# Deep Reinforcement Learning

Overview of main articles
Part 2. Policy gradient algorithms


Sergey Ivanov


September 23, 2018


MSU

# Table of contents i

# Basic policy gradient methods

# Direct optimization

Recall RL goal:

$$\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \to \max_{\theta}$$

# Direct optimization

Recall RL goal:

$$\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \to \max_{\theta}$$

Let's optimize our goal directly!

$$\nabla_\theta \mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \quad \text{— ?}$$

# Direct optimization

Recall RL goal:

$$\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \to \max_{\theta}$$

Let's optimize our goal directly!

$$\nabla_{\theta}\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \quad \text{— ?}$$

# Direct optimization

Recall RL goal:

$$\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \to \max_{\theta}$$

Let's optimize our goal directly!

$$\nabla_{\theta}\mathbb{E}_{\pi(\theta)}\mathbb{E}_{\mathcal{T}} R \quad \text{— ?}$$

**Options:**

* Metaheurisics
* Log-derivative trick[1].

[1] aka REINFORCE

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x \sim \pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx$$

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x\sim\pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx = \Big\{$$

$$\Big\} = \int \nabla_\theta \pi(x,\theta) f(x) dx$$

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x \sim \pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx = \Big\{$$

$$\Big\} = \int \nabla_\theta \pi(x,\theta) f(x) dx$$

**Problem:** and what?

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x\sim\pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx = \left\{ \quad \right\} = \int \nabla_\theta \pi(x,\theta) f(x) dx$$

**Problem:** and what?

### Log-derivative trick

$$\nabla_\theta \pi(\theta) = \pi(\theta) \nabla_\theta \log \pi(\theta)$$

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x \sim \pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx = \Big\{$$

$$\Big\} = \int \nabla_\theta \pi(x,\theta) f(x) dx =$$

**Problem:** and what?

**Log-derivative trick**

$$\nabla_\theta \pi(\theta) = \pi(\theta) \nabla_\theta \log \pi(\theta)$$

$$= \int \pi(x,\theta) \nabla_\theta \log \pi(x,\theta) f(x) dx$$

# Stochasitc estimators optimization via log-derivative trick

$$\nabla_\theta \mathbb{E}_{x \sim \pi(x,\theta)} f(x) = \nabla_\theta \int \pi(x,\theta) f(x) dx = \Big\{$$

$$\Big\} = \int \nabla_\theta \pi(x,\theta) f(x) dx =$$

**Problem:** and what?

**Log-derivative trick**

$$\nabla_\theta \pi(\theta) = \pi(\theta) \nabla_\theta \log \pi(\theta)$$

$$= \int \pi(x,\theta) \nabla_\theta \log \pi(x,\theta) f(x) dx = \mathbb{E}_{x \sim \pi(x,\theta)} \nabla_\theta \log \pi(x,\theta) f(x)$$

# From importance sampling point of view

Recall Importance Sampling. For arbitrary distribution $\phi(x)$:

$$\mathbb{E}_{x \sim \pi(x,\theta)} f(x) = \mathbb{E}_{x \sim \phi(x)} \frac{\pi(x,\theta)}{\phi(x)} f(x)$$

# From importance sampling point of view

Recall Importance Sampling. For arbitrary distribution $\phi(x)$:

$$\mathbb{E}_{x\sim\pi(x,\theta)} f(x) = \mathbb{E}_{x\sim\phi(x)} \frac{\pi(x,\theta)}{\phi(x)} f(x)$$

# From importance sampling point of view

Recall Importance Sampling. For arbitrary distribution $\phi(x)$:

$$\mathbb{E}_{x\sim\pi(x,\theta)} f(x) = \mathbb{E}_{x\sim\phi(x)} \frac{\pi(x,\theta)}{\phi(x)} f(x)$$

Let's set $\phi(x) \equiv \pi(x,\theta)$:

# From importance sampling point of view

Recall Importance Sampling. For arbitrary distribution $\phi(x)$:

$$\mathbb{E}_{x\sim\pi(x,\theta)} f(x) = \mathbb{E}_{x\sim\phi(x)} \frac{\pi(x,\theta)}{\phi(x)} f(x)$$

