Lecture 3

Topics covered so far:

- We started with finite horizon and model based algorithm based on dynamic programming.
- Later, we covered model based algorithm for infinite horizon task, value iteration and policy iteration
- We discussed value iteration based model free algorithm: Q-learning and SARSA.
- We discussed policy evaluation algorithms: Monte Carlo and TD(0).
- Function approximation based model free algorithms are covered.

To be covered in this lecture:

1) Implementation of Actor-Critic algorithm

Actor - Critic Algorithms

- Actor - Critic algorithms are based on policy iteration algorithm.
- Policy iteration algorithm has two stages:

1) Policy Evaluation

In this stage, we estimate the value function keeping the policy fixed. Value function represents the long term reward obtained for the fixed policy.

2) Policy Improvement

In this stage we use the value function to improve the current policy.

Policy evaluation and policy improvement is performed in alternation until convergence is achieved.

- Similarly every actor-critic algorithm has both policy evaluation and policy improvement steps
- Policy evaluation algorithm discussed in lecture 2 are used for performing policy evaluation in actor-critic algorithm.
- Policy gradient theorem provide the method to perform policy improvemt step for actor-critic algorithm.

Policy Gradient Theorem

- We define performance of a policy using the following objective function:

$$\eta(\pi) = \vec{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \quad \text{——— (1)}$$

- In (1), γ is the discount factor with $\gamma \in (0, 1)$.
- The objective function in (1) is called the discounted reward performance metric.
- Discounted reward is the long term discounted summation of reward
- There is another performance metric used in RL, known as average reward performance metric. (see (2))

$$\rho(\pi) = \lim_{N \to \infty} \frac{1}{N} \vec{E}\left[\sum_{t=0}^{N-1} r(s_t, a_t)\right]$$

- For this lecture, discounted reward performance metric will be considered.

- Let θ be the parameters of the policy π. The gradient of objective function $n(\pi)$ for stochastic policy is given as follows:

$$\nabla_\theta n(\pi) = \sum \rho^\pi(s) \sum \nabla_\theta \pi(a|s) Q^\pi(s,a)$$

1) Here, $\rho^\pi(s)$ is the long term discounted state visitation probability.

$$\rho^\pi(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P(S_t = s)$$

2) $\pi(a|s)$ is the stochastic policy

3) $Q^\pi(s,a)$ is the true Q-value function.

Policy improvement step :

1) Ideally we want to perform the following update for policy parameter θ.

$$\theta_{t+1} = \theta_t + \alpha_t \left(\sum_s \rho^{\pi}(s) \sum_a \nabla_\theta \pi(a|s) \, Q^{\pi}(s,a) \right)$$

2) However practically we use the following version:

$$\theta_{t+1} = \theta_t + \alpha_t \left(\frac{1}{N} \sum_{i=0}^{N-1} \nabla_\theta \log \pi(a_{t,i}|s_{t,i}) \times Q^{w}(s_{t,i}, a_{t,i}) \right)$$

Now we will understand how the pratical update rule is obtained from the exact update rule.

$$\sum_s p^{\pi}(s) \sum_a \nabla_\theta \pi(a|s) \; Q^{\pi}(s,a)$$

$$= \sum p^{\pi}(s) \sum_a \pi(a|s) \nabla_\theta \log \pi(a|s) \; Q^{\pi}(s,a)$$

(Take gradient of $\log \pi(a|s)$)

$$\approx \sum p^{\pi}(s) \sum \pi(a|s) \nabla_\theta \log \pi(a|s) \; Q^{w}(s,a)$$

(We replace $Q^{\pi}(s,a)$ with $Q^{w}(s,a)$ because $Q^{\pi}(s,a)$ is not available and we have to use approximation of Q-value which is $Q^{w}(s,a)$)

$$\approx \frac{1}{N} \sum_{i=0}^{N-1} \nabla_\theta \log(a_i|s_i) \; Q^{w}(s_i, a_i)$$

(Here we can not take the actual expection and hence we have to use empirical expectation)

- We can write the approximate objective function as:

$$\frac{1}{N} \sum_{i=0}^{N-1} \log \pi(a_i|s_i) \, Q^w(s_i, a_i)$$

- The gradient of the approximate objective function gives the approximate gradient:

