

УДК 004.85.032.26

**Г.А. Килин, Б.В. Кавалеров, Е.О. Ждановский,
И.В. Бахирев, Д.А. Опарин**

Пермский национальный исследовательский политехнический университет,
Пермь, Россия

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ РЕАЛИЗАЦИИ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Рассматривается разработанный программный комплекс, который позволяет реализовать алгоритм обучения с подкреплением для того, чтобы обучить искусственную нейронную сеть выполнять поставленные перед ней задачи. Считается, что обучение с подкреплением является частным случаем обучения с учителем, так как учителем нейронной сети является окружающая среда или созданная рабочая область, с которой взаимодействует агент в виде искусственной нейронной сети. Агент взаимодействует со средой и за каждое свое действие получает вознаграждение или наказание. Пытаясь заработать максимальное количество очков в виде суммы всех вознаграждений нейронная сеть учится взаимодействовать со средой, достигая необходимого нам результата. В статье предоставлены примеры использования как нейронных сетей, так и примеров настройки весов сети на основе обучения с подкреплением и достигнутые на их основе результаты. Данные примеры показывают актуальность применения нейронных сетей для решения поставленных задач и перспективы применения различных способов обучения искусственных нейронных сетей. Для удобства и простоты изложения материала в статье в качестве примера была рассмотрена игра крестики-нолики, в которую нейронная сеть «училась» играть, а также рассмотрены особенности вознаграждения нейронной сети, представленные в виде списка правил, при взаимодействии со средой. В качестве алгоритма оптимизации весов нейронной сети использовался генетический алгоритм как самый простой и удобный в реализации. Искусственная нейронная сеть и генетический алгоритм были написаны с нуля на языке программирования Java.

Ключевые слова: нейронные сети, нейросетевая модель, обучение с подкреплением, агент, генетический алгоритм.

**G.A. Kilin, B.V. Kavalеров, E.O. Zhdanovsky,
I.V. Bakhirev, D.A. Oparin**

Perm National Research Polytechnic University, Perm, Russian Federation

SOFTWARE COMPLEX TO IMPLEMENT REINFORCEMENT LEARNING

The article discusses developed software complex that allows implementing reinforcement learning algorithm, in order to teach the artificial neural network to perform its tasks. It is considered that reinforcement learning is a special case of training with the teacher, since the neural network teacher is the environment or the created working area, with which it interacts as an artificial neural network. The agent interacts with the environment and receives reward or punishment for each of his actions. Trying to earn the maximum number of points, in the form of a sum of all rewards, the neural network learns to

interact with the environment, achieving the desired result. The article provides examples of how to use both neural networks and examples of configuring network weights based on reinforcement learning and the results achieved on their basis. These examples show both the neural networks relevance use for solving the tasks posed, and the prospects for using artificial neural networks various training methods. For the convenience and simplicity of the material presentation in the article, as an example, we considered the tic-tac-toe game, into which the neural network learned to play, and also the features of the neural network reward presented in the form of a list of rules, when interacting with the environment. As an algorithm for optimizing the weights of a neural network, the genetic algorithm was used, as the simplest and most convenient in implementation. The artificial neural network and the genetic algorithm were written from scratch in the Java programming language.

Keywords: neural networks, neural network model, reinforcement learning, agent, genetic algorithm.

Введение. В наше время математическое моделирование стало мощным инструментом, который используется во всех сферах деятельности человека. В данной же статье будут рассмотрены модели, построенные по принципу организации и функционирования биологических сетей нервных клеток живого организма, иначе говоря, искусственные нейронные сети (ИНН). Нейронные сети способны решать широкий круг задач, они находят применение в распознавании изображений [1, 2], пешеходов [3–5], диагностики систем [6–10], в задачах идентификации газотурбинных установок [11, 12], в играх в роли искусственного интеллекта, как, например, в игре «Go» [13], а также в классификации или кластеризации данных. В зависимости от поставленной задачи выбирается архитектура обучения нейронной сети. В стандартном случае методы обучения нейронных сетей делят на два типа: обучение с учителем и обучение без учителя, но в последнее время начали выделять обучение с подкреплением.

