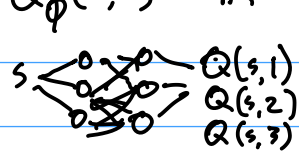
HW 4 - 1 alg from notebook
1 alg implemented

## Last Time

DQN ← Experience Replay
Freezing target Q network
NN Architecture

$Q_\phi(s,a) \to \mathbb{R}$




DPG ~ Natural Gradient

Rainbow

## Today

Actor-Critic
Exploration
Recap of RL

---

Actor-Critic

$\pi_\theta$ $Q_\phi$ or $A_\phi$ or $V_\phi$

$$\nabla_\theta U(\theta) = E\left[\sum_{k=1}^{d} \nabla_\theta \log \pi_\theta(a^{(k)}|s^{(k)})\, \gamma^{k-1}\left(r^{(k)}_{to\text{-}go} - r_{base}(s^{(k)})\right)\right]$$

$Q^{\pi_\theta}(s,a)$ $V^{\pi_\theta}(s)$ $A^{\pi_\theta}(s,a)$

$$A^{\pi_\theta}(s,a) = E\left[r + \gamma V^{\pi_\theta}(s') - V^{\pi_\theta}(s)\right]$$

$$\ell(\phi) = \frac{1}{2} E_s\left[\left(V_\phi(s) - V^{\pi_\theta}(s)\right)^2\right]$$

GAE: Generalized Advantage Estimation

$$A^{\pi_\theta}(s,a) \leftarrow \begin{array}{l} E\left[r + \gamma V_\phi(s') - V_\phi(s)\right] \leftarrow \text{Bias} \\ E\left[r^k + \gamma r^{(k+1)} + \gamma^2 r^{(k+2)} \dots \gamma^{d-1} r^{(d)} - r_{base}(s)\right] \leftarrow \text{Variance} \\ E\left[r^{(k)} + \gamma r^{(k+1)} + \dots \gamma^m V^{\pi_\theta}(s^m) - V^{\pi_\theta}(s)\right] \end{array}$$

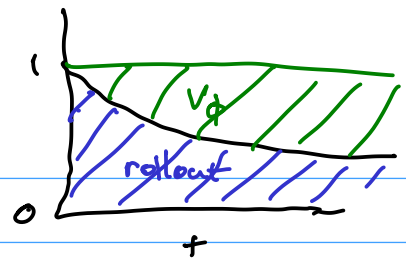
$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) = \hat{A}_t$$

$$\hat{A}^{(k)}(s,a) = E\left[\sum_{l=1}^{k} \gamma^{(l-1)} \delta_l\right]$$

$$\hat{A}^{GAE}(s,a) = \mathbb{E}\left[\sum_{l=1}^{k} (\gamma\lambda)^{(k-1)}\delta\right]$$

$\lambda \in [0,1]$ $\quad$ $\lambda = 1$ : rollout

$\lambda = 0$ : value




---

Acronyms to know

A3C Asynchronous Advantage Actor Critic

SAC $\pi(a|s) \propto e^{Q(s,a)}$

$$\pi^*_{Maxent} = \text{argmax}\, \mathbb{E}\left[\sum_t r_t + H(\pi(a|s_t))\right]$$

↑ Entropy

---

Continuous A $\quad$ $\pi(a|s)$

DPG Deterministic Policy Gradient

$a = \pi_\theta(s)$ $\quad$ $Q_\phi(s,a)$

training Q → $L(\phi) = \frac{1}{2}\mathbb{E}\left[\left(r + \gamma Q_\phi(s', \pi_\theta(s')) - Q_\phi(s,a)\right)^2\right]$

train $\pi_\theta$

training θ → $U(\theta) = \mathbb{E}_s\left[Q_\phi(s, \pi_\theta(s))\right]$

$$\nabla U(\theta) = \mathbb{E}\left[\nabla_\theta Q_\phi(s, \pi_\theta(s))\right]$$

$$= \mathbb{E}\left[\nabla_\theta \pi_\theta(s)\, \nabla_a Q_\phi(s,a)\big|_{a=\pi_\theta(s)}\right]$$

