УДК 004.85

ИСПОЛЬЗОВАНИЕ ВНУТРЕННЕЙ МОТИВАЦИИ ПРИ ОБУЧЕНИИ АГЕНТОВ ДЛЯ ИГР НА ATARI 2600

С.П. Зязюлькин
Магистрант кафедры информатики

С.Н. Нестеренков
Доцент кафедры программного обеспечения информационных технологий, кандидат технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь.
E-mail: nsn@bsuir.by.

С. П. Зязюлькин
Окончил Белорусский государственный университет в 2017 году по специальности «Прикладная информатика», магистрант второго года обучения по специальности «Информатика и технологии программирования» БГУИР.

С. Н. Нестеренков
Окончил БГУИР в 2007 году по специальности «Программное обеспечение информационных технологий», окончил магистратуру БГУИР в 2008 по специальности «Системный анализ, управление и обработка информации», окончил аспирантуру БГУИР в 2013 по специальности «Системный анализ, управление и обработка информации», окончил магистратуру БГУИР в 2013 по специальности «Экономика и управление народным хозяйством», в 2017 защитил диссертацию на соискание ученой степени кандидата технических наук по специальности «Системный анализ, управление и обработка информации».

Аннотация. Одной из основных проблем машинного обучения с подкреплением является обучение агентов в условиях отсутствия или сильной разреженности обратной связи (вознаграждений) в ответ на предпринимаемые агентом действия. В данной статье рассматриваются способы добавления агенту внутренней мотивации – дополнительного механизма вознаграждения за любопытство – с целью повышения эффективности исследования среды на примере игр для Atari 2600.

Ключевые слова: machine learning, reinforcement learning, curiosity-driven learning, Atari 2600, count-based exploration, intrinsic curiosity module, random network distillation, episodic curiosity, fast and slow exploration.

Введение. Машинное обучение с подкреплением предполагает наличие агента с некоторой политикой поведения π, взаимодействующего со средой. В каждый момент времени агент получает на вход текущее состояние среды s_t и предпринимает некоторое действие a_t. Агент получает обратную связь в виде вознаграждения r_t. Агент обучается максимизировать суммарное вознаграждение R_t, определяемое формулой.

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \quad (1)$$

где T – момент времени, в который заканчивается взаимодействие агента со средой

(например, это может быть гибель персонажа в компьютерной игре), γ – дисконтирующий коэффициент.

Ценность действия a в случае состояния среды s_t и политики π определяется как.

$$Q^{\pi}(s,a)=E[R_t \mid s_t=s,a]. \quad (2)$$

Оптимальная политика π^* предполагает выбор на каждом шаге действия с максимальной ценностью, что соответствует уравнению оптимальности Беллмана:

$$Q^*(s,a)=E\left[r+\gamma \max_{a'} Q^*(s',a') \mid s,a\right]. \quad (3)$$

Одними из популярных задач для тестирования алгоритмов машинного обучения с подкреплением являются игры для игровой приставки Atari 2600, вышедшей более 30 лет назад. Игры разнятся по сложности и частоте вознаграждений. Особенно сложными для алгоритмов машинного обучения с подкреплением являются игры с сильно разреженными вознаграждениями. Примером такой игры является Montezuma's Revenge, в которой требуется выполнять длинные и сложные последовательности действий для преодоления смертоносных препятствий и получения вознаграждения (рис. 1.).

