СБОРНИК **НАУЧНЫХ ТРУДОВ НГТУ. – 2017. – № 4 (90).** – 49–64

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.852:004.023:519.24:006.72

ПРИМЕР СЛОЖЕНИЯ ЧИСЕЛ В БИНАРНОМ ПРЕДСТАВЛЕНИИ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННОЙ СЕТИ*

Н.О. ИВАНОВ[1], Д.О. ТИНГАЙКИН[2], В.В. КОМИССАРОВ[3]

[1] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматики. E-mail: stenrulf@gmail.com*
[2] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматики. E-mail: teenguyking@gmail.com*
[3] *630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, студент кафедры автоматики. E-mail:sodiz@yandex.ru*

В статье рассматривается пример решения задачи сложения чисел в двоичном представлении при помощи нейросети. Рассматриваемая задача является задачей классификации. Она решается при помощи многослойного персептрона. Задача классификации имеет множество объектов (ситуаций), разделенных некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества. Искусственная нейронная сеть – это математическая модель в совокупности с программной или машинной реализацией, построенная подобно биологическим нейронным сетям и сетям нервных клеток биологических живых организмов. Нейронные сети не могут быть «запрограммированы» в общепринятом понимании этого слова, но они способны обучаться. Возможность обучения – важнейшее преимущество нейронных сетей перед классическими алгоритмами. Обучение нейросети – это процесс, заключающийся в поиске необходимых коэффициентов связей между нейронами. При обучении нейронной сети используются методы оптимизации. В данном случае целью оптимизации будет минимизация ошибки. Метод наименьших квадратов – математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных. МНК является одним из базовых методов регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным.

*Статья получена 18 сентября 2017 г.

Ключевые слова: метод обратного распространения ошибки, градиентный спуск, нейронная сеть, многослойный персептрон, сумматор, метод наименьших квадратов, топология нейронной сети, машинное обучение

DOI: 10.17212/2307-6879-2017-4-49-64

ВВЕДЕНИЕ

Все задачи, которые способны решать нейронные сети, связаны с обучением. Основные области применения нейронных сетей: прогнозирование, принятие решений, распознавание образов, анализ данных.

Пару десятков лет назад скорость работы нейронных сетей была слишком медленной, для того чтобы они могли использоваться в производстве, и поэтому подобные системы, как правило, использовались в проектах, связанных с компьютерным зрением, а в остальных областях применялись менее затратные алгоритмы машинного обучения.

Сложным и длительным процессом разработки нейронной сети является обучение. Для того чтобы нейронная сеть могла корректно решать поставленные задачи, необходимо провести подготовку в виде анализа выборки большого количества входных данных с заранее известными правильными или неправильными ответами. Появление разнообразных технологий ускоренного обучения дало толчок к распространению нейронных сетей.

Для обучения нами будет использоваться алгоритм нахождения локального экстремума функции с помощью движения вдоль градиента, относящийся к локальной оптимизации. Это метод градиентного спуска. Метод градиентного спуска наиболее простой в реализации из всех методов локальной оптимизации. Имеет довольно слабые условия сходимости, но зато обладает сложностью сходимости $O(n)$, что очень важно для вычислительных машин. На методе градиентного спуска базируется метод обратного распространения ошибки. Этот метод обучения многослойного персептрона предполагает два прохода по всем слоям сети: прямой и обратный. В случае прямого прохода вектор входных данных поступает на входной слой нейронной сети, а затем передается по сети по каждому слою. В итоге создается набор выходных импульсов, являющийся реакцией на входные импульсы. Прямой проход не затрагивает синаптические веса сети, т. е. они являются константными. А обратный проход, напротив, изменяет синаптические веса согласно правилу коррекции ошибок: полученные результаты сети вычитаются из ожидаемых, в результате генерируется ошибка. Затем ошибка распространяется по сети в обратном направлении синаптических связей. Синаптические веса подбираются с целью максимального приближения выходного сигнала сети к ожидаемому.

Основной проблемой реализации функции на основе нейронов является подбор такой нелинейности, которая подходит под заданный критерий точности (например, при заявленной ошибке в 20 %; далее будет использоваться абсолютное значение 0,2).

