

УДК 62-83: 531.3

Сортировка массива целых чисел с использованием нейронной сети*

А.А. ВОЕВОДА¹, В.Л. ПОЛУБИНСКИЙ², Д.О. РОМАННИКОВ³

¹ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, доктор технических наук, профессор. E-mail: voevoda@ucit.ru

² 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент. E-mail: polubinsky@tiger.cs.nstu.ru

³ 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, кандидат технических наук, доцент. E-mail: dmitry.romannikov@gmail.com

На сегодняшний момент существует достаточно большое количество различных алгоритмов и структур данных, на основании которых построены все современные программные системы. Но, с другой стороны, данные алгоритмы плохо подходят для решения задач распознавания образов, классификации и регрессии, для которых существуют специальные методы и/или применяются решения, построенные на основании нейронных сетей. Нейронные сети, по своей сущности, являются универсальными аппроксиматорами, т. е. могут с большой степенью точности повторить, например, заданную кривую. Также из этого следует, что в отличие от классических алгоритмов системы, основанные на нейронных сетях, будут всегда иметь какой-то процент ошибок из-за своей вероятностной природы. В данной работе предлагается алгоритм сортировки массива целых чисел с использованием нейронной сети. Данный алгоритм представляет собой симбиоз нейронной сети и классических алгоритмов. Приводится структурная схема алгоритма, на которой представлено несколько составных частей: нейронная сеть для поиска минимального элемента входного вектора; часть системы, которая представляет найденный минимальный элемент на место в сортированной части массива. Последняя часть системы реализована в форме классических алгоритмов, но может быть заменена вырожденным вариантом нейронной сети, в которой веса представляют варианты замены элементов массива. В работе была проведена проверка предлагаемого алгоритма. В результате из 10 000 тестов 248 оказались ошибочными, что соответствует 2,48 % ошибки. Приведены рассуждения о возможных причинах возникновения ошибок.

Ключевые слова: нейронная сеть, алгоритмы, сортировка, классификация, Matlab, вероятностные системы, обучение, вероятностные алгоритмы, персептрон

DOI: 10.17212/1814-1196-2016-2-151-157

* Статья получена 18 апреля 2016 г.

ВВЕДЕНИЕ

В настоящий момент в индустрии разработки программного обеспечения сложилась ситуация, при которой уже создана достаточно большая база для прикладного программирования, включающая в себя различные алгоритмы и структуры данных [1, 2]. Данные знания, лежащие в основе практически любой программы, позволяют создавать достаточно сложные программные продукты. С другой стороны, существующие классические алгоритмы и структуры данных плохо подходят для решения задач распознавания образов [8, 10, 12], классификации [8–12] и кластеризации [8, 14], для решения которых в последнее время применяют либо алгоритмы машинного обучения, либо глубокое обучение, основанное на многослойных сворачиваемых нейронных сетях [3–7, 12, 13, 15].

В статье предлагается симбиоз из двух подходов к созданию алгоритмов, а именно алгоритм сортировки целочисленного массива при помощи нейронной сети.

АЛГОРИТМ

Реализуемый алгоритм сортировки [1, 2] состоит из двух последовательных шагов: 1) поиска наименьшего элемента массива; 2) перестановки минимального элемента массива и элемента массива, на который ссылается указатель. Изначально данный указатель ссылается на первый элемент, а после каждого второго шага он увеличивается на единицу.

Системы, построенные на базе нейронных сетей, по своей природе имеют вероятностное решение [8], т. е. алгоритмы, построенные на их основе, будут всегда иметь какой-то процент ошибок, и по этой причине они не применяются в классическом программировании, где предполагается доказательство алгоритмов. С другой стороны, нейронные сети создавались по аналогии со строением человеческого мозга [3, 4], который способен решать алгоритмические задачи. Тогда решение алгоритмических задач при помощи нейронных сетей имеет смысл. В данном разделе приводится пример такой системы. Очевидно, что в настоящее время создание и обучение нейронной сети, имитирующей работу человеческого мозга, не представляется возможным, поэтому в данной работе сделаны некоторые упрощения.

Структурная схема нейронной сети, реализующей алгоритм сортировки, представлена на рис. 1.