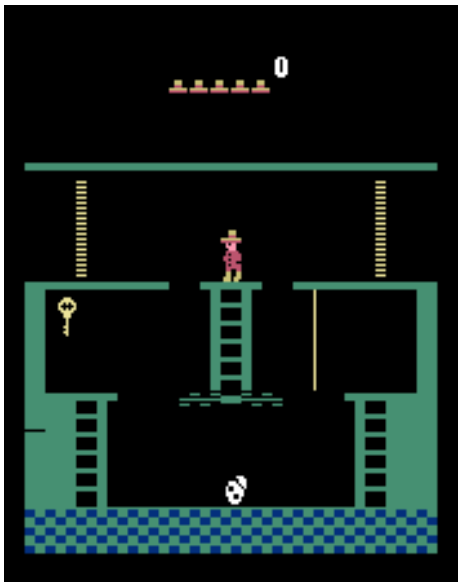

Рисунок 1. Montezuma's Revenge – игра на Atari 2600

Алгоритм машинного обучения с подкреплением, позволяющий обучить агента для простых игр на Atari 2600, был предложен ещё в 2013 году [1]. Однако игры, требующие эффективного исследования игрового пространства, представляют сложность и для современных алгоритмов машинного обучения с подкреплением [2, 3].

В классических алгоритмах машинного обучения с подкреплением исследование среды основано на стохастических политиках: модель предсказывает распределение вероятностей предпринимаемых действий или предполагает предпринятие случайного действия с некоторой вероятностью. Такой подход работает для задач с частыми вознаграждениями, однако оказывается неэффективным для задач с сильно разреженными вознаграждениями. Если случайные действия не позволяют агенту добраться до вознаграждений, агент не получает никакой обратной связи, что сводит процесс обучения на нет. Например, в игре Montezuma's Revenge для получения первого вознаграждения необходимо спуститься по лестнице, прыгнуть на верёвку, спрыгнуть с неё к следующей

лестнице, снова спуститься по лестнице, перепрыгнуть движущийся череп, подняться по лестнице и подпрыгнуть, чтобы собрать ключ. Добраться до ключа, выполняя случайные действия, крайне сложно.

Одним из способов решения проблемы исследования среды является обучение с использованием демонстраций [4]. Однако этот способ требует генерации этих самых демонстраций, а также ограничен сценариями, в которых демонстрации могут быть легко сгенерированы людьми.

Использование внутренней мотивации для эффективного исследования среды. Для эффективного исследования среды предлагается использовать внутреннюю мотивацию агента для исследования – механизм вознаграждения агента за его любопытство. Одним из первых таких механизмов является вознаграждение, основанное на счётчиках посещённых состояний – чем реже агент посещал данное состояние, тем больше вознаграждение за его посещение. Эффективность такого механизма падает с ростом числа возможных состояний среды. Например, данный подход в чистом виде является неэффективным для игр на Atari 2600, где состояние среды представлено изображением 210x160x3. Даже в случае сильного даунсэмплинга изображения число возможных вариантов изображения (состояния) остаётся очень большим. Поэтому для применения вознаграждения за любопытство, основанного на счётчиках, необходимо отображать состояние среды в некоторое внутреннее представление. Например, для такого преобразования может быть использована хеш-функция [5].

Другой вариант реализации механизма вознаграждения за любопытство базируется на предсказании следующего состояния среды на основе текущего состояния и предпринимаемого действия. Величина вознаграждения определяется ошибкой такого предсказания. Чем больше раз агент выполняет некоторое действия в определённом состоянии, тем лучше он предсказывает следующее состояние, тем меньше вознаграждение за любопытство.

Состоянием среды в случае компьютерных игр обычно является изображение игрового пространства. Предсказывание пикселей следующего кадра, коих может быть десятки тысяч, является слишком сложной проблемой. Например, рассмотрим случай, когда агенту подаются на вход изображения, на которых листва колышется под действием ветра. Моделирование движения листьев под действием ветра является сложной проблемой, не говоря уже о предсказании изображений колышущейся листвы. Более того, не всегда следующее состояние среды может быть предсказано на основе текущего состояния и предпринимаемого действия. Поэтому, как и в случае вознаграждения за любопытство, основанного на счётчиках, необходимо отображать входное изображение во внутреннее пространство признаков, менее подверженное этой проблеме.