1. ОПИСАНИЕ

Нейронные сети используются в решении различных задач [1–4, 12, 13]. В данной работе будет решаться задача классификации.

В процессе решения задач классификации с помощью многослойного персептрона возникает ряд классических проблем [5]. Это происходит из-за неоднозначности выбора параметров системы. Для решения этих проблем существуют разные подходы [4].

Далее мы будем рассматривать сложение чисел с помощью многослойного персептрона, выполняющего функцию сумматора. Для этого мы будем работать с числами в бинарном представлении.

Рассмотрим сумматоры для одно-, двух- и трехбитовых чисел и попробуем вывести закономерность в построении топологии для возможности расширения системы.

Градиентный спуск

Основная идея метода градиентного спуска [9–11] заключается в том, чтобы осуществлять оптимизацию в направлении наискорейшего спуска. Направление задается антиградиентом $-\nabla f$.

Алгоритм принимает на вход функцию от вектора действительных аргументов $f : R^n \to R$. В результате работы алгоритма вычисляется точка оптимума.

На каждом шаге алгоритма вычисляется значение, такое что

$$x^{k+1} = x^k - \lambda^k \nabla f(x^k),$$

λ^k выбирается:

- постоянной, в этом случае метод может расходиться;
- дробным шагом, т. е. длина шага в процессе спуска делится на некое число;
- наискорейшим спуском: $\lambda^{[k]} = \arg\min(f(x^{[k]} - \lambda \nabla f(x^{[x]})_\lambda$.

Критерий останова выбирается одним из следующих способов:

1) $\|x^{[k+1]} - x^{[k]}\| \le \varepsilon$;

2) $\|f(x^{[k+1]} - f(x^{[k]})\| \le \varepsilon$.

Здесь $x^{[k]} \in R^n$ – значение, полученное после k-го шага оптимизации; ε – наперед заданное положительное число.

Метод обратного распространения ошибки

Суть этого метода состоит в распространении сигналов ошибки [7, 8] от выходов сети к ее входам в направлении, обратном прямому распространению сигналов в обычном режиме работы.

Алгоритм

1. Инициализировать $\{w_{ij}\}_{i,j}$ малыми случайными значениями.
2. Для всех d от 1 до m:
   а) подать $\left\{x_i^d\right\}$ на вход сети и подсчитать выходы O_i каждого узла;
   б) для всех k выполнить $k \in Outputs$;
   в) выполнить вычисления $\delta_k = o_k(1-o_k)(t_k - o_k)$.
3. Повторить N раз действия п. 2.
4. Для каждого уровня, начиная с предпоследнего:
   а) для каждого узла j уровня вычислить

$$\delta_k = o_k(1-o_k)\sum \delta_k w_{jk}, k \in Children(j);$$

   б) для каждого ребра сети $\{i, j\}$ вычислить

$$w_{i,j} = w_{i,j} + \eta\delta_j x_{i,j} .$$

5. Выдать значения w_{ij} .

Активационная функция

В качестве активационной функции используется сигмоидальная (рис. 1):

$$f(x) = \frac{1}{1+e^{-\alpha x}}, \quad \alpha = 1 .$$

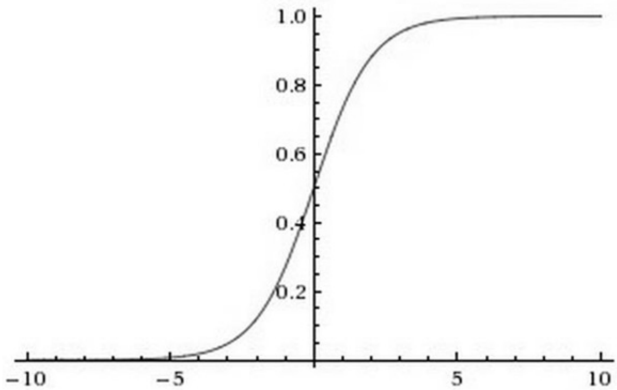

Рис. 1. Логистическая функция (сигмоида)

2. ПОСТАНОВКА ЗАДАЧИ

- На основе многослойного персептрона Розенблатта, использующего для обучения метод обратного распространения ошибки, реализовать сумматор трехбитовых чисел в бинарном представлении.
- Рассмотреть различные варианты решения данной задачи.

