

Самоорганизующиеся нейронные клеточные автоматы для обучения с подкреплением и эволюционного развития

Н. С. Мокрецов✉, Т. М. Татарникова

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

✉ nikitamokrecov6374@gmail.com

Аннотация. Показана актуальность изучения самоорганизующихся систем через моделирование клеточных автоматов, что позволяет решать различные задачи от оптимизации и управления ресурсами до прогнозирования поведения систем. Выполнен обзор клеточных автоматов, включая классические клеточные автоматы, вариационные нейронные клеточные автоматы и растущие сети развития с нейронными клеточными автоматами. Обсуждается значимость применения клеточных автоматов в моделировании сложных процессов, требующих от модели способности восстановления и переобучения при изменении свойств агента или среды. В частности показано, что внедрение самоорганизации в модель агентов обучения с подкреплением наделяет их свойствами адаптивности, что позволяет без дополнительного переобучения модели вносить изменения в исходную задачу и восстанавливать работоспособность системы в случае повреждения ее структуры. Приведены примеры применения клеточных автоматов для моделирования различных процессов и обучения агентов.

Ключевые слова: обучение с подкреплением, клеточные автоматы, самоорганизация, машинное обучение

Для цитирования: Мокрецов Н. С., Татарникова Т. М. Самоорганизующиеся нейронные клеточные автоматы для обучения с подкреплением и эволюционного развития // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 7. С. 68–75. doi: 10.32603/2071-8985-2023-16-7-68-75.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов

Original article

Self-Organizing Neural Cellular Automata for Reinforcement Learning and Evolutionary Development

N. S. Mokretsov✉, T. M. Tatarnikova

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

✉ nikitamokrecov6374@gmail.com

Abstract. The relevance of studying self-organizing systems through the modeling of cellular automata is shown, which allows solving various problems from optimization and resource management to predicting the behavior of systems. A review of cellular automata is provided, including classical cellular automata, variational neural cellular automata, and growing developmental networks with neural cellular automata. The significance of using cellular automata in modeling complex processes that require the ability of the model to recover and retrain when the properties of an agent or environment change is discussed. In particular, it is shown that the introduction of self-organization into the model of reinforcement learning agents endows them with adaptability properties, which makes it possible to make changes to the original problem without additional retraining of the model and restore the system's performance in case of damage to its structure. Examples of the use of cellular automata for modeling various processes and training agents are given.

Keywords: reinforcement learning, cellular automata, self-organization, machine learning

For citation: Mokretsov N. S., Tatarnikova T. M. Self-Organizing Neural Cellular Automata for Reinforcement Learning and Evolutionary Development // LETI Transactions on Electrical Engineering & Computer Science. 2023. Vol. 16, no. 7. P. 68–75. doi: 10.32603/2071-8985-2023-16-7-68-75.

Conflict of interest. The authors declare no conflicts of interest.

Введение. Изучение самоорганизующихся систем в контексте представляет собой современную актуальную задачу.

Самоорганизующимися называют сложные динамические системы, способные при изменении внешних или внутренних условий их функционирования и развития сохранять или совершенствовать свою организацию с учетом прошлого опыта. Они могут принимать сложные формы поведения и структуры без явных инструкций или контроля со стороны внешней среды.

Понимание и моделирование самоорганизующихся систем имеет множество приложений, включая оптимизацию при неопределенных условиях, управление ресурсами, прогнозирование поведения систем и разработку новых технологий – искусственного интеллекта, автономной робототехники и т. п. [1].

Внедрение самоорганизации в модели агентов обучения с подкреплением наделяет их свойствами адаптивности, что позволяет без дополнительного переобучения модели вносить изменения в исходную задачу с возможностью восстановления работоспособности в случае повреждения ее структуры.

К современным инструментам моделирования самоорганизующихся систем относятся вариационные нейронные клеточные автоматы [2]. Рассмотрим реализацию внедрения принципа самоорганизации в модели обучения с подкреплением через клеточные автоматы.

Вариационные нейронные клеточные автоматы. Введем необходимые понятия.