Обучение с учителем применяется, когда заранее известны входные и соответствующие им выходные значения исследуемого объекта, такие сети больше всего подходят для решения задач классификации данных. Однако данный тип обучения подходит лишь при условии, что имеется достаточный объем информации для составления обучающей и тестовой выборки. При отсутствии этого условия использование данного типа невозможно. Наиболее известным среди алгоритмов обучения с учителем является метод обратного распространения ошибки. Нейронные сети, обучающиеся без учителя, имеет смысл применять в тех случаях, когда нет никакой априорной информации о том, какими должны быть выходные значения сети. Такие сети, как правило, применяются для решения задач кластеризации данных. Однако на выполнение задач в классификации либо идентификации потребуется значительно больше времени. Нейронные сети, использующие обучение с подкреплением, получают всю необходимую информацию из взаи-

модействия с окружающей средой, соответственно данные сети аналогично адаптивны. Данный тип обучения будет рассмотрен более подробно, а также будет продемонстрирован пример обучения нейронной сети в игру крестики-нолики.

Программный комплекс. Программный комплекс позволяет реализовать искусственную нейронную сеть и генетический алгоритм, который выступает в роли алгоритма оптимизации для настройки параметров нейронной сети. Искусственные нейронные сети являются моделью мозга. Архитектура искусственных нейронных сетей является достаточно простой (рис. 1). ИНН состоит из входного слоя, скрытого слоя и выходного слоя. При этом количество нейронов во входном и скрытом слое устанавливается в зависимости от условий решаемой задачи.

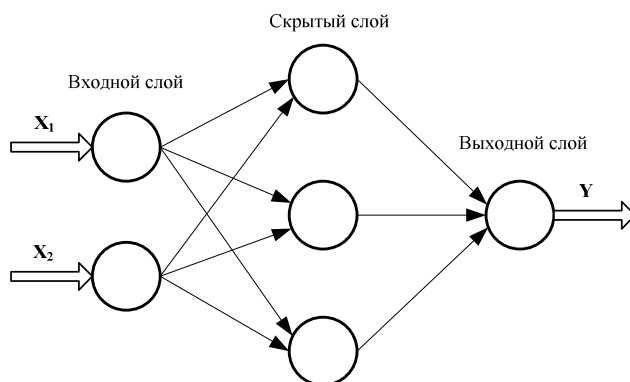


Рис. 1. Архитектура нейронной сети (X_1, X_2 – входные сигналы, Y – выходной сигнал)

На данном выше рисунке представлена сеть прямого распространения, когда выход каждого нейрона предыдущего слоя связан с каждым нейроном последующего слоя. Такая архитектура является самой простой и имеет широкое распространение [14].

Каждый слой ИНН состоит из нейронов (рис. 2). На нейрон поступают сигналы $x_1 \dots x_n$, при этом каждый сигнал имеет свой вес $w_1 \dots w_n$. Выход нейрона вычисляется следующим образом: $Y = f(S)$, где $S = \sum_{i=0}^n w_i x_i$. Немаловажным аспектом является выбор функции активации нейрона (рис. 3). На рисунке представлен гиперболический тангенс $\text{th } x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$, который относится к сигмоидальным функциям и является одной из самых распространенных функций для искусственных нейронных сетей [14].

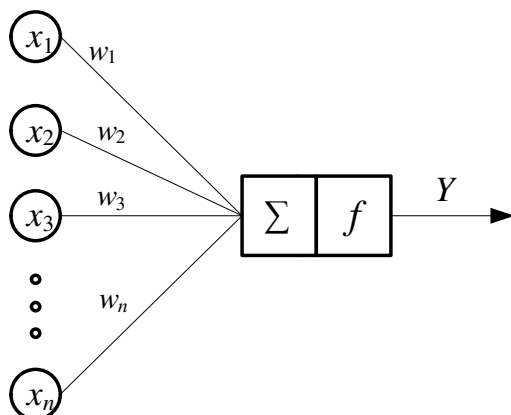


Рис. 2. Искусственный нейрон ($x_1 \dots x_n$ – входные сигналы на нейрон, Y – выходной сигнал нейрона)