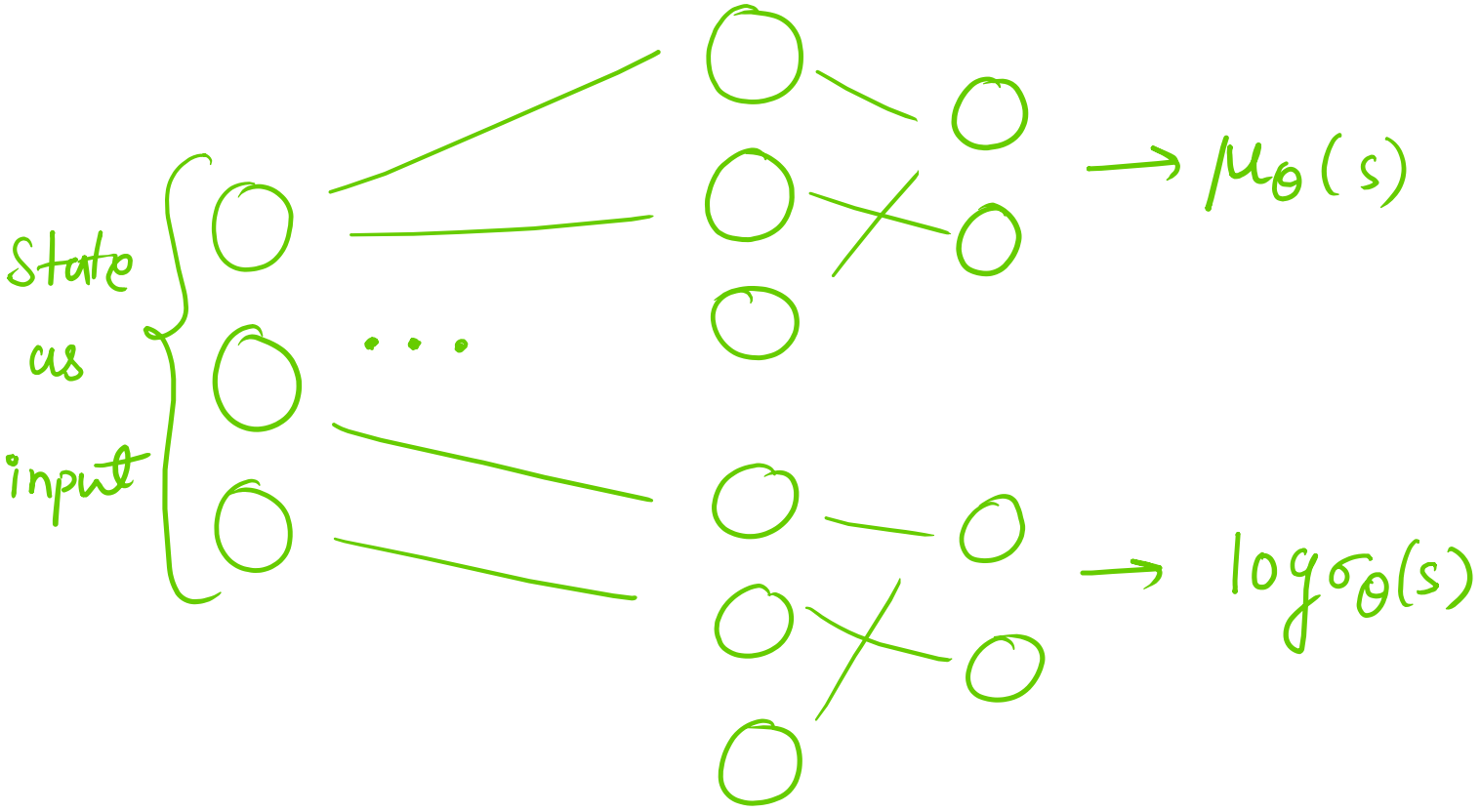
$$\frac{1}{N} \sum_{i=0}^{N-1} \nabla_\theta \log \pi(a_i|s_i) \, Q^w(s_i, a_i)$$

Representation of Policy:

- We assume that the policy follows gaussian distribution.

$$\pi(a|s) = \frac{1}{\sqrt{2\pi}\sigma_\theta(s)} \exp\left\{ -\frac{(\mu_\theta(s) - a)^2}{2\sigma_\theta(s)^2} \right\}$$

- Here $\mu_\theta(s)$ is the mean and $\sigma_\theta(s)$ is the standard deviation of the gaussian distribution for state s.
- θ is the neural network parameter.



