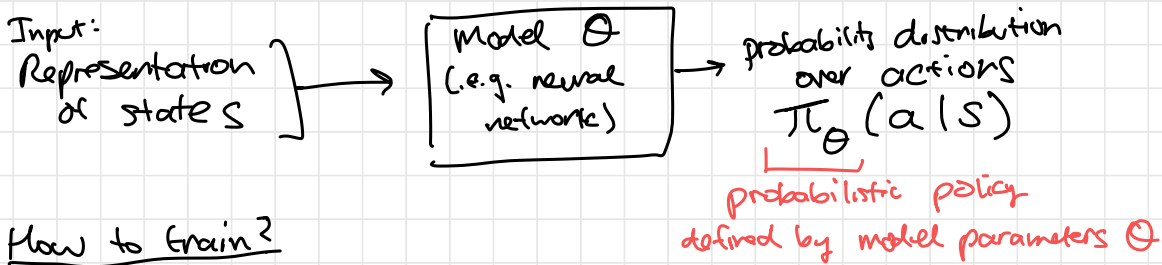


11/16/2023: Policy Gradient Methods

Previously: Q-Learning: Given (s, a) , predict $Q_{opt}(s, a)$
"regression"

Today: Policy Gradient: Given s , predict a
"classification"



How to train?

- Normal supervised learning requires knowing best action at given states as training data \times
Not known for any state
- Policy gradient: train $\pi_{\theta}(a|s)$ to achieve high total rewards

Want to maximize value of the policy π_{θ}

$$V(\theta) = \sum_{\text{trajectories } z} P(z; \theta) \cdot R(z)$$

Expected sum of rewards when using policy $\pi_{\theta}(a|s)$

Trajectory = possible sequences of $[s_1, a_1, r_1, s_2, a_2, r_2, s_3, \dots]$

Prob of z happening

Reward achieved $= \sum_{t=1}^T r_t$

Plan: $V(\theta)$ is our training objective, maximize with gradient ascent

What is $\nabla_{\theta} V(\theta)$?

$$\nabla_{\theta} V(\theta) = \sum_{\text{traj's } z} \nabla_{\theta} P(z; \theta) \cdot R(z)$$

Sum over exponentially many trajectories - infeasible

Key trick: $\nabla_{\theta} \log(P(z; \theta)) = \frac{1}{P(z; \theta)} \nabla_{\theta} P(z; \theta)$

$$\Leftrightarrow \nabla_{\theta} P(z; \theta) = P(z; \theta) \nabla_{\theta} \log P(z; \theta)$$

Plug this in to $\nabla_{\theta} V(\theta)$:

$$\begin{aligned} \nabla_{\theta} V(\theta) &= \sum_{\text{traj's } z} P(z; \theta) \nabla_{\theta} \log P(z; \theta) \cdot R(z) \\ &= \mathbb{E}_{\theta} \left[\nabla_{\theta} \log P(z; \theta) \cdot R(z) \right] \end{aligned}$$

Expected value of... this quantity

Estimate this by sampling trajectories with $\pi_{\theta}(a|s)$ and taking average of $\nabla_{\theta} \log P(z; \theta) \cdot R(z)$

What is $\nabla_{\theta} \log P(z; \theta)$?

$$\log P(z; \theta) = \underbrace{\log P(s_1)}_{\text{start state}} + \underbrace{\log \pi_{\theta}(a_1 | s_1)}_{\text{policy}} + \underbrace{\log T(s_1, a_1, s_2)}_{\text{transitions}}$$

$$\{s_1, a_1, r_1, s_2, \dots\} + \underbrace{\log \pi_{\theta}(a_2 | s_2)}_{\text{policy}} + \dots$$

Don't depend on θ
So ∇_{θ} is 0

$$\text{So } \nabla_{\theta} \log P(z; \theta) = \sum_{t=1}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)$$

Basic Policy Gradient Algorithm:

Initialize θ randomly

For each episode:

- Sample trajectory z using $\pi_{\theta}(a|s)$
- Update:

$$\theta \leftarrow \theta + \eta R(z) \underbrace{\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)}_{\approx \nabla_{\theta} V(\theta)}$$