TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

REINFORCE and Actor-Critic Algorithms

The REINFORCE Algorithm

Williams, 1992

REINFORCE is a Policy Gradient Algorithm

We assume a parameterized policy $\pi_\Phi(a|s)$.

$\pi_\Phi(a|s)$ is normalized while $Q_\Phi(s,a)$ is not.

Policy Gradient Theorem (Episodic Case)

$$\Phi^* = \underset{\Phi}{\text{argmax}}\ E_{\pi_\Phi}\ R$$

$$\nabla_\Phi\ E_{\pi_\Phi}\ R = \sum_{s_0,a_0,s_1,a_1,\ldots,s_T,a_T} \nabla_\Phi P(s_0, a_0, s_1, a_1, \ldots, s_T, a_T)\ R$$

$$\begin{aligned}
\nabla_\Phi\ P(\ldots)R = {} & P(S_0)\nabla_\Phi\ \pi(a_0)P(s_1)\pi(a_1)\cdots P(s_T)\pi(a_T)\ R \\
& + P(S_0)\pi(a_0)P(s_1)\nabla_\Phi\ \pi(a_1)\cdots P(s_T)\pi(a_T)\ R \\
& \vdots \\
& + P(S_0)\pi(a_0)P(s_1)\pi(a_1)\cdots P(s_T)\nabla_\Phi\ \pi(a_T)\ R
\end{aligned}$$

$$= P(\ldots)\left(\sum_t \frac{\nabla_\Phi\ \pi_\Phi(a_t)}{\pi_\Phi(a_t)}\right) R$$

Policy Gradient Theorem (Episodic Case)

$$\nabla_\Phi \, P(\ldots) R = \, P(\ldots) \left(\sum_t \, \frac{\nabla_\Phi \, \pi_\Phi(a_t|s_t)}{\pi_\Phi(a_t|s_t)} \right) R$$

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \, E_{\pi_\Phi} \, \left(\sum_t \, \nabla_\Phi \, \ln \pi_\Phi(a_t|s_t) \right) R$$

Policy Gradient Theorem

$$
\begin{aligned}
& \nabla_\Phi \; E_{\pi_\Phi} \; R \\
= \; & E_{\pi_\Phi} \; \left(\sum_t \; \nabla_\Phi \; \ln \pi_\Phi(a_t|s_t) \right) \quad R \\
= \; & E_{\pi_\Phi} \; \left(\sum_t \; \nabla_\Phi \; \ln \pi_\Phi(a_t|s_t) \right) \left(\sum_t r_t \right) \\
= \; & E_{\pi_\Phi} \; \sum_{t,t'} \nabla_\Phi \; \ln \pi_\Phi(a_t|s_t) \;\; r_{t'}
\end{aligned}
$$

Policy Gradient Theorem

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \sum_{t,t'} \, E_{s_t,a_t,r_{t'}} \;\; \nabla_\Phi \;\; \ln \pi_\Phi(a_t|s_t) \;\; r_{t'}$$

For $t' < t$ we have

$$\begin{aligned}
E_{r_{t'},s_t,a_t} \;\; r_{t'} \nabla_\Phi \;\; \ln \pi_\Phi(a_t|s_t) \;\; &= \;\; E_{r_{t'},s_t} \; r_{t'} \sum_{a_t} \; \pi_\Phi(a_t|s_t) \; \nabla_\Phi \;\; \ln \pi_\Phi(a_t|s_t) \\
&= \;\; E_{r_{t'},s_t} \; r_{t'} \sum_{a_t} \; \nabla_\Phi \; \pi_\Phi(a_t|s_t) \\
&= \;\; E_{r_{t'},s_t} \; r_{t'} \; \nabla_\Phi \sum_{a_t} \; \pi_\Phi(a_t|s_t) \\
&= \;\; 0
\end{aligned}$$

REINFORCE

$$\nabla_\Phi \, E_{\pi_\Phi} \, R \quad = \quad E_{\pi_\Phi} \sum_{t,\, t' \geq t} \left(\nabla_\Phi \, \ln \pi_\Phi(a_t | s_t) \right) \, r_{t'}$$

Sampling runs and computing the above sum over t and t' is Williams' REINFORCE algorithm.

