TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2020

Reinforcement Learning

Value Iteration

Definition of Reinforcement Learning

RL is defined by the following properties:

- An environment with **state**.
- State changes are influenced by **sequential decisions**.
- Reward (or loss) depends on **making decisions** that lead to **desirable states**.

Reinforcement Learning Examples

- Board games (chess or go)
- Atari Games (pong)
- Robot control (driving)
- Dialog
- Life

Policies

A policy is a way of behaving.

Formally, a (nondeterministic) policy maps a state to a probability distribution over actions.

$$\pi(a_t|s_t) \quad \text{probability of action } a_t \text{ in state } s_t$$

Imitation Learning

Construct a training set of state-action pairs (s, a) from experts.

Define stochastic policy $\pi_\Phi(s)$.

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{(s,a)\sim\mathrm{Train}} \; -\ln \pi_\Phi(a \mid s)$$

This is just cross-entropy loss where we think of a as a "label" for s.

Dangers of Imperfect Imitation Learning

Perfect imitation learning would reproduce expert behavior. Imitation learning is **off-policy** — the state distribution in the training data is different from that defined by the policy being learned.

Imitating experts, such as expert fire eaters, can be dangersous. “Don’t try this at home”.

Also, it is difficult to exceed expert performance by imitating experts. But this can happen.

Markov Decision Processes (MDPs)

An MDP consists of a set $\mathcal{S}$ of states, a set $\mathcal{A}$ of allowed actions, a reward function R and a next-state probability function P_T. We will use the following notation.

$s_t \in \mathcal{S}$ is the state at time t

$a_t \in \mathcal{A}$ is the action taken at time t.

$r_t = R(s_t, a_t) \in \mathbb{R}$ is the reward at time t

$P_T(s_{t+1}|s_t, a_t)$ is the probability of s_{t+1} given s_t and a_t.

The function $R(s, a)$ can allow for a cost of the action a.

Optimizing Reward

In RL we maximize reward rather than minimize loss.

$$\pi^* = \operatorname*{argmax}_\pi \; R(\pi)$$

$$R(\pi) \;=\; E_\pi \; \textstyle\sum_{t=0}^{T} \; r_t \qquad \text{episodic reward (go)}$$

$$\text{or} \;\; E_\pi \; \textstyle\sum_{t=0}^{\infty} \; \gamma^t r_t \qquad \text{discounted reward (financial planning)}$$

$$\text{or} \;\; \lim_{T\rightarrow\infty} \; \tfrac{1}{T} \; \textstyle\sum_{t=0}^{T} \; r_t \quad \text{asymptotic average reward (driving)}$$

The Value Function

For discounted reward:

$$V^\pi(s) = E_\pi \sum_t \gamma^t r_t \mid \pi,\ s_0 = s$$

$$V^*(s) = \sup_\pi V^\pi(s)$$

$$\pi^*(a|s) = \operatorname*{argmax}_a R(s,a) + \gamma E_{s' \sim P_T(s'|s,a)} V^*(s')$$

$$V^*(s) = \max_a R(s,a) + \gamma E_{s' \sim P_T(s'|s,a)} V^*(s')$$

Value Iteration

Suppose the state space and action space are finite.

In that case we can do value iteration.

$$V_0(s) = 0$$

$$V_{i+1}(s) = \max_a \; R(s,a) + \gamma E_{s' \sim P_T(\cdot|s,a)} \; V_i(s')$$

If all rewards are non-negative then

$$V_{i+1}(s) \geq V_i(s) \quad V_i(s) \leq V^*(s) \quad \text{so} \; \lim_{i \to \infty} V_i(s) \; \text{exists}$$

Value Iteration

Theorem: **For discounted reward**

$$V_\infty(s) \doteq \lim_{i \to \infty} V_i(s) = V^*(s)$$

Proof

$$
\begin{aligned}
\Delta &\doteq \max_s \quad V^*(s) - V_\infty(s) \\
&= \max_s \begin{pmatrix} \max_a R(s,a) + E_{s'|a} \gamma V^*(s') \\ -\max_a R(s,a) + E_{s'|a} \gamma V_\infty(s') \end{pmatrix} \\
&\leq \max_s \max_a \begin{pmatrix} R(s,a) + E_{s'|a} \gamma V^*(s') \\ -R(s,a) + E_{s'|a} \gamma V_\infty(s') \end{pmatrix} \\
&= \max_s \max_a \; E_{s'|a} \, \gamma (V^*(s') - V_\infty(s)) \\
&\leq \gamma \Delta
\end{aligned}
$$

END