↑ new

Exploration? add Gaussian Exploration

---

Alpha-Zero $\quad$ Actor-Critic with MCTS

Learn $\pi_\theta(a|s)$ $\quad$ $V_\phi(s)$

↑ Guide Tree Search $\quad$ ↑ replace rollouts

$$a = \text{argmax}\, Q(s,a) + c\,\pi_\theta(a|s)\frac{\sqrt{N(s)}}{1+N(s,a)}$$

train θ

$\pi_{MCTS}(a|s) \propto N(s,a)^\eta$

Cross entropy loss $\quad$ $L(\theta) = -\mathbb{E}_s\left[\sum_a \pi_{MCTS}(a|s) \log \pi_\theta(a|s)\right]$

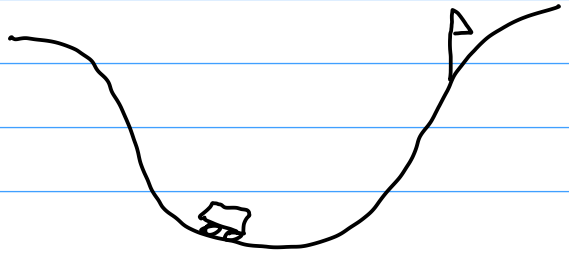
train $\phi$

$$V_{MCTS}(s) = \max_a Q(s,a)$$

$$\ell(\phi) = \frac{1}{2} \mathbb{E}_s \left[ \left( V_\phi(s) - V_{MCTS}(s) \right)^2 \right]$$

---

## Advanced Exploration



Montezuma's Revenge

**Breakout Rooms** How would you make an RL algorithm to handle MR

---

- Add incremental Rewards "Reward Shaping"
- Start state closer to goal
- Prune MCTS early
- Depth in Pyramid - Remembering

---

$$R^+(s,a) = R^{\text{extrinsic}}(s,a) + B^{\text{intrinsic}}(s,a)$$

$$B(N(s))$$

Continuous States

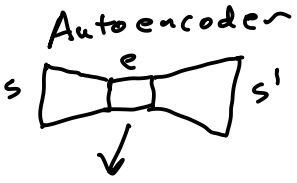
fit $p_\theta(s)$ "pseudo-count"

What Bonus

UCB $B(s,a) = c\sqrt{\frac{\log N(s)}{N(s,a)}}$

$B(s) = \sqrt{\frac{T}{N(s)}}$ ← "Unifying Count-Based Exp." Bellamare e.a.

$B(s) = \frac{1}{N(s)}$




"hash"

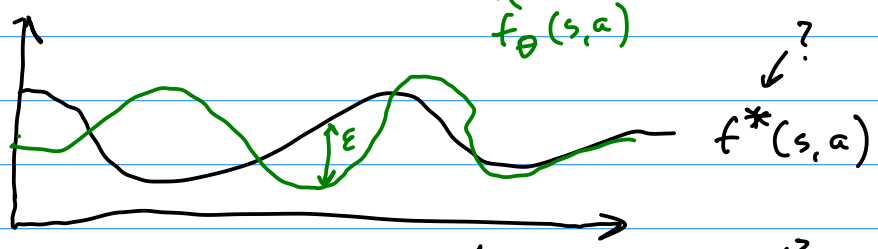
- # Exploration

compress s into k-bit code $\phi(s)$ then count $N(\phi(s))$

- Use Classifier "EX2"

$p_\theta(s) = \frac{1 - D_s(s)}{D_s(s)}$ $D_s(s)$ prop that state returns true

---




use $\varepsilon(s,a) = \|\hat{f}(s,a) - f^*(s,a)\|^2$ as bonus

→ $f^*(s,a) = s'$ "Curiosity"

→ $f^*(s,a) = f_\phi(s,a)$ where $\phi$ is a random NN

Good → "Random Network Distillation"

---

Thompson Sampling

Maintain distribution over Q ← hard

Sample $\hat{Q}$

$a = \text{argmax}\ \hat{Q}(s,a)$

- Bootstrapping

Multiple Q-networks

- Dropout

---

Information Gain VIME

---

Go-Explore

First Return, Explore

# Recap of RL

Exploration / Exploitation - Bandits, Advanced Exp.
Credit Assignment - Value Function, Eligibility / GAE
Generalization - Neural Network