УДК 004.85 URL: https://ptsj.bmstu.ru/catalog/icec/inf_tech/980.html

БАЙЕСОВСКИЕ НЕЙРОННЫЕ СЕТИ В МУЛЬТИАГЕНТНЫХ СРЕДАХ

М.С. Подмарёв maxmaster2033@gmail.com

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Рассмотрены особенности поведения байесовских нейронных сетей в мультиагентных средах. Представлены обучение с подкреплением, мультиагентное обучение с подкреплением и поведение байесовского Actor-Critic в мультиагентных средах. Рассмотрено обучение с помощью байесовских нейронных сетей. Показаны архитектура и иерархия байесовской нейронной сети. Описаны компоненты агента, его взаимодействие со средой тестирования. Для проведения тестирования выбрана среда Mujoco. Выделены особенности и аспекты среды. Проведены тесты для проверки поведения агентов под управлением байесовской нейронной сети с использованием алгоритма Actor-Critic. Продемонстрированы результаты тестирования, по итогам которых определены успешность работы алгоритма, достижение успехов в обучении и взаимодействия агентов со средой. Проделанная работа является доказательством того, что байесовские нейронные сети и байесовский алгоритм Actor-Critic способны обучаться и достигать поставленных результатов мультиагентных средах.

Ключевые слова: мультиагентная среда, машинное обучение, мультиагентное обучение с подкреплением, нейронные сети, байесовские нейронные сети, байесовский алгоритм Actor-Critic

Введение. Машинное обучение — одна из самых актуальных тем в наше время. Оно применяется для упрощения и автоматизации многих аспектов жизни человека.

Существует три типа обучения: обучение с учителем, обучение без учителя, обучение с подкреплением [1].

При *обучении с учителем* нейросеть получает специальный набор данных, в котором заранее отмечено, что эти данные означают.

Суть *обучения без учителя* заключается в том, что нейросеть получает на входе неразмеченные данные и старается сама найти в них общие признаки и связи.

Обучение с подкреплением — способ машинного обучения, в ходе которого испытуемая система (агент) обучается, взаимодействуя с некоторой средой и получая за это награду [2].

Мультиагентное обучение с подкреплением. Мультиагентное обучение с подкреплением (англ. *Multi-agent Reinforcement Learning*, MARL) — совокупность алгоритмов и моделей машинного обучения с подкреплением, которые дают возможность обучать мультиагентные системы [3]. Особенность MARL состоит в том, что все агенты всегда подвергают среду своим действиям. Агенты $i = 1, 2, \ldots, N$ обуча-

 ISSN 2541-8009

ются в процессе взаимодействия. Они получают от среды положительные или отрицательны вознаграждения $R_t^1, R_t^2, \ldots, R_t^N$ и новые состояния $S_{t+1}^1, S_{t+1}^2, \ldots, S_{t+1}^N$, которые зависят от действий $a_t^1, a_t^2, \ldots, a_t^N$ агентов (рис. 1).

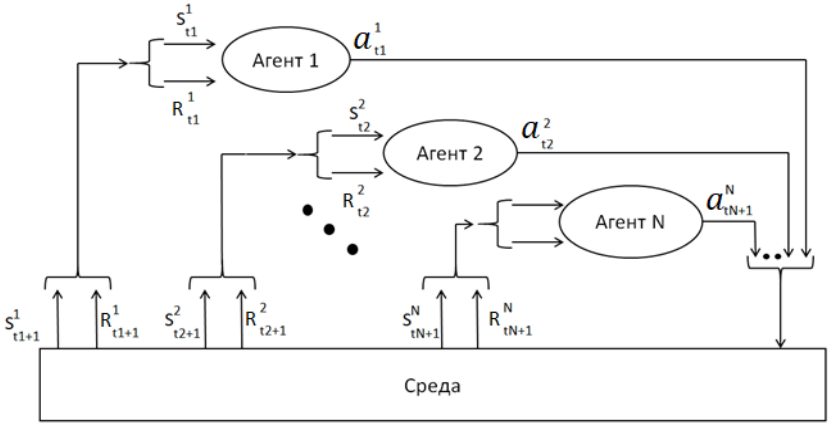


Рис. 1. Модель обучения с подкреплением

В обучении с подкреплением все взаимодействия происходят между двумя основными составляющими — агентом и средой. Агент применяет действия, а среда реагирует на действие агента и возвращает ему измененное состояние и вознаграждение.