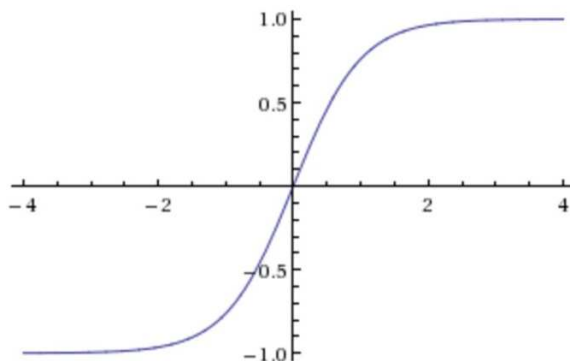


Рис. 3. Гиперболический тангенс

Обучение с подкреплением. Нейронная сеть будет показывать хороший результат только в случае корректного обучения. Обучение нейронной сети является одним из самых сложных этапов при работе с нейронными сетями [14]. Рассмотрим один из алгоритмов обучения нейронной сети, а именно обучение с подкреплением (рис. 4).

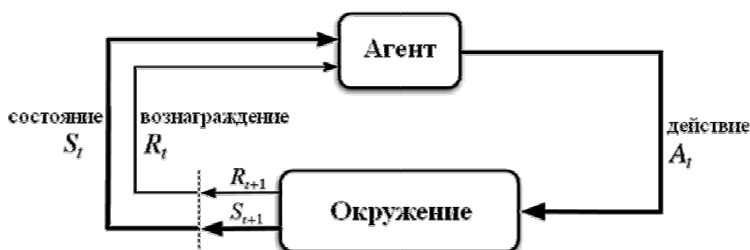


Рис. 4. Обучение с подкреплением

Обучение с подкреплением – совокупность методов машинного обучения, в ходе которых так называемый агент обучается, взаимодействуя с окружающей средой. Нейронная сеть взаимодействует с определённой областью или окружающей средой, применяя различные действия, переходя в новое состояние. В зависимости от этих действий и переходов агент получает числовое вознаграждение, которое может быть как положительным, так и отрицательным. Какое именно вознаграждение получит сеть, решается из установленных правил, если установить неоптимальный набор правил, то необходимого результата либо будет сложно достичь, либо невозможно в принципе.

Формально простейшая модель обучения с подкреплением состоит из следующих пунктов:

- 1) множества состояний окружения S ;
- 2) множества действий A ;
- 3) множества скалярных «выигрышей».

В произвольный момент времени t агент характеризуется состоянием $s_t \in S$ и множеством возможных действий $A(s_t)$. Выбирая действие $a \in A(s_t)$, он переходит в состояние s_{t+1} и получает выигрыш R_t . Основываясь на таком взаимодействии с окружающей средой, агент, обучающийся с подкреплением, должен выработать стратегию $\pi: S \rightarrow A$, позволяя максимизировать величину вознаграждения $R = R_0 + R_1 + \dots + R_n$. Благодаря данному подходу удается добиться потрясающих результатов в решении довольно сложных задач [15–17]. Из данных примеров можно сделать вывод, что алгоритм обучения с подкреплением универсален и способен решать множество поставленных задач.

Крестики-нолики. В качестве игры для проверки работы алгоритма обучения с подкреплением была выбрана игра крестики-нолики, так как благодаря минимуму правил она достаточно проста в реализации.

1. **Вознаграждение.** В ходе взаимодействия со средой агент в виде нейронной сети совершает действия, за которые получает как положительное, так и отрицательное вознаграждение. Для игры были реализованы следующие правила получения вознаграждения:

1) *Неверный ход.* Под неверным ходом подразумевается ход в ячейку, которая уже занята либо самим агентом, либо противником. Такая ситуация возможна, так как изначально нейронная сеть совершенно не обучена и может совершать ход в занятую ячейку. Поскольку в реальной игре такой ситуации не должно возникать, этот случай

является самым «плохим» вариантом, и агент получает отрицательное вознаграждение: $R_i = R_{i-1} - 100 \cdot i$, где R_i – результирующее вознаграждение после i хода; i – номер хода.