Let's set $\phi(x) \equiv \pi(x,\theta)$:

$$\nabla_\theta \mathbb{E}_{x\sim\phi(x)} \frac{\pi(x,\theta)}{\phi(x)} f(x) = \mathbb{E}_{x\sim\phi(x)} \frac{\nabla_\theta \pi(x,\theta)}{\phi(x)} f(x)$$

**Note:** that is the same gradient as with log-derivative trick[2].

[2] really? Could it even happen otherwise?

# REINFORCE

Let's apply log-derivative trick to our goal!

$$\nabla_\theta \mathbb{E}_{\pi(\theta)} \mathbb{E}_{\mathcal{T}} R = \mathbb{E}_{\pi(\theta)} \nabla_\theta \log \pi(\theta) \mathbb{E}_{\mathcal{T}} R$$

# REINFORCE

Let's apply log-derivative trick to our goal!

$$\nabla_\theta \mathbb{E}_{\pi(\theta)} \mathbb{E}_{\mathcal{T}} R = \mathbb{E}_{\pi(\theta)} \nabla_\theta \log \pi(\theta) \mathbb{E}_{\mathcal{T}} R \approx$$

We can estimate this gradient using Monte-Carlo by playing, let's say, one game:

$$\approx \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) R$$

# Problems of REINFORCE

- $\times$ Doesn't work.

# Problems of REINFORCE

- $\times$ Doesn't work.
    - **Reason:** *high variance* of Monte-Carlo gradient estimation.

# Problems of REINFORCE

× Doesn't work.

- **Reason:** *high variance* of Monte-Carlo gradient estimation.
- you can play more than one game for one gradient step, but that doesn't help much.

# Baseline

**Proprosition**

For arbitrary distribution $\pi(\theta)$:

$$\mathbb{E}\nabla_\theta \log \pi(\theta) = \int \nabla_\theta \pi(\theta) = \nabla_\theta \int \pi(\theta) = \nabla_\theta 1 = 0$$

# Baseline

**Proprosition**

For arbitrary distribution $\pi(\theta)$:

$$\mathbb{E}\nabla_\theta \log \pi(\theta) = \int \nabla_\theta \pi(\theta) = \nabla_\theta \int \pi(\theta) = \nabla_\theta 1 = 0$$

Adding $\mathbb{E}\nabla_\theta \log \pi(\theta) b$ for some $b$ to gradient estimate will not lead to bias, but may change variance.

# Lowest variance baseline

**Theorem**

$$b = \frac{\mathbb{E}(\nabla_\theta \log \pi(\theta))^2 R}{\mathbb{E}(\nabla_\theta \log \pi(\theta))^2}$$

is the baseline which leads to the lowest variance.

# Lowest variance baseline

**Theorem**

$$b = \frac{\mathbb{E}(\nabla_\theta \log \pi(\theta))^2 R}{\mathbb{E}(\nabla_\theta \log \pi(\theta))^2}$$

is the baseline which leads to the lowest variance.

* similar to average reward, which is also a good baseline.

# Carefull REINFORCE

**Strange thing:** our gradient estimate depends on $R$, which includes reward in the first state $r(s_0)$, where we haven't performed any actions.[3]

[3] did we make any mistake?

# Carefull REINFORCE

**Strange thing:** our gradient estimate depends on $R$, which includes reward in the first state $r(s_0)$, where we haven't performed any actions.[3]

Let's untangle our goal:

$$\nabla_\theta \mathbb{E}_{p(s_1)} \left( r(s_1) + \mathbb{E}_{a_1 \sim \pi(s_1, \theta)} \mathbb{E}_{p(s_2 | s_1, a)} \left[ r(s_2) + \ldots \right] \right)$$

[3] did we make any mistake?

