

RL: POLICY GRADIENT METHODS

Ошибка аппроксимации

DQN обучается на основе ошибки:

$$L \approx E \left[Q(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') \right]$$

	True	A	B
$Q(s_0, a_0)$	1	1	2
$Q(s_0, a_1)$	2	2	1
$Q(s_1, a_0)$	3	3	3
$Q(s_0, a_1)$	100	50	100

Вопрос: какой из этих алгоритмов лучше?

Ошибка аппроксимации

DQN обучается на основе ошибки:

$$L \approx E \left[Q(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') \right]$$

	True	A	B
$Q(s_0, a_0)$	1	1	2
$Q(s_0, a_1)$	2	2	1
$Q(s_1, a_0)$	3	3	3
$Q(s_0, a_1)$	100	50	100

Политика
лучше

Меньше MSE

Ошибка аппроксимации

DQN обучается на основе ошибки:

$$L \approx E \left[Q(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') \right]$$

	True	A	B
$Q(s_0, a_0)$	1	1	2
$Q(s_0, a_1)$	2	2	1
$Q(s_1, a_0)$	3	3	3
$Q(s_0, a_1)$	100	50	100

Q learning выберет более плохую политику B

Политика лучше

Меньше MSE

Выводы

Очень часто q значения найти сложнее чем оптимальное действие

Мы можем избежать обучения q функции, заменив это обучением политики $\pi_{\theta}(a|s)$

Вопрос: какой алгоритм мы уже проходили на основе политики?

$\text{argmax}[$
Q(s, pet the tiger)
Q(s, run from tiger)
Q(s, provoke tiger)
Q(s, ignore tiger)

$]$



$$\pi(run|s)=1$$



Политика

2 типа

☐ Детерминистическая политика

$$a = \pi_{\theta}(s)$$

☐ Стохастическая политика

$$a \sim \pi_{\theta}(s)$$

Вопрос: в каких случаях стохастическая политика лучше?

Политика

2 типа

❑ Детерминистическая политика

Генетические алгоритмы
Детерминистический градиент политики

Одинаковый действий каждый раз
 $a = \pi_{\theta}(s)$

❑ Стохастическая политика

Метод кросс энтропии
Градиент политики

Семплирование помогает исследовать
 $a \sim \pi_{\theta}(s)$

Вопрос: как представить политику в непрерывном случае?


Политика

2 типа

☐ Детерминистическая политика

Генетические алгоритмы
Детерминистический градиент политики

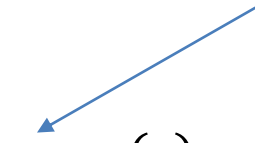
Одинаковый действий каждый раз


$$a = \pi_{\theta}(s)$$

☐ Стохастическая политика

Метод кросс энтропии
Градиент политики

Семплирование помогает исследовать


$$a \sim \pi_{\theta}(s)$$

Категориальная, нормальная, смесь нормальных, и пр.

Два подхода

❑ Value based:

Ищем функцию ценности

$$Q_{\theta}(s, a) \quad \text{или} \quad V_{\theta}(s)$$

Инференс политики

$$a = \underset{a}{\operatorname{argmax}} Q_{\theta}(s, a)$$

❑ Policy based:

Находим политику в явном виде

$$\pi_{\theta}(a|s) \quad \text{или} \quad \pi_{\theta}(s) \rightarrow a$$

Метод кроссэнтропии

- Инициализировать веса

- Цикл:

Сэмплируем N лучших сессий

$$elite = [(s_0, a_0), (s_1, a_1), \dots, (s_k, a_k)]$$

$$w_{i+1} = w_i + \alpha \nabla \left[\sum_{s_i, a_i \in Elite} \log \pi_{w_i}(a_i | s_i) \right]$$

Policy gradient: основная идея

- ❑ Почему так сложно?
- ❑ Будем максимизировать reward напрямую

Цель

□ Ожидаемый reward:

$$J = \underset{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a) \\ \dots}}{\mathbb{E}} R(s, a, s', a', \dots)$$

□ Ожидаемый reward с учетом дисконта:

$$J = \underset{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}}{\mathbb{E}} G(s, a)$$

Цель

□ Ожидаемый reward:

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a) \\ \dots}} R(s, a, s', a', \dots)$$

□ Ожидаемый reward с учетом дисконта: $G(s, a) = r + \gamma G(s', a')$

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}} G(s, a)$$

Цель

□ Для простоты рассмотрим один шаг

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_{\theta}(a|s) R(s, a) da ds$$

Цель

□ Для простоты рассмотрим один шаг

$$J = \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_{\theta}(a|s) R(s, a) da ds$$

Частота посещения
состояния


Reward для 1 шага сессии

Вопрос: как вычислить?

