Approximate Policy Iteration and Variants

Chen-Yu Wei

Policy Iteration

For $k = 1,\ 2, \ldots$

$\quad$ Calculate $Q^{\pi_k}(s,a) \quad \forall s,a$

$\quad$ $\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_k}(s,a) \quad \forall s$

Asynchronous Policy Iteration

For $k = 1,\ 2, \ldots$

Pick any state $\hat{s}$

Calculate $Q^{\pi_k}(\hat{s}, a) \quad \forall a$

$\pi_{k+1}(\hat{s}) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_k}(\hat{s}, a)$

and $\pi_{k+1}(s) = \pi_k(s) \quad \forall s \neq \hat{s}$

$V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s) \quad \forall s$

$$\mathbb{E}_{s\sim\rho}\left[V^{\pi_{k+1}}(s)\right] - \mathbb{E}_{s\sim\rho}\left[V^{\pi_k}(s)\right]$$

$$= \sum_{s,a} d_\rho^{\pi_{k+1}}(s)\left[\pi_{k+1}(a|s) - \pi_k(a|s)\right] Q^{\pi_k}(s,a)$$

$$= \sum_a d_\rho^{\pi_{k+1}}(\hat{s})\left(\underline{\pi_{k+1}(a|\hat{s})} - \pi_k(a|\hat{s})\right) \underline{Q^{\pi_k}(\hat{s},a)}$$

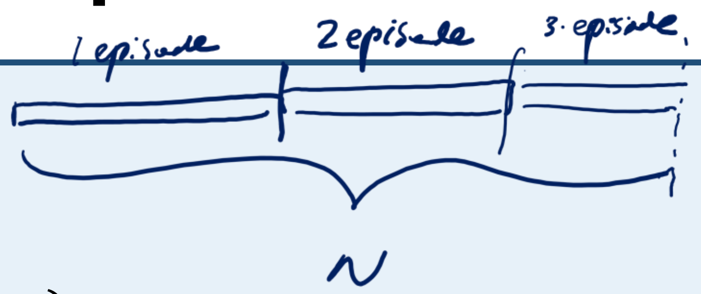
$$= \underline{d_\rho^{\pi_{k+1}}(\hat{s})}\left(\underbrace{\max_a Q^{\pi_k}(\hat{s},a)}\ - \underbrace{\sum_a \pi_k(a|\hat{s}) Q^{\pi_k}(\hat{s},a)}\right)$$

$$\geq 0$$

Asynchronous Policy Iteration

- To improve policy, we may just evaluate Q^{π_k} on a particular state s.
- Of course, a **real improvement** is made only when $\exists a$ s.t. $Q^{\pi_k}(s,a) - V^{\pi_k}(s)$ is large.
- This is **different from Value Iteration**, where ideally, we would like to find Q_{k+1} such that $Q_{k+1}(s,a) \approx R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\max_{a'} Q_k(s',a')\right]$ $\forall s, a$
- VI-based algorithm like DQN usually requires **stronger function approximation** that can generalize to unseen state.

Policy Iteration with Samples



For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Evaluate $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a)$ for $s = s_1, \ldots, s_N$ and all a

or $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a) - b_k(s)$ for $s = s_1, \ldots, s_N$ and all a

Policy Evaluation

Update θ_{k+1} from θ_k using the estimators $\{Z_k(s_i, a)\}_{i=1}^N$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

Why can we independently optimize the policy on each state?

Essentially treating **states** as **contexts**, but replacing $R(x, a)$ by $Q^{\pi_{\theta_k}}(s, a)$

Policy Evaluation

Policy Evaluation

(s, a, r, s')

Given: a policy π

Evaluate $V^{\pi}(s)$ or $Q^{\pi}(s, a)$ for certain (states, actions)

✓ **On-policy policy evaluation**: the learner can execute π to evaluate π

✗ **Off-policy/offline policy evaluation**: the learner can only execute some $\pi_b \neq \pi$, or can only access some existing dataset to evaluate π

Use cases:

- Approximate policy iteration: $\pi_k(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_{k-1}}(s, a)$
- Estimate the value of a policy before deploying it in the real world, e.g., COVID-related border measures, economic recovery policies, or policy changes in recommendation systems.