Клеточные автоматы (КА) – это вычислительные системы, вдохновленные процессом клеточного роста и дифференцировки, где «клетки» итеративно обновляют свое состояние на основе состояния соседних клеток и правил КА. Вычислительная система представляется пространством из прилегающих друг к другу клеток – ячеек, образующих решетку. Размерность решетки может быть любой – конечной или бесконечной, причем для первой предусматривается закольцованность при достижении границ пространства. Каждая клетка описывается состоянием и окрестностью. Например, двумерный КА представляет собой

множество конечных автоматов с координатами (i, j) размещения на плоскости

$$\sigma_{ij} \in \Sigma \equiv \{0, 1, \dots, m\},$$

где Σ – множество конечных автоматов на плоскости; m – число конечных автоматов на плоскости.

Окрестность определяется множеством соседей, которое задается рангом – расстоянием от текущей клетки. Например, окрестность фон Неймана описывается как

$$\mathbb{N}_N^1(i, j) = \{(k, l) \mid |i - k| + |j - l| \leq 1\},$$

где k, l – координаты клетки, соседней с текущей в окрестности фон Неймана, а окрестность Мура – как

$$\mathbb{N}_M^1(i, j) = \{(k, l) \mid |i - k| \leq 1, |j - l| \leq 1\}.$$

Состояния автоматов меняются согласно правилам перехода

$$\sigma_{ij}(t+1) = \varphi(\sigma_{k,l}(t) \mid (k, l) \in \mathbb{N}(i, j)),$$

где φ – функция перехода клетки σ из состояния в момент времени t в состояние в момент времени $(t+1)$; $\mathbb{N}(i, j)$ – окрестность клетки σ_{ij} .

Обычно правила перехода устанавливаются одинаковыми для всех автоматов. За один шаг работы КА выполняется обход всех клеток и определяются их новые состояния с учетом текущего состояния и окрестностей. Перед стартом работы КА случайно или предопределенно устанавливается начальное состояние клеток.

Даже при соблюдении простых правил КА может демонстрировать сложное поведение всего с четырьмя простыми правилами «Игры жизни Конвея», полной по Тьюрингу.

Нейронные клеточные автоматы (НКА) – это КА, где состояния ячеек есть векторы, а правила КА параметризуются и изучаются нейронными сетями [3].

Вариационный автокодер (ВА) – это оригинальная вероятностная генерирующая модель, которая воспроизводит данные с использованием модели скрытой переменной, такой, что

$$p(x) = \int_z p_\theta(x \mid z) p(z),$$

где $p(z)$ – фиксированный априорный закон; $p_\theta(x \mid z)$ – обученный декодер.

Параметры модели вычисляются посредством максимизации нижней границы правдоподобия $\log p(x)$. Они могут быть эффективно получены с использованием амортизированного вариационного вывода [4].

НКА учатся генерировать изображения, трехмерные структуры и функциональные артефакты, способные восстанавливаться при получении повреждений. Однако НКА может генерировать и регенерировать только один артефакт, на котором он обучен, не обладая различными порождающими свойствами современных вероятностных генеративных моделей.

В качестве решения этой проблемы в [5] предложена архитектура вариационных нейронных клеточных автоматов (ВНКА) – архитектура на основе НКА, которая устраняет эти ограничения, полагаясь только на локальные взаимодействия и самоорганизацию. На высоком уровне описываемая генеративная модель объединяет НКА с вариационным автоматом (ВА), используя НКА в качестве декодера в ВА.

Основной деятельностью в настоящее время являются эксперименты со стандартным декодером НКА и новой архитектурой НКА, которая дублирует все клетки каждые M шагов. Это приводит к повышению производительности генеративного моделирования и имеет вычислительное преимущество, заключающееся в том, что ВНКА вычисляет обновления только для живых клеток на каждом этапе, которых на ранних стадиях роста относительно немного.

Это не единственное существующее решение. Например, ВА-НКА [6], несмотря на название, не относится к ВА, поскольку не определяет порождающую модель и не использует вариационный вывод. Скорее, это автокодер, который декодирует набор весовых коэффициентов, которые параметризуют НКА, восстанавливающий изображения. Существует также аналогичная архитектура, использующая автокодер для декодирования весов НКА, которая затем генерирует изображения.

В отличие от обоих этих методов, рассматриваемый в публикации метод – это действительно генеративная модель, т. е. использует один изученный НКА и отбирает начальное состояние исходных клеток из предыдущего $p(z)$. Традиционный НКА определяет, «живая» или «мертвая» клетка, основываясь на состоянии клетки и состояниях клеток ее соседей. Это позволяет НКА

«расти» из небольшого начального количества живых клеток. Описываемый ВНКА использует другой подход, основанный на клеточном митозе.

