

Программная реализация алгоритма обучения нейронной сети

О.А. Кабышев, М.П. Маслаков, А.М. Кабышев

Северо-Кавказский горно-металлургический институт (государственный технологический университет)

Аннотация: В статье рассматриваются вопросы, связанные с проектированием устройств управления технологическим оборудованием на основе искусственных нейронных сетей. Разработан алгоритм процесса обучения нейронных сетей, основанный на методе Уидроу-Хоффа, предназначенный для автоматизированного проектирования искусственных нейронных сетей, функционирующих в среде микропроцессорной системы или персонального компьютера. Выполнена программная реализация алгоритма обучения трехслойной нейронной сети с использованием языка программирования C# на платформе Microsoft.NET. В программе используется алгоритм обучения нейронной сети с «учителем». В статье показан интерфейс взаимодействия разработанной программы с «учителем» (проектировщиком нейронных сетей). Приведен пример разработки искусственной нейронной сети, выполняющей функцию сумматора двоичных чисел. Полученные в статье результаты могут быть в дальнейшем использованы при разработке компьютерной системы автоматизированного проектирования искусственных нейронных сетей, а также могут найти применение при разработке программного обеспечения для микропроцессорных систем управления технологическим оборудованием.

Ключевые слова: Искусственная нейронная сеть, обучение, алгоритм, язык программирования, смещение нейрона, сумматор, входные сигналы, выходные сигналы, синаптические связи, нейрон.

Искусственные нейронные сети находят применение для обработки сигналов в системах управления и диагностики режимов работы технологического оборудования [1]. Применение в системах управления комбинационных логических автоматов, выполненных на основе искусственных нейронных сетей, позволяет разрабатывать технологическое оборудование, адаптирующееся к конкретным условиям технологического процесса.

Функционирование нейронных сетей невозможно без обучения нейронов, входящих в их состав. Обученная искусственная нейронная сеть должна находить зависимости между входными и выходными данными и выполнять их обобщение. В процессе обучения необходимо найти баланс между способностью сети выдавать правильные результаты в ответ на обучающую выборку, и способностью правильного ответа на данные,

которые отсутствовали в обучающей выборке. При этом достаточно трудоемким процессом, требующим большого объема вычислений, является поиск величин весовых коэффициентов синаптических связей и смещений нейронов сети [2, 3]. Поэтому при разработке и реализации нейронных сетей целесообразно использовать возможности современных средств микропроцессорной и вычислительной техники с соответствующим программным обеспечением, реализующим определенный алгоритм обучения. Алгоритмам обучения нейронных сетей посвящено большое количество разработок и публикаций [4-6].

На рис.1 показана схема процесса обучения нейронной сети, в которой использован алгоритм обратного распространения ошибки на основе метода Уидроу-Хоффа [3-5]. Программная реализация этой схемы в вычислительной среде микропроцессорной системы или персонального компьютера позволяет автоматизировать процесс разработки искусственных нейронных сетей.

В процессе обучения, согласно алгоритму, на вход нейронной сети подаются сигналы из массива обучающих выборок "x", а также сети предоставляется информация, в виде массива эталонных значений "y0", о том, как необходимо реагировать на каждую обучающую выборку. Эта информация задается разработчиком нейронной сети в блоке №1 алгоритма. В блоке №2 формируются начальные значения весов синаптических связей "w" и смещений (порогов) "b" нейронов сети. В блоке №3 разработчик задает скорость обучения "α" (выбирается число в диапазоне: $0 < \alpha < 1$) и количество итераций "n", эти величины влияют на качество обучения [7]. Вычисление текущих состояний "s" и выходных сигналов "y" нейронов, входящих в состав сети, осуществляется в блоке №4. Текущее состояние каждого нейрона сети вычисляется по формуле: $s = \sum_{i=1}^m x_i * w_i + b$, где: x_i - входной i -

i -й сигнал; w_i - вес синаптической связи по которой передается i -й сигнал; b - смещение нейрона; m - количество входных сигналов. Величины сигналов " y " на выходах нейронов зависят от величин текущего состояния " s " и от того, какая функция активации применяется в нейронах [2, 4]. Вычисление ошибок и корректировка весовых коэффициентов выполняется в блоке №5.

