[2.3] PPO

David Quarel

oh no lots of maths in this one :(

June 9, 2026

Assumptions

- **Action space** $\mathcal{A}$ is still discrete, and finite (though PPO works for continuous action spaces, unlike DQN)
- **State space** $\mathcal{S}$ is continuous (a subspace of $\mathbb{R}^n$), and each state is a vector $\mathbf{s} \in \mathcal{S}$.
- Environment is **episodic**, and agent always starts at/near **starting state** $\mathbf{s}_0$. We generate a full episode, and then afterwards learn.
- Define measure of performance (**joy**) as

$$J(\boldsymbol{\theta}) := V_{\pi_{\boldsymbol{\theta}}}(\mathbf{s}_0)$$

- Policy $\pi_{\boldsymbol{\theta}} \equiv \pi(\cdot|\mathbf{s}; \boldsymbol{\theta})$ is a conditional probability distribution $\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta\mathcal{A}$ over actions, mapping state to logits.
- Gradient $\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}$ is well-defined.
- Choose $\boldsymbol{\theta}$ to maximize joy $J(\boldsymbol{\theta})$ via gradient *ascent*

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

- Original PPO paper uses loss for everything. I use **loss** L for things we **minimize**, and **joy** J for things we **maximize**.

Aside: Expectations hide stuff!

The constant uses of $\mathbb{E}$ can hide a lot of what's going on when you implement it in practice. Usually wish to compute expected value of some function f of trajectories τ.

$$\mathbb{E}_\tau[f(\tau)] = \sum_\tau \Pr(\tau|\boldsymbol{\theta}) f(\tau)$$

and $\hat{\mathbb{E}}[f(\tau)] \equiv \frac{1}{|\mathcal{B}|} \sum_{b=1}^{|\mathcal{B}|} [f(\tau^b)]$ for when this is **approximated by sampling** .

REINFORCE

Recall (VPG)

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) G(\tau) \right]$$

Algorithm 1 Vanilla Policy Gradient ($\gamma = 1$)

Input: Environment p, Number of episodes M

1: **for** episode $= 1$ to M **do**
2: Generate episode $s_0, a_0, r_1, s_1, \ldots, s_{T-1}, a_{T-1}, r_T s_T$
3: Define $G_t = \sum_{i=t}^{T-1} r_i$ for $0 \leq t \leq T-1$
4: Define gain $J(\boldsymbol{\theta}) = \sum_{t=0}^{T-1} G_t \log \pi_{\boldsymbol{\theta}}(a_t|s_t)$
5: Gradient **ascent** step $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
6: **end for**

VPG generates a trajectory, learns, then discards it, generates another.

Clever trick 1: The future cannot affect the past

- In REINFORCE, each $\log \pi_\theta(a_t|s_t)$ is multiplied by the full return $G(\tau)$.
- But a_t cannot influence rewards that happened **before** it.
- Can be shown that we can replace $G(\tau)$ with the **reward-to-go**:

$$G_t := \sum_{t'=t+1}^{T} r_{t'} = r_{t+1} + \ldots + r_T$$

  which (in expectation) is the same as the Q-value $Q_{\pi_\theta}(s_t, a_t)$.
- This gives a lower-variance, unbiased estimator.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\, G(\tau)\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\, G_t\right] \\
&= \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)\, Q_{\pi_\theta}(s_t, a_t)\right]
\end{aligned}$$

Theorem: By the EGLP Lemma, (i.e. math) we have

$$\mathbb{E}_{a_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t|s_t) b(s_t)] = 0$$

for any baseline function b depending only on state.