# Carefull REINFORCE

**Strange thing:** our gradient estimate depends on $R$, which includes reward in the first state $r(s_0)$, where we haven't performed any actions.[3]

Let's untangle our goal:

$$\nabla_\theta \mathbb{E}_{p(s_1)} \left( r(s_1) + \mathbb{E}_{a_1 \sim \pi(s_1, \theta)} \mathbb{E}_{p(s_2 | s_1, a)} \left[ r(s_2) + \ldots \right] \right) =$$

After carefully applying log-derivative trick:

$$= \mathbb{E} \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^T r(s_{t'}) \right)$$

[3] did we make any mistake?

# Carefull REINFORCE

**Strange thing:** our gradient estimate depends on $R$, which includes reward in the first state $r(s_0)$, where we haven't performed any actions.[3]

Let's untangle our goal:

$$\nabla_\theta \mathbb{E}_{p(s_1)} \left( r(s_1) + \mathbb{E}_{a_1 \sim \pi(s_1, \theta)} \mathbb{E}_{p(s_2 | s_1, a)} \left[ r(s_2) + \ldots \right] \right) =$$

After carefully applying log-derivative trick:

$$= \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) \right)$$

✓ that's much better!

[3] did we make any mistake?

# Actor-critic

**Note:** $\sum_{t'=t+1}^{T} r(s_{t'})$ is estimation of $Q^{\pi}(s_t, a_t)$!

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) \right)$$

# Actor-critic

**Note:** $\sum_{t'=t+1}^{T} r(s_{t'})$ is estimation of $Q^{\pi}(s_t, a_t)$!

$$
\begin{aligned}
\nabla &= \mathbb{E} \sum_{t}^{T} \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) \right) = \\
&= \mathbb{E} \sum_{t}^{T} \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) Q^{\pi}(s_t, a_t)
\end{aligned}
$$

# Actor-critic

**Note:** $\sum_{t'=t+1}^{T} r(s_{t'})$ is estimation of $Q^\pi(s_t, a_t)$!

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) \right) =$$

$$= \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) Q^\pi(s_t, a_t)$$

Better estimation of $Q^\pi(s, a)$
should lead to lower variance.

# Actor-critic

**Note:** $\sum_{t'=t+1}^{T} r(s_{t'})$ is estimation of $Q^{\pi}(s_t, a_t)$!

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) \right) =$$

$$= \mathbb{E} \sum_{t}^{T} \nabla_{\theta} \log \pi(a_t \mid s_t, \theta) Q^{\pi}(s_t, a_t)$$

Better estimation of $Q^{\pi}(s, a)$
should lead to lower variance.

* $\pi$ is an *actor*
* estimate of $Q^{\pi}(s, a)$ is a *critic*

# Advantage Actor Critic

$$\nabla = \mathbb{E}\sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) Q^\pi(s_t, a_t)$$

Let's insert some baseline:

# Advantage Actor Critic

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( Q^\pi(s_t, a_t) - b \right)$$

Let's insert some baseline:

# Advantage Actor Critic

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( Q^\pi(s_t, a_t) - b \right)$$

Let's insert some baseline:

* Recall average $Q^\pi(s_t, a_t)$ is a good baseline.

# Advantage Actor Critic

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left(Q^\pi(s_t, a_t) - b\right)$$

Let's insert some baseline:

* Recall average $Q^\pi(s_t, a_t)$ is a good baseline.
* Recall $\mathbb{E} Q^\pi(s_t, a_t) = V^\pi(s_t)$

# Advantage Actor Critic

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( Q^\pi(s_t, a_t) - b \right)$$

Let's insert some baseline:

* Recall average $Q^\pi(s_t, a_t)$ is a good baseline.
* Recall $\mathbb{E} Q^\pi(s_t, a_t) = V^\pi(s_t)$
* Recall definition $A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$

# Advantage Actor Critic

$$\nabla = \mathbb{E} \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) A^\pi(s_t, a_t)$$

Let's insert some baseline:

* Recall average $Q^\pi(s_t, a_t)$ is a good baseline.
* Recall $\mathbb{E} Q^\pi(s_t, a_t) = V^\pi(s_t)$
* Recall definition $A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$

# Critic set up

Critic can be a second neural net!