В итоге, чем позже агент совершит свой неверный ход, тем меньше баллов он получает.

Изначально алгоритм предусматривал окончание игры в случае возникновения такой ситуации. Данный подход показал свою малую эффективность, так как почти все необученные агенты совершали такие ходы, получая отрицательное вознаграждение и заканчивая игру. В итоге было проблематично оценить умения агента взаимодействовать со средой. Поэтому было принято решение, что в ходе игры, когда агент совершает ход в уже занятую ячейку, то игра не заканчивается, а ход совершается случайным образом в любую свободную ячейку. При этом агент так же получает отрицательное вознаграждение.

2) *Проигрыш*. Поскольку проигрыш не является целью обучения агента, то за данную ситуацию он аналогично получает отрицательное вознаграждение: $R_i = R_{i-1} - 100/i$. Чем позже агент проигрывает, тем больше баллов он получает.

3) *Ничья*. Для данного исхода игры было принята следующая стратегия: $R_i = R_{i-1} - 50$, так как конечная цель – научить агента побеждать.

4) *Верный ход* является полной противоположностью «неверного хода». За каждый верный ход агент получает вознаграждение в виде: $R_i = R_{i-1} + 100i$.

5) *Выигрыш*. Самым успешным исходом является выигрыш агента, за который он получает максимально возможное вознаграждение в виде: $R_i = R_{i-1} + 1000$.

2. *Противник*. Для роли противника была написана простая программа, которая совершает ход в случайную свободную ячейку игрового поля.

3. *Процесс игры*. Было принято, что первым ходит агент, а уже потом совершает свой ход противник. В итоге, поскольку поведение противника является случайным, количество очков, набранных агентом, может сильно отличаться от игры к игре. Для того чтобы усреднить получаемый выигрыш, было принято решение играть каждому агенту не менее 100 игр. По результатам данных игр происходит финальная оценка вознаграждения.

4. Процесс обучение. Процесс обучения игрового агента заключается в настройке весов искусственной нейронной сети. Одним из способов настройки весов ИНН является генетический алгоритм. Генетический алгоритм (ГА) – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Генетический алгоритм представляет собой комбинированный подход. Механизмы скрещивания и мутации в каком-то смысле реализуют переборную часть метода, а отбор лучших решений – градиентный спуск [18].