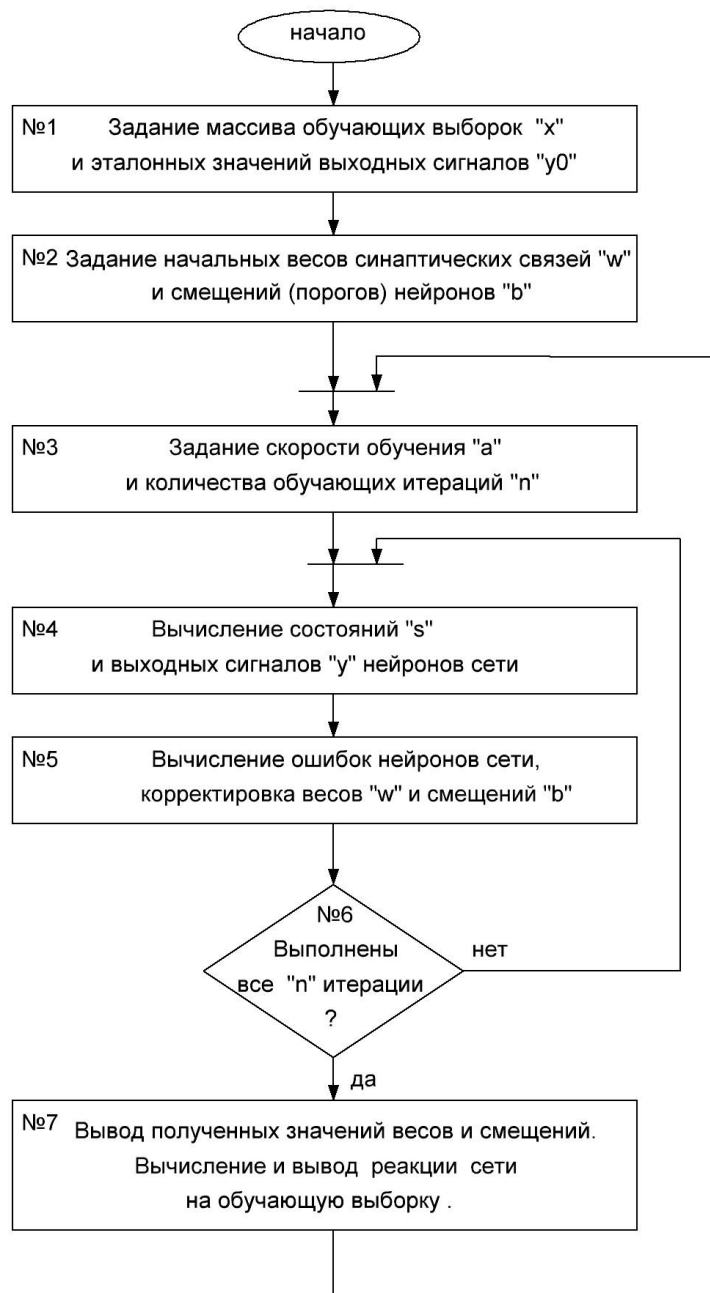


Рис. 1. Алгоритм обучения искусственной нейронной сети

При использовании в нейронах логистической функции активации ошибка, между эталонным значением выходного сигнала " y_0 " и его реальным значением " y ", рассчитывается по формулам:

$$\delta_j = (y_0 - y) * y_j * (1 - y_j) \text{ - для нейронов выходного слоя,}$$

$$\delta_j = y_j * (1 - y_j) * \sum_i \delta_i * w_{ij} \text{ - для остальных слоев,}$$

где: δ_j - ошибка нейрона с номером j ; y_j - выход нейрона с номером j ; i – номер слоя, который посылает ошибку по сети в обратном направлении (с выхода сети на вход).

Рассчитываются величины, на которые надо изменить весовые коэффициенты: $\Delta w_{ij} = \alpha * \delta_j * x_i$, где: α - скорость обучения; x_i - сигнал, поступающий от нейрона с номером i ; δ_j - ошибка нейрона j .

Выполняется коррекция весов синаптических связей путем прибавления величин Δw_{ij} к предыдущим значениям весовых коэффициентов. Для корректировки смещений (порогов) нейрона с номером j используется формула: $b_j = b_j + \alpha * \delta_j$.

В блоке №6 контролируется количество выполненных итераций обучения.

