Захватывающая история о градиенте стратегии в RL

NSU Hackspace

5 августа 2022 г.

1 Основные понятия и задачи RL

Начнём разговор с описания окружающего ландшафта и посмотрим ещё раз на основные элементы RL.

Опр. *Траектория* $\tau = s_0, a_0, r_0, \ldots$ *– последовательность состояний* s_t*, между которыми перемещается агент, совершая действия* a_t *и получая вознаграждение* r_t*. Конечную траекторию будем также называть эпизодом.*

При выборе очередного действия, агент использует стратегию π_θ.

Опр. *Стратегия* π_θ *– отображение пространства состояний на пространство действий. Для каждого состояния* s_t *стратегия возвращает распределение вероятностей выбора того или иного действия* $\pi_\theta(s_t)$*.*

В роли стратегии π_θ выступает какая-то хитрая функция с набором параметров θ, покрутив которые можно обучить её нужному поведению. Для удобства мы можем считать, что это нейросеть.

Эффективность стратегии π_θ определяется вознаграждением, которое агент получил при движении вдоль траектории τ, порожденной π_θ.

Опр. *Функция вознаграждения* $R_t(\tau)$ *представляет собой сумму всех вознаграждений* r_t*, которые получил агент, начиная с момента времени* t *и до конца эпизода* T*. Коэффициент обесценивания* γ *регулирует влияние отдаленных шагов на текущий момент времени* t

$$R_t(\tau) = \sum_{t'=t}^{T} \gamma^{t'-t} r_t$$

Если траекторию рассматриваем целиком, то обозначение слегка упрощается (исчезает индекс t):

$$R(\tau) := R_0(\tau) = \sum_{t=0}^{T} \gamma^t r_t$$

В сложных средах траектории будут сильно различаться даже для одной и той же стратегии π_θ, поэтому для оценки поведения агента удобнее усреднять вознаграждение по набору траекторий.

Опр. *Целевая функция* $J(\pi_\theta)$ *– это среднее вознаграждение агента по всем траекториям, порожденным с помощью стратегии* π_θ*.*

$$J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[R(\tau)\right] = E_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Возникает естественное желание, изменяя параметры θ стратегии π_θ, максимизировать целевую функцию, чтобы добиться наибольшего вознаграждения для агента.

$$\max_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[R(\tau)\right]$$

Т.к. мы договорились считать π_θ нейросетью, то набор параметров θ – это просто веса сети.

2 Градиентные затруднения

В первом приближении карта местности определена. Цель: увеличить среднее вознаграждение, которое получает агент в процессе взаимодействия со средой, руководствуясь стратегией π_θ. Математическим языком эту цель можно выразить в виде задачи оптимизации

$$\max_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[R(\tau)\right] \tag{2.1}$$

Как это можно сделать? Первый ответ, который приходит в голову специалистам по ML, воспользоваться градиентом целевой функции $J(\pi_\theta)$ по набору параметров θ: $\nabla_\theta J(\pi_\theta)$! Схему градиентного подъема можно записать вот так

$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\pi_\theta)$$

Но как воспользоваться ей на практике?

Посмотрим внимательно на уравнение нашей задачи оптимизации (2.1). Ключевым элементом выражения является функция вознаграждения $R(\tau)$. Что же нам известно о ней? Как она зависит от θ?

К сожалению, мы не знаем об $R(\tau)$ почти ничего, она выступает в роли загадочного черного ящика. Выражение $R(\tau) = \sum_{t=0}^{T}\gamma^t r_t$ не получится продифференцировать по θ. Но связь между θ и $R(\tau)$ можно почувствовать, сэмплируя траектории $\tau\sim\pi_\theta$ и вычисляя вдоль них вознаграждение $R(\tau)$. То есть, нужная информация хранится в распределении $\tau\sim\pi_\theta$ и нужно найти способ до неё добраться.

3 Теорема о градиенте

Такой способ существует и называется теоремой о градиенте стратегии. Для удобства восприятия, мы сперва разберем её в упрощенных обозначениях, а затем подставим интересующие выражения и выведем основной результат.

Пусть определены следующие объекты: функция $f(x)$, условное (или параметризованное) распределение вероятностей $p(x|\theta)$ и математическое ожидание $E_{x\sim p(x|\theta)}\left[f(x)\right]$. Найдем градиент математического ожидания по параметру θ

$$\begin{aligned}
&\nabla_\theta E_{x\sim p(x|\theta)}\left[f(x)\right] = && \text{(сперва выпишем определение мат. ожидания)}\\
&= \nabla_\theta \int f(x)p(x|\theta)dx = && \text{(вносим } \nabla_\theta \text{ под знак интеграла)}\\
&= \int \nabla_\theta\left(f(x)p(x|\theta)\right)dx = && \text{(применяем оператор дифференцирования)}\\
&= \int f(x)\nabla_\theta p(x|\theta)dx = && \text{(домножим на единицу в виде } \tfrac{p(x|\theta)}{p(x|\theta)} \text{)}\\
&= \int f(x)p(x|\theta)\frac{\nabla_\theta p(x|\theta)}{p(x|\theta)}dx = && \text{(вносим } \tfrac{1}{p(x|\theta)} \text{ под оператор дифференцирования)}\\
&= \int f(x)p(x|\theta)\nabla_\theta \log p(x|\theta)dx = && \text{(собираем обратно выражение для мат. ожидания)}\\
&= E_{x\sim p(x|\theta)}\left[f(x)\nabla_\theta \log p(x|\theta)\right]
\end{aligned}$$

За счет смены порядка операторов дифференцирования и интегрирования, а также пары математических трюков удалось получить тождество, в котором градиент находится под интегралом и действует только на распределение $p(x|\theta)$, явным образом зависящее от параметра θ.

$$\nabla_\theta E_{x\sim p(x|\theta)}\left[f(x)\right] = E_{x\sim p(x|\theta)}\left[f(x)\nabla_\theta \log p(x|\theta)\right] \tag{3.1}$$

На функцию $f(x)$ при этом накладываются минимальные требования: мы хотим, чтобы она была интегрируемой. Тогда интеграл для мат. ожидания можно оценивать численно с помощью выборок $x \sim p(x|\theta)$.