Помимо этого предложены и иные мощные генеративные модели изображений, основанные на фреймворке вариационного автокодировщика. Обычно выделяют два типа декодеров: авторегрессионный и неавторегрессионный. Авторегрессионный декодер условно связывает свои выходные данные с предыдущими выходами, обычно расположенными выше и слева от указанного пикселя. Авторегрессионные декодеры настолько эффективны в прямом моделировании $p(x)$, что имеют тенденцию игнорировать латентные коды, что приводит к плохому восстановлению изображений [7].

Кроме того, они дорогостоящие в выборке, так как она должна происходить попиксельно. Декодер ВНКА не авторегрессионный, что делает выборку из него дешевле в плане затрачиваемых вычислительных и временных ресурсов. В отличие от других решений на основе глубоких сверток, ВНКА определяет относительно простую функцию декодирования, которая рассматривает только непосредственное окружение ячейки (пикселя) на каждом шаге и применяется итеративно на протяжении нескольких шагов. ВНКА, таким образом, стремится обучиться самоорганизующемуся генеративному процессу.

Каждые M шагов НКА обновляет все дублирующиеся ячейки, создавая новые, инициализированные в их собственном текущем состоянии справа и под ними самими, при этом «отодвигая» существующие ячейки в сторону. После K удвоений и $M(K + 1)$ шагов обновления НКА, конечные состояния ячеек используются для определения параметров вероятностного распределения, которое используется для моделирования данных.

Чтобы оптимизировать стабильность и процесс восстановления повреждений, ВНКА адаптирует процедуру обучения восстановлению повреждений к выбранным условиям генеративного моделирования [3]. Во время обучения половина каждой обучающей партии отбирается из набора предыдущих выборок и соответствующих им состояний ячеек после запуска НКА, z_T . Половина из этих состояний набора z_T подвергается повреждению за счет установки состояний ячеек равными нулю в случайном квадрате $H/2 \times W/2$. После того как НКА выполнит T шагов, все новые состояния z_T помещаются в набор, затем набор

перемешивается и сохраняются первые P выборов, где P соответствует фиксированному размеру текущего набора. Эффект заключается в том, что половина каждого обучающего пакета нормальная, а половина соответствует запуску НКА для дополнительных T шагов на предыдущем образце, из которых половина повреждена. Для половины, которая соответствует запуску НКА для дополнительных T шагов, вычисление градиента усекается, что соответствует усеченному обратному распространению во времени.

Для визуализации результатов ВНКА на рис. 1–3 продемонстрированы сравнительные данные о скрытых пространствах моделей ВНКА и глубокой сверточной базовой модели (модель, которая используется в качестве отправной точки при решении задачи машинного обучения, на которую настраиваются и тестируются более сложные модели). Используется улучшенная версия метода стохастического вложения соседей – SNE (Stochastic Neighbor Embedding). SNE – это метод снижения размерности, используемый для визуализации многомерных данных в двух или трех измерениях. Он позволяет сохранить локальную структуру и расстояния между близкими точками в многомерном пространстве при переходе к пространству меньшей размерности.

SNE строит вероятностную модель, где вероятность того, что точка i – соседняя для точки j , пропорциональна гауссовскому распределению от расстояния между i и j в многомерном пространстве. Затем SNE строит аналогичную вероятностную модель в пространстве меньшей размерности и минимизирует расхождение между двумя моделями, используя градиентный спуск. SNE позволяет сохранить расстояния между близкими точками в многомерном пространстве при переходе к пространству меньшей размерности. Визуализация данных в двух или трех измерениях с помощью SNE позволяет визуально оценить структуру и распределение данных, что может быть полезно в задачах машинного обучения и анализа данных.

T-SNE (t-Distributed Stochastic Neighbor Embedding) – это улучшенная версия метода снижения размерности SNE, которая позволяет сохранять не только локальную структуру данных, но и глобальную структуру. Он использует распределение t-Стюдента для моделирования вероятности выбора соседей в многомерном пространстве, что позволяет более эффективно сохранять глобальную структуру данных. Аналогично SNE,

t-SNE строит две вероятностные модели для многомерного и меньшего по размерности и минимизирует расхождение между ними, используя градиентный спуск.