В блоке №7 выводятся, для визуального контроля, полученные в процессе обучения величины весов синаптических связей и смещений нейронов сети и выполняется расчет реакции обученной сети на сигналы обучающей выборки.

Программная реализация рассмотренного алгоритма выполнена на языке программирования C# (платформа Microsoft.NET) [8, 9, 10].

Ниже приведена основная часть кода разработанной программы:

```
// задание типа переменных и начальных значений весов
double w11=0, w21=0, w12=0, w22=0, w13=0, w23=0, b1=0, b2=0, b3=0, a, s1, s2, s3, error3, error2, error1, y1, y2;
int n; // число итераций
// интерфейс ввода обучающей выборки "x1"
double[] x1 = new double[4] {
double.Parse(textBox11.Text),
double.Parse(textBox12.Text),
double.Parse(textBox14.Text),
double.Parse(textBox13.Text)};
// интерфейс ввода обучающей выборки "x2"
double[] x2 = new double[4] {
double.Parse(textBox18.Text),
double.Parse(textBox17.Text),
double.Parse(textBox16.Text),
double.Parse(textBox15.Text)};
// интерфейс ввода эталонных значений выхода сети
double[] y0 = new double[4] {
double.Parse(textBox22.Text),
double.Parse(textBox21.Text),
double.Parse(textBox20.Text),
double.Parse(textBox19.Text)};
double[] y = new double[4]; // массив выходных значений сети
// процедура функции вычисления
for (int j = 1; j <= n; j++)
{for (int k = 1; k <= 4; k++)
{// вычисление текущих состояний "s" и выходных значений "y" нейронов
s1 = (w11 * x1[k]) + (w21 * x2[k]) + b1;
```

```
y1=1/(1+Math.Exp(-s1));
s2 = (w12 * x1[k]) + (w22 * x2[k]) + b2;
y2 = 1 / (1 + Math.Exp(-s2));
s3 = (w13 * y1) + (w23 * y2) + b3;
y[k] = 1 / (1 + Math.Exp(-s3));
// вычисление ошибок
error3 = (y0[k] - y[k]) * (1 - y[k]) * y[k];
error1 = y1 * (1 - y1) * (error3 * w13);
error2 = y2 * (1 - y2) * (error3 * w23);
// корректировка весов и смещений
w11 += a * error1 * x1[k];
w12 += a * error1 * x2[k];
w21 += a * error2 * x1[k];
w22 += a * error2 * x2[k];
w13 += a * error3 * y1;
w23 += a * error3 * y2;
b1 += a * error1;
b2 += a * error2;
b3 += a * error3;}}
// вывод результатов обучения сети
textBox1.Text = w11.ToString("0.00");
textBox2.Text = w21.ToString("0.00");
textBox3.Text = b1.ToString("0.00");
textBox6.Text = w12.ToString("0.00");
textBox5.Text = w22.ToString("0.00");
textBox4.Text = b2.ToString("0.00");
textBox9.Text = w13.ToString("0.00");
textBox8.Text = w23.ToString("0.00");
textBox7.Text = b3.ToString("0.00");
textBox26.Text = y[0].ToString("0.00");
textBox25.Text = y[1].ToString("0.00");
textBox24.Text = y[2].ToString("0.00");
textBox23.Text = y[3].ToString("0.00");
```

Программа адаптирована для обучения нейронной сети, показанной на рис.2. Сеть состоит из трех нейронов, входные сигналы "x1, x2" образуют первый слой сети, нейроны 1 и 2 входят в состав второго (промежуточного) слоя, нейрон с номером 3 находится в выходном слое, сигнал на его выходе характеризует реакцию сети на входные сигналы. В состав нейронов входят преобразователи, устанавливающие логистическую зависимость выходных сигналов нейронов "y1, y2, y" от их текущих состояний "s" [2].