3. ОСНОВНАЯ ЧАСТЬ

Нейроны обучаемой сети используют сигмоидальную функцию активации. Обучение выполняется при помощи самописной программы на языке C++. Начальные условия задаются равномерным распределением из диапазона {–0.1, 0.1}, непосредственно само обучение выполняется за 5000 эпох с использованием метода обратного распространения ошибки.

Кроме того, рассматривался пример сложения двух чисел, представленных в бинарном виде тремя разрядами.

Далее в статье будем руководствоваться обозначениями:

$a_0 a_1 \ldots a_{n-1} a_n$, где a_i – i-й бит первого числа;

$b_0 b_1 \ldots b_{n-1} b_n$, где b_i – i-й бит второго числа.

Однобитный сумматор

Набор возможных комбинаций при сложении двух битовых чисел выглядит следующим образом:

Операция суммирования может быть представлена через элементарные логические операции.

Как видно из таблицы, первый бит можно получить из входных значений посредством операции «исключающее ИЛИ»:

$$XOR \Leftrightarrow (a_0 \wedge \neg b_0) \vee (\neg a_0 \wedge b_0) .$$

Второй бит можно получить из входных значений посредством операции «логическое И».

Таблица истинности для однобитного сложения

<table>
<tr><th rowspan="3">Первое слагаемое</th><th rowspan="3">Второе слагаемое</th><th colspan="3">Сумма</th></tr>
<tr><th rowspan="2">Десятичное представление</th><th colspan="2">Двоичное представление</th></tr>
<tr><th>Первый бит</th><th>Второй бит</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Операция «И» может быть выполнена на однослойном персептроне с двумя входами и одним выходом [1–3], однако операция «исключающее ИЛИ» является классической задачей, которая не может быть решена на однослойном персептроне. Впрочем, она может быть решена на двухслойном персептроне [4]. Следовательно, нам потребуется сеть с одним скрытым слоем, содержащим два нейрона.

Расчет коэффициентов был произведен с использованием метода обратного распространения ошибки программой, написанной на языке С++.

Решение имеет неоднозначность, так как коэффициенты для решения этой задачи могут быть подобраны различным образом [3].

Двухбитное сложение

По аналогии с однобитовым сложением мы вывели формулы для каждого из битов при сложении трехбитовых чисел.

Так же как и в примере с однобитовым сумматором (рис. 2), попытаемся спрогнозировать топологию двухбитового сумматора исходя из логических формул для каждого бита.

Первый бит: $(a_0 \wedge a_1 \wedge b_1) \vee (a_0 \wedge b_0) \vee (a_1 \wedge b_0 \wedge b_1)$.

Второй бит: $(a_0 \wedge a_1 \wedge b_0 \wedge b_1) \vee (a_0 \wedge \neg a_1 \wedge \neg b_0) \vee$
$\vee (a_0 \wedge \neg b_0 \wedge \neg b_1) \vee (\neg a_0 \wedge a_1 \wedge \neg b_0 \wedge b_1) \vee$
$\vee (\neg a_0 \wedge \neg b_0 \wedge b_0) \vee (\neg a_0 \wedge b_0 \wedge \neg b_1)$.

Третий бит: $(a_1 \wedge \neg b_1) \vee (\neg a_1 \wedge b_1)$.

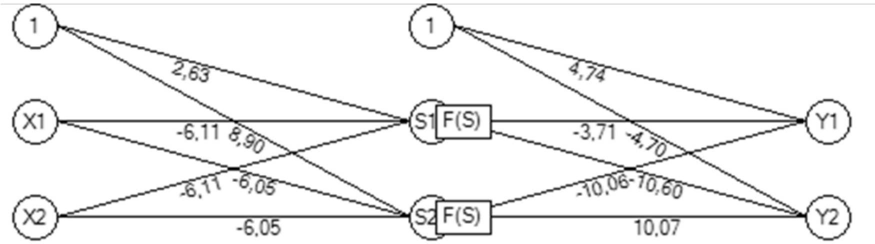

Рис. 2. Однобитный сумматор

По логическим формулам первого бита спроектируем часть топологии (рис. 3).

