УДК 004.032.26

Д. Хорват, кандидат технических наук
Институт прикладной механики и мехатроники машиностроительного факультета
Словацкого технического университета, Братислава

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ НЕЙРОНА, ПОЛУЧЕННАЯ С ПОМОЩЬЮ ОПЕРАЦИОННОГО УСИЛИТЕЛЯ

В статье описана программная и аппаратная реализация математической модели нейрона.

Ключевые слова: нейронная сеть, обучение Хебба, операционный усилитель.

Введение

Главной задачей настоящей работы было заменить программную реализацию математической модели нейрона, основанную на использовании компьютерной программы? аппаратным решением, обеспечивающим более быструю (параллельную) обработку входных данных за счет использования возможностей аппаратных средств. Поэтому программная реализация модели нейрона было заменена аппаратной с использованием операционного усилителя (далее ОУ), разработанного фирмой Cypres, комплект PSoC 4 – Pioneer Kit.

Биологический нейрон

Нейроны являются основными элементами, из которых построена нервная система. Каждый нейрон соединен с сотнями и даже тысячами других нейронов. Отдельные нейроны связаны между собой через нервные окончания, так называемые синапсы. В результате нейроны образуют пространственную сеть, аналогичную вычислительной. Нейроны служат для обработки, хранения и передачи информации.

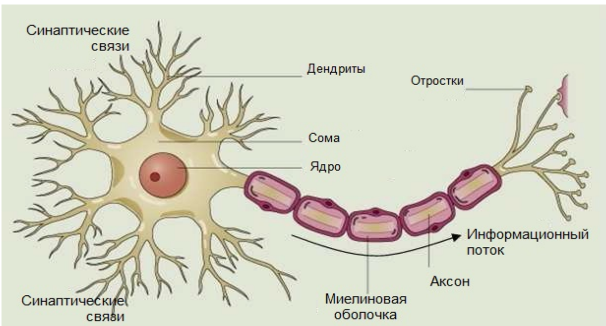

Рис. 1. Схематическое изображение нейрона [1]

Технический нейрон – перцептрон

Технический нейрон представляет собой значительно упрощенную форму биологического нейрона. Является основной структурной и вычислительной единицей искусственной нейронной сети. Самая распространенная модель искусственного нейрона (перцептрон) представляется как пороговая величина (элемент), в которой используется так называемая скачкообразная передаточная функция сигма (рис. 2). В этой модели нейрон получает данные от других внешних источников и вычисляет «взвешенную» сумму n входных сигналов x и формирует на выходе сигнал величины 1, если эта сумма превышает определенный порог, и 0, если она его не превышает [2].

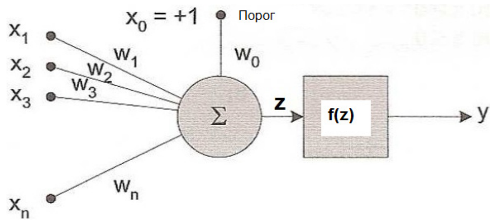

Рис. 2. Схема технического нейрона [3]

Нейрон перерабатывает входную информацию на выходную, наиболее часто в соответствии с выражением [4]:

$$z_j = \sum_{i=0}^{n} w_i x_i \tag{1}$$

или

$$z_j = \sum_{i=0}^{n} w_i x_i + b\,, \tag{2}$$

где $w_0 \cdot x_0 = b$; z – выходящая информация; $f(z)$ – передаточная функция сигма (линейная и сигмоидальная); x_i – входящие сигналы (входящие данные); w_i – синаптические весы входящих сигналов; b – порог чувствительности (bias).

Значения скачкообразной передаточной функции сигма:

$f(z) = 0$ при $z <= 0$, $f(z) = 1$ при $z > 0$.

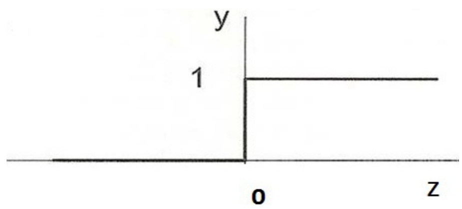

Рис. 3. Скачкообразная передаточная функция

Выражение (3) для бинарного порогового элемента является уравнением прямой линии, перпендикулярной к вектору **w** синаптического веса входного сигнала.