Value Iteration for V^{π} / Q^{π}

Input: π

For $k = 1,\ 2, \ldots$

$$\forall s, \qquad V_k(s) \leftarrow \sum_a \pi(a|s) \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{k-1}(s') \right)$$

Input: π

For $k = 1,\ 2, \ldots$

$$\forall s, a, \qquad Q_k(s,a) \leftarrow R(s,a) + \gamma \sum_{s',a'} P(s'|s,a)\, \pi(a'|s') Q_{k-1}(s',a')$$

On-Policy Policy Evaluation

Temporal Difference (TD) Learning for V^π

uniform $((s_1, a, s_2), (s_2, a, s_2))$

For $k = 1,\ 2, \ldots$

$\boxed{\text{Collect } \{(s_i, a_i, r_i, s_i')\}_{i=1}^N \text{ using policy } \pi}$

$$\theta_k \leftarrow \theta_{k-1} - \alpha\, \nabla_\theta \frac{1}{N} \sum_{i=1}^{N} \left(V_\theta(s_i) - r_i - \gamma V_{\theta_{k-1}}(s_i') \right)^2 \Bigg|_{\theta = \theta_{k-1}}$$

No target network needed because this is an **on-policy** problem.

This algorithm is also called TD(0)

TD(λ), $\lambda \in [0, 1]$

Temporal Difference (TD) Learning for Q^π

For $k = 1, \; 2, \dots$

Collect $\{(s_i, a_i, r_i, s_i')\}_{i=1}^N$ using policy π

$$\theta_k \leftarrow \theta_{k-1} - \alpha \, \nabla_\theta \frac{1}{N} \sum_{i=1}^{N} \left(Q_\theta(s_i, a_i) - r_i - \gamma \sum_a \pi(a|s_i') Q_{\theta_{k-1}}(s_i', a') \right)^2 \Bigg|_{\theta=\theta_{k-1}}$$

No target network needed because this is an on-policy problem.

Monte Carlo Estimation

Start from $(s_1, a_1) = (\hat{s}, \hat{a})$ and
execute policy π until the episode ends and obtain trajectory

$$s_1 = \hat{s}, a_1 = \hat{a}, r_1, s_2, a_2, r_2, \dots, s_\tau, a_\tau, r_\tau$$

Let $G = \sum_{h=1}^{\tau} \gamma^{h-1} r_h$

$\mathbb{E}(G)$ is an unbiased estimator for $Q^\pi(\hat{s}, \hat{a})$

MC estimator: unbiased, higher variance

TD estimator: biased, lower variance

A Family of Estimators

Suppose we have a **state-value function estimation** $V_\phi(s) \approx V^\pi(s)$

Suppose we also have a **trajectory** $s_1, a_1, r_1, \ldots, s_\tau, a_\tau, r_\tau$ generated by π where $s_{\tau+1}$ is a terminal state

The following are all valid estimators of $Q^\pi(s_1, a_1)$:

$$G_{1:1} = r_1 + \gamma V_\phi(s_2)$$

$$G_{1:2} = r_1 + \gamma r_2 + \gamma^2 V_\phi(s_3)$$

…

$$G_{1:\tau-1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} V_\phi(s_\tau)$$

$$G_{1:\tau} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_\tau$$

$$G_{1:\tau+1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_\tau$$

…

$$G_{1:\infty} =$$

same

more biased
lower variance

more unbiased
higher variance

A Family of Estimators

And the following are estimators of $Q^{\pi}(s_1, a_1) - V_{\phi}(s_1)$ (baseline)

$$A_{1:1} = r_1 + \gamma V_{\phi}(s_2) - V_{\phi}(s_1)$$

…

$$A_{1:\tau-1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} V_{\phi}(s_{\tau}) - V_{\phi}(s_1)$$

$$A_{1:\tau} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_{\tau} - V_{\phi}(s_1)$$

$$A_{1:\tau+1} = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{\tau-1} r_{\tau} - V_{\phi}(s_1)$$

…

Below, we will introduce a way to combine these estimators.

