Reinforcement Learning 1

Denis Basharin

Постановка задачи

Учимся принимать хорошие последовательности действий, без понимания фундаментального устройства мира

Задача включает в себя

- Optimization
- Generalization
- Exploration
- Delayed consequences

RL vs Supervised Learning

RL

- **Optimization**

  Sparse reward function

- **Generalization**

  yep

- **Exploration**

  unknown consequences for actions

- **Delayed consequences**

  reward for decisions may come afterwards

Supervised Learning

- **Optimization**

  differentiable loss function

- **Generalization**

  yep

- **Exploration**

  true labels for dataset

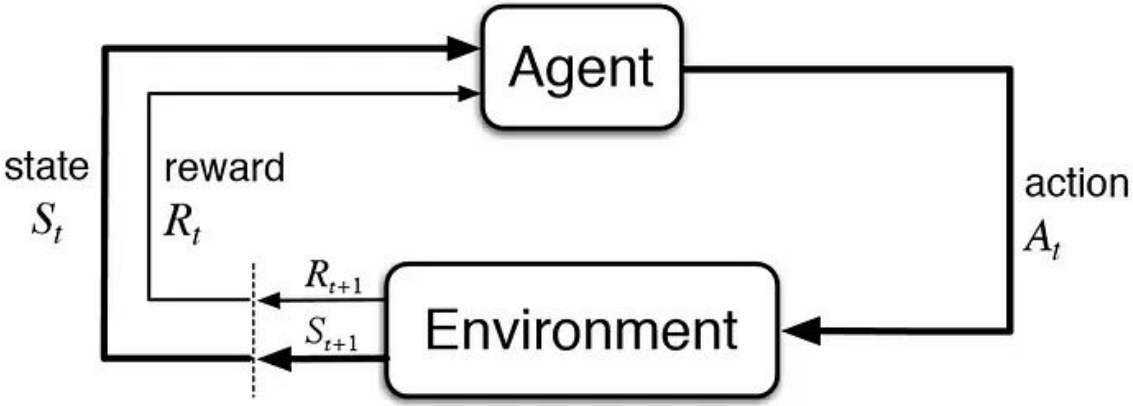
- **Delayed consequences**

  none

Основные определения

Основные определения

- Среда (*environment*)
- Агент (*agent*)
- Действие (*action*)
- Состояние (*state*)
- Награда (*reward*)



- Наблюдение (*observation)*

Основные определения

Марковское свойство:

$$p(S_{t+1}|S_t, A_t) = p(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0)$$

Состояние state полностью определяет все неслучайные процессы окружения

Основные определения

- Политика (*policy*)

$$\pi(a|s)$$

- v-function (*state-value function*)

$$v_\pi(s) = E\left[G_t|S_t = s\right] = E\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s\right]$$

- q-function (*action-value function)*

$$q_\pi(s,a) = E\left[G_t|S_t = s, A_t = a\right] = E\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle|S_t = s, A_t = a\right]$$

Кросс-энтропийный метод

Общий фреймворк

1. Инициализируем политику

2. Семплируем n сессий
3. Оцениваем все n сессий по v-функции
4. Выбираем m элитных сессий
5. Обновляем политику на основании данных элитных сессий

6. Повторяем 2-5 пока не сойдемся

Табличный случай

Конечное число state, action

- политика - таблица state x action

- семпл - обычный семпл дискретного распределения

- $$\pi^{new}(a|s) = \frac{\sum_{a_i, s_i \in Elites} I\{a_i = a, s_i = s\}}{\sum_{a_i, s_i \in Elites} I\{s_i = s\}}$$

А если слишком много возможных состояний?

Конечное число state, action

- политика - классификатор $state \rightarrow action\ probabilities$

- семпл - обычный семпл дискретного распределения

- $\pi^{new}(a|s) = argmax_{\pi} \sum_{a,s \in Elites} \log \pi(a|s)$

  argmax делать сложно, оптимизируем моделью

А если слишком много возможных состояний?

