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

Diagram

Description automatically generated

In the Reinforce algorithm we take the Agent, in the cartpole problem that would be the

moveable platform. As the platform moves, the pole starts to move this represents a change in

the environment. The goal is balance, the AI changes the state in a guess and it either passes or

fails. If it passes the agent recognizes the action and it creates the reward for future memory as

possible. It conducts this experiment many times over.

* **Explain how actor-critic approaches differ from value- and policy-based approaches**.

Policies take a state in the environment and then matches possible actions from that given

state, and then in a value policy expects to see a good return value based on the state it’s in. All

that to say that in policy-based approaches there are expected to be specified acts to take in a

given state, where as A2C is given a free range of actions to take so as not to specify the actions

and state. But can learn by playing with actions regardless of the state of the environment.

* **Explain how the cartpole problem can be solved using the A2C algorithm**. Consider using pseudocode, UML, diagrams, Diagram

  Description automatically generatedor flowcharts to help illustrate your solution.

The Idea behind the A2C, or actor too critic, is to take the actor, or moveable foundation, and

take an action, and see how it changes in the new Environment. This could be any type of action

as the actor decides. After the actor takes the action, it looks at the critic for feedback. The Critic

then looks at the state to see if the change is successful. If not, it throws it out and if it, is it sends

positive feedback to the Actor who then says it did an excellent job.

So, in the cartpole problem the actor moves the platform in any direction, then the critic

takes the change in the environment and matches it with a success value. If its bad it tells the

platform not to do it again, and if its good it gives the platform the message that its good.

* **Explain how policy gradient approaches differ from value-based approaches, such as Q-learning**.

In Policy gradient approach is over time to try to take actions and check the state against a

changeable max reward based on its state. Where the Neural network keeps the changes and

overtime creates a deep neural network. The Policy gradient looks for the optimal policy first,

which is helpful if the sample is too large. Whereas Value based approach like Q-learning looks

at the neural network for the ultimate valued reward instead of the short term which is good if the

sample size is not too large or constantly changing. Q learning estimates the action for a change

in state, looking for the best possible state.

References

# [Sergios Karagiannakos](https://medium.com/@SergiosKar?source=post_page-----6dd7dfd0acb8--------------------------------), November 16,2018, The idea behind Actor-Critics and how A2C and A3C improve them,

# https://towardsdatascience.com/the-idea-behind-actor-critics-and-how-a2c-and-a3c-improve-them-6dd7dfd0acb8

[Dhanoop Karunakaran](https://medium.com/@dhanoopkarunakaran?source=post_page-----84bde440c816--------------------------------)

Jun 4, 2020

# REINFORCE — a policy-gradient based reinforcement Learning algorithm

https://medium.com/intro-to-artificial-intelligence/reinforce-a-policy-gradient-based-reinforcement-learning-algorithm-84bde440c816