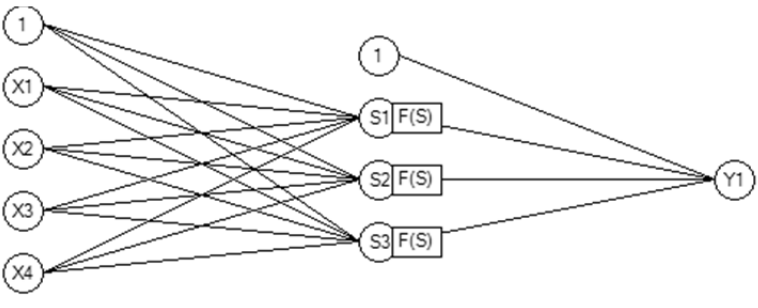

Рис. 3. Топология для первого бита

Часть сети для первого бита имеет три нейрона в скрытом слое. Логическая формула для первого бита самая короткая. Очевидно, что сеть, спроектированная исходя из логических формул, будет избыточной.

Поэтому эвристическим путем было выяснено, что наиболее стабильной сетью с разумным количеством нейронов является сеть с четырьмя нейронами в скрытом слое (рис. 4).

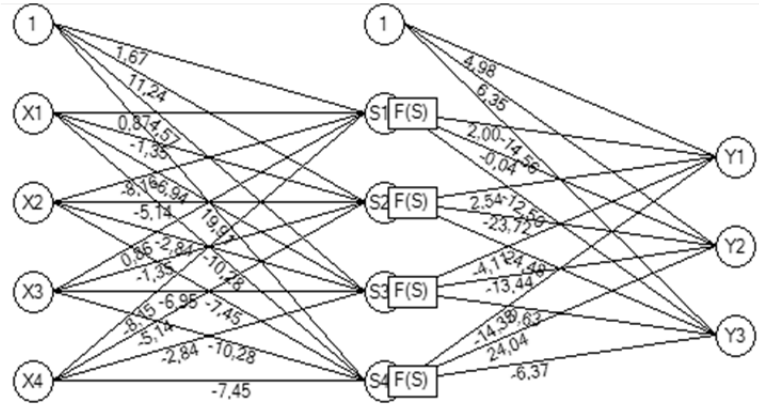

Рис. 4. Двухбитный сумматор

Трехбитное сложение

Для трехбитного сумматора вывести логические формулы достаточно затруднительно, поэтому мы решили выполнить тестирование различных топологий.

Мы провели 100 экспериментов для каждой из рассмотренных топологий и вывели функцию ошибки по методу наименьших квадратов, а также разницы между ожидаемым и полученным ответом. Сети обучалось 5000 эпох.

$$error_i = \frac{\sum_{k=0}^{exper} \sum_{j=0}^{iter} (derror_{i,j,k})^2}{2 * exper},$$

где *derrors* – вектор ошибок на выходном слое, *exper* – число экспериментов, *iter* – число итераций на каждой эпохе, $error_i$ – ошибка в i-й эпохе (средняя по экспериментам).

Как можно заметить по графику (рис. 6), после 1000 *epoch* скорость уменьшения ошибки резко снижается, и на 5000 *epoch* нейросеть (рис. 5) не сходится, имея ошибку ~ 0,2.