Theorem: By the EGLP Lemma, (i.e. math) we have

$$\mathbb{E}_{a_t \sim \pi_{\boldsymbol{\theta}}}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) b(s_t)] = 0$$

for any baseline function b depending only on state. So, can add/subtract any such b to decrease variance $\equiv$ speed up learning

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \Big(Q_{\pi_{\boldsymbol{\theta}}}(s_t, a_t) - b(s_t) \Big) \right]$$

Theorem: By the EGLP Lemma, (i.e. math) we have

$$\mathbb{E}_{a_t \sim \pi_{\boldsymbol{\theta}}}[\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) b(s_t)] = 0$$

for any baseline function b depending only on state. So, can add/subtract any such b to decrease variance $\equiv$ speed up learning

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \Big(Q_{\pi_{\boldsymbol{\theta}}}(s_t, a_t) - b(s_t) \Big) \right]$$

Clever trick 2: Choose $b(s_t) = \widehat{V}_\pi(s_t)$ as (some estimate of) the value function.

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) A_{\pi_{\boldsymbol{\theta}}}(s_t, a_t) \right]$$

where $\widehat{A}_\pi(s, a) := Q_\pi(s, a) - \widehat{V}_\pi(s)$ is the **advantage** function

- how much better taking action a is compared to sampling from $\pi(\cdot|s)$.
- (assuming optimal value function) with no advantage ($A_\pi = 0$), stop learning, ($\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0$).

Aside: n-step update

Recall TD(0): $\widehat{V}_\pi(s_t) \leftarrow \widehat{V}_\pi(s_t) + \alpha\delta_t$ where

$$\delta_t \equiv \delta_{t:t+1} = r_{t+1} + \gamma \widehat{V}_\pi(s_{t+1}) - \widehat{V}_\pi(s_t)$$

Aside: n-step update

Recall TD(0): $\widehat{V}_\pi(s_t) \leftarrow \widehat{V}_\pi(s_t) + \alpha\delta_t$ where

$$\delta_t \equiv \delta_{t:t+1} = r_{t+1} + \gamma \widehat{V}_\pi(s_{t+1}) - \widehat{V}_\pi(s_t)$$

Why not include the information from two time-steps?

$$\delta_{t:t+2} = r_{t+1} + \gamma r_{t+2} + \gamma^2 \widehat{V}_\pi(s_{t+2}) - \widehat{V}_\pi(s_t)$$

Aside: n-step update

Recall TD(0): $\widehat{V}_\pi(s_t) \leftarrow \widehat{V}_\pi(s_t) + \alpha\delta_t$ where

$$\delta_t \equiv \delta_{t:t+1} = r_{t+1} + \gamma \widehat{V}_\pi(s_{t+1}) - \widehat{V}_\pi(s_t)$$

Why not include the information from two time-steps?

$$\delta_{t:t+2} = r_{t+1} + \gamma r_{t+2} + \gamma^2 \widehat{V}_\pi(s_{t+2}) - \widehat{V}_\pi(s_t)$$

In general, we can perform n-step return updates:

$$\delta_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \widehat{V}_\pi(s_{t+n}) - \widehat{V}_\pi(s_t)$$

Aside: n-step update

Recall TD(0): $\widehat{V}_\pi(s_t) \leftarrow \widehat{V}_\pi(s_t) + \alpha\delta_t$ where

$$\delta_t \equiv \delta_{t:t+1} = r_{t+1} + \gamma \widehat{V}_\pi(s_{t+1}) - \widehat{V}_\pi(s_t)$$

Why not include the information from two time-steps?

$$\delta_{t:t+2} = r_{t+1} + \gamma r_{t+2} + \gamma^2 \widehat{V}_\pi(s_{t+2}) - \widehat{V}_\pi(s_t)$$

In general, we can perform n-step return updates:

$$\delta_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \widehat{V}_\pi(s_{t+n}) - \widehat{V}_\pi(s_t)$$

Or full rollout to the end of the episode (Monte Carlo (MC) update):

$$\delta_{t:T} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^{T-(t+1)} r_T - \widehat{V}_\pi(s_t)$$

where T is the final time step of the episode.

