https://medium.com/@walkerastro41/policy-gradient-methods-vs-q-learning-c9f513f63d3d>