Цель

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_\theta(a|s) R(s, a) da ds$$


Семплируем N сессий
текущей политики


$$J \approx \frac{1}{N} \sum_{i=0}^N R(s, a)$$

Цель

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_{\theta}(a|s) R(s, a) da ds$$

Семплируем N сессий
текущей политики


$$J \approx \frac{1}{N} \sum_{i=0}^N R(s, a)$$

Как сейчас оптимизировать политику?

Цель

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_\theta(a|s) R(s, a) da ds$$

$$J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s, a \in \mathcal{Z}_i} R(s, a)$$

Мы не знаем как сделать расчет $\frac{\partial J}{\partial \theta}$

Оптимизация

Конечные разности

- Небольшое изменение политики и оценка:

$$\nabla J \approx \frac{J_{\theta+\epsilon} - J_{\theta}}{\epsilon}$$

Стохастическая оптимизация

- Метод кросс энтропии
- Максимизация элитных действий

Оптимизация

Конечные разности

- Небольшое изменение политики и оценка:

$$\nabla J \approx \frac{J_{\theta+\epsilon} - J_{\theta}}{\epsilon}$$

Стохастическая оптимизация

- Метод кросс энтропии
- Максимизация элитных действий

Какие видите проблемы?

Оптимизация

Конечные разности

- Небольшое изменение политики и оценка:

$$\nabla J \approx \frac{J_{\theta+\epsilon} - J_{\theta}}{\epsilon}$$

Очень шумно, особенно
если оба J семплируются

Стохастическая оптимизация

- Метод кросс энтропии
- Максимизация элитных действий

“квантильная сходимость”
Проблемы со стохастическими
MDP

Какие видите проблемы?

Цель

$$J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} R(s, a) = \int_s p(s) \int_a \pi_\theta(a|s) R(s, a) da ds$$

Что бы мы хотели:

- аналитический градиент
- простая/стабильная аппроксимация

Логарифмический трюк

- Простая математика

$$\nabla \ln \pi(z) = ???$$

Логарифмический трюк

- Простая математика

$$\nabla \ln \pi(z) = \frac{1}{\pi(z)} \nabla \pi(z)$$

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$

Policy gradient

- Аналитический вывод

$$\nabla J = \int_s p(s) \int_a \nabla \pi_\theta(a|s) R(s, a) da ds$$

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$


Policy gradient

- Аналитический вывод

$$\nabla J = \int_s p(s) \int_a \nabla \pi_\theta(a|s) R(s, a) da ds$$

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$


$$\nabla J = \int_s p(s) \int_a \pi_\theta(a|s) \nabla \ln \pi_\theta(a|s) R(s, a) da ds$$

На что похожа последняя формула?

Policy gradient

- Аналитический вывод

$$\nabla J = \int_s p(s) \int_a \nabla \pi_\theta(a|s) R(s, a) da ds$$

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$


$$\nabla J = \int_s p(s) \int_a \pi_\theta(a|s) \nabla \ln \pi_\theta(a|s) R(s, a) da ds$$

Мат ожидание

REINFORCE (бандит)

□ Инициализировать веса θ_0 NN

□ В цикле

- Семплируем N сессий по текущей политике
- Оцениваем градиент политики

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$

$$\nabla J = \frac{1}{N} \sum_{i=0}^N \nabla \ln \pi_{\theta}(a|s) R(s, a)$$

- Обновляем веса

$$\theta_{i+1} \rightarrow \theta_i + \alpha \nabla J$$

Дисконтированное вознаграждение

□ Меняем R на Q ($Q(s, a) = E[G(s, a)]$)

$$\nabla J = \int_s p(s) \int_a \nabla \pi_\theta(a|s) Q(s, a) da ds$$

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$


$$\nabla J = \int_s p(s) \int_a \pi_\theta(a|s) \nabla \ln \pi_\theta(a|s) Q(s, a) da ds$$