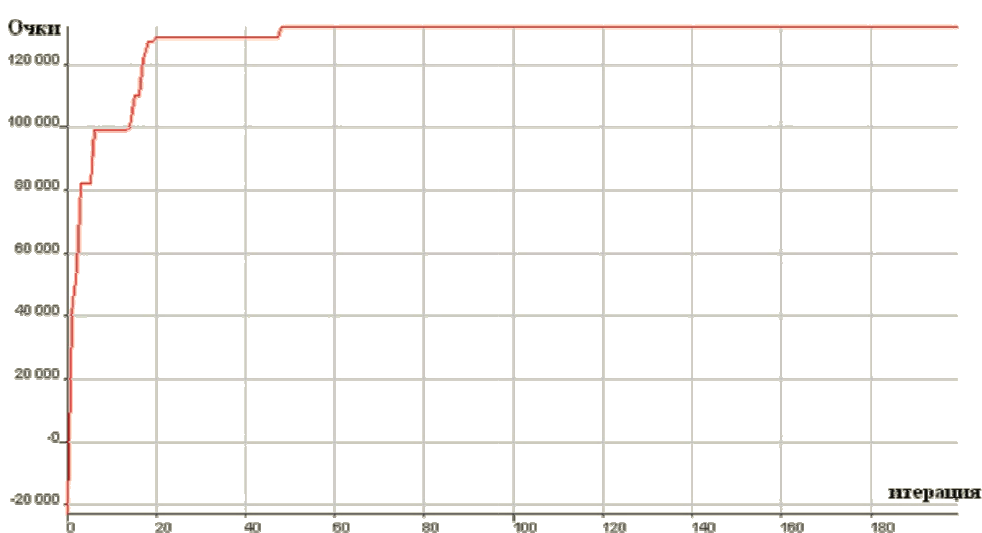


Рис. 5. График изменения количества набранных очков лучшим агентом из популяции за 100 игр (по оси ординат – очки, по оси абсцисс – количество итераций генетического алгоритма)

В самом начале формируется «популяция» из 100 особей, где каждая особь является «хромосомой», в которой содержатся все веса одной единственной нейронной сети. В ходе работы генетического алгоритма [18] происходят итерационные процессы отбора, кроссинговера и мутации. Эти процессы приводят к формированию новых популяций и тому, что некоторые особи в этих популяциях начинают лучше играть в игру. Работа генетического алгоритма повторяется до тех пор, пока не достигнуто условие останова.

5. Результаты обучения. На рис. 5 представлен график изменения счета за 100 игр лучшего агента среди всей популяции.

В таблице дано сравнение обученной и необученной нейронных сетей при отыгрыше 1000 игр.

Результаты обучения нейронной сети

Наименование	Победа	Ничья	Проигрыш
Необученная нейронная сеть	292	343	365
Обученная нейронная сеть	741	127	132

Необученной нейронной сети удается добиться таких результатов только благодаря тому, что алгоритм игры предусматривает, что при неправильном ходе агента ход в итоге совершается в случайную свободную ячейку. Фактически в данном случае происходит игра между алгоритмами, которые ходят в случайную свободную ячейку. Как результат, распределение побед, ничьих и проигрышей примерно одинаковое.

В случае обученной нейронной сети наглядно видно, что количество побед значительно возросло, а количество ничьих и проигрышей уменьшилось. При этом данная нейронная сеть научилась выигрывать следующим образом: пытаясь сразу совершить ходы для достижения победы по диагонали и не пытаясь совершать другие действия для достижения победы. Данная стратегия обусловлена тем, что при игре против такого простого противника и при наборе реализованных правил ее достаточно, чтобы выигрывать в большинстве случаев и получать максимально возможное вознаграждение.

Выводы. Достаточно часто возникает ситуация когда для обучения ИНС мало обучающих примеров либо они отсутствуют полностью. Особенно это заметно при обучении ИНС компьютерным играм [19, 20], где количество возможных состояний агента необычайно велико. Именно в этих случаях на помощь приходит обучение с подкреплением, где отсутствуют обучающие примеры, и ИНС в виде агента в среде самостоятельно обучается, достигая необходимого результата. Находя свое применение в решении практических задач, например, для предсказания потребления электроэнергии [21–23] и предсказания трафика [24–27], искусственные нейронные сети еще раз показывали свою практическую ценность и необходимость. Разработанный программный комплекс является тем самым инструментом, который позволяет решать необходимый круг задач, которые уже определяет специалист, работающий в данной области.

Окружение в виде игры крестики-нолики является достаточно простым примером среды, чтобы разобраться в особенностях функционирования алгоритма обучения с подкреплением, понять его основные преимущества и недостатки, а также определить круг задач, в которых применение данного алгоритма является целесообразным.