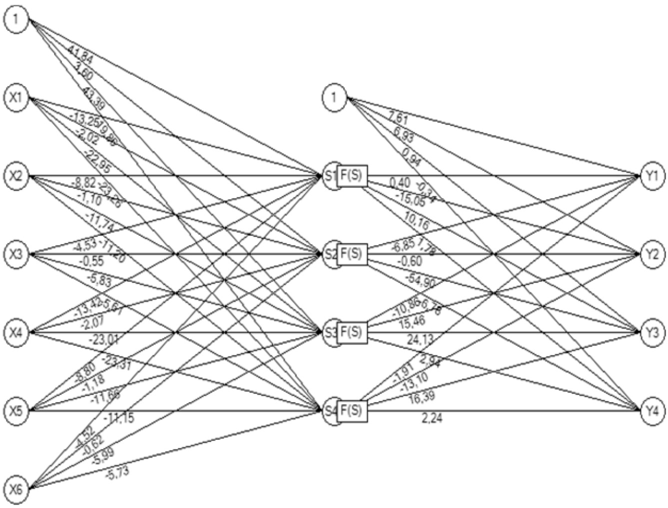

Рис. 5. Трехбитный сумматор. Четыре нейрона в скрытом слое

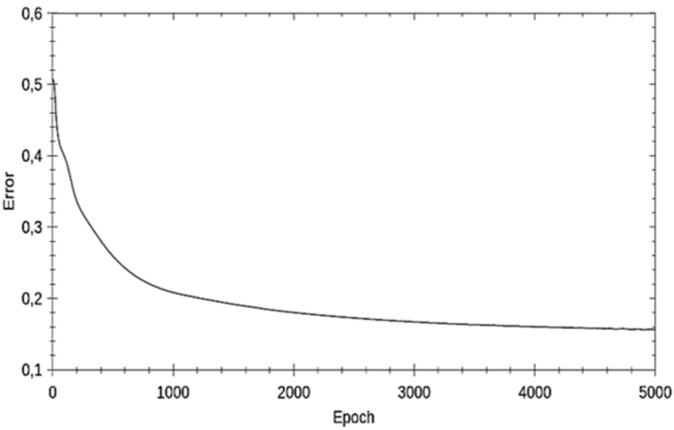

Рис. 6. График ошибки, полученной по МНК, для нейросети с четырьмя нейронами в скрытом слое

По графику (рис. 8) видно, что сеть (рис. 7) с данной топологией имеет меньшую ошибку, однако имеет проблемы со сходимостью, как и предыдущая (рис. 5).

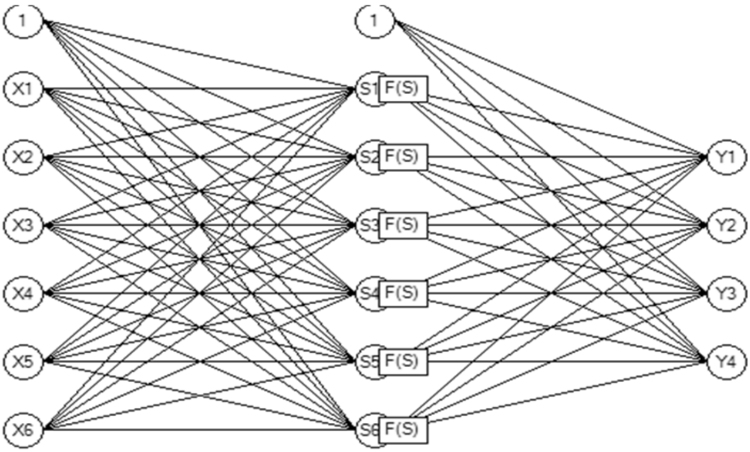

Рис. 7. Трехбитный сумматор. Шесть нейронов в скрытом слое

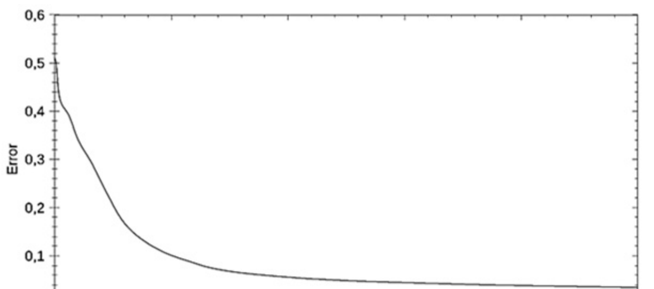

Рис. 8. График ошибки, полученной по МНК, для нейросети с шестью нейронами в скрытом слое