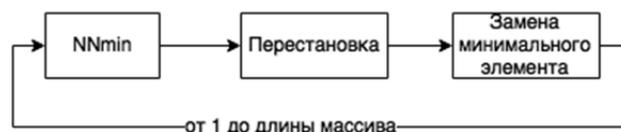


Рис. 1. Схематичное представление алгоритма сортировки с использованием нейронной сети

В данной структурной схеме, изображенной на рис. 2, *NNmin* – нейронная сеть, реализующая выявление минимального элемента. Результатом работы сети *NNmin* является такой вектор, что с большой вероятностью *i*-й эле-

мент вектора минимальный. Помимо нейронной сети, основанной на многослойном перцептроне [15], в данной ее реализации также присутствует звено задержки z^{-1} , через которое передаются входные элементы вектора для выбора минимального элемента. Выбор осуществляется путем умножения выходного значения нейронной сети $NNmin$ на входной элемент, при этом выходное значение вектора нейронной сети предварительно округляется либо до единицы, либо до нуля. Далее вектор чисел поступает на часть нейронной сети, которая изображена на рис. 2 справа. В данной части сети начальные веса получены не путем обучения сети, а заранее определены. Изначально все веса имеют нулевые значения, кроме тех, которые ведут в нейрон А (рис. 2). Таким образом, из всего множества чисел входного вектора минимальное число поступает в нейрон А.

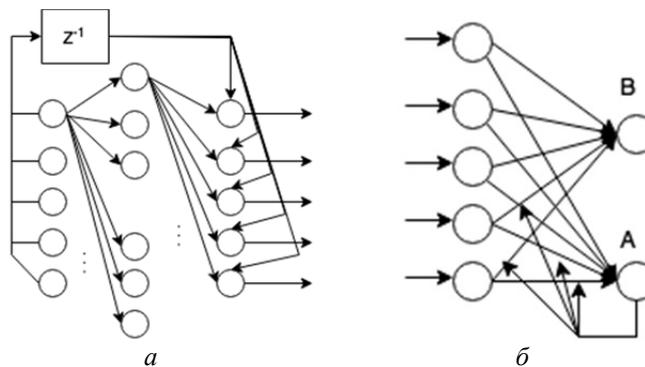


Рис. 2. Схема нейронной сети:

a – $NNmin$; *б* – часть системы, в которой выполняется перестановка значений массива

После этого веса меняются согласно следующему принципу: те веса, которые были единичными, становятся нулевыми, а веса от входного вектора к следующему нейрону (в данном случае В) становятся единичными. Таким образом, выбирается минимальный элемент входного вектора на каждый проход сети. Под частью замены минимального элемента выполняется подстановка. Далее выявленный минимальный элемент заменяется на максимальное число из диапазона входных значений.

Рассмотрим пример сортировки произвольно составленного вектора $v = [3 \ 2 \ 7 \ 9 \ 6]$, $v_i \in [0, 10]$ на примере итерации его сортировки согласно рассматриваемому алгоритму. На рис. 3 и 4 представлены поэтапные состояния системы, которая реализует рассматриваемый алгоритм.

Первая итерация рассматриваемого алгоритма сортировки начинается с определения минимального значения с помощью нейронной сети (рис. 3), выходом которой является вектор той же длины, что и входной вектор, в котором содержатся вероятности того, что соответствующий элемент является минимальным. Для рассматриваемого вектора v таким элементом является цифра 2 с вероятностью 0.8279. Вектор полученных вероятностей округляется до единицы или нуля и далее умножается на соответствующие элементы, которые поступают, проходя через звено задержки z^{-1} . Следующим этапом является перестановка определенного минимального элемента в выходной отсортированный вектор. Это достигается с помощью сети, структура

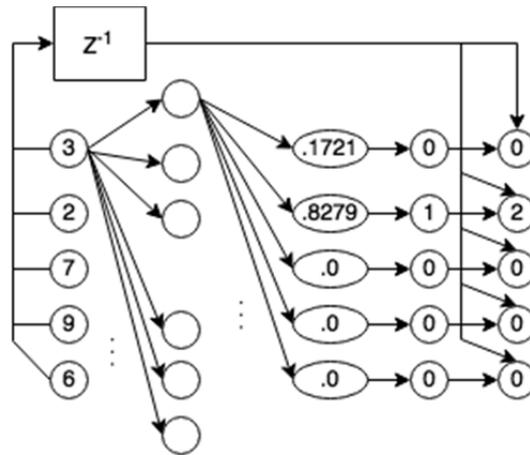


Рис. 3. Нейронная сеть выбора наименьшего элемента входного вектора для вектора $v = [3 \ 2 \ 7 \ 9 \ 6]$. Минимальный элемент «2» определен с вероятностью 0.8279

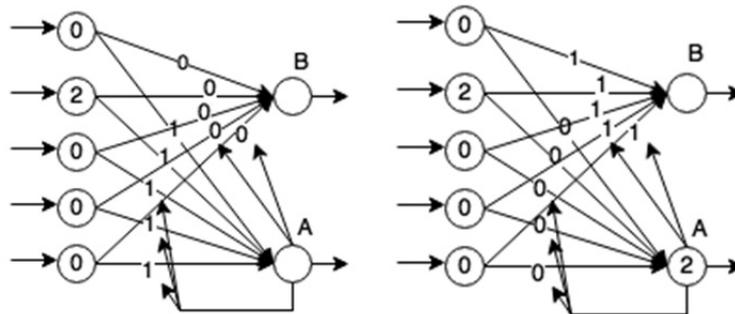


Рис. 4. Часть нейронной сети перестановки элементов массива: слева изображено начальное состояние, справа – состояние после перестановки