Библиографический список

1. Convolutional networks for images, speech, and time series / Y. LeCun, Y. Bengio [et al.] // *The handbook of brain theory and neural networks*. – 1995. – Vol. 3361. – № 10. – P. 1995.
2. What is the best multi-stage architecture for object recognition? / K. Jarrett [et al.] // *Computer Vision, 2009 IEEE 12th International Conference on. IEEE*. – 2009. – P. 2146–2153.
3. Pedestrian detection with unsupervised multi-stage feature learning / P. Sermanet [et al.] // *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. – 2013. – P. 3626–3633.
4. Pedestrian detection based on deep convolutional neural network with ensemble inference network / H. Fukui [et al.] // *Intelligent Vehicles Symposium (IV), 2015 IEEE*. – 2015. – P. 223–228.
5. Switchable deep network for pedestrian detection / P. Luo [et al.] // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. – 2014. – P. 899–906.
6. Khomfoi S., Tolbert L.M. Fault diagnostic system for a multilevel inverter using a neural network // *IEEE Transactions on Power Electronics*. – 2007. – Vol. 22. – № 3. – P. 1062–1069.
7. Samanta B., Al-Balushi K.R. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features // *Mechanical systems and signal processing*. – 2003. – Vol. 17. – № 2. – P. 317–328.
8. Jayaswal P., Verma S.N., Wadhvani A.K. Development of EBP-Artificial neural network expert system for rolling element bearing fault diagnosis // *Journal of Vibration and Control*. – 2011. – Vol. 17. – № 8. – P. 1131–1148.
9. Artificial neural networks in medical diagnosis / F. Amato, A. López Rodríguez, E.M. PeñaandMéndez, P. Vañhara, A. Hampl, J. Havel // *Journal Appl. Biomed*. – 2013. – № 11. – P. 47–58.
10. Saravanan N., Siddabattuni V.N.S.K., Ramachandran K.I. Fault diagnosis of spur bevel gear box using artificial neural network (ANN)

and proximal support vector machine (PSVM) // *Applied Soft Computing*. – 2010. – Vol. 10. – № 1. – P. 344–360.

11. Artificial neural network–based system identification for a single-shaft gas turbine / H. Asgari [et al.] // *Journal of Engineering for Gas Turbines and Power*. – 2013. – Vol. 135. – № 9. – P. 092601.

12. Asgari H., Chen X.Q., Sainudiin R. Modelling and simulation of gas turbines // *International Journal of Modelling, Identification and Control*. – 2013. – Vol. 20. – № 3. – P. 253–270.

13. Mastering the game of Go with deep neural networks and tree search / D. Silver [et al.] // *Nature*. – 2016. – Т. 529. – № 7587. – С. 484–489.

14. Хайкин С. Нейронные сети: полный курс. – 2-е изд. – М.: Вильямс, 2008.

15. Device Placement Optimization with Reinforcement Learning / A. Mirhoseini [et al.] // *International Conference on Machine Learning*. – 2017. – P. 2430–2439.

16. Karthik Narasimhan, Tejas Kulkarni, Regina Barzilay. Language understanding for text-based games using deep reinforcement learning // *Proc. of Empirical Methods in Natural Language Processing*. – EMNLP, 2015.

17. Evans R., Gao J. Deepmind AI reduces Google data centre cooling bill by 40 % // *DeepMind blog*. – 2016. – Vol. 20.

18. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – М.: Горячая линия-Телеком, 2006. – 452 с.

19. Wong S. Competitive Reinforcement Learning in Atari Games // *AI 2017: Advances in Artificial Intelligence: 30th Australasian Joint Conference, Melbourne, VIC, Australia, August 19–20, 2017, Proceedings*. – Springer, 2017. – Vol. 10400. – P. 14.

20. Human-level control through deep reinforcement learning / V. Mnih [et al.] // *Nature*. – 2015. – Vol. 518. – № 7540. – P. 529.

21. Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis / K. Li. [et al.] // *Energy and Buildings*. – 2015. – Vol. 108. – P. 106–113.

22. Platon R., Dehkordi V. R., Martel J. Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis // *Energy and Buildings*. – 2015. – Vol. 92. – P. 10–18.

23. Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings / Y.T. Chae [et al.] // *Energy and Buildings*. – 2016. – Vol. 111. – P. 184–194.

24. An improved fuzzy neural network for traffic speed prediction considering periodic characteristic / J. Tang [et al.] // *IEEE Transactions on Intelligent Transportation Systems*. – 2017. – Vol. 18. – № 9. – P. 2340–2350.

