Разработка интеллектуальных агентов компьютерных игр

Повторение: Value-based Methods

- Аппроксимация action-value функции через нейросеть

$$Q(s,a) \approx Q_\theta(s,a)$$

- Берем действия, которые максимизируют функцию

Повторение: Value-based Methods

Плюсы

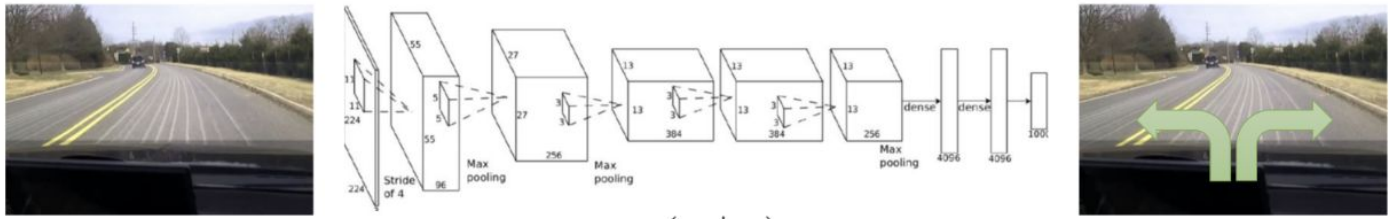
- Легко генерировать политику
- Достаточно близок к реальным целям
- Легко понимаемый метод, есть хорошие алгоритмы

Минусы

- Все еще не реальные цели
- Может зафокуситься на релевантных данных
- Небольшие ошибки в value могут привести к огромным ошибкам в политике

Параметрическая политика

- $$\pi_\theta(a \mid s) = \mathbb{P}(a_t = a \mid s_t = s; \theta)$$



$\mathbf{s}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ $\mathbf{a}_t$

Параметрическая политика

$$\theta^* = argmax_\theta J(\theta) = argmax_\theta \mathbb{E}_{p_\theta(\tau)}[\sum_{t=0}^{T} \gamma^t R_t] = argmax_\theta \mathbb{E}_{p_\theta(\tau)}[G(\tau)]$$

$$J(\theta) = \mathbb{E}_{p_\theta(\tau)}[G(\tau)] = \int p_\theta(\tau) G(\tau) d\tau$$

$$p_\theta(\tau) = p(s_0)\pi_\theta(a_0 \mid s_0)p(s_1 \mid s_0, a_0)\ldots$$

Градиент

$$p_\theta(\tau) = p(s_0)\pi_\theta(a_0 \mid s_0)p(s_1 \mid s_0, a_0)\dots$$

$$\nabla J(\theta) = \nabla \mathbb{E}_{p_\theta(\tau)}[G(\tau)] = \int \nabla p_\theta(\tau) G(\tau) d\tau$$

Log-derivative trick: $\nabla p_\theta(\tau) = p_\theta(\tau)\dfrac{\nabla p_\theta(\tau)}{p_\theta(\tau)} = p_\theta(\tau)\nabla \log p_\theta(\tau) =$

$$= p_\theta(\tau) \sum_{t=0}^{T} \nabla \log \pi_\theta(a_t \mid s_t)$$

$$\nabla J(\theta) = \nabla \mathbb{E}_{p_\theta(\tau)}[G(\tau)] = \mathbb{E}_{p_\theta(\tau)}[\nabla \log p_\theta(\tau) G(\tau)]$$

REINFORCE (1992)

- Берем N траекторий из среды с текущей политикой
- Считаем градиент с Монте-Карло

$$\nabla J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{t=0}^{T} \nabla \log \pi_\theta(a_{i,t} \mid s_{i,t}) G(\tau_i) \right]$$

- Делаем шаг

$$\theta_{k+1} = \theta_k + \alpha \nabla J(\theta_k)$$

REINFORCE (1992)

- Не используем буфер
- Старые сэмлы не используются в обновлении

Policy-based

Плюсы

- Оптимизирует реальные цели
- Легко расширяется
- Изучается стохастической политикой
- Нет знаний до
- Легко обучается напрямую

Минусы

- Может застрять в локальном оптиуме
- Менее эффективный
- Высокая дисперсия

Actor-Critic

- А давайте смерджим два метода и уменьшим дисперсию
- Actor - определяет действие из политики
- Critic - с помощью value функции оценивает на сколько хорошо актор выбирает действия

Actor-Critic