Конечное число state, action

- политика - классификатор $state \rightarrow action\ probabilities$

  model = RandomForestClassifier()

- семпл - обычный семпл дискретного распределения

- $\pi^{new}(a|s) = argmax_{\pi} \sum_{a,s \in Elites} \log \pi(a|s)$

  model.fit(elite_states, elite_actions)

А если и количество действий континуально?

Зафиксируем распределение action при state

a|s $\sim N(\mu_\theta(s), const)$ - нормальное распределение

$\sim P_\theta(s)$ - любое параметризованное распределение

Предсказываем распределение с помощью нейросети

А если и количество действий континуально?

Конечное число state, action

- политика - регрессор $state \rightarrow action\ values$

- семпл нормального распределения

- $\pi^{new}(a|s) = argmax_\theta \sum_{a_i, s_i \in Elites} \log \pi_\theta(a|s)$

  argmax решить сложно, поэтому используем приближенное решение

А если и количество действий континуально?

Конечное число state, action

- политика - регрессор $state \rightarrow action\ values$

  model = RandomForestRegressor()

- семпл нормального распределения

- $\pi^{new}(a|s) = argmax_\theta \sum_{a_i, s_i \in Elites} \log \pi_\theta(a|s)$

  model.fit(elite_states, elite_actions)

Другой подход

Можем считать модель π_w и добиваться exploration также через семплирование весов:

1) $w \sim N(\mu, \sigma)$

2) sample and evaluate n models π_{w_i}
3) select top m examples, reassign μ and σ

4) repeat 2-3

Примеры

https://github.com/udacity/deep-reinforcement-learning/tree/master/cross-entropy

Action

force ∈ [-1;1]

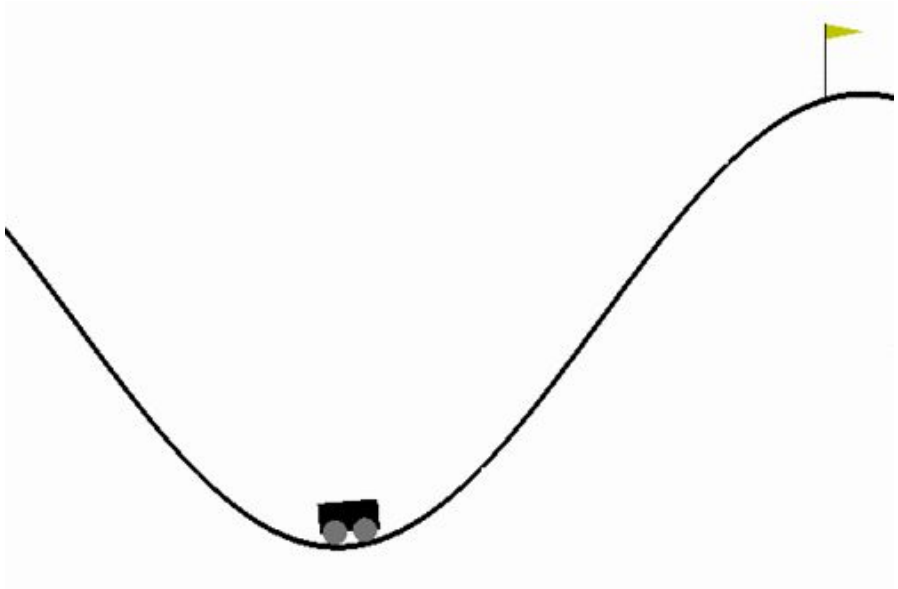
State

(velocity, position)

Reward

$-0.1\ action^2$

100 if goal is reached



Потенциальные сложности в RL

- Reward hacking
- Exploitation vs Exploration
- Credit assignment problem

Источники

https://www.youtube.com/watch?v=FgzM3zpZ55o&list=PLoROMvodv4rOSOPzutgyCTapiGlY2Nd8u

https://www.youtube.com/watch?v=BwLIPEUkjxQ&list=PL4_hYwCyhAvY7k32D65q3xJVo8X8dc3Ye&index=8

https://www.youtube.com/watch?v=JgvyzIkgxF0

https://www.youtube.com/watch?v=eMxOGwbdqKY

example: https://github.com/udacity/deep-reinforcement-learning/tree/master/cross-entropy

picture: https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292