REINFORCE (с дисконтированием)

□ Policy Gradient

$$\nabla J = \mathop{\mathbb{E}}_{\substack{s \sim p(s) \\ a \sim \pi_{\theta}(s|a)}} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

□ Аппроксимация семплированием

$$\nabla J \approx 1/N \sum_{i=0}^N \sum_{s, a \in z_j} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

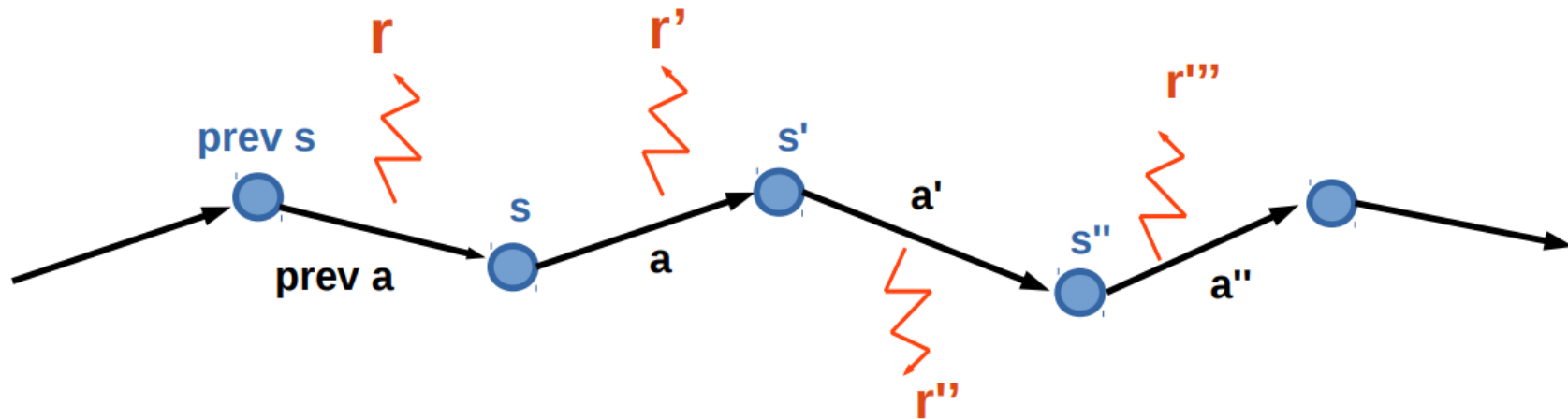
REINFORCE алгоритм

- ❑ Мы можем оценить Q используя G

$$Q_{\pi}(s_t, a_t) = E_s[G(s_t, a_t)]$$

- ❑ Аппроксимация семплированием

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots$$



Дисконтированный reward

$$\begin{aligned} G_t &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots = r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots) \\ &= r_t + \gamma G_{t+1} \end{aligned}$$

Мы можем использовать эту формулу для расчета всех rewards за линейное время

REINFORCE алгоритм

□ Инициализировать веса θ_0 NN

□ В цикле

- Семплируем N сессий по текущей политике
- Оцениваем градиент политики

$$\pi(z) \nabla \ln \pi(z) = \nabla \pi(z)$$

$$\nabla J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

- Обновляем веса

$$\theta_{i+1} \rightarrow \theta_i + \alpha \nabla J$$

REINFORCE алгоритм

- Инициализировать веса θ_0 NN

Q: это off или on policy?

- В цикле

- Семплируем N сессий по текущей политике
- Оцениваем градиент политики

$$\nabla J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

- Обновляем веса

$$\theta_{i+1} \rightarrow \theta_i + \alpha \nabla J$$

REINFORCE алгоритм

□ Инициализировать веса θ_0 NN

□ В цикле

Действие по текущей политике
= on policy

- Семплируем N сессий по текущей политике
- Оцениваем градиент политики

$$\nabla J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

- Обновляем веса

$$\theta_{i+1} \rightarrow \theta_i + \alpha \nabla J$$

Value-based vs policy based

Value-based	Policy-based
Q-learning, SARSA	REINFORCE, CEM
Решает сложные задачи	Решает простые задачи
Искусственное исследование	Встроенное исследование
Обучается на основе частичного опыта (временная разница)	Встроенная стохастичность
Оценивает стратегию бесплатно	Поддержка непрерывного пространства действий
	Учится только на полных сессиях?