25. Habtemichael F.G., Cetin M. Short-term traffic flow rate forecasting based on identifying similar traffic patterns // *Transportation Research Part C: Emerging Technologies*. – 2016. – Vol. 66. – P. 61–78.

26. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling / F. Moretti [et al.] // *Neurocomputing*. – 2015. – Vol. 167. – P. 3–7.

27. Kumar K., Parida M., Katiyar V.K. Short term traffic flow prediction in heterogeneous condition using artificial neural network // *Transport*. – 2015. – Vol. 30. – № 4. – P. 397–405.

References

1. LeCun Y., Bengio Y. [et al.]. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 1995, vol. 3361, no. 10, P. 1995.

2. Jarrett K. [et al.]. What is the best multi-stage architecture for object recognition? *Computer Vision, 2009 IEEE 12th International Conference on. IEEE*, 2009, pp. 2146-2153.

3. Sermanet P. [et al.]. Pedestrian detection with unsupervised multi-stage feature learning. *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 3626-3633.

4. Fukui H. [et al.]. Pedestrian detection based on deep convolutional neural network with ensemble inference network. *Intelligent Vehicles Symposium (IV), 2015 IEEE*, 2015, pp. 223-228.

5. Luo P. [et al.]. Switchable deep network for pedestrian detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 899-906.

6. Khomfoi S., Tolbert L.M. Fault diagnostic system for a multilevel inverter using a neural network. *IEEE Transactions on Power Electronics*, 2007, vol. 22, no. 3, pp. 1062-1069.

7. Samanta B., Al-Balushi K.R. Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. *Mechanical systems and signal processing*, 2003, vol. 17, no. 2, pp. 317-328.

8. Jayaswal P., Verma S.N., Wadhvani A.K. Development of EBP-Artificial neural network expert system for rolling element bearing fault diagnosis. *Journal of Vibration and Control*, 2011, vol. 17, no. 8, pp. 1131-1148.

9. Amato F., López Rodríguez A., PeñaandMéndez E.M., Vañhara P., Hampl A., Havel J. Artificial neural networks in medical diagnosis. *Journal Appl. Biomed*, 2013, no. 11, pp. 47-58.

10. Saravanan N., Siddabattuni V.N.S.K., Ramachandran K.I. Fault diagnosis of spur bevel gear box using artificial neural network (ANN) and proximal support vector machine (PSVM). *Applied Soft Computing*, 2010, vol. 10, no. 1, pp. 344-360.

11. Asgari H. [et al.]. Artificial neural network-based system identification for a single-shaft gas turbine. *Journal of Engineering for Gas Turbines and Power*, 2013, vol. 135, no. 9, P. 092601.

12. Asgari H., Chen X.Q., Sainudiin R. Modelling and simulation of gas turbines. *International Journal of Modelling, Identification and Control*, 2013, vol. 20, no. 3, pp. 253-270.

13. Silver D. [et al.]. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, vol. 529, no. 7587, pp. 484-489.

14. Khaikin S. Neironnye seti [Neural networks]. 2nd ed. Moscow: Vil'iams, 2008.

15. Mirhoseini A. [et al.]. Device Placement Optimization with Reinforcement Learning. *International Conference on Machine Learning*, 2017, pp. 2430-2439.

16. Karthik Narasimhan, Tejas Kulkarni, Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. *Proc. of Empirical Methods in Natural Language Processing. EMNLP*, 2015.

17. Evans R., Gao J. Deepmind AI reduces Google data centre cooling bill by 40 %. *DeepMind blog*, 2016, vol. 20.

18. Rutkovskaia D., Pilin'skii M., Rutkovskii L. Neironnye seti, geneticheskie algoritmy i nechetkie sistemy [Neural networks, genetic algorithms and indistinct systems]. Moscow. Goriachaia liniia-Telekom, 2006. 452 p.

19. Wong S. Competitive Reinforcement Learning in Atari Games. *AI 2017: Advances in Artificial Intelligence: 30th Australasian Joint Conference, Melbourne, VIC, Australia, August 19-20, 2017, Proceedings*. Springer, 2017, vol. 10400. P. 14.