$$\sum_{i=0}^{n} w_i x_i + b = 0\,. \tag{3}$$

Прямая линия задается уравнением (рис. 4):

$$w_1 \cdot x_1 + w_2 \cdot x_2 + b \cdot 1 = 0\,. \tag{4}$$

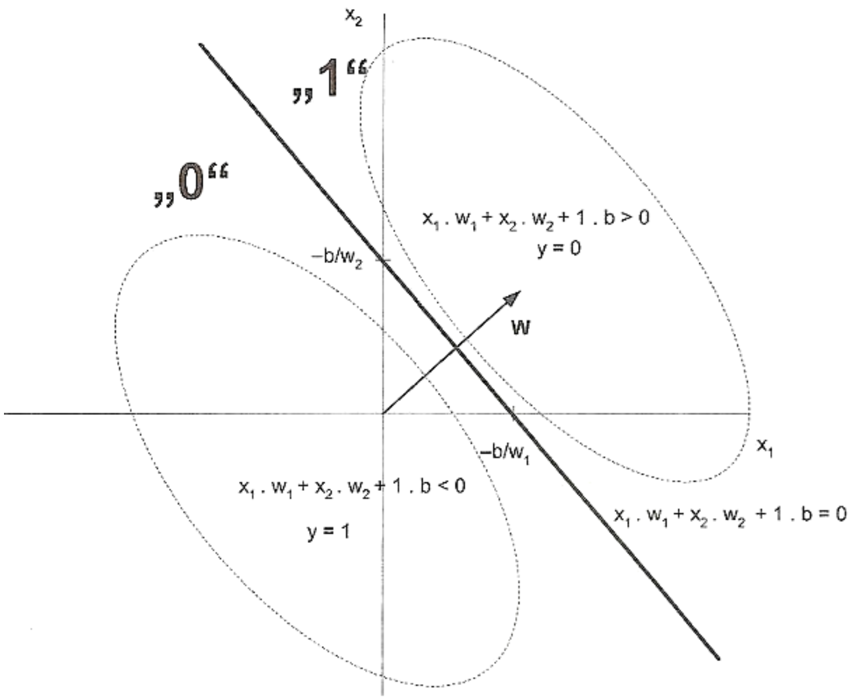

Рис. 4. Разделение плоскости входного пространства (порговой величины) прямой линией на два класса («0» и «1») для случая нейрона с двумя входами [5]

В случае искусственного нейрона (нейронной сети) применяются следующие способы обучения: обучение с «учителем» (сравнение выходящего сигнала с заданным значением) и обучение «без учителя» (самообучение). Обучение представляет собой процесс, с помощью которого получают весовые значения входящих сигналов (вектора *w*), позволяющие разделить их на два требуемых класса. Самым распространенным методом обучения является обучение Хебба (алгоритм Хебба), которое было создано по принципу обучения биологических систем, в процессе обучения которых происходит модификация весов.

Алгоритм обучения включает [6]:

1. Инициализация (определение) весов (небольших случайных значений).
2. Постепенное (последовательное) представление входящего сигнала и заданного выходящего сигнала.
3. Расчет потенциала нейрона.
4. Расчет весов:

выход = заданный выход: $wi(t+1) = wi(t)$

(конец расчета)

выход = 0, но должен был быть 1:

$$wi(t+1) = wi(t) + ni \cdot delta \cdot xi(t)$$

výstup = 1, но должен был быть 0 :

$$wi(t+1) = wi(t) - ni \cdot delta \cdot xi(t)$$

5. Повторение процесса обучения (до получения нулевого или минимального отклонения выходящего значения от ожидаемого).

При определении весов конкретные значения отдельных весов и порогов чувствительности находятся из интервала <0,1>. Значение $x_0 = 1$.

Обучение весов производится в соответствии с выражением [7]:

$$wi(t+1) = wi(t) + ni \cdot delta \cdot xi(t),$$

где ni – коэффициент обучения из интервала $0 < ni <= 1$. Иногда используется коэффициент ni, значения которого изменяются (уменьшаются) в зависимости от времени, а именно $ní = k / t$, где k – постоянная, а t – время.

delta – разница между реальным (ожидаемым) и рассчитанным значением:

$delta = o(t) - y(t)$.

Следует отметить, что однослойные нейронные сети могут быть использованы только для решения так называемых линейно разделимых задач (**И/*AND***, **ИЛИ/*OR***). Для решения линейно неразделимых задач (***XOR***) необходимо использовать многослойную нейронную сеть.

Использование при обработке логических операций

Решение логической операции И/AND

w = [0,2; 0,3] – инициализация весов является случайной;

$b = 0{,}1$;

$ní = 0{,}5$.

Расчет весов и порога чувствительности продолжается то тех пор, пока разница между реальным (ожидаемым) и рассчитанным значением, т. е. *delta*, не достигнет нуля для всех входных векторов X (в нашем случае для четырех) (рис. 5, табл. 1, 2).