Агент — это кто-либо, вступающий во взаимодействие со средой, выполняя определенные действия в ней и получая за них вознаграждения, в зависимости от наблюдений среды.

Среда — это все, что находится вне агента. Взаимодействия со средой ограничено действиями, наблюдениями и вознаграждениями.

Действия — это то, что агент может делать в среде. Действия должны удовлетворять возможностям среды.

Вознаграждения — скалярное значение, получаемое от среды. Смысл вознаграждения состоит в том, чтобы научить агента какой-либо стратегии.

Наблюдения — изменения в среде, происходящие после выполнения агентом действий.

Actor-Critic. Алгоритм Actor-Critic — это тип алгоритма обучения с подкреплением, который сочетает в себе аспекты как методов, основанных на политике (Actor), так и методов, основанных на ценностях (Critic) [4]. Этот гибридный подход разработан для устранения ограничений каждого метода при индивидуальном использовании. В рамках взаимодействия «актер — критик» агент («актер») изучает политику для принятия решений, а функция оценки («критик») оценивает действия, предпринятые Actor.

В этой гибридной конфигурации субъект берет на себя роль лица, принимающего решения, выбирая действия на основе текущей политики. Одновременно критик оценивает ценность или качество этих действий. Эта двойная роль позволяет алгоритму находить баланс между исследованием и эксплуатацией, используя сильные стороны, как политических, так и ценностных функций.

Байесовский алгоритм Actor-Critic. Теорема (формула) Байеса позволяет выяснить вероятность события A при условии, что произошло связанное с ним другое событие B [5]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

Здесь $P(A)$ — априорная вероятность (начальное представление о вероятности события A; $P(B|A)$ — вероятность наступления события B при условии, что событие A произошло; $P(B)$ — маргинальная вероятность (вероятность наступления события B); $P(A|B)$ — апостериорная вероятность (обновленное представление о вероятности события A после учета новых данных B).

Формула Байеса позволяет обновлять представление о мире, когда появляется новая информация.

Структура байесовской сети G представляет собой ориентированный ациклический граф, узлами которого служат случайные величины $X_1, \ldots, X_n$. Пусть Pa_{Xi}^G обозначает «родителей» X_i в G, а NonDescendants X_i обозначает переменные в графе, которые не являются «потомками» X_i. Тогда G кодирует следующий набор предположений условной независимости, называемых локальной независимостью и обозначаемых $I_l(G)$: для каждой переменной $X_i : \left(X_i \perp \text{NonDescendants}\, X_i | \; Pa_{X_i}^G\right)$. Другими словами, локальная независимость утверждает, что каждый узел X_i условно независим от своих «не-потомков», включая его «родителей». Более того, байесовскую сеть можно представить как цепное правило условных вероятностей:

$$P(x_1 \cap \ldots \cap x_n) = P(x_1)P(x_2|x_1)\ldots P(x_n|x_1 \cap \ldots \cap x_{n-1}).$$

Здесь x_i — значение переменной X_i, $P(x_i) = P(X_i = x_i)$.

Предполагаемая архитектура алгоритма Actor-Critic на основе байесовских сетей представлена на рис. 2.