Optimizing Discrete Decisions with Non-Differentiable Loss

The REINFORCE algorithm is used generally for non-differentiable loss functions.

For example error rate and BLEU score are non-differentiable — they are defined on the result of discrete decisions.

$$\Phi^* = \underset{\Phi}{\text{argmax}} \; E_{w_1,\ldots,w_n \sim P_\Phi} \; \text{BLEU}$$

The Variance Issue

REINFORCE typically suffers from high variance of the gradient samples requiring very small learning rates and very long convergence times.

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \sum_{t,\, t' \geq t} E_{s_t, a_t, r_{t'}} \, (\nabla_\Phi \, \ln \pi_\Phi(a_t | s_t)) \quad r_{t'}$$

We have to consider

- the variation over $r_{t'}$ given s_t, a_t
- the variation over the choice of s_t, a_t

Reducing the Variance over $r_{t'}$

The Policy Gradient Theorem

"Policy Gradient Methods for Reinforcement Learning with Function Approximation" Sutton, McAllester, Singh, Mansour, 2000, cited by 2,841 as of March 2020.

"Actor-Critic Algorithms", Konda and Tsitsilas, 2000, cited by 776.

These two papers both appeared at NeurIPS 2000 and are essentially identical. The first is just easier to read.

Reducing the Variance over $r_{t'}$

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \sum_{t,t'} \, E_{s_t,a_t,r_{t'}} \, \nabla_\Phi \, \ln \pi_\Phi(a_t|s_t) \;\; r_{t'}$$

$$= \sum_{t} \, E_{s_t,a_t} \, \nabla_\Phi \, \ln \pi_\Phi(a_t|s_t) \;\; \sum_{t' \geq t} E_{r_{t'} \mid s_t,a_t} \, r_{t'}$$

$$= \, E_{\pi_\Phi} \, \sum_{t} \, (\nabla_\Phi \, \ln \pi_\Phi(a_t|s_t)) \;\; Q^{\pi_\Phi}(s_t, a_t)$$

$$Q^\pi(s,a) \, = \, E_\pi \, \sum_{t} \, r_t \mid s_0 = s, \;\; a_0 = a$$

Reducing the Variance over $r_{t'}$

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \sum_t \, E_{s_t, a_t} \, \left(\nabla_\Phi \, \ln \pi_\Phi(a_t | s_t) \right) Q^{\pi_\Phi}(s_t, a_t)$$

The point is that we can now approximate Q^{π_Φ} with neural network Q_Φ where the networks π_Φ and Q_Φ can use different, perhaps overlapping, parts of Φ.

We reduced the variance at the cost of approximating the expected future reward.

The Actor-Critic Algorithm

$$\nabla_\Phi \, E_{\pi_\Phi} \, R \approx E_{\pi_\Phi} \sum_t \left(\nabla_\Phi \, \ln \pi_\Phi(a_t|s_t)\right) \; Q_\Phi(s_t, a_t)$$

π_Φ is the "actor" and Q_Φ is the "critic"

The Actor-Critic Algorithm

$$\nabla_\Phi \; E_{\pi_\Phi} \; R \approx \; E_{\pi_\Phi} \; \sum_t \; (\nabla_\Phi \; \ln \pi_\Phi(a_t|s_t)) \quad Q_\Phi(s_t, a_t)$$

We can sample an episode and then do

$$\Phi \; \texttt{+=} \; \sum_t \eta_1 \, (\nabla_\Phi \; \ln \pi_\Phi(a_i|s_i)) \, Q_\Phi(s_t, a_t)$$

$$\Phi \; \texttt{-=} \; \sum_t \eta_2 \; \nabla_\Phi \; \left(Q_\Phi(s_t, a_t) - \sum_{t' \geq t} r_{t'} \right)^2$$

The two updates typically apply to different (but perhaps overlapping) subsets of the parameters Φ.