Методы SNE, t-SNE широко используются для визуализации высокомерных данных в двух или трех измерениях – изображения, звуковые файлы или текстовые документы. Визуализация может помочь понять глобальную структуру и распределение данных, а также обнаружить скрытые паттерны, которые не могут быть обнаружены с помощью других методов.

На рис. 1 показано несколько линейных интерполяций между случайными цифрами в пространстве скрытых переменных ВНКА, обученного на бинаризованном MNIST (датасет, состоящий из изображений рукописных цифр от 0 до 9, в котором каждый пиксель изображения – бинарный, т. е. черный или белый). В случае MNIST точки в пространстве скрытых переменных обычно представляют собой векторы, полученные на выходе автокодировщика, который был обучен на изображениях цифр MNIST. Каждый вектор представляет собой скрытое представление (эмбединг) соответствующего изображения. Таким образом, две точки в пространстве скрытых переменных могут соответствовать двум разным изображениям цифр MNIST, а линейная интерполяция между этими точками может приводить к генерации новых изображений, которые представляют собой комбинацию (интерполяцию) двух исходных изображений.

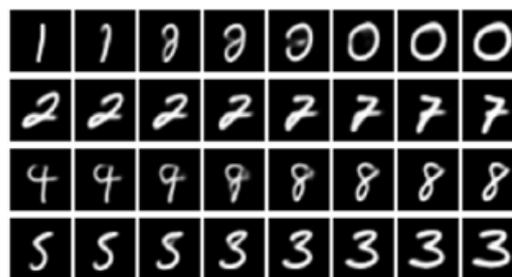


Рис. 1. Примеры линейных интерполяций между случайными рукописными цифрами
Fig. 1. Examples of linear interpolations between random handwritten digits

Рис. 2 и 3 показывают результат применения t-SNE для 5000 цифр из датасета, выбранных случайным образом для ВНКА и для глубокой сверточной базовой модели (модель, используемая в качестве отправной точки при решении задачи машинного обучения, на которую настраиваются и на которой тестируются более сложные модели) соответственно. На них точки различных оттенков соот-

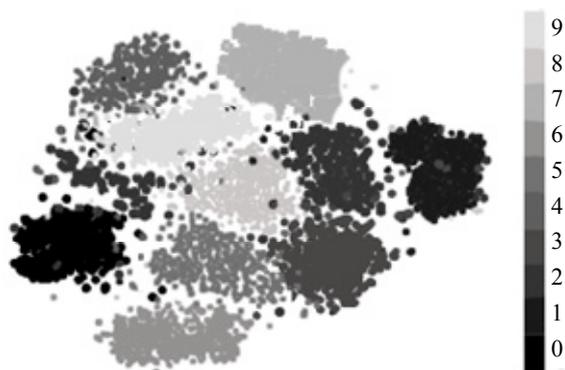


Рис. 2. Результат применения t-SNE для случайных 5000 цифр для вариационного нейронного клеточного автомата

Fig. 2. The result of using t-SNE for random 5000 digits for a variational neural cellular automaton

ветствуют разным цифрам. Соответствие цвета цифре указано в правой части рис. 2 и 3.

Видно, что скрытое пространство ВНКА, изображенное на рис. 2, имеет более выраженную структуру t-SNE (более явные границы между кластерами, более плотную группировку объектов внутри кластеров и более отдаленные отношения между кластерами), чем скрытое пространство рис. 3, и более четкое разделение цифровых кодировок. Это показывает, что ВНКА лучше различает цифры в различных рукописных вариациях.

Метод HyperNCA при построении моделей обучения с подкреплением. Описанный подход построения нейронных сетей помимо построения генеративных моделей показывает результаты при построении прочих моделей обучения с подкреплением. Предлагается новый «гиперсетевой» подход к созданию искусственных нейронных сетей на основе НКА.

Метод HyperNCA [8] позволяет «выращивать» нейронные сети, способные решать распространенные задачи обучения с подкреплением. Описывается как подход, рассмотренный ранее [3], может быть использован для построения сетей развития, способных преобразовывать свои веса для решения вариаций исходной задачи обучения с подкреплением.