Aside: n-step update

Complete return $G_t \equiv G_{t:T} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{T-t-1} r_T$. Can write approximations using n-step returns:

$$G_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \hat{V}_\pi(s_{t+n})$$

Aside: n-step update

Complete return $G_t \equiv G_{t:T} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{T-t-1} r_T$. Can write approximations using n-step returns:

$$G_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \hat{V}_\pi(s_{t+n})$$

In TD(0), we use the single-step return:

$$\delta_t = G_{t:t+1} - \hat{V}_\pi(s_t)$$
$$G_{t:t+1} = r_{t+1} + \gamma \hat{V}_\pi(s_{t+1})$$

Aside: n-step update

Complete return $G_t \equiv G_{t:T} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{T-t-1} r_T$. Can write approximations using n-step returns:

$$G_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \hat{V}_\pi(s_{t+n})$$

In TD(0), we use the single-step return:

$$\delta_t = G_{t:t+1} - \hat{V}_\pi(s_t)$$
$$G_{t:t+1} = r_{t+1} + \gamma \hat{V}_\pi(s_{t+1})$$

Can now write n-step TD update as:

$$\delta_{t:t+n} = G_{t:t+n} - \hat{V}_\pi(s_t)$$

Aside: n-step update

Complete return $G_t \equiv G_{t:T} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{T-t-1} r_T$. Can write approximations using n-step returns:

$$G_{t:t+n} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n \hat{V}_\pi(s_{t+n})$$

In TD(0), we use the single-step return:

$$\delta_t = G_{t:t+1} - \hat{V}_\pi(s_t)$$
$$G_{t:t+1} = r_{t+1} + \gamma \hat{V}_\pi(s_{t+1})$$

Can now write n-step TD update as:

$$\delta_{t:t+n} = G_{t:t+n} - \hat{V}_\pi(s_t)$$

Can be shown that n-step return $G_{t:t+n}$ are all valid targets for TD updates.

- Smaller n: Lower variance, higher bias
- Larger n: Higher variance, lower bias
- ...is there a way to get the benefits of both?

Mixing updates

Can have any mixture of n-step returns as a valid target for TD updates. For example,

$$\delta_t = \frac{G_{t:t+1} + G_{t:t+2}}{2} - \hat{V}_\pi(s_t)$$

is a valid TD update (an average of 1-step and 2-step returns).

[1] Note that $\sum_{n=0}^{\infty} \lambda^n = \frac{1}{1-\lambda}$, so we scale weighting by $1 - \lambda$.

Mixing updates

Can have any mixture of n-step returns as a valid target for TD updates. For example,

$$\delta_t = \frac{G_{t:t+1} + G_{t:t+2}}{2} - \hat{V}_\pi(s_t)$$

is a valid TD update (an average of 1-step and 2-step returns).
TD(λ) mixes over all n-step returns, weighting[1] $G_{t:t+n}$ by weight $w_n \propto \lambda^n$

$$\delta_t^\lambda = w_1 G_{t:t+1} + w_2 G_{t:t+2} + w_3 G_{t:t+3} + \ldots - \hat{V}_\pi(s_t)$$

The further forward in the future, the less weight it has.

[1] Note that $\sum_{n=0}^{\infty} \lambda^n = \frac{1}{1-\lambda}$, so we scale weighting by $1 - \lambda$.

Mixing updates

Can have any mixture of n-step returns as a valid target for TD updates. For example,

$$\delta_t = \frac{G_{t:t+1} + G_{t:t+2}}{2} - \hat{V}_\pi(s_t)$$

is a valid TD update (an average of 1-step and 2-step returns).
TD(λ) mixes over all n-step returns, weighting[1] $G_{t:t+n}$ by weight $w_n \propto \lambda^n$

$$\delta_t^\lambda = w_1 G_{t:t+1} + w_2 G_{t:t+2} + w_3 G_{t:t+3} + \ldots - \hat{V}_\pi(s_t)$$

The further forward in the future, the less weight it has.

TD(λ) update

$$\delta_t^\lambda = G_t^\lambda - \hat{V}_\pi(s_t) \text{ where } G_t^\lambda = (1-\lambda)\sum_{n=0}^{\infty} \lambda^n G_{t:t+n+1}$$

[1] Note that $\sum_{n=0}^{\infty} \lambda^n = \frac{1}{1-\lambda}$, so we scale weighting by $1-\lambda$.

Mixing updates

Can have any mixture of n-step returns as a valid target for TD updates. For example,

$$\delta_t = \frac{G_{t:t+1} + G_{t:t+2}}{2} - \hat{V}_\pi(s_t)$$

is a valid TD update (an average of 1-step and 2-step returns).
TD(λ) mixes over all n-step returns, weighting[1] $G_{t:t+n}$ by weight $w_n \propto \lambda^n$

$$\delta_t^\lambda = w_1 G_{t:t+1} + w_2 G_{t:t+2} + w_3 G_{t:t+3} + \ldots - \hat{V}_\pi(s_t)$$

The further forward in the future, the less weight it has.

TD(λ) update

$$\delta_t^\lambda = G_t^\lambda - \hat{V}_\pi(s_t) \text{ where } G_t^\lambda = (1-\lambda)\sum_{n=0}^{\infty} \lambda^n G_{t:t+n+1}$$

Exercise: Prove TD(λ) for $\lambda = 0$ is TD(0), and that TD(1) is MC.

[1] Note that $\sum_{n=0}^{\infty} \lambda^n = \frac{1}{1-\lambda}$, so we scale weighting by $1-\lambda$.

Eligibility Traces

Because math one can compute the TD(λ) update rule

$$\begin{aligned}\hat{V}_\pi(s_t) &\leftarrow \hat{V}_\pi(s_t) + \alpha\delta_t^\lambda \\ &\leftarrow \hat{V}_\pi(s_t) + \alpha\left((1-\lambda)\left[\sum_{n=0}^{\infty}\lambda^n G_{t:t+n+1}\right] - \hat{V}_\pi(s_t)\right)\end{aligned}$$

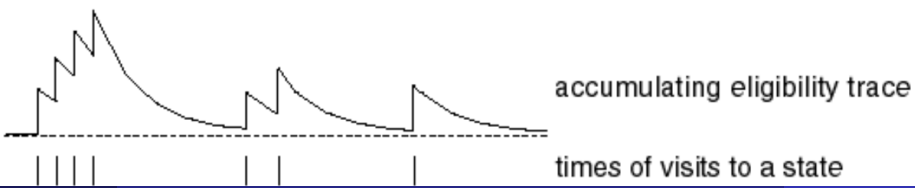
very efficiently as

$$\hat{V}_\pi(s_t) \leftarrow \hat{V}_\pi(s_t) + \alpha\delta_t E_t(s_t)$$

where E is the **eligibility trace**

$$\begin{aligned}E^0(s) &:= 0 \\ E^t(s) &:= \gamma\lambda E^{t-1}(s) + [\![s = s_t]\!]\end{aligned}$$

and $\delta_t = G_{t:t+1} - \hat{V}_\pi(s_t)$ is the standard 1-step TD error. We use this for PPO advantage A_t.



Actor-Critic VPG

Approximate advantage as $\hat{A}_t \equiv \hat{A}_{\pi_\theta}(s_t, a_t) = \hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t)$.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

where:

Actor-Critic VPG

Approximate advantage as $\hat{A}_t \equiv \hat{A}_{\pi_\theta}(s_t, a_t) = \hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t)$.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

where:

- Approx Q_{π_θ} using empirical return, $\hat{Q}_{\pi_\theta}(s_t, a_t) := G_t = \sum_{t'=t+1}^{T} r_{t'}$

Actor-Critic VPG

Approximate advantage as $\hat{A}_t \equiv \hat{A}_{\pi_\theta}(s_t, a_t) = \hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t)$.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

where:

- Approx Q_{π_θ} using empirical return, $\hat{Q}_{\pi_\theta}(s_t, a_t) := G_t = \sum_{t'=t+1}^{T} r_{t'}$
- Approx V_{π_θ} using **critic** network $\hat{V}_{\pi_\theta}(s_t) := V_\phi(s_t)$ with parameters ϕ

Actor-Critic VPG

Approximate advantage as $\hat{A}_t \equiv \hat{A}_{\pi_\theta}(s_t, a_t) = \hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t)$.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

where:

- Approx Q_{π_θ} using empirical return, $\hat{Q}_{\pi_\theta}(s_t, a_t) := G_t = \sum_{t'=t+1}^{T} r_{t'}$
- Approx V_{π_θ} using **critic** network $\hat{V}_{\pi_\theta}(s_t) := V_\phi(s_t)$ with parameters ϕ
- Approximate the expectation $\mathbb{E}_{\tau \sim \pi_\theta}$ by sampling $\hat{\mathbb{E}}$

Actor-Critic VPG

Approximate advantage as $\hat{A}_t \equiv \hat{A}_{\pi_\theta}(s_t, a_t) = \hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t)$.

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(\hat{Q}_{\pi_\theta}(s_t, a_t) - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

where:

- Approx Q_{π_θ} using empirical return, $\hat{Q}_{\pi_\theta}(s_t, a_t) := G_t = \sum_{t'=t+1}^{T} r_{t'}$
- Approx V_{π_θ} using **critic** network $\hat{V}_{\pi_\theta}(s_t) := V_\phi(s_t)$ with parameters ϕ
- Approximate the expectation $\mathbb{E}_{\tau \sim \pi_\theta}$ by sampling $\hat{\mathbb{E}}$

$$\nabla_\theta J(\theta) \approx \hat{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) \left(G_t - \hat{V}_{\pi_\theta}(s_t) \right) \right]$$

- π_θ is the **actor** and V_ϕ is the **critic**.

Critic Loss

The critic is simply trained against the returns from the environment. We define the **value function** loss or **critic loss** as

$$L^{VF}(\phi) = \hat{\mathbb{E}}_t[V_\phi(s_t) - G_t^\lambda]^2$$

where G_t^λ is the target for TD(λ) introduced for eligibility traces.

PPO: The big picture

PPO is just Actor-Critic VPG with a bag of tricks/heuristics on top:

- Use importance sampling to allow off-policy training.
- Clip the importance sampling ratio so new policy doesn't drift too far.
- Use Actor-Critic method for stability:
  - Actor chooses the actions
  - Critic learns the value of states, and provides feedback to the actor.
- Add an entropy bonus to prevent the policy converging on deterministic too fast.

PPO: The big picture

PPO is just Actor-Critic VPG with a bag of tricks/heuristics on top:

- Use importance sampling to allow off-policy training.
- Clip the importance sampling ratio so new policy doesn't drift too far.
- Use Actor-Critic method for stability:
  - Actor chooses the actions
  - Critic learns the value of states, and provides feedback to the actor.
- Add an entropy bonus to prevent the policy converging on deterministic too fast.

Q: Is the bag of tricks well motivated/principled?

PPO: The big picture

PPO is just Actor-Critic VPG with a bag of tricks/heuristics on top:

- Use importance sampling to allow off-policy training.
- Clip the importance sampling ratio so new policy doesn't drift too far.
- Use Actor-Critic method for stability:
  - Actor chooses the actions
  - Critic learns the value of states, and provides feedback to the actor.
- Add an entropy bonus to prevent the policy converging on deterministic too fast.

Q: Is the bag of tricks well motivated/principled?
A: Somewhat? Importance sampling is principled, clipping is kind of hacky, and entropy bonus really feels like an afterthought, but it works!

Importance Sampling

Recall VPG Joy estimator:

$$J(\boldsymbol{\theta}) \approx \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \hat{A}_t \right]$$

- Inefficient to generate trajectories and then learn one at a time.
- Sample lots of trajectories, and then learn from them (perhaps many times).
- But during learning, data generated from $\pi_{\boldsymbol{\theta}_{old}}$, but policy drifts to $\pi_{\boldsymbol{\theta}}$.

$$J(\boldsymbol{\theta}) \not\approx \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_{old}}} \left[\sum_{t=0}^{T-1} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \hat{A}_t \right]$$

- VPG is an on-policy method, so we can't use it.
- We have $\tau \sim \pi_{\boldsymbol{\theta}_{old}}$, but we need $\tau \sim \pi_{\boldsymbol{\theta}}$.

Importance Sampling

Importance Sampling

Idea: Want to estimate $\mathbb{E}_{x\sim p}[f(x)]$ using samples from q.

$$\mathbb{E}_{x\sim p}[f(x)] = \mathbb{E}_{x\sim q}\left[\frac{p(x)}{q(x)} f(x)\right] \tag{1}$$

Importance Sampling

We can log the probabilities $\pi_{\theta_{old}}$ that were used to generate the trajectories:

$$\nabla J(\boldsymbol{\theta}) \approx \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \hat{A}_t \right]$$

where $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ is the importance ratio. Note that $r_t(\boldsymbol{\theta}_{\text{old}}) = 1$.

Importance Sampling

We can log the probabilities $\pi_{\theta_{old}}$ that were used to generate the trajectories:

$$\nabla J(\boldsymbol{\theta}) \approx \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \hat{A}_t \right] = \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)} \hat{A}_t \right]$$

where $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ is the importance ratio. Note that $r_t(\boldsymbol{\theta}_{\text{old}}) = 1$.

Importance Sampling

We can log the probabilities $\pi_{\theta_{old}}$ that were used to generate the trajectories:

$$\nabla J(\boldsymbol{\theta}) \approx \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{T-1}\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(a_t|s_t)\hat{A}_t\right] = \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{T-1}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)}\hat{A}_t\right]$$

$$= \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}_{old}}}\left[\frac{P(\tau|\pi_{\boldsymbol{\theta}})}{P(\tau|\pi_{\boldsymbol{\theta}_{old}})}\sum_{t=0}^{T-1}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)}\hat{A}_t\right]$$

where $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ is the importance ratio. Note that $r_t(\boldsymbol{\theta}_{\text{old}}) = 1$.

Importance Sampling

We can log the probabilities $\pi_{\theta_{old}}$ that were used to generate the trajectories:

$$\nabla J(\boldsymbol{\theta}) \approx \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{T-1}\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(a_t|s_t)\hat{A}_t\right] = \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_{t=0}^{T-1}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)}\hat{A}_t\right]$$

$$= \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}_{old}}}\left[\frac{P(\tau|\pi_{\boldsymbol{\theta}})}{P(\tau|\pi_{\boldsymbol{\theta}_{old}})}\sum_{t=0}^{T-1}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)}\hat{A}_t\right]$$

$$\approx \hat{\mathbb{E}}_{\tau\sim\pi_{\boldsymbol{\theta}_{old}}}\left[\sum_{t=0}^{T-1}\frac{\cancel{\pi_{\boldsymbol{\theta}}(a_t|s_t)}}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}\frac{\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\cancel{\pi_{\boldsymbol{\theta}}(a_t|s_t)}}\hat{A}_t\right]$$

where $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ is the importance ratio. Note that $r_t(\boldsymbol{\theta}_{\text{old}}) = 1$.

Importance Sampling

We can log the probabilities $\pi_{\theta_{old}}$ that were used to generate the trajectories:

$$\nabla J(\theta) \approx \hat{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right] = \hat{\mathbb{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T-1}\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\hat{A}_t\right]$$

$$= \hat{\mathbb{E}}_{\tau\sim\pi_{\theta_{old}}}\left[\frac{P(\tau|\pi_\theta)}{P(\tau|\pi_{\theta_{old}})}\sum_{t=0}^{T-1}\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_\theta(a_t|s_t)}\hat{A}_t\right]$$

$$\approx \hat{\mathbb{E}}_{\tau\sim\pi_{\theta_{old}}}\left[\sum_{t=0}^{T-1}\frac{\cancel{\pi_\theta(a_t|s_t)}}{\pi_{\theta_{old}}(a_t|s_t)}\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\cancel{\pi_\theta(a_t|s_t)}}\hat{A}_t\right]$$

$$= \hat{\mathbb{E}}_{\tau\sim\pi_{\theta_{old}}}\left[\sum_{t=0}^{T-1}\frac{\nabla_\theta \pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_{\tau\sim\pi_{\theta_{old}}}\left[\sum_{t=0}^{T-1}\nabla_\theta r_t(\theta)\hat{A}_t\right]$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the importance ratio. Note that $r_t(\theta_{\text{old}}) = 1$.