В работе [6] был предложен механизм вознаграждения за любопытство на основе предсказания следующего состояния, именуемый ICM. Он предполагает использование двух моделей. Одна модель (Inverse Model) отображает состояние среды во внутреннее пространство признаков, а также предсказывает по внутренним признакам текущего и следующего состояния предпринятое действие. Минимизация ошибки предсказания предпринятого действия по состояниям позволяет получить внутреннее пространство признаков, учитывающее только элементы, на которые может влиять агент или которые влияют на агента. Вторая модель (Forward Model) учится предсказывать признаки следующего состояния по признакам текущего состояния и предпринятому действию. Величина вознаграждения за любопытство определяется формулой.

$$r_t^i = \frac{\eta}{2} \|\hat{\varphi}(s_{t+1}) - \varphi(s_{t+1})\|_2^2 \quad (4)$$

где η – масштабирующий коэффициент, $\hat{\varphi}(s_{t+1})$ и $\varphi(s_{t+1})$– предсказанные и истинные

признаки следующего состояния соответственно (рис. 2.).

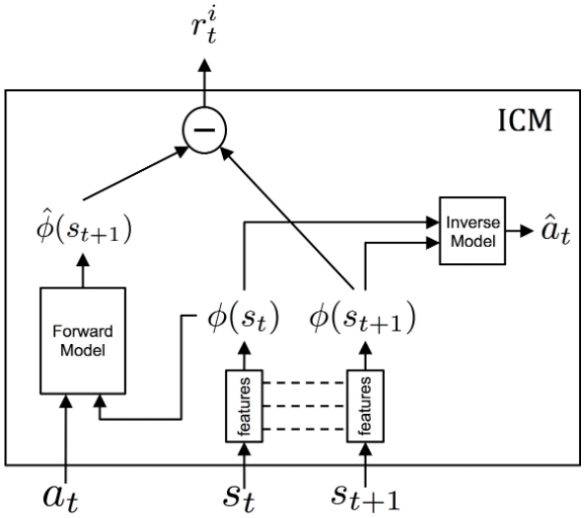

Рисунок 2. Intrinsic Curiosity Module (ICM)

Вознаграждение за любопытство, основанное на предсказании следующего состояния, сталкивается с рядом проблем. Первое, агенты застревают в процессе исследования среды в состояниях со стохастическим шумом. Наличие шума в следующих состояниях сильно затрудняет или вовсе делает невозможным предсказание, т. е. значительно уменьшает зависимость следующего состояния от текущего состояния и действий агента. Эта проблема также известна как white noise или Noisy-TV problem [7]. Второе, ограничения архитектуры модели могут не позволять точно предсказывать следующее состояние среды. Например, число слоёв нейронной сети или нейронов в них слишком мало для решения данной задачи.

Решение Noisy-TV problem было предложено в статье [8]. В ней описан механизм вознаграждения за любопытство, именуемый RND, который также базируется на предсказании следующего состояния, однако исключает зависимость предсказания от текущего состояния среды и действий агента. Идея заключается в использовании двух нейронных сетей. Первая (целевая) инициализируется случайными весами. Вторая нейронная сеть (предсказывающая) учится предсказывать не следующее состояние, а выходы целевой сети. Для редко посещаемых состояний выходы предсказывающей и целевой сетей будут сильно различаться. Чем больше ошибка предсказания, тем выше вознаграждение за любопытство. Величина награды за любопытство вычисляется по формуле.

$$r_t^i = \left\| f'(s_{t+1}) - f(s_{t+1}) \right\|_2^2 \quad (5)$$

где f' и f – предсказывающая и целевая сети соответственно (рис. 3.).