которой приведена на рис. 4, где представлена сеть, на вход которой поступает вектор, содержащий минимальный элемент. В данной сети веса получены не путем обучения сети, а заданы однозначно: на первоначальном этапе (рис. 4 слева) все веса, соединяющие элементы входного вектора и выходной элемент (A на рис. 4), имеют значение 1, все остальные – 0. Данная структура позволяет переместить значение из входного вектора в отсортированный вектор. После перемещения веса меняются таким образом, что все веса, идущие в следующий элемент отсортированного вектора, становятся равными единице, а остальные – нулю (рис. 4 справа). В результате первой итерации алгоритма первым элементом алгоритма было определено число 2, что соответствует действительности. На последующих итерациях алгоритма размерность входного вектора не изменится, но минимальный элемент должен быть заменен на максимальный элемент исследуемого диапазона. Таким образом, новое значение для вектора будет иметь вид $v = [3 \ 10 \ 7 \ 9 \ 6]$.

РЕЗУЛЬТАТЫ И ЗАКЛЮЧЕНИЕ

Предлагаемый алгоритм был реализован на языке Matlab и проверен на 10 000 сгенерированных примеров для входного вектора из 10 элементов для чисел из диапазона от нуля до 10. В результате проверки было установлено, что процент неправильно отсортированных входных векторов равен 2.48 % (248 раз). Данный процент можно объяснить тем, что нейронная сеть была натренирована на входных векторах со случайным распределением элементов, а при последовательной замене минимального элемента на максимальное значение из входного диапазона к концу вектора остаются только числа из верхнего диапазона значений.

Рассматриваемый алгоритм пригоден для сортировки вектора целых чисел, но может быть применен и для сортировки дробных чисел при соответствующей модификации сети, в которой числа будут анализироваться поразрядно нейронной сетью.

СПИСОК ЛИТЕРАТУРЫ

1. Introduction to algorithms / T. Cormen, C. Leiserson, R. Rivest, C. Stein. – 3rd ed. – Cambridge, MA: MIT Press, 2009. – 1328 p.
2. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы: пер. с англ. – М.: Вильямс, 2016. – 400 с.
3. Haykin S. Neural networks and learning machines. – 3rd ed. – New York: Pearson Education, 2009. – 938 p.
4. Bishop C. Pattern recognition and machine learning. – New York: Springer, 2007. – 729 p.
5. Richert W., Coelho L.P. Building machine learning systems with Python. – 2nd ed. – Birmingham, UK: Packt, 2015. – 290 p.
6. Mohri M., Rostamizadeh A., Talwalkar A. Foundations of machine learning. – Cambridge, MA: MIT Press, 2012. – 427 p.
7. Hastie T., Tibshirani R., Friedman J. The elements of statistical learning. – 2nd ed. – New York: Springer, 2009. – 764 p.
8. Ackley D.H., Hinton G.E., Sejnowski T.J. A learning algorithm for Boltzmann machines // Cognitive Science. – 1985. – Vol. 9. – P. 147–169.
9. Bahdanau D., Cho K., Bengio Y. Neural machine translation by jointly learning to align and translate // 3rd International Conference on Learning Representations ICLR'2015: conference paper. – San Diego, 2015. – P. 1–15.
10. Behnke S. Learning iterative image reconstruction in the neural abstraction pyramid // International Journal Computational Intelligence and Applications. – 2015. – Vol. 1 (4). – P. 427–438.
11. Gated feedback recurrent neural networks / J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio // Proceedings of the 32nd International Conference on Machine Learning ICML'15. – Lille, France, 2015. – P. 2067–2075.
12. Deep generative image models using a Laplacian pyramid of adversarial networks / E. Denton, S. Chintala, A. Szlam, R. Fergus // NIPS Proceedings. – Montreal, Canada, 2015. – P. 1–9.
13. Frey B.J. Graphical models for machine learning and digital communication. – Cambridge, MA: MIT Press, 1998. – 195 p.
14. Goodfellow I.J., Courville A., Bengio Y. Scaling up spike-and-slab models for unsupervised feature learning // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2013. – Vol. 35 (8). – P. 1902–1914.
15. Block H.D. The Perceptron: a model for brain functioning // Reviews of Modern Physics. – 1962. – Vol. 34 (1). – P. 123–135.

