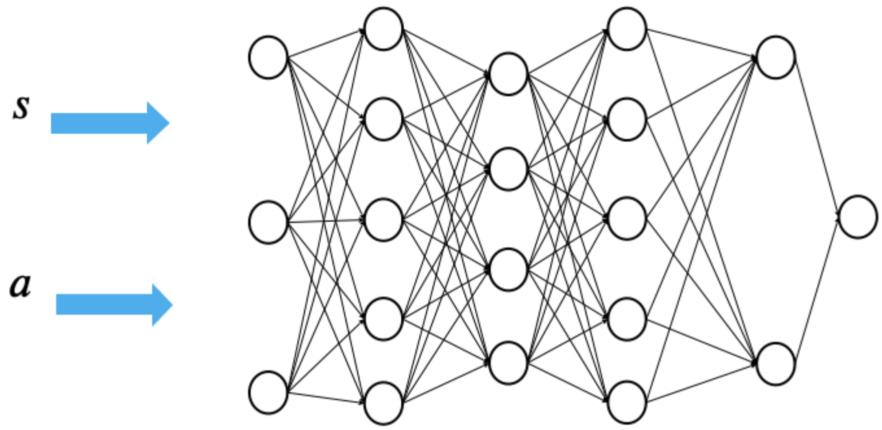
CAS CS 440
Lec 34

Policy Learning V: Reinforcement Learning V

1. The Advantage Function (A2C)
    a. What happened to using the Bellman equation?
    b. We can include it for $G_{\theta'}$
        i. $G_{\theta'}$ just estimates the “value” of the current choice
        ii. But we know choices are related!
            1. How good is a choice?
            2. $A(s_t, a_t) = Q_{\theta'}(s_t, a_t) - V_{\theta''}(s_t)$
        iii. This is called the advantage function
            1. Do we need two function-approximations, one for and one for
            2. No!
        iv. $Q(s_t, a_t) = \mathbb{E}\left[R(s_{t+1}) + \gamma V(s_{t+1})\right]$
        v. Therefore

$$A(s_t, a_t) = R(s_{t+1}) + \gamma V_{\theta''}(s_{t+1}) - V_{\theta''}(s_t)$$

    c. We only need $V_{\theta''}$ and $\pi_\theta$!
2. Offline Actor-Critic RL
    a. Could also do this offline
        i. Sample N trajectories (i.e. play N games) and record trajectories
        ii. Build supervised learning datasets
        iii. For k training iterations:
            1. Update $\pi_\theta$ and $G_{\theta'}$
        iv. Repeat
    b. Useful for scaling:
        i. A3C (Async A2C) = play lots of games in parallel (in eplore step), update after
3. NN Q-Function

a. 


b. NN takes place of tabular Q-function

c. How to apply updates?

   i. If we had a table:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(R(s) + \gamma \max_{a'} Q(s',a') - Q(s,a)\right)$$

   ii. Since we have a NN…cannot adjust Q value directly (like in above equation):

$$\theta_i \leftarrow \theta_i + \alpha\left(R(s) + \gamma \max_{a'} \hat{Q}_\theta(s',a') - \hat{Q}_\theta(s,a)\right)\frac{\partial \hat{Q}_\theta(s,a)}{\partial \theta_i}$$

d. Gradient descent!

   i. Error (loss function) =

$$R(s) + \gamma \max_{a'} \hat{Q}_\theta(s',a') - \hat{Q}_\theta(s,a)$$

4. Training a NN Q-function

   a. For now, let us assume exporation function f is randomized impl.

      i. With some probability p, ignore policy and choose random action

      ii. With remaining probability 1-p, follow policy

   b. Let us observe the transition (s, a, s') where we have recorded R(s) and R(s')

      i. Fix the policy (i.e. just use the NN):

         1. Calculate TD error by iterating over possible actions a' in s'

$$e = R(s) + \gamma \max_{a'} \hat{Q}_\theta(s',a') - \hat{Q}_\theta(s,a)$$

      ii. Backprop e through the network treating e like it is any other error function