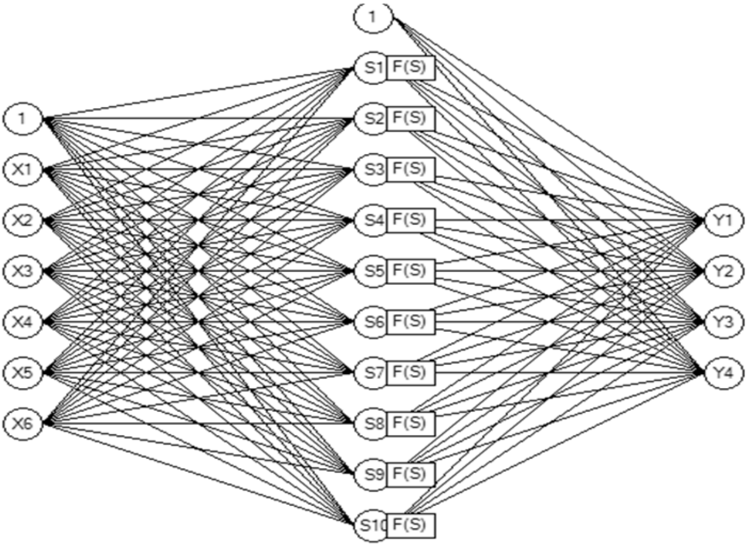

Рис. 9. Трехбитный сумматор. Десять нейронов в скрытом слое

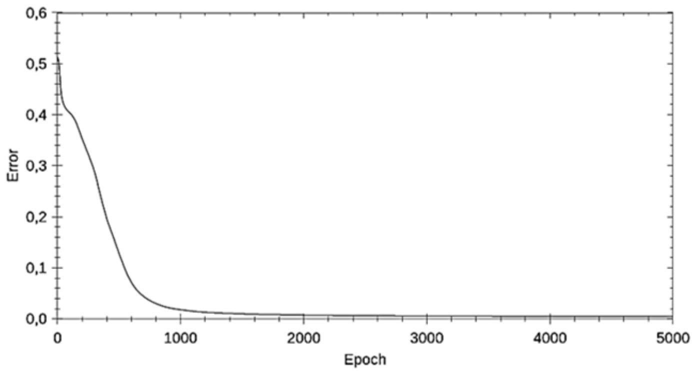

Рис. 10. График ошибки, полученной по МНК, для нейросети
с десятью нейронами в скрытом слое.

По графику (рис. 10) видно, что сеть (рис. 9) с данной топологией имеет сходимость на 1000 *epoch*.

Аналогичные эксперименты были проведены на сетях, содержащих от 1 до 12 нейронов скрытом слое. Было получено значение доли правильных ответов в зависимости от числа нейронов в скрытом слое. Результаты экспериментов представлены на графике (рис. 11).

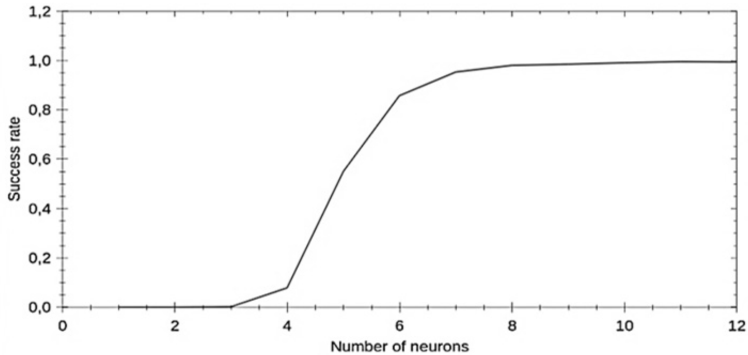

Рис. 11. Зависимость успеха обучения от числа в скрытом слое

Из графика (рис. 11) видно, что для реализации трехбитного сумматора на классическом многослойном персептроне необходимо использовать количество нейронов в скрытом слое ≥ 8 .

БЛАГОДАРНОСТЬ

Авторы выражают искреннюю благодарность профессору кафедры автоматики А.А. Воеводе за помощь при выполнении работ, а также полезное обсуждение полученных результатов.

ЗАКЛЮЧЕНИЕ

Было показано, что прогнозирование сети исходя из логических формул для каждого бита результата сложения является нетривиальной задачей. Если подходить непосредственно к реализации, то количество нейронов при проектировании сети будет значительно больше, чем число нейронов, полученных при классическом способе обучения с помощью метода обратного распространения ошибки и метода градиентного спуска.

Для трехбитного сумматора были рассмотрены топологии многослойного персептрона с одним скрытым слоем и различным числом нейронов в нем. По результатам экспериментов сделан вывод о том, что минимальной надежной топологией является топология с восемью нейронами в скрытом слое.

СПИСОК ЛИТЕРАТУРЫ

1. *Розенблатт Ф.* Принципы нейродинамики: перцептроны и теория механизмов мозга. – М.: Мир, 1965. – 480 с.