Воевода Александр Александрович, доктор технических наук, профессор Новосибирского государственного технического университета. Основные направления научных исследований: теория автоматического управления, сети Петри. Имеет более 200 публикаций. E-mail: voevoda@ucit.ru

Полубинский Владимир Львович, кандидат технических наук, доцент Новосибирского государственного технического университета. Основное направление научных исследований – биотехнические измерительные системы. Имеет более 100 публикаций. E-mail: polubinsky@tiger.cs.nstu.ru

Романников Дмитрий Олегович, кандидат технических наук, доцент Новосибирского государственного технического университета. Основное направление научных исследований – машинное обучение. Имеет более 40 публикаций. E-mail: dmitry.romannikov@gmail.com

Sorting the array of integers using a neural network *

*A.A. VOEVODA*¹, *V.L. POLUBINSKY*², *D.O. ROMANNIKOV*³

¹ *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, D.Sc. (Eng.), Professor. E-mail: voevoda@ucit.ru*

² *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, PhD (Eng.), Associate Professor. E-mail: polubinsky@tiger.cs.nstu.ru*

³ *Novosibirsk State Technical University, 20 K. Marx Prospekt, Novosibirsk, 630073, Russian Federation, PhD (Eng.), Associate Professor. E-mail: dmitry.romannikov@gmail.com*

Currently, quite a number of various algorithms and data structures are available on which all modern software systems are built. On the other hand, these algorithms are poorly suited for solving problems of pattern recognition, classification, and regression for which specific methods exist and/or solutions based on neural networks are applied. Neural networks are per se universal approximators, i.e. they can reproduce a predetermined curve with a high degree of accuracy. It also follows that unlike classical algorithms, systems based on neural networks will always have a certain percentage of errors due to their probabilistic nature. In this paper, an algorithm of sorting an array of integers using a neural network is proposed. This algorithm is a symbiosis of a neural network and classical algorithms. The paper presents a block diagram of the algorithm which comprises several parts such as a neural network to search for a minimal element of the input vector and a part of the system which places the found minimal element in the sorted part of the array. The last part of the system is implemented in the form of classical algorithms, but can be replaced by a degenerate version of a neural network in which weights are the replacements of array elements. The verification of the proposed algorithm was carried out. As a result, 248 out of 10,000 measurements were wrong, which corresponds to a 2.48% error. Some possible causes of this error rate are also discussed in the paper.

Keywords: neural network, algorithms, sorting, classification, Matlab, probabilistic systems, learning, probabilistic algorithms, perceptron

DOI: 10.17212/1814-1196-2016-2-151-157

REFERENCES

1. Cormen T., Leiserson C., Rivest R., Stein C. *Introduction to algorithms*. 3rd ed. Cambridge, MA, MIT Press, 2009. 1328 p.
2. Aho A.V., Hopcroft J.E., Ullman J.D. *Структуры данных и алгоритмы* [Data structures and algorithms]. Translated from English. Moscow, Williams Publ., 2016. 400 p.

* Received 18 April 2016.

3. Haykin S. *Neural networks and learning machines*. 3rd ed. New York, Pearson Education, 2009. 938 p.
4. Bishop C. *Pattern recognition and machine learning*. New York, Springer, 2007. 729 p.
5. Richert W., Coelho L.P. *Building machine learning systems with Python*. 2nd ed. Birmingham, UK, Packt, 2015. 290 p.
6. Mohri M., Rostamizadeh A., Talwalkar A. *Foundations of machine learning*. Cambridge, MA, MIT Press, 2012. 427 p.
7. Hastie T., Tibshirani R., Friedman J. *The elements of statistical learning*. 2nd ed. New York, Springer, 2009. 764 p.
8. Ackley D.H., Hinton G.E., Sejnowski T.J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985, vol. 9, pp. 147–169.
9. Bahdanau D., Cho K., Bengio Y. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations ICLR'2015*: conference paper. – San Diego, 2015. – P. 1–15.
10. Behnke S. *Learning iterative image reconstruction in the neural abstraction pyramid*. *International Journal Computational Intelligence and Applications*, 2015, vol. 1 (4), pp. 427–438.
11. Chung J., Gülçehre Ç., Cho K., Bengio Y. Gated feedback recurrent neural networks. *Proceedings of the 32nd International Conference on Machine Learning ICML'15*, Lille, France, 2015, pp. 2067–2075.
12. Denton E., Chintala S., Szlam A., Fergus R. Deep generative image models using a Laplacian pyramid of adversarial networks. *NIPS Proceedings*, Montreal, Canada, 2015, pp. 1–9.
13. Frey B.J. *Graphical models for machine learning and digital communication*. Cambridge, MA, MIT Press, 1998. 195 p.
14. Goodfellow I.J., Courville A., Bengio Y. Scaling up spike-and-slab models for unsupervised feature learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, vol. 35 (8), pp. 1902–1914.
15. Block H.D. The Perceptron: a model for brain functioning. *Reviews of Modern Physics*, 1962, vol. 34 (1), pp. 123–135.