Value-based vs policy based

Value-based	Policy-based
Q-learning, SARSA	REINFORCE, CEM
Решает сложные задачи	Решает простые задачи
Искусственное исследование	Встроенное исследование
Обучается на основе частичного опыта (временная разница)	Встроенная стохастичность
Оценивает стратегию бесплатно	Поддержка непрерывного пространства действий
	Учится только на полных сессиях?

REINFORCE: основы

□ Инициализировать веса θ_0 NN

□ В цикле

- Семплируем N сессий по текущей политике
- Оцениваем градиент политики

$$\nabla J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} \nabla \ln \pi_{\theta}(a|s) Q(s, a)$$

Что лучше для обучения: случайное действие в хорошем состоянии или хорошее действие в плохом состоянии?

REINFORCE: основы

Мы можем вычесть произвольную функцию $b(s)$

$$\begin{aligned}\nabla J &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) (Q(s|a) - b(s)) = \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) Q(s|a) - \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) b(s)\end{aligned}$$

REINFORCE: основы

Мы можем вычесть произвольную функцию $b(s)$

$$\begin{aligned}\nabla J &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) (Q(s|a) - b(s)) = \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) Q(s|a) - \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) b(s)\end{aligned}$$

Мы можем упростить второй член?

Заметим, что $b(s)$ не зависит от a

REINFORCE: основы

Мы можем вычесть произвольную функцию $b(s)$

$$\begin{aligned}\nabla J &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) (Q(s|a) - b(s)) = \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) Q(s|a) - \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) b(s) \\ &\quad \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) b(s) = b(s) \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) = 0\end{aligned}$$

REINFORCE: основы

Мы можем вычесть произвольную функцию $b(s)$

$$\begin{aligned}\nabla J &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) (Q(s|a) - b(s)) = \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) Q(s|a) - \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) b(s) \\ &= \mathbb{E}_{\substack{s \sim p(s) \\ a \sim \pi_\theta(s|a)}} \nabla \ln \pi_\theta(a|s) Q(s|a)\end{aligned}$$

Направление градиента не меняется

REINFORCE: основы

- ❑ Направление градиента ∇J остается таким же
- ❑ Дисперсия может меняться

Дисперсия градиента: $Var[Q(s, a) - b(s)]$ - случайная величина

$$Var[Q(s, a)] - 2Cov[Q(s, a), b(s)] + Var[b(s)]$$

REINFORCE: основы

- ❑ Направление градиента ∇J остается таким же
- ❑ Дисперсия может меняться

Дисперсия градиента: $Var[Q(s, a) - b(s)]$ - случайная величина

$$Var[Q(s, a)] - 2Cov[Q(s, a), b(s)] + Var[b(s)]$$

Если $b(s)$ коррелирует с $Q(s, a)$, дисперсия уменьшается

REINFORCE: основы

- ❑ Направление градиента ∇J остается таким же
- ❑ Дисперсия может меняться

Дисперсия градиента: $Var[Q(s, a) - b(s)]$ - случайная величина

$$Var[Q(s, a)] - 2Cov[Q(s, a), b(s)] + Var[b(s)]$$

Вопрос: какую $b(s)$ можем использовать?

REINFORCE: основы

- ❑ Направление градиента ∇J остается таким же
- ❑ Дисперсия может меняться

Дисперсия градиента: $Var[Q(s, a) - b(s)]$ - случайная величина

$$Var[Q(s, a)] - 2Cov[Q(s, a), b(s)] + Var[b(s)]$$

Наивный подход: b = скользящая средняя Q по всем (s, a) ,

$$Var[b(s)] = 0, Cov[Q, b] > 0$$

REINFORCE: основы

- ❑ Направление градиента ∇J остается таким же
- ❑ Дисперсия может меняться

Дисперсия градиента: $Var[Q(s, a) - b(s)]$ - случайная величина

$$Var[Q(s, a)] - 2Cov[Q(s, a), b(s)] + Var[b(s)]$$

Наивный базовый подход: b = скользящая средняя Q по всем (s, a) ,
 $Var[b(s)]=0$, $Cov[Q, b]>0$

REINFORCE: основы

□ Более удачный выбор $b(s) = V(s)$

$$\nabla J \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in z_j} \nabla \ln \pi_{\theta}(a|s) (Q(s,a) - V(s))$$

Вопрос: как будем предсказывать $V(s)$?