# Critic set up

Critic can be a second neural net!

**Options:**

* approximate $A^{\pi}(s, a)$

# Critic set up

Critic can be a second neural net!

**Options:**

* approximate $A^\pi(s, a)$
* approximate $Q^\pi(s, a)$[4]

[4] can we just use Q-learning for this?

# Critic set up

Critic can be a second neural net!

**Options:**

* approximate $A^\pi(s, a)$
* approximate $Q^\pi(s, a)$[4]
* approximate $V^\pi(s)$:

[4] can we just use Q-learning for this?

# Critic set up

Critic can be a second neural net!

**Options:**

* approximate $A^\pi(s, a)$
* approximate $Q^\pi(s, a)$[4]
* approximate $V^\pi(s)$:

$$Q^\pi(s_t, a_t) - V^\pi(s_t) \approx r(s_{t+1}) + V^\pi(s_{t+1}) - V^\pi(s_t)$$

[4] can we just use Q-learning for this?

# Critic set up

Critic can be a second neural net!

**Options:**

* approximate $A^{\pi}(s, a)$
* approximate $Q^{\pi}(s, a)$[4]
* approximate $V^{\pi}(s)$:

$$Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) \approx r(s_{t+1}) + V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$$

  ✓ the least complex one! [5]

[4] can we just use Q-learning for this?
[5] why?

# Critic learning

For given state $s$ we can calculate a target $y = V^{\pi}(s) \approx \sum_{t'=t+1}^{T} r(s_{t'})$.

At the end of the game, make a step of gradient descent to teach critic.

# Critic learning

For given state $s$ we can calculate a target $y = V^{\pi}(s) \approx \sum_{t'=t+1}^{T} r(s_{t'})$.

At the end of the game, make a step of gradient descent to teach critic.

**Problem:** the batch is highly correlated.

# Critic learning

For given state $s$ we can calculate a target $y = V^{\pi}(s) \approx \sum_{t'=t+1}^{T} r(s_{t'})$.

At the end of the game, make a step of gradient descent to teach critic.

**Problem:** the batch is highly correlated.

* well, play more games.

# Critic learning

For given state $s$ we can calculate a target $y = V^{\pi}(s) \approx \sum_{t'=t+1}^{T} r(s_{t'})$.

At the end of the game, make a step of gradient descent to teach critic.

**Problem:** the batch is highly correlated.

* well, play more games.
    :( for one step of gradient descent, yeah...

# Critic learning

For given state $s$ we can calculate a target $y = V^{\pi}(s) \approx \sum_{t'=t+1}^{T} r(s_{t'})$.

At the end of the game, make a step of gradient descent to teach critic.

**Problem:** the batch is highly correlated.

* well, play more games.
  
  :( for one step of gradient descent, yeah...

**Alternative:** $y = V^{\pi}(s) \approx r(s') + V^{\pi}(s')$

# A2C: Complete algorithm

**Advantage Actor-Critic (A2C) Algorithm:**

- get $(s, a, r, s')$

# A2C: Complete algorithm

**Advantage Actor-Critic (A2C) Algorithm:**

- get $(s, a, r, s')$
- update critic $\hat{V}(s)$ using target $r + \hat{V}(s')$

# A2C: Complete algorithm

**Advantage Actor-Critic (A2C) Algorithm:**

- get $(s, a, r, s')$
- update critic $\hat{V}(s)$ using target $r + \hat{V}(s')$
- evaluate $\hat{A}(s, a) = r + \hat{V}(s') - \hat{V}(s)$

# A2C: Complete algorithm

**Advantage Actor-Critic (A2C) Algorithm:**

- get $(s, a, r, s')$
- update critic $\hat{V}(s)$ using target $r + \hat{V}(s')$
- evaluate $\hat{A}(s, a) = r + \hat{V}(s') - \hat{V}(s)$
- update policy using estimate of gradient $\nabla_\theta \log \pi(a \mid s, \theta)\hat{A}(s, a)$

**Advantage Actor-Critic (A2C) Algorithm:**

- get $(s, a, r, s')$
- update critic $\hat{V}(s)$ using target $r + \hat{V}(s')$
- evaluate $\hat{A}(s, a) = r + \hat{V}(s') - \hat{V}(s)$
- update policy using estimate of gradient $\nabla_\theta \log \pi(a \mid s, \theta)\hat{A}(s, a)$

Check out this comic about A2C!

# A2C: Resume

- ✓ lower variance!

# A2C: Resume

- ✓ lower variance!
- × yet policy gradient estimates are not unbiased anymore![6]

[6] why?

# A2C: Resume

- ✓ lower variance!
- × yet policy gradient estimates are not unbiased anymore![6]
- × batch_size = 1

[6] why?

# A2C: Resume

- ✓ lower variance!
- × yet policy gradient estimates are not unbiased anymore![6]
- × batch_size $= 1$
  - ✓ do gradient descent step every $N$ game steps.

[6] why?

# A2C: Resume

- ✓ lower variance!
- × yet policy gradient estimates are not unbiased anymore![6]
- × batch_size = 1
  - ✓ do gradient descent step every $N$ game steps.
  - ✓ play several games in parallel.

[6] why?

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) - V^\pi(s_t) \right)$$

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^T r(s_{t'}) - V^\pi(s_t) \right)$$

✓ unbiased gradient

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) - V^\pi(s_t) \right)$$

- ✓ unbiased gradient
- × higher variance

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) - V^\pi(s_t) \right)$$

✓ unbiased gradient

× higher variance

Or use a compromise:

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{t+N} r(s_{t'}) + V^\pi(s_{t+N}) - V^\pi(s_t) \right)$$

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^T r(s_{t'}) - V^\pi(s_t) \right)$$

✓ unbiased gradient

× higher variance

Or use a compromise:

$$\nabla = \mathbb{E} \sum_t^T \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{t+N} r(s_{t'}) + V^\pi(s_{t+N}) - V^\pi(s_t) \right)$$

× new hyperparameter $N$

# Eligibility trace

We may use critic only for baseline:

$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{T} r(s_{t'}) - V^\pi(s_t) \right)$$

✓ unbiased gradient

× higher variance

Or use a compromise:

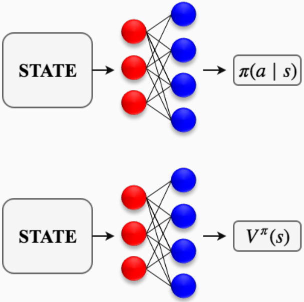
$$\nabla = \mathbb{E} \sum_{t}^{T} \nabla_\theta \log \pi(a_t \mid s_t, \theta) \left( \sum_{t'=t+1}^{t+N} r(s_{t'}) + V^\pi(s_{t+N}) - V^\pi(s_t) \right)$$

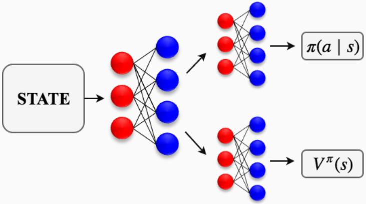
× new hyperparameter $N$

✓ regulates trade-off between variance and bias

# Dealing with two networks

Option 1: just two neural nets



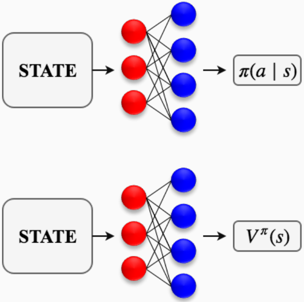



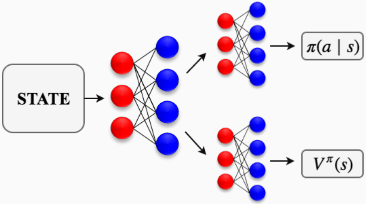

Option 2: shared feature extractor

# Dealing with two networks

Option 1: just two neural nets

- × obviously redundant




Option 2: shared feature extractor

- × may be unstable

# NEXT: A3C