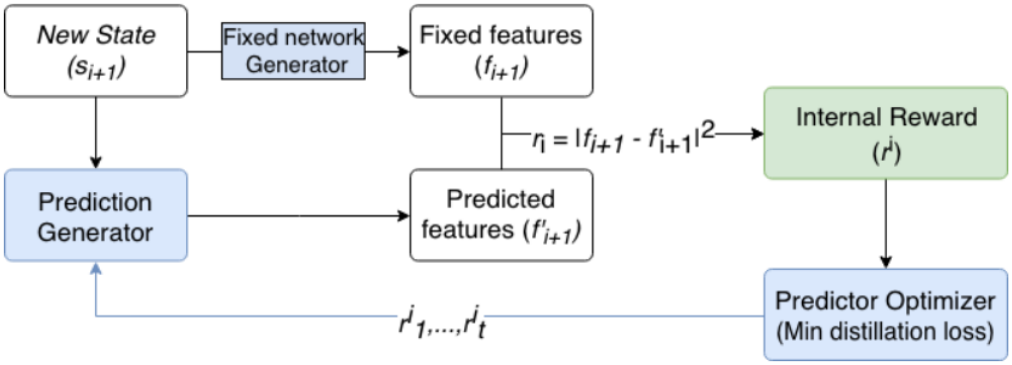

Рисунок 3. Random Network Distillation (RND)

Другой способ решения noisy-TV problem, именуемый EC, был предложен в статье [9]. Он базируется на следующей идее. Вознаграждение агента за любопытство определяется сложностью достижения состояния из уже посещённых агентом в текущем эпизоде взаимодействия со средой. Сложность достижения определяется числом действий, которое необходимо предпринять для перехода в данное состояние.

Для хранения посещённых состояний используется эпизодическая память. В начале каждого эпизода взаимодействия со средой эпизодическая память пуста. На каждом шаге при помощи нейронной сети (Embedding network) текущее состояние среды отображается во внутреннее пространство признаков, затем выполняется вычисление схожести текущего состояния с состояниями из эпизодической памяти при помощи ещё одной нейронной сети (Comparator network). Результаты сохраняются в буфер достижимости, на основе данных которого вычисляется схожесть текущего состояния с ранее посещёнными. Один из простых вариантов вычисления схожести текущего состояния – максимум из величин из буфера достижимости, однако такой вариант чувствителен к выбросам, поэтому вместо него рекомендуется использовать 90-й перцентиль. Величина вознаграждения за любопытства вычисляется по формуле.

$$r_t^i = \alpha(\beta - F(c_1, c_2, ..., c_n)) \quad (6)$$

где c_i – схожесть текущего состояния с i-м состоянием из эпизодической памяти, F – агрегирующая функция схожести, α – масштабирующий коэффициент, β – коэффициент, определяющий знак вознаграждения за любопытство.

После вычисления величины вознаграждения внутренние признаки текущего состояния сохраняются в эпизодическую память, если величина вознаграждения за любопытство превышает некоторый порог новизны. Добавление всех посещённых агентом состояний в эпизодическую память будет приводить к ситуации, когда текущее состояние всегда схоже с последними посещёнными, т. е. величина вознаграждения за любопытство будет близка к нулю. Также использование порога позволяет снизить избыточность хранимой в эпизодической памяти информации, т. к. в ней не будут содержаться схожие состояния (рис. 4.).

