Notes from Deep Reinforcement Learning

* In the reinforcement learning problem, we have an agent interacting with an environment, which provides numeric reward signals.
* Goal: Learn how to take actions to maximize the reward.
* The environment gives the agent a state, .
* In turn the agent takes an action, .
* The environment gives back a reward, , as well as its next state, .
* This process repeats until the environment produces a terminal state which ends the episode.
* Example: Cart-Pole Problem
  + Objective: Balance a pole on top of a movable cart
  + State: Angle, angular speed, position horizontal velocity
  + Action: Horizontal force applied to cart
  + Reward: 1 at each time step if the pole is upright
* Example: Atari game:
  + Objective: Complete the game with the highest score
  + State: Raw pixel inputs of the game state
  + Action: Game controls
  + Reward: Score increase/decrease at each time step

**Markov Decision Process (MDP)**

* Markov property: Current state completely characterizes the state of the world
* To mathematically formalize the RL problem we define a tuple
  + : Set of possible states
  + : Set of possible actions
  + : Distribution of reward given (state, action) pair
  + : Transition probability
    - i.e. Distribution over next state given
  + : Discount factor

**MDP Algorithm:**

At time step t=0, environment samples initial state

// Look at the first frame

for until done {

Agent selects action

// Observe the current game controls

Environment samples reward

  // Observe the current score of the game.

Environment samples next state

// Observe next frame

Agent receives reward and next state

// Update the games score and draw next frame to screen

}  // end loop

* Policy is a function mapping from set of states to set of actions , that specifies what action to take in each state
  + is program logic that determines what buttons to press based on what frames of video the algorithm sees
* Objective: Find the optimal policy that maximizes cumulative discount reward.
  + i.e. learn the best function that maximizes the "cost" function.
    - The cost is called the reward
      * really the sum of weighted rewards weighted by the discount factor to give higher weight to rewards that happened recently
      * I think gamma is between zero and one, so as increases it makes have less influence on the cost:
* The optimal policy I the policy that results in us taking an action that is most likely to get us to our terminal state.
  + Optimal policy tells us that whatever state we are in what is the action that we should take to maximize the sum of rewards we will get.
* Our objective is to find an optimal policy that maximizes the sum or rewards.
* How do we handle the randomness (initial state, transition probability, etc.)?
  + We maximize the expected sum of rewards: