ОБУЧЕНИЕ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ С ПОДКРЕПЛЕНИЕМ

В. В. Реентович

Белорусский государственный университет, г. Минск;
vladrrv@gmail.com;
науч. рук. – В. В. Краснопрошин, д-р техн. наук, проф.

В работе рассмотрена проблема обучения глубоких нейронных сетей с подкреплением. Предложен подход, в котором для обучения агента используется не только среда, но и ее нейросетевая модель, обучаемая одновременно с агентом. Проведено экспериментальное сравнение предложенного подхода с известным методом Actor-Critic. Результаты экспериментов показали, что предложенный подход требует почти в два раза меньше обучающих данных, чем метод Actor-Critic.

Ключевые слова: искусственный интеллект, нейронные сети, глубокое обучение, обучение с подкреплением.

ВВЕДЕНИЕ

В условиях интенсивного развития информационных технологий актуальной является задача интеллектуального анализа данных, а также разработка интеллектуальных систем реального времени. Искусственные нейронные сети (ИНС) являются одним из основных инструментов для решения этих задач. При работе с ИНС наиболее трудоемкий процесс – это их обучение. Различают три вида обучения в зависимости от входной информации – с учителем, без учителя и с подкреплением. В работе рассматриваются способы обучения с подкреплением, которые применяются, когда информация подкрепляется сигналами от внешней среды. Преимущество такого обучения состоит в способности обучаемого агента адаптироваться к изменениям в среде, что позволяет применять его в интеллектуальных системах реального времени.

Существуют различные методы обучения глубоких нейросетей с подкреплением. Однако все эти методы имеют одну общую проблему, которая связана с большими объемами обучающей выборки. Для решения этой проблемы предлагается способ обучения с подкреплением на основе модели среды, которая обучается параллельно с агентом.

ПОДХОД К ОБУЧЕНИЮ НА ОСНОВЕ МОДЕЛИ СРЕДЫ

Описание алгоритма. Предлагаемый алгоритм обучения с подкреплением, основанный на модели среды, содержит две основные подзадачи:

обучение стратегии агента параллельно с обучением модели среды и планирование стратегии с помощью модели. Как стратегия, так и модель представлены соответствующими нейронными сетями.

Взаимодействие агента с детерминированной средой условно можно описать как некоторую функцию вида (1), где s – текущее состояние среды, a – выбранное агентом действие, s' – следующее состояние и r – награда за этот переход. При эпизодичном характере взаимодействия со средой в s' входит также флаг терминальности для завершения эпизода.

Задача модели среды $\hat{f}$, представленной нейросетью с обучаемыми параметрами θ, состоит в предсказании награды $\hat{r}$ и следующего состояние $\hat{s}'$ по текущему состоянию s и выбранному действию a (2).

$$f(s,a) = (r,s') \tag{1}$$

$$\hat{f}(s,a|\theta) = (\hat{r},\hat{s}') \tag{2}$$

В качестве цели оптимизации при обучении модели естественным образом можно взять степень различия пар (r,s') и $(\hat{r},\hat{s}')$, например $L_1(\hat{s}',s') + L_2(\hat{r},r)$, где L_i – некоторая функция потерь.

Для того, чтобы обучить такую модель, необходимо сначала получить выборку переходов (s,a,r,s'). Для этого был применен подход «experience replay memory» из метода DQN [1]: агент запоминает переходы при взаимодействии со средой, и нейросетевая модель итеративно обучается алгоритмом Adam на случайных выборках из памяти.

В предлагаемом подходе для обучения стратегии агента $\pi(a|s,\omega)$ за основу был взят метод A2C (Advantage Actor-Critic) [2]. Особенность состоит в том, что обучение актора и критика происходит не только на «реальных» эпизодах (в среде), но и на «воображаемых» эпизодах (в модели среды). Одновременное обучение стратегии и модели, а также планирование стратегии с помощью модели среды позволит уменьшить объем необходимых для обучения данных.

Таким образом, можно сформулировать общую схему метода обучения с подкреплением, основанного на модели среды:

шаг 1. Инициализировать параметры θ модели среды $\hat{f}$ и параметры ω стратегии агента π;

шаг 2. Исполнить стратегию, получить опыт взаимодействия с реальной средой (множество переходов вида (s,a,r,s'));

шаг 3. Использовать накопленный в памяти опыт для обучения как модели среды, так и стратегии агента (актора и критика);

шаг 4. Использовать модель среды для планирования стратегии;

шаг 5. Повторять шаги 2-4 необходимое число раз.

Модель среды. На вход нейросети, отвечающей за модель среды, поступает конкатенированная пара векторов (s, a), а на выходе нейросеть выдает конкатенированную пару $(\hat{r}, \hat{s}')$.

Нейросеть модели среды включает в себя два скрытых слоя по 128 нейронов в каждом. В качестве функции активации для этих слоев была выбрана ReLU. Выходные векторы $\hat{s}'$ и $\hat{r}$ являются проекцией последнего скрытого слоя без функции активации. Кроме того, на выходном слое присутствует еще один нейрон с сигмоидной функцией активации. Он нужен при работе с эпизодичными задачами, чтобы предсказывать терминальность состояния. Он выдает значения в диапазоне от 0 до 1, которые интерпретируются как вероятность того, что состояние терминальное.