Policy Evaluation step:

- Let the current policy be π.
- Our goal is to estimate the Q-value value function for policy π.
- Q-value follows the following bellman equation:

$$Q^{\pi}(s,a) = E\left[R(s,a) + \gamma Q^{\pi}(s',a') \mid s,a\right]$$

Q. Why do we want to estimate the Q-value?

→ $$Q^{\pi}(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t) \mid s_0 = s, a_0 = a\right]$$

Q-value for (s,a) pair represents the long term reward that the agent will get if it starts from state s and takes action a and then follow policy π.

Hence, Q-value helps us estimate the performance of policy π.

Once we know the current performance then only it can be improved.

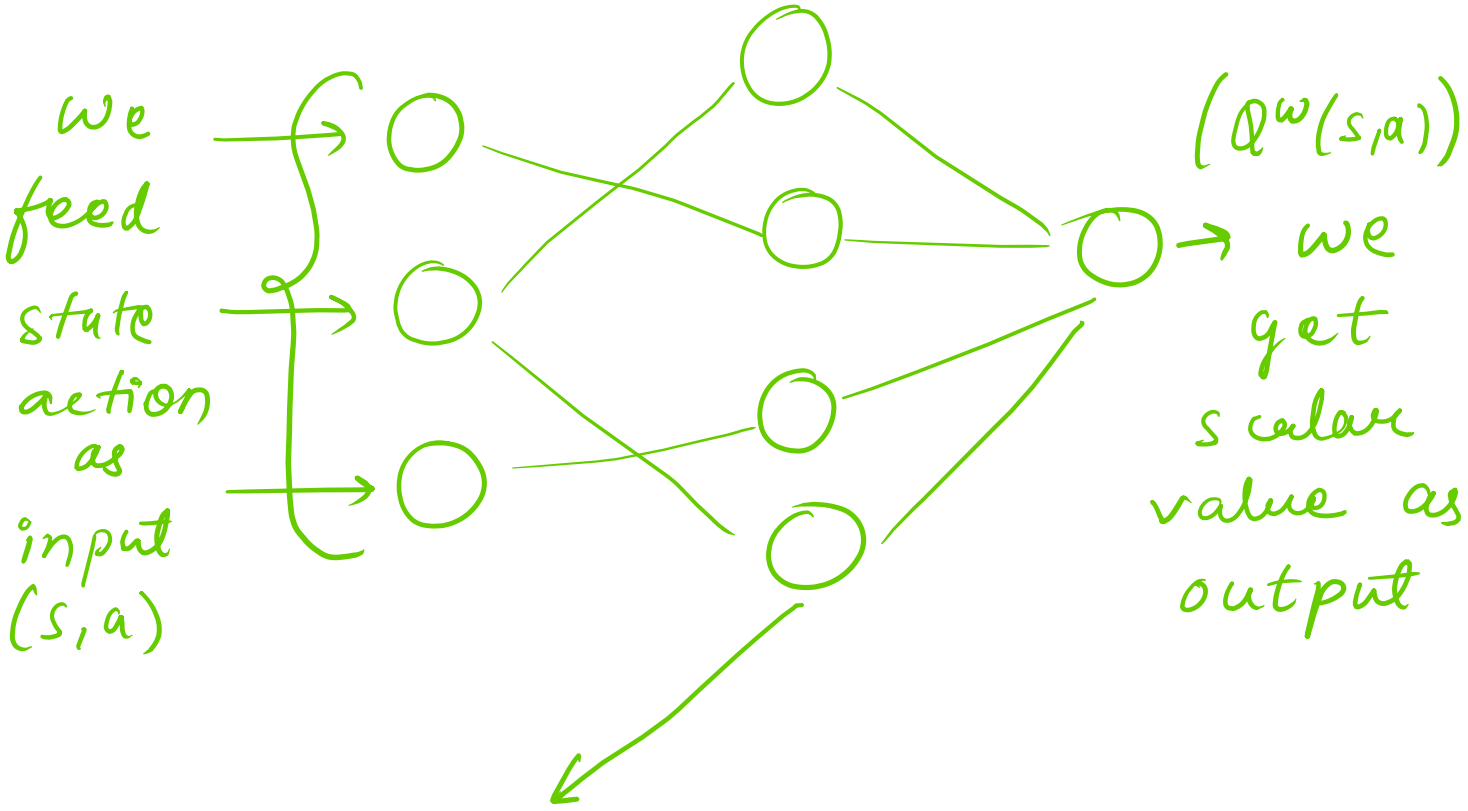
Hence policy evaluation step is indispensible for policy improvement.