Чтобы вернуться обратно к градиенту целевой функции $\nabla_\theta J(\pi_\theta)$, подставим в получившееся тождество (3.1) выражения для стратегии π_θ. Теперь в роли x выступает траектория τ, $f(x) = R(\tau)$, а $p(x|\theta) = p(\tau|\theta)$.

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[R(\tau)\nabla_\theta \log p(\tau|\theta)\right] \tag{3.2}$$

В целом выражение выглядит хорошо, но появился множитель $p(\tau|\theta)$, который выражает вероятность возникновения траектории τ, при условии, что стратегия задана набором параметров θ. Как его вычислить?

Мы знаем, что для каждого шага t вероятность действия a_t определяется стратегией и равна $\pi_\theta(a_t|s_t)$, а вероятность перехода между состояниями s_t и s_{t+1} определяется как $p(s_{t+1}|s_t, a_t)$. Тогда вероятность осуществления всей траектории τ равна произведению вероятностей этих переходов.

$$p(\tau|\theta) = \prod_{t\geq 0} p(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t) \tag{3.3}$$

Загадка множителя $p(\tau|\theta)$ разгадана, но попробуем ещё немного улучшить уравнение градиента, прологарифмировав выражение (3.3) и вычислив градиент по θ

$$\log p(\tau|\theta) = \log \prod_{t\geq 0} p(s_{t+1}|s_t, a_t)\pi_\theta(a_t|s_t) = \sum_{t\geq 0}\left(\log p(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t)\right)$$

$$\nabla_\theta \log p(\tau|\theta) = \nabla_\theta \sum_{t\geq 0}\left(\log p(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t)\right) = \nabla_\theta \sum_{t\geq 0} \log \pi_\theta(a_t|s_t)$$

$$\nabla_\theta \log p(\tau|\theta) = \nabla_\theta \sum_{t\geq 0} \log \pi_\theta(a_t|s_t) \tag{3.4}$$

В равенстве (3.4) удалось избавиться от вероятностей, связанных с переходами между состояниями, которые не зависят от стратегии, т.е. агент не может на них повлиять.

Собрав все множители под знаком суммы, получаем формулировку теоремы о градиенте стратегии.

Теорема 3.1 (теорема о градиенте стратегии)**.**

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[\sum_{t\geq 0} R(\tau)\nabla_\theta \log \pi_\theta(a_t|s_t)\right] \tag{3.5}$$

На практике имеет смысл выполнить ещё одно преобразование, хотя оно уже и не будет тождественным. В выражении (3.5) каждое слагаемое содержит коэффициент $R(\tau)$, зависящий от полной траектории. Т.е. информация о порядке действий a_t не используется. Это можно исправить, если считать вознаграждения по отдельности для каждого временного шага t. Для этого мы заменим $R(\tau)$ на $R_t(\tau)$.

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} R_t(\tau)\nabla_\theta \log \pi_\theta(a_t|s_t)\right] \tag{3.6}$$

Получившиеся выражения (3.5) и (3.6) в общем случае не получится аналитически проинтегрировать, но на самом деле этого и не требуется. Основная ценность уравнений градиента заключается в том, что с их помощью можно определять направление, в котором следует изменить стратегию π_θ, чтобы увеличить совокупное вознаграждение. Нужно просто собирать информацию о пройденных траекториях и пересчитывать значения суммы для всех шагов.

Последнее замечание касается множителя $\nabla_\theta \log \pi_\theta(a_t|s_t)$. Т.к. π_θ является по сути нейросетью, возвращающей вероятностные распределения для действий на каждом шаге, значит с вычислением градиента способен справиться любой DL-фреймворк.

4 Пример алгоритма

В настоящее время достаточно много продвинутых RL-алгоритмов в том или ином виде опираются на градиент стратегии. В качестве примера разберем простейший из них, который называется REINFORCE. Это алгоритм, использующий для обучения только актуальный опыт, то есть текущую траекторию.

```python
def reinforce(env, pi, n_episode, gamma=1.0):
    """
    Алгоритм REINFORCE
    @param env: имя среды Gym
    @param pi: сеть, аппроксимирующая стратегию
    @param n_episode: количество эпизодов
    @param gamma: коэффициент обесценивания
    """
    for episode in range(n_episode):
      log_probs = []
      rewards = []
      state = env.reset()
      while True:
        action, log_prob = pi.get_action(state)
        next_state, reward, is_done, _ = env.step(action)
        log_probs.append(log_prob)
        rewards.append(reward)
        if is_done:
           returns = []
           Gt, k = 0, 0
           for reward in rewards[::-1]:
              Gt += gamma ** k * reward
              k += 1
              returns.append(Gt)
        returns = torch.tensor(returns)
        pi.update(returns, log_probs)
```

5 Использованные источники