Отметим, что переходов в терминальное состояние намного меньше, чем переходов в нетерминальное. Для решения этой проблемы был применен oversampling. Пакеты для обучения модели составлялись из двух равных частей – выборки переходов в нетерминальные состояния и такой же по размеру выборки переходов в терминальные. В результате каждый пакет будет сбалансирован по классу терминальности.

ЭКСПЕРИМЕНТЫ

Методика проведения. В качестве прикладной задачи для проведения сравнительных экспериментов была выбрана известная задача «CartPole-v0», которая заключается в обучении агента «выживанию» в одноименной среде [3]. Каждый эпизод длится не более 200 шагов, за каждый шаг награда +1. Задача считается «решенной», когда получено среднее вознаграждение не меньше 195,0 за 50 эпизодов подряд.

Программная реализация проведена с использованием языка программирования Python и фреймворков PyTorch и OpenAI Gym.

Моделирование среды. Модель на выходе имеет три компоненты: вектор состояния, награда и терминальность. Для первых двух компонент задана функция потерь в виде среднеквадратичной ошибки, для последней, в силу бинарности этой компоненты, – в виде кросс-энтропии.

Ниже на рисунке представлены графики изменения соответствующих функций потерь в течение обучения модели за более чем 800 шагов. На всех графиках наблюдается уменьшение значения функции потерь с выходом на плато, что говорит о корректности процесса обучения.

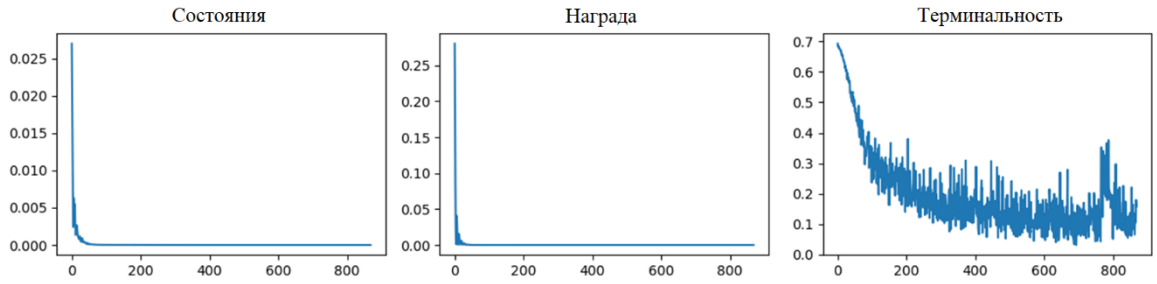

Рис. Динамика функции потерь при обучении модели среды

Обучение агента. Агент сначала обучался алгоритмом A2C, не используя модель, т.е. только на «реальных» эпизодах. Во втором эксперименте обучение проводилось предложенным способом по следующей схеме. Первые k эпизодов обучалась только стратегия агента (с целью накопления пакета размера k для обучения модели), далее обучались и стратегия, и модель, причем через каждые d эпизодов происходило планирование стратегии на «воображаемых» эпизодах (в модели среды), длившееся n эпизодов.

Приведем в таблице ниже сравнение результатов обучения методом Actor-Critic и разработанным подходом, основанным на модели среды, при разных наборах параметров k, d, n.

Таблица

Сравнение результатов обучения двумя методами

Метод обучения		Число эпизодов обучения	
		«реальных»	«воображаемых»
Метод A2C		261	–
Наш метод	k=32, d=20, n=50	110	200
	k=64, d=20, n=50	131	150
	k=32, d=40, n=100	122	200
	k=64, d=40, n=100	136	300

Исходя из полученных результатов экспериментов, можно сделать вывод, что предложенный способ требует для обучения почти в два раза меньше данных, чем метод Actor-Critic без модели среды.

ВЫВОДЫ

Таким образом, был предложен оригинальный подход к обучению с подкреплением. Его особенностью является нейросетевая модель среды, которая обучается и затем используется для планирования стратегии поведения агента. При этом процесс обучения модели происходит параллельно с обучением стратегии. С помощью экспериментов было проде-

монстрировано, что предложенный подход к обучению имеет почти двухкратное преимущество по объему обучающих данных по сравнению с известным методом Actor-Critic.

Библиографические ссылки

1. *Mnih V.* Playing Atari with Deep Reinforcement Learning / V. Mnih et al. [Электронный ресурс] // ArXiv preprint. – 2013. – Режим доступа: https://arxiv.org/pdf/1312.5602. – Дата доступа: 08.11.2019.
2. *Mnih, V.* Asynchronous Methods for Deep Reinforcement Learning / V. Mnih et al. [Электронный ресурс] // ArXiv preprint. – 2016. – Режим доступа: https://arxiv.org/pdf/1602.01783v2. – Дата доступа: 16.11.2019.
3. OpenAI Gym. Среда «CartPole-v0» [Электронный ресурс]. Режим доступа: https://gym.openai.com/envs/CartPole-v0. – Дата доступа: 08.11.2019.