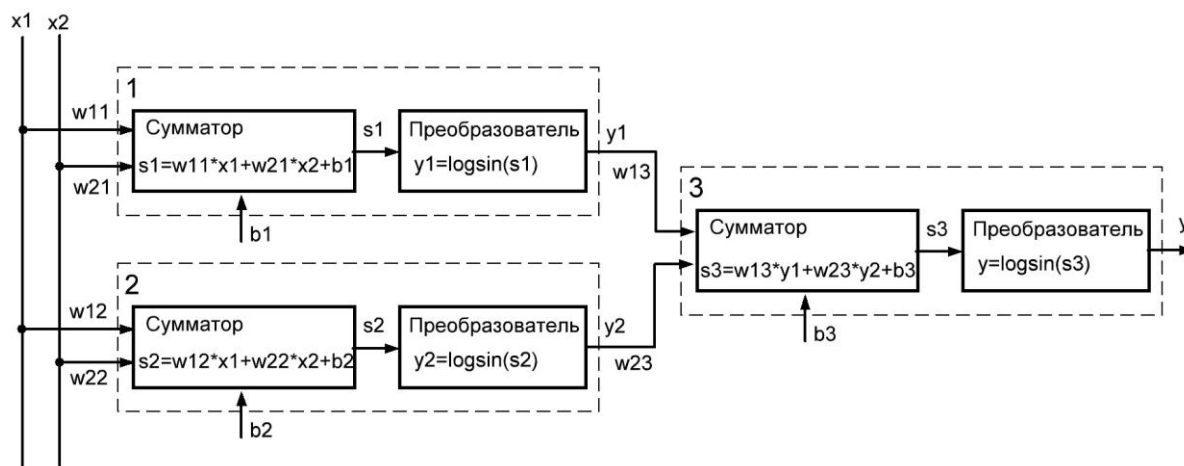


Рис. 2. Структурная схема нейронной сети

Интерфейс обеспечивающий взаимодействие программы с оператором (разработчиком нейронной сети) представлен на рис. 3. Для обучения нейронной сети необходимо ввести обучающую выборку, ожидаемые выходные данные («Эталонный выход») и «нажать» кнопку «Вычислить» (рис.3). Веса синаптических связей нейронов автоматически корректируются и в поле «Выходной сигнал» отобразится полученный результат работы нейронной сети. Изменяя параметры «Скорость обучения» и «Количество итераций», можно корректировать скорость и точность обучения нейронной сети.

Обучающая выборка				Синаптические веса						Скорость обучения		
X1	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	W11	<input type="text" value="-5,39"/>	W12	<input type="text" value="5,05"/>	W13	<input type="text" value="-9,98"/>	а	<input type="text" value="0,9"/>
X2	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	W21	<input type="text" value="-6,43"/>	W22	<input type="text" value="-1,82"/>	W23	<input type="text" value="1,73"/>	Количество итераций	
Эталонный выход				b1	<input type="text" value="8,57"/>	b2	<input type="text" value="1,19"/>	b3	<input type="text" value="3,11"/>	п <input type="text" value="10000"/>		
Y0	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="button" value="Вычислить"/>							
Выходной сигнал												
Y	<input type="text" value="0,00"/>	<input type="text" value="0,01"/>	<input type="text" value="0,01"/>	<input type="text" value="0,99"/>	<input type="button" value="Очистить"/>							

Рис.3. Интерфейс программы

В качестве примера, с помощью разработанной программы, был проведен процесс обучения нейронной сети, выполняющей функцию сумматора двоичных чисел, схема и таблица истинности сумматора представлены на рис. 4. Схема выполнена на основе нейронной сети, показанной на рис. 2.