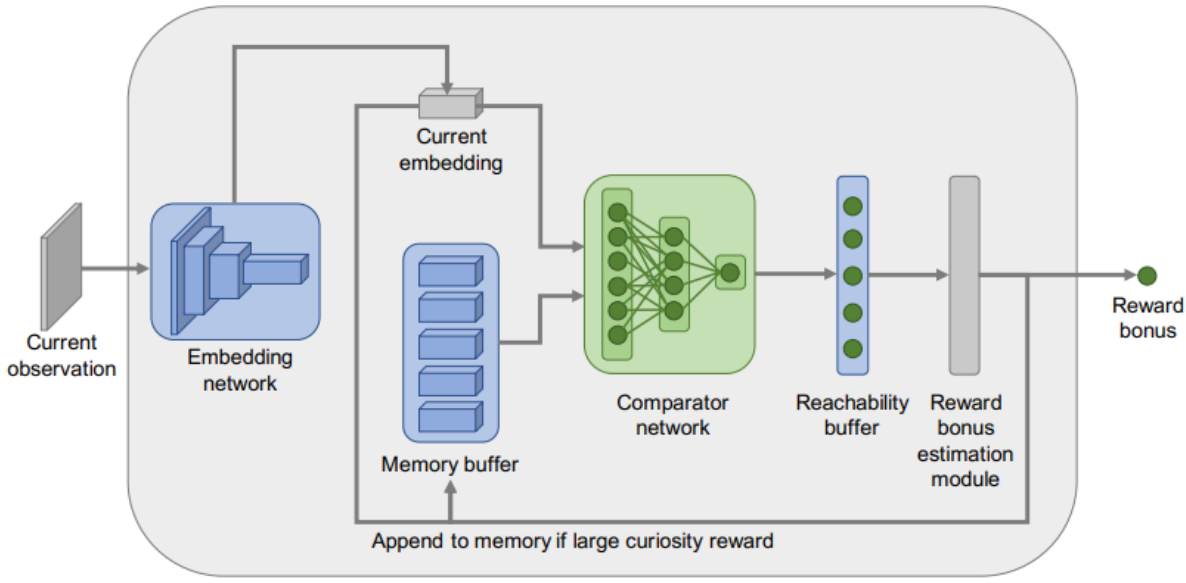

Рисунок 4. Episodic Curiosity (EC)

Авторы работы [10] предлагают базировать механизм вычисления награды за любопытство на ошибке реконструкции текущего состояния по его контексту. Под контекстом состояния понимается его некоторый преобразованный вид. В работе предлагается два способа преобразования: даунсэмплинг входного изображения и замена нескольких регионов исходного изображения стохастическим шумом.

Для вычисления ошибки реконструкции состояния используется индекс структурного сходства (SSIM) [11]. Величина вознаграждения за любопытство имеет вид.

$$r_t^i = 1 - \left[\frac{1}{P} \sum_{i=1}^{P} L\left(s_t^i, \widehat{s_t^i}\right) \Gamma\left(s_t^i, \widehat{s_t^i}\right) S\left(s_t^i, \widehat{s_t^i}\right) \right] \quad (7)$$

где s_t^i и $\widehat{s_t^i}$ – i-я область исходного состояния и его реконструкции соответственно. Яркость (L), контраст (Γ) и структура (S) вычисляются по формулам.

$$L(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \Gamma(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, S(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (8)$$

где μ_x, μ_y, σ_x, σ_y определяют среднюю интенсивность и среднеквадратическое отклонение интенсивности пикселей, σ_{xy} – коэффициент корреляции между соответствующими пикселями, C_1, C_2, C_3 – небольшие константы для повышения вычислительной устойчивости.

Также в работе [10] предлагается разделить величину вознаграждения за любопытство r_t^i на две компоненты: быструю r_t^{fast} и медленную r_t^{slow}. Быстрая компонента вознаграждения за любопытство отвечает за локальное исследование и быстро уменьшается для новых состояний. В отличие от быстрой компоненты, медленная компонента отвечает за глобальное исследование и остаётся большой продолжительное время, что поощряет исследование труднодоступных состояний среды. Итоговый вид величины вознаграждения за любопытство:

$$r_t^i = \alpha r_t^{fast} + \beta r_t^{slow} \quad (9)$$

где α и β – коэффициенты, позволяющие задать значимость каждой из компонент.

Для быстрой и медленной компонент вознаграждения за любопытство используются разные контексты состояния и разные нейронные сети для их реконструкции. Такая модель получила название FaSo (рис. 5.).