Подход HyperNCA состоит из НКА, работающих на многомерной подложке, значения которой интерпретируются как веса другой сети. НКА используется в качестве математической модели, аналогичной развитию биологических организмов. Достигается это за счет того, что НКА обучается руководить процессом разработки сети политик, которая будет контролировать агента обучения с подкреплением. Процесс роста политической сети –

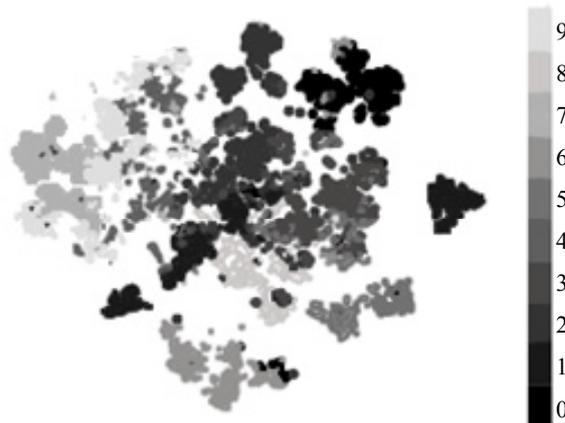


Рис. 3. Результат применения t-SNE для случайных 5000 цифр для глубокой сверточной базовой модели

Fig. 3. The result of applying t-SNE for random 5000 digits for a deep convolutional base model

это самоорганизующийся процесс, окончательная конфигурация которого возникает исключительно в результате локальных взаимодействий.

НКА состоит из трехмерных свертков, за которыми следует плотный слой (один из типов слоев нейронных сетей, в котором каждый нейрон связан со всеми нейронами предыдущего слоя). Для создания сети политик используется следующий алгоритм:

1. Случайным образом инициализируем подложку (высокоразмерное пространство, на котором работает нейронная сеть, представляет собой матрицу или тензор, состоящий из значений, которые интерпретируются как веса другой нейронной сети), выбирая значения из равномерного случайного распределения.

2. НКА наносится на подложку за конечное число этапов.

3. Значения одного из каналов подложки интерпретируются как веса сети политик P .

4. Результирующая политика P оценивается в задаче обучения с подкреплением T .

5. Результирующая пригодность P в задаче T определяет эволюцию весов НКА.

Подложка имеет форму $L \times C \times W$, где L – количество слоев сети политик; C – количество каналов сверточных слоев НКА; W – размер пространства наблюдения задачи (или размер пространства действий, если он больше, чем размер сети политик). Этот универсальный подход позволяет выращивать сети любого размера по глубине и ширине изменением формы подложки.

В данном подходе используется CMA-ES – стратегия эволюции адаптации ковариационной матрицы [9], алгоритм оптимизации популяции на основе «черного ящика», для обучения

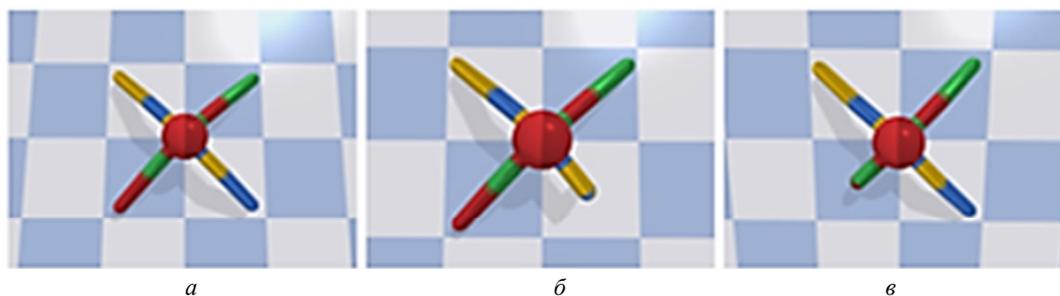


Рис. 4. Четырехногий агент при получении различных повреждений: *a* – отсутствие повреждений; *б* – повреждена задняя правая нога; *в* – повреждена задняя левая нога

Fig. 4. Four-legged agent when receiving various damage: *a* – no damage; *b* – damage to the hind right leg; *v* – damage to the hind left leg

HyperNCA. Алгоритмы эволюционных стратегий способны находить эффективные решения для задач обучения подкреплению как прямой оптимизации весовых коэффициентов политики, так и с кодированием политики в виде локальных правил обучения. Методы «черного ящика» – такие, как CMA-ES, имеют то преимущество, что не требуют вычисления градиентов и легко распараллеливаются.