20. Mnih V. [et al.]. Human-level control through deep reinforcement learning. *Nature*, 2015, vol. 518, no. 7540, P. 529.

21. Li K. [et al.]. Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis. *Energy and Buildings*, 2015, vol. 108, pp. 106-113.

22. Platon R., Dehkordi V. R., Martel J. Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. *Energy and Buildings*, 2015, vol. 92, pp. 10-18.

23. Chae Y.T. [et al.]. Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings. *Energy and Buildings*, 2016, vol. 111, pp. 184-194.

24. Tang J. [et al.]. An improved fuzzy neural network for traffic speed prediction considering periodic characteristic. *IEEE Transactions on Intelligent Transportation Systems*, 2017, vol. 18, no. 9. pp. 2340-2350.

25. Habtemichael F.G., Cetin M. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. *Transportation Research Part C: Emerging Technologies*, 2016, vol. 66, p. 61-78.

26. Moretti F. [et al.]. Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling. *Neurocomputing*, 2015, vol. 167, p. 3-7.

27. Kumar K., Parida M., Katiyar V.K. Short term traffic flow prediction in heterogeneous condition using artificial neural network. *Transport*, 2015, vol. 30, no. 4, p. 397-405.

Сведения об авторах

Килин Григорий Александрович (Пермь, Россия) – старший преподаватель, магистр кафедры «Электротехника и электромеханика» Пермского национального исследовательского политехнического университета (614990, Пермь, Комсомольский пр. 29, e-mail: thisisforasm@rambler.ru).

Кавалеров Борис Владимирович (Пермь, Россия) – доктор технических наук, доцент, заведующий кафедрой «Электротехника и электромеханика» Пермского национального исследовательского политехнического университета (614990, Комсомольский проспект 29, e-mail: kbv@pstu.ru).

Ждановский Евгений Олегович (Пермь, Россия) – аспирант, магистр кафедры «Электротехника и электромеханика» Пермского национального исследовательского политехнического университета (614990, Комсомольский проспект 29, e-mail: zhdanovskiy.e@gmail.com).

Бахирев Иван Владимирович (Пермь, Россия) – старший преподаватель, магистр кафедры «Электротехника и электромеханика» Пермского национального исследовательского политехнического университета (614990, Комсомольский проспект 29, e-mail: bahirevy@mail.ru).

Опарин Денис Андреевич (Пермь, Россия) – старший преподаватель, магистр кафедры «Электротехника и электромеханика» Пермского национального исследовательского политехнического университета (614990, Комсомольский проспект 29, e-mail: dlowarp@gmail.com).

About the author

Kilin Grigorij Aleksandrovich (Perm, Russian Federation) – senior lecturer, master of the Electro technical and electromechanical department, Perm national research polytechnic university, 614990, Komsomol'skij prospekt 29, e-mail: thisisforasm@rambler.ru.

Kavalerov Boris Vladimirovich (Perm, Russian Federation) – Doctor of Technical Sciences, head of the Electro technical and electromechanical department, Perm national research polytechnic university, 614990, Komsomol'skij prospekt 29, e-mail: kbv@pstu.ru.

Zhdanovskij Evgenij Olegovich (Perm, Russian Federation) – post-graduate student, master of the Electro technical and electromechanical department, Perm national research polytechnic university, 614990, Komsomol'skij prospekt 29, e-mail: zhdanovskiy.e@gmail.com.

Bahirev Ivan Vladimirovich (Perm, Russian Federation) – senior lecturer, master of the Electro technical and electromechanical department, Perm national research polytechnic university, 614990, Komsomol'skij prospekt 29, e-mail: bahirevy@mail.ru.

Oparin Denis Andreevich (Perm, Russia) – senior lecturer, master of the Electro technical and electromechanical department, Perm national research polytechnic university, 614990, Komsomol'skij prospekt 29, e-mail: dlowarp@gmail.com.

Получено 09.07.2018