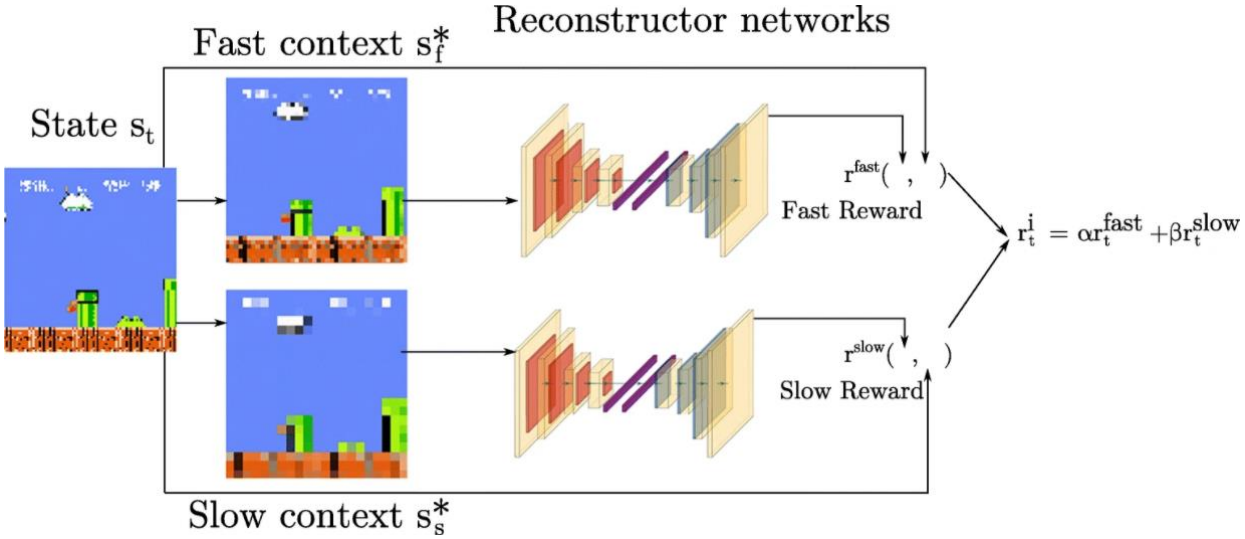

Рисунок 5. Fast and Slow exploration (FaSo)

Величина вознаграждения за любопытство может сильно варьироваться в процессе обучения даже для одного и того же состояния. Для решения этой проблемы необходимо выполнять нормализацию величины вознаграждения за любопытство. Один из способов нормализации – делить на скользящее среднеквадратическое отклонение вознаграждения или суммарного вознаграждения за любопытство:

$$\bar{r}_t^i = \frac{r_t^i}{\sigma(r^i)}, \hat{r}_t^i = \frac{r_t^i}{\sigma(R^i)} \tag{10}$$

В случае FaSo рекомендуется выполнять нормализацию независимо для каждой из компонент.

Добавление вознаграждения за любопытство к основному вознаграждению делает величину вознаграждения более зашумлённой, что усложняет предсказание величины вознаграждения нейронной сетью. Для решения этой проблемы рекомендуется использовать отдельные «головы» нейронной сети с общими базовыми слоями или вовсе отдельные нейронные сети для предсказания основного вознаграждения и вознаграждения за любопытство. Например, в случае алгоритма PPO рекомендуемая архитектура сети имеет следующий вид (рис. 6.).