$\sum_{i=1}^{\infty}(1-\lambda)\lambda^{i-1}=1$

Balancing Bias and Variance

$G_{1:1}$ lower variance, higher bias

$G_{1:2}$

$\vdots$

$G_{1:\tau}$ all estimators of $Q^{\pi}(s_1,a_1)$

$\vdots$

$G_{1:\infty}$ higher variance, lower bias

$$G_1(\lambda) = (1-\lambda)\sum_{i=1}^{\infty}\lambda^{i-1}G_{1:i}$$

$$= (1-\lambda)\left(G_{1:1} + \lambda G_{1:2} + \lambda^2 G_{1:3} + \cdots + \lambda^{\tau-1}G_{1:\tau} + \lambda^{\tau}G_{1:\tau+1} + \lambda^{\tau+1}G_{1:\tau+2} + \cdots\right)$$

$G_{1:1}$

$1 \quad \lambda \quad \lambda^2 \quad + \lambda^3 + \cdots$

$= 1+\lambda+\lambda^2+\cdots+\lambda^{\infty} = \frac{1}{1-\lambda}$

$G_{1:1} + \lambda G_{1:2} + \lambda^2 G_{1:3} + \cdots$

$\frac{G_{1:1} + \lambda G_{1:2} + \cdots}{1+\lambda+\lambda^2+\cdots} = (1-\lambda)(G_{1:1}+\cdots)$

$\frac{1}{1-\lambda}$

$$A_1(\lambda) = (1-\lambda)\sum_{i=1}^{\infty}\lambda^{i-1}A_{1:i}$$ **(Generalized Advantage Estimation)**

$A_{1:i} = (G_{1:i} - V_{\phi}(s_i))$

$$= (1-\lambda)\left(A_{1:1} + \lambda A_{1:2} + \lambda^2 A_{1:3} + \cdots + \lambda^{\tau-1}A_{1:\tau} + \lambda^{\tau}A_{1:\tau+1} + \lambda^{\tau+1}A_{1:\tau+2} + \cdots\right)$$

Computational time $\approx 1+2+\cdots+\tau \approx \Theta(\tau^2)$

$$A_1(\lambda) = G_1(\lambda) - V_{\phi}(s_1)$$

Computing Generalized Advantage Estimator (GAE)

$$A_1(\lambda) \approx Q^{\pi_{\theta_k}}(s_1,a_1) - V_\phi(s_1) = (1-\lambda)(G_{1:1} + \lambda G_{1:2} + \cdots + \lambda^{\tau-1} G_{1:\tau} + \cdots)$$

$$A_2(\lambda) \approx Q^{\pi_{\theta_k}}(s_2,a_2) - V_\phi(s_2)$$

$$A_{m-1}$$

$$A_m(\lambda) \approx Q^{\pi_{\theta_k}}(s_m,a_m) - V_\phi(s_m) = (1-\lambda)(G_{m:m})$$

$$A_N(\lambda) \approx Q^{\pi_{\theta_k}}(s_N,a_N) - V_\phi(s_N)$$

$m-1$ is an end of a episode

m is a start of a new episode

Focusing on calculating $A_1(\lambda), A_2(\lambda), \cdots, A_\tau(\lambda)$

[we can calcute all of them in $O(\tau)$ time]

$$A_\tau(\lambda) = (1-\lambda)(A_{\tau:\tau} + \lambda A_{\tau:\tau+1} + \lambda^2 A_{\tau:\tau+2} + \cdots) = A_{\tau:\tau} = r_\tau + \gamma \overset{0}{\cancel{V_\phi(s_{\tau+1})}} - V_\phi(s_\tau) = \delta_\tau$$

$$A_{\tau-1}(\lambda) = (1-\lambda)(A_{\tau-1:\tau-1} + \lambda A_{\tau-1:\tau} + \lambda^2 A_{\tau-1:\tau+1} + \cdots) =$$