В качестве базовой модели используется Линейная гиперсеть [10], которая состоит из нескольких линейных вложений (компонентов Гиперсети, которые соответствуют слоям целевой нейронной сети) и матрицы, генерирующей линейные веса для целевой нейронной сети. Каждое вложение в Гиперсети соответствует слою в целевой сети. Однако, так как целевые нейронные сети невелики, в данном случае используются многие вложения и объединяются выходы генератора весов для большей компактности модели. Другими словами, используется более широкая Гиперсеть, чем необходимо для целевой сети, чтобы получить более универсальную модель, которая может быть использована для обучения различных целевых сетей.

В качестве примера использования HyperNCA приводится модель четырехногого агента, который обучается передвижению в трехмерном пространстве. Основная идея эксперимента заключается в случайном изменении длины конечностей либо в отключении конечностей, что имитирует получение повреждений. После этого наблюдают, как агент переучивается выполнять задачу в данных условиях. Задача считается выполненной, если агенту удалось сдвинуться по любой из осей хотя бы на 1000 единиц. На рис. 4 продемонстрированы примеры различных состояний агента: в левой части повреждения отсутствуют, в следующих двух продемонстрированы повреждения различных ног.

На рис. 5 демонстрируется результат работы агента после обнуления подмножества синаптических весов. Границы временного шага, в котором обнулялись веса, обозначены линиями 1 и 2. Видно, что после первоначального снижения происходит быстрое восстановление. Награда представляет собой количество очков, заработанных агентом в ходе выполнения задачи (безразмерная величина). Стремление к увеличению награды позволяет агенту выбирать действия на пути к поставленной цели.

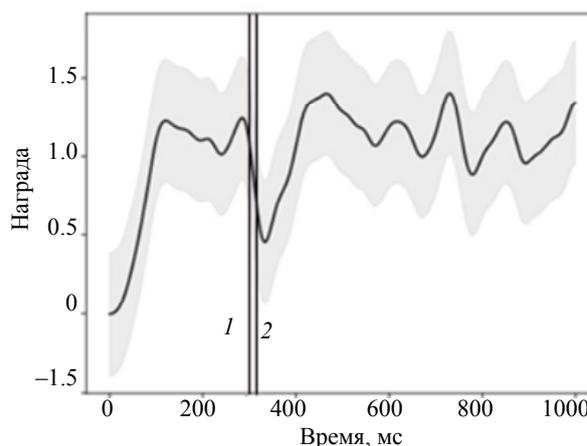


Рис. 5. Восстановление после нарушения весов сети

Fig. 5. Recovering from Network Weight Violations

Заключение. Рассмотрены различные аспекты клеточных автоматов, включая классические, вариационные нейронные клеточные автоматы и HyperNCA. Показано их использование для создания различных моделей обучения с подкреплением.

Рассмотрены вариационные нейронные клеточные автоматы, которые представляют собой расширение классических клеточных автоматов, позволяющее создавать более сложные и разнообразные модели, демонстрирующие свойства восстановления при повреждении.

Рассмотрена модель HyperNCA – новый тип клеточных автоматов для создания более слож-

ных и универсальных моделей. Приведен пример использования HyperNCA для моделирования различных процессов и обучения агентов.

Показана значимость применения клеточных автоматов в моделировании сложных процессов,

требующих от модели способности восстановления и переобучения при изменении свойств агента или среды.

Список литературы

1. A brief survey of deep reinforcement learning / K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath // IEEE Intern. Conf. on Neural Networks (ICNN). IEEE, 2017. P. 4365–4373. doi: 10.1109/MSP.2017.2743240.
2. Матюшкин И. В., Заплетина М. А. Обзор по тематике клеточных автоматов на базе современных отечественных публикаций // Компьютерные исследования и моделирование. 2019. Т. 11, № 1. С. 9–57. doi: 10.20537/2076-7633-2019-11-1-9-57.
3. Randazzo E., Mordvintsev A., Fouts C. Growing steerable neural cellular automata. URL: https://www.researchgate.net/publication/368688124_Growing_Steerable_Neural_Cellular_Automata (дата обращения 21.05.23).
4. Rezende D. J., Mohamed S., Wierstra D. Stochastic backpropagation and approximate inference in deep generative models // Proc. of the Intern. Conf. on Machine Learning (ICML), Beijing, China. 2014. P. 1278–1286.
5. Wolfram S. Cellular automata and complexity: Collected papers. CRC Press, 2018. 608 p.
6. Лобанов А. И. Модели клеточных автоматов // Компьютерные исследования и моделирование. 2010. Т. 2, № 3. С. 273–293.
7. Fixing a broken elbo / A. Alemi, B. Poole, I. Fischer, J. Dillon, R. Saurous, K. Murphy // Intern. Conf. on Machine Learning. PMLR, 2018. P. 159–168.
8. HyperNCA: Growing developmental networks with neural cellular automata / E. Najarro, Sudhakaran, S. C. Glanois, S. Risi // Proc. of the Conf. on Artificial Life (ALIFE 2021). Cambridge: MIT Press. 2021. P. 349–356.
9. Hansen N., Ostermeier A. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation // Proc. of IEEE Intern. Conf. on Evolutionary Computation. IEEE, 1996. P. 312–317.
10. Haldar T., Chowdhury D. R. Design of hash function using two dimensional cellular automata // Proc. of the Fifth Intern. Conf. on Mathematics and Computing. Springer, 2021. P. 33–45.