Representation of Q-value

- We represent the Q-value either using a linear function or using a neural network.

- If we are using linear function, we write Q-value as $Q^{\omega}(s,a) = \phi(s,a)^T \omega$. $\phi(s,a)$ is the feature vector and for linear function representation we have to manully decide $\phi(s,a)$.

- We use neural network in order to avoid designing feature representation manually.



The last layer provides the feature representation.

- We take the help of bellman equation to find the best approximation of Q-value.

- We use the following error (Bellman error) to update Q-value parametre:

$$\mathcal{E}(w) = \sum_s \rho^\pi(s) \sum_a \pi(a|s) \Big(r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum \pi(a'|s') Q^w(s',a') - Q^w(s,a) \Big)^2$$

$$= E^\pi \left[\left(r(s,a) + \gamma Q^w(s',a') - Q^w(s,a) \right)^2 \right]$$

(Instead of actual expectation we evaluate empirical expectation)

$$\approx \frac{1}{N} \sum_{i=0}^{N-1} \Big(r(s_i, a_i) + \gamma Q^w(s_i', a_i') - Q^w(s_i, a_i) \Big)^2$$

Here, $\rho^\pi(s)$ is the long term discounted state visitation probability

a_i is the action corresponding to state s_i

s_i' is the next state obtained after s_i

and a_i' is the action taken in state s_i'.

- We have the following update rule for Q-value parameter ω:

$$\omega_{t+1} = \omega_t - \alpha_t \nabla_\omega \hat{\mathcal{E}}(\omega)$$

$$\hat{\mathcal{E}}(\omega) = \frac{1}{N} \sum_{i=0}^{N-1} \left(r(s_i, a_i) + \gamma Q^\omega(s_i', a_i') - Q^\omega(s_i, a_i) \right)^2$$

Note: Policy is called Actor and Q-value function is called Critic

Actor - Critic Algorithm

1. Initialize actor parameter θ, and critic parameter w.
   Initialize target critic parameter $\bar{w} \leftarrow w$.
2. Repeat the following for a few episode
   2.1 Initialize s_0, $a_0 \sim \pi(\cdot|s_0)$
   2.2 Repeat until the end of episode or maximum episode step limit.
      2.2.1 Observe r_t and s_{t+1}
      2.2.2 $a_{t+1} \sim \pi(\cdot|s_{t+1})$
      2.2.3 Store $\{s_t, a_t, s_{t+1}, a_{t+1}\}$ in a buffer
3. For N_c iterations perform below steps:
   3.1 Sample $\{s_i, a_i, s_i', a_i'\}_{i=0}^{N-1}$
   3.2 Calculate error $\hat{\mathcal{E}}(w)$ as

$$\hat{\mathcal{E}}(\omega) = \frac{1}{N}\sum_{i=0}^{N-1}\left(r(s_i,a_i) + \gamma\, Q^{\bar{\omega}}(s_i',a_i') - Q^{\omega}(s_i,a_i)\right)^2$$

3.3 Update critic parameter as:

$$\omega_{t+1} = \omega_t - \alpha_t \nabla_\omega \hat{\mathcal{E}}(\omega)$$

4. For N_a iterations perform below steps:

4.1 sample $\{s_i, a_i, s_i', a_i'\}_{i=0}^{N-1}$

4.2 Calculate $\hat{n}(\theta)$ as:

$$\hat{n}(\theta) = \frac{1}{N}\sum_{i=0}^{N-1} \log \pi(a_i|s_i)\, Q^{\omega}(s_i,a_i)$$

4.3 Update actor parameter as:

$$\theta_{t+1} = \theta_t - \beta_t \nabla_\theta \hat{n}(\theta)$$

5. With certain frequency update target critic as:

$$\bar{\omega}_{t+1} = \bar{\omega}_t(1-\tau) + \tau\, \omega_{t+1}$$

6. Repeat step 2 onwards

Remarks:

1. Actor Critic algorithm resembles policy iteration because it involves policy evaluation and policy improvement step.
2. AC algorithm described here is on-policy, model free and uses stochastic policy.
3. Target critic is used to stabalize training.
4. Here policy evaluation is based on TD(0) but Monte Carlo policy evaluation could also be used here.