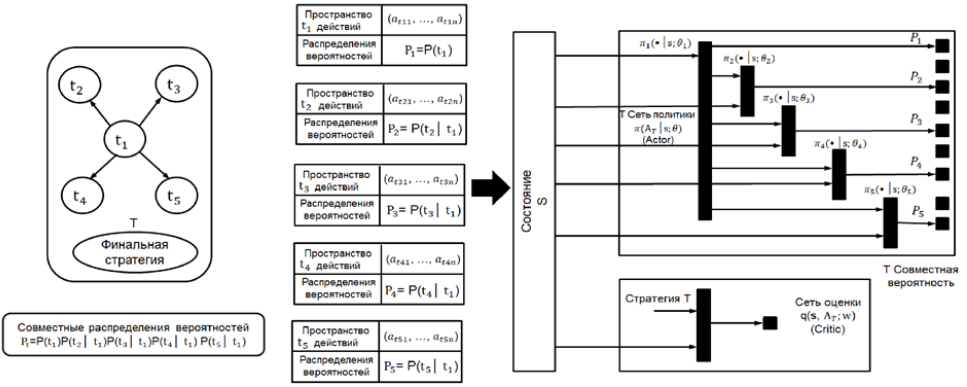
Данный метод включает концепцию максимальной энтропии в алгоритм глубокого обучения с подкреплением Actor-Critic — Soft Actor-Critic (SAC) [6]. Согласно аддитивности энтропии, энтропия системы может представлять собой сумму энтропии нескольких независимых подсистем. Для каждого шага итерации мягкой политики совместная политика π будет рассчитывать значение, позволяющее максимизировать сумму энтропии π_i подсистем

 ISSN 2541-8009

в сетях байесовской стратегии (Bayesian Strategy Network — BSN), используя уравнение целевой функции

$$J_V(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t, \mathcal{A}_t) \sim \rho_{\pi_i}} \left[r(s_t, \mathcal{A}_t) + \frac{a}{m} \sum_{i=1}^{m} \mathcal{H}\left(\pi_i(\bullet \mid s_t)\right) \right].$$

Здесь s_t — состояние; $\mathcal{A}_t$ — совместное действие; m — количество тактик; $\bullet$ — обобщенное обозначение действия; $\mathbb{E}_{(s_t, \mathcal{A}_t) \sim \rho_{\pi_i}}$ — математическое ожидание по распределению ρ_{π_i} состояний s_t и действий $\mathcal{A}_t$ согласно стратегии π_i; $r(s_t, \mathcal{A}_t)$ — функция вознаграждения, которая определяет вознаграждение за нахождение в состоянии s_t и выполнение действия $\mathcal{A}_t$; $\mathcal{H}\left(\pi_i(\bullet \mid s_t)\right)$ — энтропия стратегии π_i в состоянии s_t. Весовые и соответствующие температурные параметры *a* для каждого воздействия одинаковы в каждой подсистеме.



а *б*

Рис. 2. Реализация модели архитектуры Actor-Critic на основе байесовских сетей:
а — байесовские стратегические сети; *б* — модель Actor-Critic

Мягкое значение *Q* (*Q*-функцию) можно вычислить итеративно, начиная с любой функции *Q*: $S \times A \to R$ и неоднократно применяя модифицированный резервный оператор Беллмана T^{π}. В мягком байесовском алгоритме «актер — критик» (Bayesian Soft Actor-Critic — BSAC) T^{π} определяется уравнением

$$T^{\pi} Q(s_t, \mathcal{A}_t) \triangleq r(s_t, \mathcal{A}_t) + \mathbb{E}_{s_{t+1} \sim \rho}\left[V(s_{t+1})\right].$$

Учитывая, что при оценке каждой субполитики применяются одно и то же значение *Q* и вес, функцию значения мягкого состояния можно представить в виде уравнения

$$V(s_t) = \mathbb{E}_{\mathcal{A}_t \sim \pi}\left[Q(s_t, \mathcal{A}_t) - \frac{1}{m}\sum_{i=1}^{m} \log \pi_i(a_{i_t} | \ s_t) \right].$$

Здесь a_{i_t} — действие, которое выбирается конкретной стратегией π_i в состоянии s_t.

В частности, на каждом этапе улучшения субполитики π_i для каждого состояния мы обновляем соответствующую политику в соответствии с уравнением

$$\pi_{new} = \arg\min D_{KL}\left(\frac{1}{m}\prod_{i=1}^{m} \pi_i'(\bullet | s_t) \ \middle| \ \frac{\exp\left(Q^{\pi_{old}}(s_t, \bullet)\right)}{Z^{\pi_{old}}(s_t)} \right).$$

Здесь π_{new} — новая стратегия; D_{KL} — дивергенция Кульбака — Лейблера, которая измеряет «расстояние» между двумя распределениями; π_{old} — старая стратегия; $Z^{\pi_{old}}(s_t)$ — нормировочная константа, необходимая для того, чтобы результирующее распределение было корректным вероятностным распределением.