PPO Clip Joy

$$J(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_{old}}} \left[\sum_{t=0}^{T-1} r_t(\boldsymbol{\theta}) \hat{A}_t \right]$$

- Last dirty hack: try to make only a small update by clipping $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ to be in the range $[1-\epsilon, 1+\epsilon]$.

$$J^{small}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_{old}}} \left[\sum_{t=0}^{T-1} \text{clip}(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon) \hat{A}_t \right]$$

PPO Clip Joy

$$J(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_{old}}} \left[\sum_{t=0}^{T-1} r_t(\boldsymbol{\theta}) \hat{A}_t \right]$$

- Last dirty hack: try to make only a small update by clipping $r_t(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t|s_t)}$ to be in the range $[1-\epsilon, 1+\epsilon]$.

$$J^{small}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}_{old}}} \left[\sum_{t=0}^{T-1} \text{clip}(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon) \hat{A}_t \right]$$

- Be conservative, and take smaller of the clipped and unclipped objective to get a lower bound on the unclipped objective. See Section 3 of PPO paper.

PPO Clip Joy

$$J^{CLIP}(\boldsymbol{\theta}) = \hat{\mathbb{E}}_t \left[\sum_{t=0}^{T-1} \min \left(r_t(\boldsymbol{\theta}) \hat{A}_t, \text{clip}(r_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right]$$

for **magic number** $\epsilon = 0.2$.

Entropy Bonus

- We also add an **entropy bonus** to incentivise exploration by increasing the **entropy** of the distribution. The entropy $H_\pi(s)$ of a policy π in state s is defined as

$$H_\pi(s) = \sum_{\substack{a \in \mathcal{A} \\ \pi(a|s) > 0}} \pi(a|s) \log \frac{1}{\pi(a|s)}$$

- Entropy can be thought of as a measure of how "random" the distribution is,
- Entropy bounded by $0 \leq H_\pi \leq \log |\mathcal{A}|$.
- Min entropy of 0 means a deterministic policy (all the probability mass on one output).
  - Can also log $H_\pi(s)$ during training. If it crashes to zero, model probably got stuck.
- Max entropy of $\log |\mathcal{A}|$ means the policy is uniformly guessing at random.

Entropy Bonus

In practice, entropy bonus is computed as an average:

$$
\begin{aligned}
J^H(\boldsymbol{\theta}) &:= \hat{\mathbb{E}}_t \left[H_{\pi_{\boldsymbol{\theta}}}(s_t) \right] \\
&= \frac{1}{|\mathcal{B}|} \sum_{b=1}^{|\mathcal{B}|} \sum_{t=0}^{T-1} \left[\sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s_t^b) \log \frac{1}{\pi_{\boldsymbol{\theta}}(a|s_t^b)} \right]
\end{aligned}
$$

Note: We sum over all actions a, no dependency on the actual action a_t taken!

PPO at long last!

PPO Loss/Gain functions ($\sum$ and $\mathbb{E}$ suppressed)

$$J_t^{CLIP}(\theta) = \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \quad \text{[clip joy]}$$

$$L_t^{VF}(\phi) = (V_\phi(s_t) - G_t^\lambda)^2 \quad \text{[critic loss]}$$

$$J_t^H(\theta) = H_{\pi_\theta}(s_t) \quad \text{[entropy joy]}$$

Combine them all, with magic numbers $c_{VF} \approx 0.5$, $c_H = 0.01$, and $\epsilon = 0.2$.

PPO Objective

$$J^{PPO}(\theta, \phi) = \hat{\mathbb{E}}_t[J_t^{CLIP}(\theta) - c_{VF} L_t^{VF}(\phi) + c_H J_t^H(\theta)]$$

Maximise $J^{PPO}(\theta, \phi)$ with respect to policy parameters θ and critic parameters ϕ.

Rollout phase