Reducing the Variance over s_t

Thoerem:

$$\nabla_\Phi \, E_{\pi_\Phi} \, R = \sum_t \, E_{s_t, a_t} \, \left(\nabla_\Phi \, \ln \pi_\Phi(a_t|s_t)\right) \left(Q^{\pi_\Phi}(s_t, a_t) - V^{\pi_\Phi}(s_t)\right)$$

$$V^{\pi_\Phi}(s) = E_{a \sim \pi_\Phi(a|s)} \, Q^{\pi_\Phi}(s, a)$$

$Q^{\pi_\Phi}(s, a) - V^{\pi_\Phi}(s)$ is the "advantage" of deterministically using a rather than sampling an action.

Proof

We have the following for any function $V(s)$ of states.

$$
\begin{aligned}
& E_{s_t,a_t} \ \ (\nabla_\Phi \ \ln \pi_\Phi(a_t|s_t))\, V(s_t) \\
= \ & E_{s_t} \sum_{a_t} (\pi_\Phi(a_t|s_t) \ \nabla_\Phi \ \ln \pi_\Phi(a_t|s_t))\, V(s_t) \\
= \ & E_{s_t} \sum_{a_t} (\nabla_\Phi \pi_\Phi(a_t|s_t))\, V(s_t) \\
= \ & E_{s_t}\, V(s_t) \nabla_\Phi \sum_{a_t} \ \pi_\Phi(a_t|s_t) \ = \ 0
\end{aligned}
$$

Advantage-Actor-Critic Algorithm

$$\nabla_\Phi \, E_{\pi_\Phi} \, R \approx \, E_{\pi_\Phi} \sum_t \, (\nabla_\Phi \, \ln \pi_\Phi(a_t|s_t)) \;\; (Q_\Phi(s_t, a_t) - V_\Phi(s_t))$$

We can sample an episode and then do

$$\Phi \; \texttt{+=} \sum_t \eta_1 \, (\nabla_\Phi \, \ln \pi_\Phi(a_i|s_i)) \, (Q_\Phi(s_t, a_t) - V_\Phi(s_t))$$

$$\Phi \; \texttt{-=} \sum_t \eta_2 \, \nabla_\Phi \, \left(Q_\Phi(s_t, a_t) - \sum\nolimits_{t' \geq t} \, r_{t'} \right)^2$$

$$\Phi \; \texttt{-=} \sum_t \eta_3 \, \nabla_\Phi \, \left(V_\Phi(s_t) - Q_\Phi(s_t, a) \right)^2$$

Asynchronous Methods for Deep RL (A3C)

Mnih et al., Arxiv, 2016 (Deep Mind)

$\tilde{\Phi} = \Phi$ (retrieve global Φ)

using policy $\pi_{\tilde{\Phi}}$ compute $s_t, a_t, r_t, \ldots, s_{t+K}, a_{t+K}, r_{t+K}$

$$R_i = \sum_{\delta=0}^{D} \gamma^{i+\delta} r_{(i+\delta)}$$

$$\Phi \texttt{ += } \eta \sum_{i=t}^{t+K-D} \left(\nabla_{\tilde{\Phi}} \ln \pi_{\tilde{\Phi}}(a_i|s_i)\right) \left(R_i - V_{\tilde{\Phi}}(s_i)\right)$$

$$\Phi \texttt{ -= } \eta \sum_{i=t}^{t+K-D} \nabla_{\tilde{\Phi}} \left(V_{\tilde{\Phi}}(s_i) - R_i\right)^2$$

Issue: Policies must be Exploratory

The optimal policy is deterministic — $a(s) = \text{argmax}_a \, Q(s, a)$.

However, a deterministic policy never samples alternative actions.

Typically one forces a random action some small fraction of the time.

Issue: Discounted Reward

DQN and A3C use discounted reward on episodic or long term problems.

Presumably this is because actions have near term consequences.

This should be properly handled in the mathematics, perhaps in terms of the mixing time of the Markov process defined by the policy.

Observation: Continuous Actions are Differentiable

In problems like controlling an inverted pendulum, or robot control generally, a continuous loss can be defined and the gradient of loss of with respect to a deterministic policy exists.

More Videos

https://www.youtube.com/watch?v=g59nSURxYgk
https://www.youtube.com/watch?v=rAai4QzcYbs

END