Более того, оценка мягкой политики и поочередное выполнение улучшения мягкой политики на каждой итерации мягкой субполитики гарантируют сходимость оптимальной максимальной энтропии среди комбинации субполитик. Используются аппроксиматоры функций как для функции *Q*, так и для каждой субполитики, оптимизируя сети с помощью стохастического градиентного спуска. Вместо использования одной сети политик алгоритм BSAC делит основную политику и генерирует из нее несколько простых подполитик на основе BSN для адаптации к определенной модели.

Тестирование байесовского алгоритма Actor-Critic в среде. Здесь оценивается эффективность предлагаемого агента BSAC в нескольких сложных средах непрерывного контроля с комбинациями действий. Был выбран физический движок Mujoco для моделирования экспериментов в среде OpenAI Gym. В данном эксперименте используется стандартный тестовый домен непрерывного управления — Hopper-v2.

В соответствующих экспериментах анализируется поведение hopper с использованием BSN. Например, три политики BSN (действие бедра t_1, действие колена t_2 и действие лодыжки t_3) в Hooper-v2 (рис. 3).

Кроме того, для соответствующих моделей BSAC можно формализовать их совместную политику действий:

$$P(t_1, t_2, t_3) = P(t_1)P(t_2 | \ t_1)P(t_3 | \ t_2);$$

 ISSN 2541-8009

$$P(t_1,t_2,t_3,t_4,t_5) = P(t_1)P(t_2|\ t_1)P(t_3|\ t_1)P(t_4|\ t_2)P(t_5|\ t_3);$$

$$P(t_1,t_2,t_3,t_4,t_5) = P(t_1)P(t_2|\ t_1)P(t_3|\ t_1)P(t_4|\ t_1)P(t_5|\ t_1).$$

С увеличением сложности поведения и стратегии агента разложение сложного поведения на простые действия или тактики и организация их в виде подходящего BSN, построение соответствующей модели совместной политики в BSAC может существенно повысить эффективность обучения.

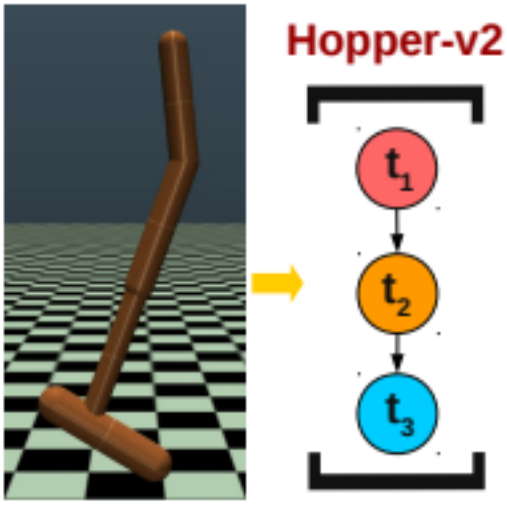


Рис. 3. BSN модель в Hopper-v2

BSAC обеспечивает подход к созданию более подходящей совместной политики, соответствующей распределению стоимости, повышая эффективность конвергенции и производительность модели. В частности, традиционный подход обучения с подкреплением заключается в том, чтобы указать унимодальное распределение политики, сосредоточенное на максимальном значении Q, и распространить его на соседние действия, чтобы обеспечить продвижение для исследования. Особенно исследование смещено в сторону локального прохода, там агент уточняет свою политику и полностью игнорирует другие. Если можно разработать подходящее совместное распределение политик, состоящее из нескольких простых распределений политик, соответствующее подходящему распределению значений Q, это существенно повысит эффективность выборки при обучении агента.

Заключение. В данной научно-исследовательской работе был изучен новый метод обучения с подкреплением — байесовский Actor-Critic (BSAC) и рассмотрены его основные особенности.