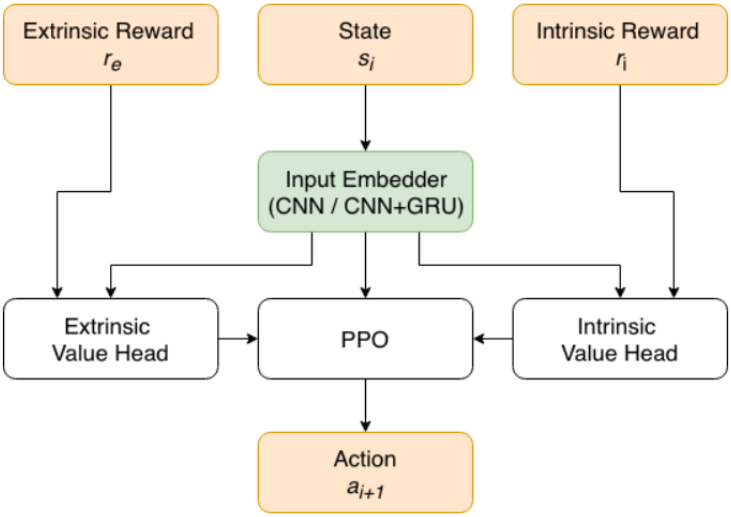

Рисунок 6. PPO с вознаграждением за любопытство

Заключение. Рассмотренные механизмы внутренней мотивации – дополнительного вознаграждения за любопытство – значительно повышают эффективность исследования среды агентами. Они позволяют обучать агентов, превосходящих человека на играх на Atari 2600 с сильно разреженными вознаграждениями, например Montezuma's Revenge. Изучение способов интеграции механизмов внутренней мотивации в различные алгоритмы машинного обучения с подкреплением, а также поиск новых механизмов внутренней мотивации остаются открытыми вопросами для дальнейшего изучения.

Список литературы

[1] Mnih, V. Playing Atari with Deep Reinforcement Learning / V. Mnih, K. Kavukcuoglu, D. Silver, [и др.] // arXiv:1312.5602.

[2] Зязюлькин, С.П. Использование DQN для обучения агентов игр (Atari 2600) / С.П. Зязюлькин, С.Н. Нестеренков // BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: сб. материалов VI Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 20-21 мая 2020 года) : в 3 ч. Ч. 2 / редкол.: В. А. Богуш [и др.]. – Минск: Бестпринт, 2020. – С. 274-280.

[3] Зязюлькин, С.П. Использование actor-critic алгоритмов при обучении агентов для игр на ATARI 2600 / С.П. Зязюлькин, С.Н. Нестеренков // Информационные технологии и системы 2020 (ИТС 2020) = Information Tehnologies and Systems 2020 (ITS 2020): материалы междунар. науч. конф., Минск, 18 ноября 2020 г. / Белорус. гос. ун-т информатики и радиоэлектроники; редкол.: Л. Ю. Шилин [и др.]. – Минск: БГУИР, 2020. – С. 74-75.

[4] ,Hester, T. Deep Q-learning from demonstrations / T. Hester, M. Vecerik, O. Pietquin [и др.] // In Proc. of AAAI. – 2018.

[5] ,Tang, H. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning / H. Tang, R. Houthooft, D. Foote [и др.] // arXiv:1611.04717.

[6] Pathak, D. Curiosity-driven Exploration by Self-supervised Prediction / D. Pathak, P. Agrawal, A.A. Efros, T. Darrell // arXiv:1705.05363.

[7] Burda, Y. Large-Scale Study of Curiosity-Driven Learning / Y. Burda, H. Edwards, D. Pathak, [и др.] // arXiv:1808.04355.

[8] Burda, Y. Exploration by Random Network Distillation / Y. Burda, H. Edwards, A. Storkey, O. Klimov // arXiv:1810.12894.

[9] Savinov, N. Episodic Curiosity through Reachability / ,N. Savinov, A. Raichuk, R. Marinier [и др.] // arXiv:1810.02274.

[10] Bougie, N. Fast and slow curiosity for high-level exploration in reinforcement learning / N. Bougie, R. Ichise // Appl Intell 51. – 2021. – P. 1086-1107.

[11] Wang Z. Image quality assessment: from error visibility to structural similarity / Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli // IEEE Transactions on Image Processing 13 (4). – 2004. – P. 600-612.

USING INTRINSIC MOTIVATION TO TRAIN AGENTS FOR ATARI 2600 GAMES

S.P. ZYAZYULKIN
Master student of the department of Informatics

S.N. NESTERENKOV
PhD Associate professor of the department of The software of information technologies

Belarusian State University of Informatics and Radioelectronics,
Republic of Belarus
E-mail: nsn@bsuir.by

Abstract. Training agents, when external feedback (reward) to actions is sparse or nonexistent, is a major challenge for reinforcement learning. This article considers ways of adding intrinsic motivation (an additional mechanism for rewarding curiosity) to improve exploration efficiency on Atari 2600 games.

Keywords: machine learning, reinforcement learning, curiosity-driven learning, Atari 2600, count-based exploration, intrinsic curiosity module, random network distillation, episodic curiosity, fast and slow exploration.