$$\vdots$$

$$A_2(\lambda) = \cdots =$$

$$A_1(\lambda) = (1-\lambda)(A_{1:1} + \lambda A_{1:2} + \lambda^2 A_{1:3} + \cdots) = \delta_1 + \lambda\gamma A_2(\lambda)$$

$$= (1-\lambda)(\cancel{\delta_1} + \lambda(\cancel{\delta_1} + \gamma\delta_2) + \lambda^2(\cancel{\delta_1} + \gamma\delta_2 + \gamma^2\delta_3) + \cdots)$$

$$= \delta_1 + (1-\lambda)\lambda\gamma(\delta_2 + \lambda(\delta_2 + \gamma\delta_3) + \lambda^2(\delta_2 + \gamma\delta_3 + \gamma^2\delta_4) + \cdots) \qquad = A_2(\lambda)$$

$$A_{i:j} = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{j-i} r_j + \gamma^{j-i+1} V_\phi(s_{j+1}) - V_\phi(s_i)$$

$$= [r_i + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)] + \gamma [r_{i+1} + \gamma V_\phi(s_{i+2}) - V_\phi(s_{i+1})] + \gamma^2 [r_{i+2} + \gamma V_\phi(s_{i+3}) - V_\phi(s_{i+2})]$$

$$+ \cdots + \gamma^{j-i} [r_j + \gamma V_\phi(s_{j+1}) - V_\phi(s_j)]$$

$\boxed{\text{Generalized Advantage estimator}}$

$$= \delta_i + \gamma \delta_{i+1} + \gamma^2 \delta_{i+2} + \cdots + \gamma^{j-i} \delta_j$$

$$A_\tau(\lambda) = \delta_\tau = r_\tau + \cancel{\gamma V_\phi(s_{\tau+1})}^{\to 0} - V_\phi(s_\tau)$$

For $m < \tau$: $\boxed{A_m(\lambda)} = \delta_m + \lambda \gamma A_{m+1}(\lambda)$, where $\delta_m = r_m + \gamma V_\phi(s_{m+1}) - V_\phi(s_m)$

$$\boxed{Q^{\pi_k}(s_m, a_m) - V_\phi(s_m)} \qquad \boxed{Q^{\pi_F}(s_{m+1}, a_{m+1}) - V_\phi(s_{m+1})}$$

GAE (Generalized Advantage Estimation)

Let $(s_1, a_1, r_1, s_1', s_2, a_2, r_2, s_2', \dots, s_N, a_N, r_N, s_N')$ be a trajectory collected with policy π, where $s_i' = s_{i+1}$ if s_i' is not a terminal state, and $s_{i+1} \sim \rho$ otherwise.

Also, let V_ϕ be a given state-value estimation.

Then the following procedure can estimate $A_i \approx Q^\pi(s_i, a_i) - V_\phi(s_i)$

Parameter: λ (controlling variance-bias tradeoff)

For $i = N, N-1, \dots, 1$:

If s_i' is a terminal state:

$\delta_i = r_i - V_\phi(s_i)$

$A_i = \delta_i$

Else:

$\delta_i = r_i + V_\phi(s_{i+1}) - V_\phi(s_i)$

$A_i = \delta_i + \lambda\gamma A_{i+1}$

Schulman et al. High-Dimensional Continuous Control Using Generalized Advantage Estimation. 2015.

Using GAE in the Policy Iteration Framework

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Evaluate $Z_k(s, a) \approx Q^{\pi_{\theta_k}}(s, a) - V_\phi(s)$ for $s = s_1, \ldots, s_N$ and all a

$$\Rightarrow\ Z_k(s_i, a) = \frac{\mathbb{I}\{a_i = a\}}{\pi_{\theta_k}(a|s_i)}\,\hat{A}_k(s_i, a_i)$$

Policy Evaluation

Update θ_{k+1} from θ_k using the estimator $\{Z_k(s_i, a)\}_{i=1}^N$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

Training the Baseline V_ϕ (in iteration k)