Actor-critic

- ❑ Будем искать (обучать) $V(s)$ и $\pi_{\theta}(a|s)$
- ❑ И надеемся, что получим лучшее решение



Advantage actor-critic

- ❑ Идея: Обучаем $V_\theta(s)$ и $\pi_\theta(a|s)$
- ❑ Используем $V_\theta(s)$, чтобы быстрее обучить $\pi_\theta(a|s)$

Вопрос: как мы можем оценить $A(s,a)$ из (s,a,r,s') и V функцию

Advantage actor-critic

- ❑ Идея: Обучаем $V_\theta(s)$ и $\pi_\theta(a|s)$
- ❑ Используем $V_\theta(s)$, чтобы быстрее обучить $\pi_\theta(a|s)$

$$A(s, a) = Q(s, a) - V(s)$$

$$Q(s, a) = r + \gamma V(s')$$

$$A(s, a) = r + \gamma V(s') - V(s)$$

Advantage actor-critic

- ❑ Идея: Обучаем $V_\theta(s)$ и $\pi_\theta(a|s)$
- ❑ Используем $V_\theta(s)$, чтобы быстрее обучить $\pi_\theta(a|s)$

$$A(s, a) = Q(s, a) - V(s)$$

$$Q(s, a) = r + \gamma V(s')$$

$$A(s, a) = r + \gamma V(s') - V(s)$$

Также: n-step
версия

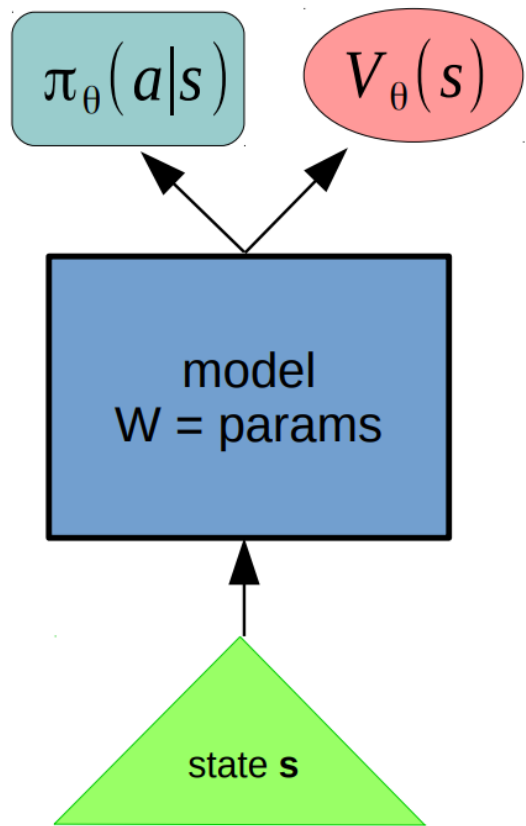
Advantage actor-critic

- ❑ Идея: Обучаем $V_\theta(s)$ и $\pi_\theta(a|s)$
- ❑ Используем $V_\theta(s)$, чтобы быстрее обучить $\pi_\theta(a|s)$

$$A(s, a) = r + \gamma V(s') - V(s)$$

$$\nabla J_{actor} \approx \frac{1}{N} \sum_{i=0}^N \sum_{s, a \in \mathcal{Z}_j} \nabla \ln \pi_\theta(a|s) A(s, a)$$

Advantage actor-critic



□ Улучшение политики

$$\nabla J_{actor} \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} \nabla \ln \pi_{\theta}(a|s) A(s, a)$$

□ Улучшение ценности

$$\nabla J_{critic} \approx \frac{1}{N} \sum_{i=0}^N \sum_{s,a \in \mathcal{Z}_j} (V_{\theta}(s) - [r + \gamma V_{\theta}(s')])$$

Непрерывное пространство действий

- ☐ Автономный автомобиль
- ☐ Управление роботом

Как изменим алгоритм?