2. *Минский М., Пейперт С.* Персептроны. – М.: Мир, 1971. – 261 с.

3. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning internal representations by error propagation // Parallel Distributed Processing. – Cambridge, MA: MIT Press, 1986. – Vol. 1. – P. 318–362.

4. *Воевода А.А., Романников Д.О.* Синтез нейронной сети для решения логико-арифметических задач // Труды СПИИРАН. – 2017. – Вып. 5 (54). – 205–223.

5. *Хайкин С.* Нейронные сети. Полный курс. – 2-е изд. – М.: Вильямс, 2006. – 1104 с.

6. *Горбань А.Н., Россиев Д.А.* Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996. – 276 с. – ISBN 5-02-031196-0.

7. *Wasserman P.D.* Experiments in translating Chinese characters using backpropagation // Proceedings of the Thirty-Third IEEE Computer Society International Conference. – Washington, D.C.: Computer Society Press of the IEEE, 1988. – P. 399–402.

8. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация. – М.: Мир, 1985. – 509 с.

9. *Максимов Ю.А., Филлиповская Е.А.* Алгоритмы решения задач нелинейного программирования. – М.: МИФИ, 1982. – 52 с.

10. *Максимов Ю.А.* Алгоритмы линейного и дискретного программирования. – М.: МИФИ, 1980. – 72 с.

11. *Городецкий С.Ю., Гришагин В.А.* Нелинейное программирование и многоэкстремальная оптимизация. – Н. Новгород: Изд-во Нижегор. ун-та, 2007. – С. 357–363.

12. *Воевода А.А., Романников Д.О.* Асинхронный алгоритм сортировки массива чисел с использованием ингибиторных сетей Петри // Труды СПИИРАН. – 2016. – № 5 (28). – С. 198–213.

13. *Воевода А.А., Полубинский В.Л., Романников Д.О.* Сортировка массива целых чисел с использованием нейронной сети // Научный вестник НГТУ. – 2016. – № 2 (63). – С. 151–157.

Иванов Никита Олегович, студент кафедры автоматики Новосибирского государственного технического университета по направлению «Информатика и вычислительная техника». E-mail: stenrulf@gmail.com

Тингайкин Денис Олегович, студент кафедры автоматики Новосибирского государственного технического университета по направлению «Информатика и вычислительная техника». E-mail: teenguyking@gmail.com

Комиссаров Валерий Владимирович, студент кафедры автоматики Новосибирского государственного технического университета по направлению «Информатика и вычислительная техника». E-mail: sodiz@yandex.ru

Example of adding numbers in a binary representation using neural networks*

N.O. Ivanov[1], D.O. Tingajkin[2], V.V. Komissarov[3]

[1] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, student of the automation department. E-mail: stenrulf@gmail.com*
[2] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, student of the automation department. E-mail: teenguyking@gmail.com*
[3] *Novosibirsk State Technical University, 20 Karl Marx Avenue, Novosibirsk, 630073, Russian Federation, student of the automation department. E-mail: sodiz@yandex.ru*

The article considers an example of solving the problem of adding numbers in a binary representation using a neural network. The problem under consideration is a classification problem. It is solved by means of a multilayer perceptron. The classification problem has many objects (situations), divided in some way into classes. A finite set of objects is defined for which it is known which classes they belong to. This set is called a training sample. The class affiliation of the remaining objects is not known. It is required to construct an algorithm capable of classifying an arbitrary object from the original set. An artificial neural network is a mathematical model in conjunction with a software or machine implementation, built like biological neural networks, nerve cell networks of biological living organisms.Neural networks can not be "programmed" in the conventional sense of the word, but they are able to train. The possibility of training the most important advantage of neural networks over the "classical" algorithms. Training a neural network is a process that involves finding the necessary coefficients of connections between neurons.When training a neural network, optimization methods are used. In this case, the optimization goal will be to minimize the error. The method of least squares is a mathematical method used to solve various problems, based on minimizing the sum of squares of deviations of certain functions from the unknown variables. OLS is one of the basic methods of regression analysis for estimating unknown parameters of regression models from sample data.

