Actor-Critic Reinforcement Learning Models

Pawan Jayakumar

May 2018

1 Terminology

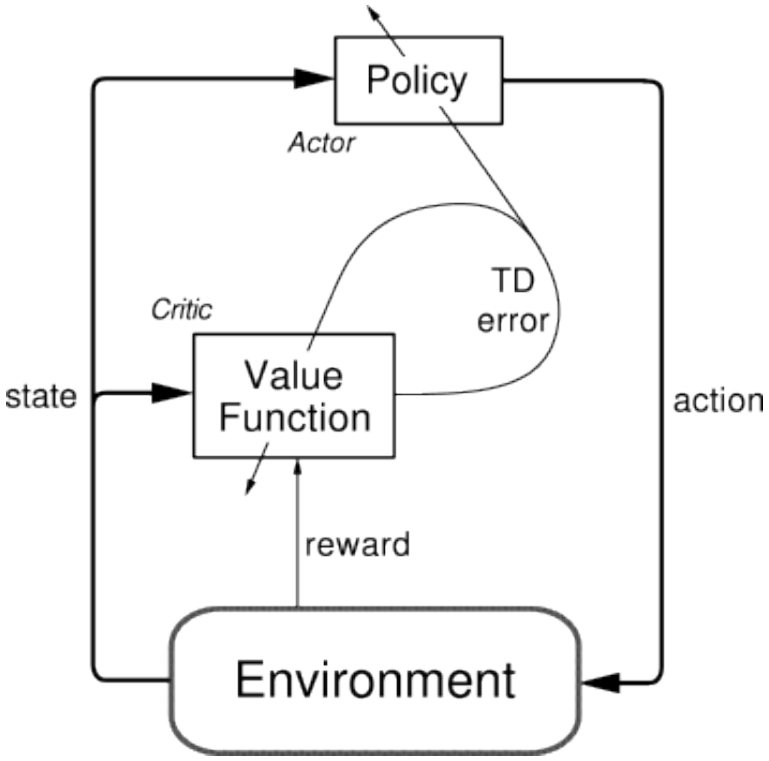
Any terms related to RL will be defined here in case you don't know them:
Action space: The set of possible actions the agent can take in state S
Q value: The Value of taking an action A in state S which is defined as the reward+discounted future rewards
Return: Discounted future rewards defined as $\sum_{k=0}^{\infty} \gamma^k r_t + k + 1$ Note: this calculates V_t which is used to approximate return in AC models
Reward: A scalar that the agent recieves from the environment that the agent tries to maximize
State: A unique situation in the environment
Policy: The method used to calculate what action to take given state S

2 Actor-Critic Models

2.1 Intuition

Problem In Q learning, the basic principle is finding the value of taking an action in a particular state aka the Q value. The problem with this is that this only works if there are relatively few states and actions.

Solution The way to solve this is to have two different models which as the name suggests, are the actor and the critic. The actor model uses a policy to take an action in the current state. The critic then receives the reward and using a value function, finds the temporal difference error to update both itself and the policy. Both models can be represented using a neural network with the input being environment information and the output layer of the actor model being a soft max of action probabilities.



Temporal Difference Learning The temporal difference error equation is very similar to the errors used in back propagation. Below is the TD(λ) equation which is used to find the error

$$\Delta w_t = \alpha(V_{k+1} - V_k) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w V_t \tag{1}$$

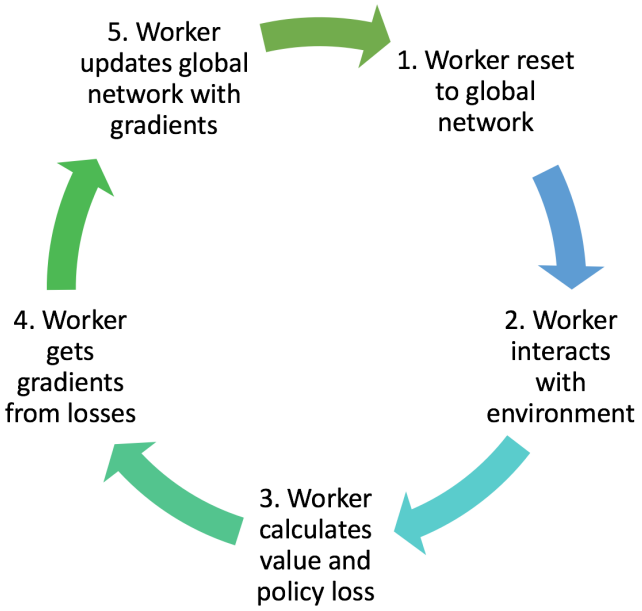
The change in weight w at time t is equal to the learning rate α times the difference between the value of the next time-step and the value of the current time step. This is multiplied by the sum of the exponentially discounted previous values of the partial derivative of the prediction with respect to w. The older partial derivatives effect on the update of the current w is exponentially decreased.
Notice that to find the error at time t, the V_{t+1} must be known. This means that when you update the weights, you are updating the previous iteration. Although that seems like a problem, we store that change in the sum part of the equation so the next Δw will be updated according to what happened previously. This is why it is called temporal-difference.

2.2 Implementation

2.3 A3C variant

Significance The introduction of A3C rendered DQN obsolete in 2014 because they are just better. A3C stands for Asynchronous Advantage Actor-Critic.

Asynchronous meaning it utilizes several agents to learn more efficiently. It can take advantage of multicore CPU's by putting one agent on one thread.



Advantage meaning the agent can determine how much better the action it took turned out to be than expected. This means it can make the policy focus on where the network's predictions are lacking. The advantage can be calculated by the equation $A = R - V(s)$ where R is the return.