BSAC предусматривает взаимодействие сетей байесовских стратегий и мягкого алгоритма Actor-Critic и позволяет разложить сложную стратегию или совместную

политику на несколько простых подполитик. Здесь демонстрируется работа на стандартных тестах непрерывного управления, таких как Hopper в Mujoco со средой OpenAI Gym. Результаты демонстрируют потенциал и значимость предложенной архитектуры BSAC благодаря достижению более эффективного выборочного обучения и более высокой производительности по сравнению с современными методами глубокого обучения с подкреплением. Кроме того, реализация BSAC на реальных роботах также представляет собой интересную задачу. Это поможет нам разработать надежные вычислительные модели для роботизированных систем, таких как управление передвижением роботов, планирование и навигация с участием нескольких роботов, а также роботизированные поисково-спасательные миссии.

Литература

[1] Лапань М. *Глубокое обучение с подкреплением. AlphaGo и другие технологии.* Санкт-Петербург, Питер, 2020, 496 с.

[2] Саттон Р.С., Барто Э.Дж. *Обучение с подкреплением.* Москва, ДМК Пресс, 2020, 552 с.

[3] Алфимцев А.Н. *Мультиагентное обучение с подкреплением.* Москва, МГТУ им. Н.Э. Баумана, 2021, 222 с.

[4] Konda V. *Actor-Critic Algorithms.* Available at: https://dspace.mit.edu/bitstream/handle/1721.1/8120/51552606-MIT.pdf?sequence=2 (accessed April 15, 2024).

[5] Дауни А.Б. *Байесовские модели.* Москва, ДМК Пресс, 2019, 184 с.

[6] Qin Yang, Ramviyas Parasuramana. A Strategy-Oriented Bayesian Soft Actor-Critic Model. *The 14th International Conference on Ambient Systems, Networks and Technologies (ANT)*, 2023, Leuven, Belgium. Available at: https://arxiv.org/pdf/2303.04193.pdf (accessed March 7, 2023).

Поступила в редакцию 25.02.2024

Подмарёв Максим Станиславович — студент кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Алфимцев Александр Николаевич, доктор технических наук, профессор кафедры «Информационные системы и телекоммуникации», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:
Подмарёв М.С. Байесовские нейронные сети в мультиагентных средах. *Политехнический молодежный журнал*, 2024, № 03 (92).
URL: https://ptsj.bmstu.ru/catalog/icec/inf_tech/980.html

 ISSN 2541-8009

BAYESIAN NEURAL NETWORKS IN THE MULTI-AGENT ENVIRONMENT

M.S. Podmaryov maxmaster2033@gmail.com

Bauman Moscow State Technical University, Moscow, Russian Federation

The paper considers behavioral features of the Bayesian neural networks in the multi-agent environment. It presents reward learning, multi-agent reward learning, and the Bayesian Actor-Critic behavior in the multi-agent environment. Learning using the Bayesian neural networks is analyzed. The Bayesian neural network architecture and hierarchy are demonstrated. The paper describes agent components and their interaction with the testing environment. The Mujoco environment is selected in testing. The environment features and aspects are highlighted. Tests were conducted to check behavior of the agents controlled by the Bayesian neural network using the Actor-Critic algorithm. Test results are presented, they prove the algorithm efficiency, achieving successful learning, and interaction between the agents and the environment. The work performed is acknowledging that the Bayesian neural networks and the Bayesian Actor-Critic algorithm are capable of learning and achieving set results in the multi-agent environment.

Keywords: multi-agent environment, machine learning, multi-agent reward learning, neural networks, Bayesian neural networks, Bayesian Actor-Critic algorithm

Received 25.04.2024

Podmaryov M.S. — Student, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Alfimtsev A.N., Dr. Sci. (Eng.), Professor, Department of Information Systems and Telecommunications, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:
Podmaryov M.S. Bayesian neural networks in the multi-agent environment. *Politekhnicheskiy molodezhnyy zhurnal*, 2024, no. 03 (92). (In Russ.). URL: https://ptsj.bmstu.ru/catalog/icec/inf_tech/980.html