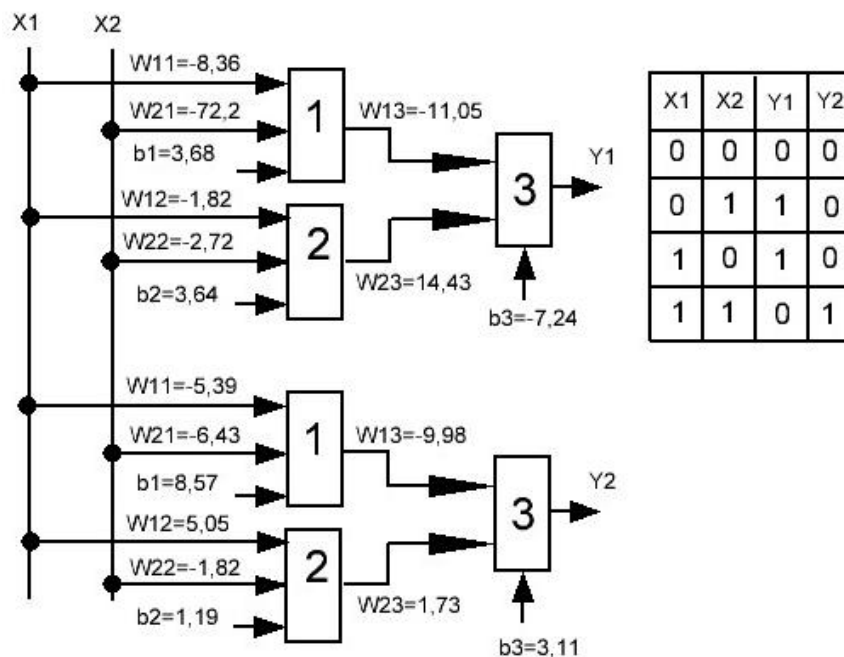


Рис. 4. Схема сумматора

Полученные в статье результаты могут найти применение при разработке программного обеспечения для микропроцессорных систем управления технологическим оборудованием. Рассмотренный алгоритм и текст программы могут быть использованы при разработке автоматизированной системы проектирования искусственных нейронных сетей.

Литература

1. Minsky M. L., Papert S. A. Perceptrons. — Cambridge: MIT Press, 1969. 258 с.

2. Хайкин С. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. М.: Издательский дом «Вильямс», 2006. 1104 с.
3. Тархов, Д.А. Нейронные сети. Модели и алгоритмы. Кн.18. Справочное издание. (Серия 'Нейрокомпьютеры и их применение'): - М.:Радиотехника, 2005. 256 с.
4. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия – Телеком, 2002. 382 с.
5. Романов Д.Е. Нейронные сети обратного распространения ошибки // Инженерный вестник Дона, 2009, №3. URL: ivdon.ru/magazine/archive/n3y2009/143/.
6. Лиля В.Б. Алгоритм и программная реализация адаптивного метода обучения искусственных нейронных сетей // Инженерный вестник Дона, 2012, №1. URL: ivdon.ru/magazine/archive/n1y2012/626/.
7. Handbook of neural network signal processing/ Edited by Yu Hen Hu, Jenq-Neng Hwang. – Boca Raton; London; New York, Washington D.C.: CRC press, 2001. 384 с.
8. Головки В.А., под ред. проф. Галушкина А.И. Нейронные сети: обучение, организация и применение, ИПРЖР, Москва, 2001. 256 с.
9. Фаронов В. В., Создание приложений с помощью С#: Руководство программиста. М.: Эксмо, 2008. 575 с.
10. Пахомов Б. И. С# для начинающих. СПб.: БХВ-Петербург, 2014. 432 с

References

1. Minsky M. L., Papert S. A. Perceptrons. Cambridge: MIT Press, 1969. 258 p.
 2. Khaykin S. Neyronnye seti: polnyy kurs [Neural Networks: a complete course], 2-e izdanie. Per. s angl. M.: Izdatel'skiy dom «Vil'yams», 2006. 1104 p.
-



3. Tarkhov, D.A. Neyronnye seti. Modeli i algoritmy [Neural networks. Models and algorithms]. Kn.18. Spravochnoe izdanie. (Seriya 'Neyrokomp'yutery i ikh primeneniye'): M.: Radiotekhnika, 2005. 256 p.

4. Kruglov V.V., Borisov V.V. Iskusstvennye neyronnye seti. Teoriya i praktika [Artificial neural networks. Theory and practice]. M.: Goryachaya liniya Telekom, 2002. 382 p.

5. Romanov, D.E. Inzhenernyj vestnik Dona, 2009, №3. URL: ivdon.ru/magazine/archive/n3y2009/143/.

6. Lila V.B. Inzhenernyj vestnik Dona, 2012, №1. URL: ivdon.ru/magazine/archive/n1y2012/626/.

7. Yu Hen Hu, Jenq-Neng Hwang. Boca Raton; London; New York, Washington D.C.: CRC press, 2001. 384 p.

8. Golovko V.A., pod red. prof. Galushkina A.I. Neyronnye seti: obuchenie, organizatsiya i primeneniye [Neural networks: Training, organization, and application], IPRZhR, Moscow, 2001. 256 p.

9. Faronov V. V., Sozdanie prilozheniy s pomoshch'yu C#: Rukovodstvo programmista [Creating Applications with C#: A Programmer's Guide]. M.: Eksmo, 2008. 575 p.

10. Pakhomov B. I. C# dlya nachinayushchikh [C# for beginners]. SPb.: BKhV-Peterburg, 2014. 432 p.