Keywords: backpropagation, gradient descent, optimization methods, multilayer perceptron, adder, the method of least squares, neural network topology, machine training

DOI: 10.17212/2307-6879-2017-4-49-64

* *Received 18 September 2017.*

REFERENCES

1. Rosenblatt F. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Washington, Spartan Books, 1962 (Russ. ed.: Rozenblatt F. *Printsipy neirodinamiki: pertseptrony i teoriya mekhanizmov mozga*. Moscow, Mir Publ., 1965. 480 p.).

2. Minsky M., Papert S. *Perceptrons: an introduction to computational geometry*. Cambridge, MA, London, MIT Press, 1969 (Russ. ed.: Minskii M., Peipert S. *Perseptrony*. Moscow, Mir Publ., 1971. 261 p.).

3. Rumelhart D.E., Hinton G.E., Williams R.J. Learning internal representations by error propagation. *Parallel Distributed Processing*. Cambridge, MA, MIT Press, 1986, vol. 1, pp. 318–362.

4. Voevoda A.A., Romannikov D.O. Sintez neironnoi seti dlya resheniya logiko-arifmeticheskikh zadach [Synthesis of neural network for solving logical-arithmetic problems]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2017, iss. 5 (54), pp. 205–223.

5. Haykin S. *Neural networks*. Upple Saddle River, Prentice Hall, 1999 (Russ. ed.: Khaikin S. *Neironnye seti. Polnyi kurs*. 2nd ed. Moscow, Williams Publ., 2006. 1104 p.).

6. Gorban' A.N., Rossiev D.A. *Neironnye seti na personal'nom komp'yutere* [Neural networks on a personal computer]. Novosibirsk, Nauka Publ., 1996. 276 p.– ISBN 5-02-031196-0.

7. Wasserman P.D. Experiments in translating Chinese characters using backpropagation. *Proceedings of the Thirty-Third IEEE Computer Society International Conference*. Washington, D.C., Computer Society Press of the IEEE, 1988, pp. 399–402.

8. Gill P.E., Murray W., Wright M.H. *Practical optimization*. London, New York, Academic Press, 1981 (Russ. ed.: Gill F., Myurrei U., Rait M. *Prakticheskaya optimizatsiya*. Moscow, Mir Publ., 1985. 509 p.).

9. Maksimov Yu.A., Fillipovskaya E.A. *Algoritmy resheniya zadach nelineinogo programmirovaniya* [Algorithms for solving nonlinear programming problems]. Moscow, MEPI Publ., 1982. 52 p.

10. Maksimov Yu.A. *Algoritmy lineinogo i diskretnogo programmirovaniya* [Algorithms of linear and discrete programming]. Moscow, MEPI Publ., 1980. 72 p.

11. Gorodetskii S.Yu., Grishagin V.A. *Nelineinoe programmirovanie i mnogoekstremal'naya optimizatsiya* [Nonlinear programming and multi-extremal optimization]. Nizhnii Novgorod, Lobachevsky State University of Nizhni Novgorod Publ., 2007, pp. 357–363.

12. Voevoda A.A., Romannikov D.O. Asinkhronnyi algoritm sortirovki massiva chisel s ispol'zovaniem ingibitornykh setei Petri [Asynchronous sorting algorithm for array of numbers with the use of inhibitory Petri nets]. *Trudy SPIIRAN – SPIIRAS Proceedings*, 2014, iss. 3 (34), pp. 218–232.

13. Voevoda A.A., Polubinskii V.L., Romannikov D.O. Sortirovka massiva tselykh chisel s ispol'zovaniem neironnoi seti [Sorting the array of integers using a neural network]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Science bulletin of the Novosibirsk state technical university*, 2016, no. 2 (63), pp. 151–157.

Для цитирования:

Иванов Н.О., Тингайкин Д.О., Комиссаров В.В. Пример сложения чисел в бинарном представлении с использованием нейронной сети // Сборник научных трудов НГТУ. – 2017. – № 4 (90). – С. 49–64.

For citation:

Ivanov N.O., Tingajkin D.O., Komissarov V.V. Primer slozheniya chisel v binarnom predstavlenii s ispol'zovaniem neironnoi seti [Example of adding numbers in a binary representation using neural networks]. *Sbornik nauchnykh trudov Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta – Transaction of scientific papers of the Novosibirsk state technical university*, 2017, no. 4 (90), pp. 49–64.