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - r_1 - \gamma V_{\phi_k}(s_i') \right)^2 \Bigg|_{\phi=\phi_k} \qquad \text{TD(0)}$$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - G_i(\lambda; \phi_k) \right) \Bigg|_{\phi=\phi_k} \quad \text{where } G_i(\lambda; \phi_k) = A_i(\lambda; \phi_k) + V_{\phi_k}(s_i) \qquad \text{TD}(\lambda)$$

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - \sum_{h=i}^{\tau(i)} \gamma^{h-i} r_i \right) \Bigg|_{\phi=\phi_k} \qquad \text{TD(1)}$$

Approximate Policy Iteration and Variants

PPO

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Requires training a separate V_ϕ

Define $Z_k(s_i, a) = \frac{\mathbb{I}\{a_i = a\}}{\pi_{\theta_k}(a|s_i)}\,\hat{A}_k(s_i, a_i)$

Use another inner for-loop to solve the argmax with gradient ascent

$$\theta_{k+1} = \operatorname*{argmax}_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{a} \pi_\theta(a|s_i) Z_k(s_i, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_{\theta_k}(\cdot\,|s_i), \pi_\theta(\cdot\,|s_i)\big) \right) \right\}$$

$$\approx \operatorname*{argmax}_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)}\,\hat{A}_k(s_i, a_i) - \frac{1}{\eta} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \right) \right\}$$

Schulman et al. Proximal Policy Optimization Algorithms. 2017.

PPO with Clipping

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\boxed{} - \frac{1}{\eta} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} - 1 - \log \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \right) \right\}$$

$$\boxed{\min \left\{ \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \hat{A}_k(s_i, a_i), \qquad \operatorname{clip}_{[1-\epsilon, 1+\epsilon]} \left(\frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_k}(a_i|s_i)} \right) \hat{A}_k(s_i, a_i) \right\}}$$

Schulman et al. Proximal Policy Optimization Algorithms. 2017.

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

In standard A2C, $\hat{A}_k(s_i, a_i) = r_i + \gamma V_{\phi_k}(s_i') - V_{\phi_k}(s_i)$ (GAE estimator with $\lambda = 0$)

and ϕ_k is trained with TD(0):

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_{\color{red}{\phi}}(s_i) - r_1 - \gamma V_{\color{blue}{\phi_k}}(s_i')\right)^2 \Bigg|_{\phi=\phi_k}$$

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

In standard PG, $\hat{A}_k(s_i, a_i) = \sum_{h=i}^{\tau(i)} \gamma^{h-i} r_i - V_{\phi_k}(s_i)$ (GAE estimator with $\lambda = 1$)

A2C (Advantage Actor Critic) / PG

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

$$\theta_{k+1} = \theta_k - \eta \left(\nabla_\theta \log \pi_\theta(a_i|s_i)\right)\Big|_{\theta=\theta_k} \hat{A}_k(s_i, a_i)$$

In general, one can use GAE with any λ to calculate $\hat{A}_k(s_i, a_i)$, with V_ϕ calculated from TD(λ') with any λ'.

Summary: Algorithms based on Policy Iteration

- The algorithms are almost the same as those we introduced for contextual bandits
  - PPO / NPG
  - A2C / PG
- The only change is replacing $r(x_i, a_i) - b(x_i)$ by Advantage Estimator:
  - $\lambda = 0$: $\quad r(s_i, a_i) + \gamma V_\phi(s_{i+1}) - V_\phi(s_i)$
  - $\lambda = 1$: $\quad r(s_i, a_i) + \gamma r(s_{i+1}, a_{i+1}) + \gamma^2 r(s_{i+2}, a_{i+2}) + \cdots + \gamma^{\tau - i} r(s_\tau, a_\tau) - V_\phi(s_i)$
  - Any $\lambda \in [0,1]$: calculated by the GAE procedure
- The baseline $V_\phi(s)$ tries to track $V^{\pi_\theta}(s)$ where π_θ is the current policy
  - It is trained with a separate procedure TD(λ')

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_{\phi}(s_i) - r_1 - \gamma V_{\phi_k}(s_i') \right)^2 \Bigg|_{\phi = \phi_k} \qquad \text{TD(0)}$$