Информация об авторах

Мокрецов Никита Сергеевич – аспирант кафедры информационных систем СПбГЭТУ «ЛЭТИ».
E-mail: v.nikitamokrecov6374@gmail.com

Татарникова Татьяна Михайловна – д-р техн. наук, профессор кафедры информационных систем СПбГЭТУ «ЛЭТИ».
E-mail: v.tm-tatarn@yandex.ru
<http://orcid.org/0000-0002-6419-0072>

References

1. A brief survey of deep reinforcement learning / K. Arulkumaran, M. P. Deisenroth, M. Brundage, A. A. Bharath // IEEE Intern. Conf. on Neural Networks (ICNN). IEEE, 2017. P. 4365–4373. doi: 10.1109/MSP.2017.2743240.
2. Matjushkin I. V., Zapletina M. A. Obzor po tematike kletochnyh avtomatov na baze sovremennyh oteche-stvennyh publikacij // Komp'juternye issledovaniya i modelirovanie. 2019. T. 11, № 1. S. 9–57. doi: 10.20537/2076-7633-2019-11-1-9-57. (In Russ.).
3. Randazzo E., Mordvintsev A., Fouts C. Growing steerable neural cellular automata. URL: https://www.researchgate.net/publication/368688124_Growing_Steerable_Neural_Cellular_Automata (data obrashheniya 21.05.23).
4. Rezende D. J., Mohamed S., Wierstra D. Stochastic backpropagation and approximate inference in deep generative models // Proc. of the Intern. Conf. on Machine Learning (ICML), Beijing, China. 2014. P. 1278–1286.
5. Wolfram S. Cellular automata and complexity: Collected papers. CRC Press, 2018. 608 p.
6. Lobanov A. I. Modeli kletochnyh avtomatov // Komp'juternye issledovaniya i modelirovanie. 2010. T. 2, № 3. S. 273–293. (In Russ.).
7. Fixing a broken elbo / A. Alemi, B. Poole, I. Fischer, J. Dillon, R. Saurous, K. Murphy // Intern. Conf. on Machine Learning. PMLR, 2018. P. 159–168.
8. HyperNCA: Growing developmental networks with neural cellular automata / E. Najarro, Sudhakaran, S. C. Glanois, S. Risi // Proc. of the Conf. on Artificial Life (ALIFE 2021). Cambridge: MIT Press. 2021. P. 349–356.
9. Hansen N., Ostermeier A. Adapting arbitrary normal mutation distributions in evolution strategies: the covari-

ance matrix adaptation // Proc. of IEEE Intern. Conf. on Evolutionary Computation. IEEE, 1996. P. 312–317.

the Fifth Intern. Conf. on Mathematics and Computing. Springer, 2021. P. 33–45.

10. Haldar T., Chowdhury D. R. Design of hash function using two dimensional cellular automata // Proc. of

Information about the authors

Nikita S. Mokretsov – postgraduate student of Department of the Information Systems of Saint Petersburg Electrotechnical University.

E-mail: nikitamokrecov6374@gmail.com

Tatiana M. Tatarnikova – Dr Sci. (Tech.) Professor, Department of the Information Systems of Saint Petersburg Electrotechnical University.

E-mail: tm-tatarn@yandex.ru

<http://orcid.org/0000-0002-6419-0072>

Статья поступила в редакцию 25.05.2023; принята к публикации после рецензирования 05.06.2023; опубликована онлайн 25.09.2023.

Submitted 25.05.2023; accepted 05.06.2023; published online 25.09.2023.