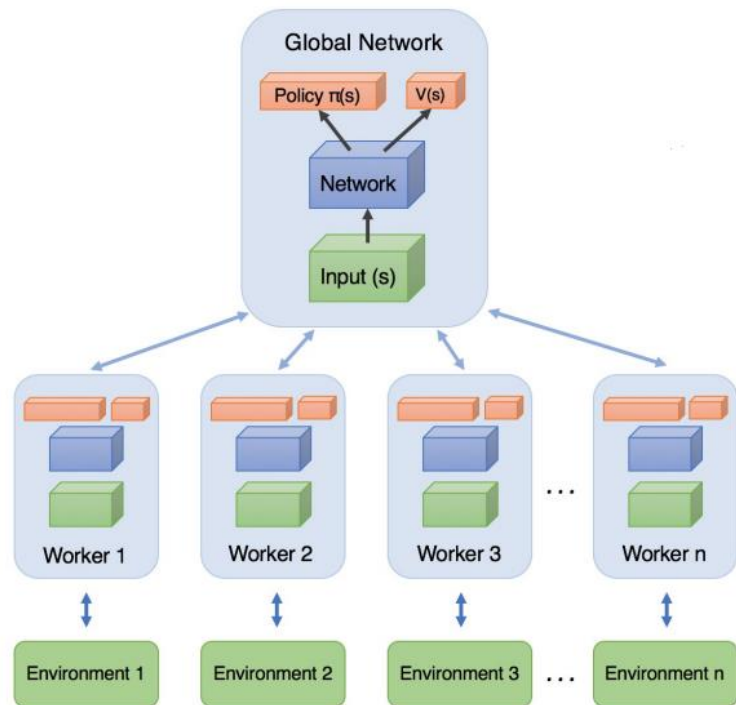
Непрерывное пространство действий

- ☐ Автономный автомобиль
- ☐ Управление роботом

Как изменим алгоритм?

Рассмотрим другую формулу для политики, например нормальное распределение

Асинхронный advantage actor-critic

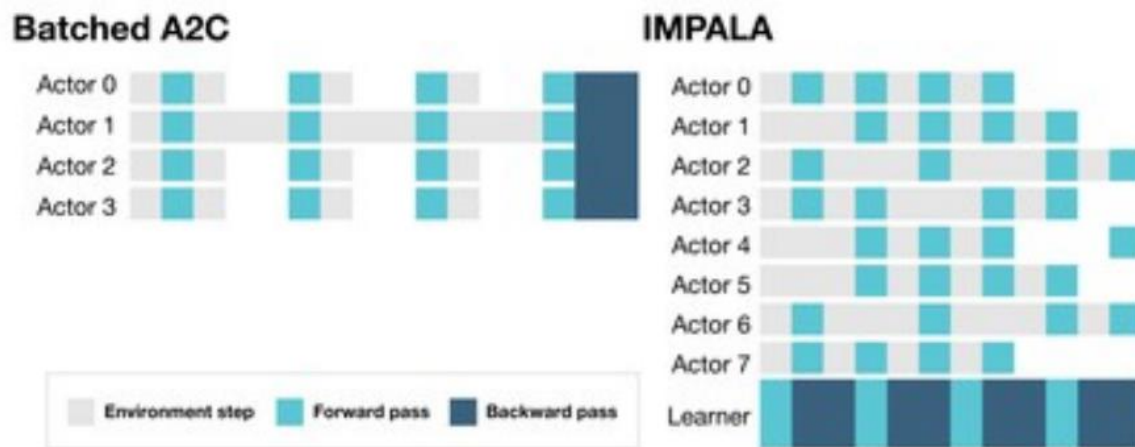


- Параллельные игровые сессии
- Асинхронное обучение на нескольких процессорах
- Без experience replay
- LSTM policy
- N-step advantage
- Без experience replay

<https://arxiv.org/abs/1602.01783>

IMPALA

- С массовым параллелизмом
- Раздельные процессы actor и critic
- Небольшое повторение опыта с выборкой по важности



<https://arxiv.org/abs/1802.01561>