# Notes on Deep Reinforcement Learning through Policy Optimization by Pieter Abbeel and John Schulman

Compiled by D. Gueorguiev 2/17/2024

## Policy Optimization

A diagram of a agent and environment

Description automatically generated

A diagram of a network

Description automatically generated

A diagram of a function

Description automatically generated

Consider control policy parametrized by parameter vector . We want to maximize the expected reward over a given time interval represented by its discrete time points by selecting the optimal policy – that is the policy with such parameter vector which maximizes .

Often we choose a policy from a stochastic policy class which has smoothing effect on the problem dynamics compared to deterministic policy. We denote the stochastic policy with which represents the probability of action when in state . Finding optimal policy instead of finding the *state-value function* and the *action-value function* has advantages. Recall, is the expected return when starting in state and following policy thereafter. On other side, represents the expected return starting from state taking action and following policy thereafter.

A policy is defined to be better than or equal to a policy if its expected return is greater than or equal to that of for all states. We write:

We denote with any one of the optimal policies which have the same *state-value function*, denoted with , and defined as:

Optimal policies also share the same *optimal action-value function*, denoted with and defined similarly:

## Literature

[Deep Reinforcement Learning through Policy Optimization, Pieter Abbeel, John Schulman, OpenAI, Berkeley AI Research Lab, NIPS, 2016](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/docs/nips-tutorial-policy-optimization-Schulman-Abbeel.pdf)

[Human-level control through deep reinforcement learning, Volodymyr Mnih et al, Nature, 2015](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Human-level_control_through_deep_reinforcement_learning_Mnih_2015.pdf)

[Deep Reinforcement Learning with Double Q Learning, Hado van Hasselt et al, Google DeepMind, 2015](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Deep_Reinforcement_Learning_with_Double_Q-learning_Hasselt_2015.pdf)

[The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning, RY Rubinstein, DP Kroese, 2004](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/books/The_Cross_Entropy_Method_A_Unified_Approach_Rubinstein_Kroese_2004.pdf)

[Dueling Network Architectures for Deep Reinforcement Learning, Z Wang et al, Google DeepMind, 2015](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Dueling_Network_Architectures_for_Deep_Reinforcement_Learning_2015.pdf)

[Continuous Deep Q-Learning with Model-based Acceleration, S. Gu et al, U. of Cambridge, Google DeepMind, 2016](https://github.com/dimitarpg13/reinforcement_learning_and_game_theory/blob/main/articles/ReinforcementLearning/Continuous_Deep_Q-Learning_with_Model-based_Acceleration_Gu_2016.pdf)