Таблица 1. **Входные векторы и ожидаемые выходящие значения функции И/*AND***

x_2	x_1	O
0	0	0
0	1	0
1	0	0
1	1	1

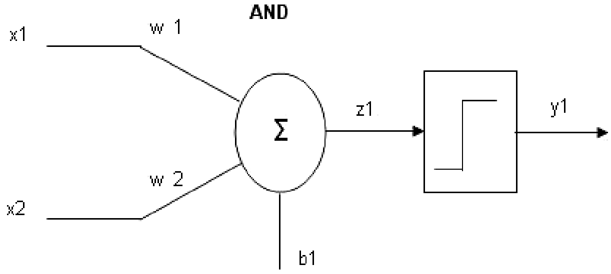

Рис. 5. Пример для нейрона с двумя входами и с передаточной функцией сигма

Таблица 2. **Расчет весов**

Логическая функция AND											
Нейрон 0	Ряд	x[0]	w[0]=b	x[1]	w[1]	x[2]	w[2]	Выход	z	y	delta
	1	1	0.1	1	0.2	1	0.3	1	0.6	1	0
	2	1	0.1	0	0.2	1	0.3	0	0.4	1	-1
	3	1	-0.4	1	0.2	0	-0.2	0	-0.2	0	0
	4	1	-0.4	0	0.2	0	-0.2	0	-0.4	0	0
	5	1	-0.4	1	0.2	1	-0.2	1	-0.4	0	1
	6	1	0.1	0	0.7	1	0.3	0	0.4	1	-1
	7	1	-0.4	1	0.7	0	-0.2	0	0.3	1	-1
	8	1	-0.9	0	0.2	0	-0.2	0	-0.9	0	0
	9	1	-0.9	1	0.2	1	-0.2	1	-0.9	0	1
	10	1	-0.4	0	0.7	1	0.3	0	-0.1	0	0
	11	1	-0.4	1	0.7	0	0.3	0	0.3	1	-1
	12	1	-0.9	0	0.2	0	0.3	0	-0.9	0	0
	13	1	-0.9	1	0.2	1	0.3	1	-0.4	0	1
	14	1	-0.4	0	0.7	1	0.8	0	0.4	1	-1
	15	1	-0.9	1	0.7	0	0.3	0	-0.2	0	0
	16	1	-0.9	0	0.7	0	0.3	0	-0.9	0	0
	17	1	-0.9	1	0.7	1	0.3	1	0.1	1	0
	18	1	-0.9	0	0.7	1	0.3	0	-0.6	0	0
	19	1	-0.9	1	0.7	0	0.3	0	-0.2	0	0
	20	1	-0.9	0	0.7	0	0.3	0	-0.9	0	0

В последних четырех рядах табл. 2 видно, что *delta* = 0 для всех комбинаций входных векторов и ожидаемых выходных данных. Обучение сети завершено.

Результат: w=[0,7; 0,3], b = 0,9,

$x_1 = -b/w_1$, если $x_2 = 0$,

$x_2 = -b/w_2$, если $x_1 = 0$.

Получена прямая, соединяющая точки x_1 и x_2, которая разделила плоскость на классы «0» и «1» (рис. 6):

$X_1 = 0{,}9/0{,}7=1{,}28$,

$X_2 = 0{,}9/0{,}3=3$.

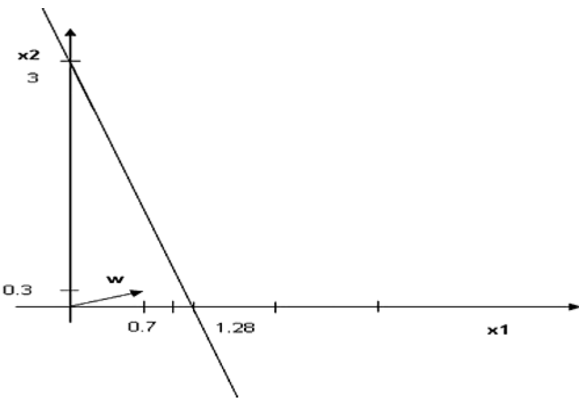

Рис. 6. Разделение плоскости на классы «0» и «1» [8]

Решение логической операции XOR

Поскольку функция ***XOR*** является линейно неразделимой (нельзя разделить плоскость на классы «0» и «1»), для ее решения необходимо использовать многослойную нейронную сеть (табл. 3).

Таблица 3. **Входные векторы и ожидаемые выходящие значения функции *XOR***

X_2	X_1	O
0	0	0
0	1	1
1	0	1
1	1	0

$$O = x_1 \cdot (inv)\, x_2 + (inv)\, x_1 \cdot x_2 .$$

Ожидаемые выходящие данные функции ***NAND*** и **ИЛИ/*OR*** идут на вход нейрона, реализующего функцию **И/*AND***. Результатом являются значения, соответствующие функции ***XOR*** (табл. 4).

Таблица 4. **Входные векторы и ожидаемые выходящие значения функции *NAND***

X_2	X_1	O_1
0	0	1
0	1	1
1	0	1
1	1	0

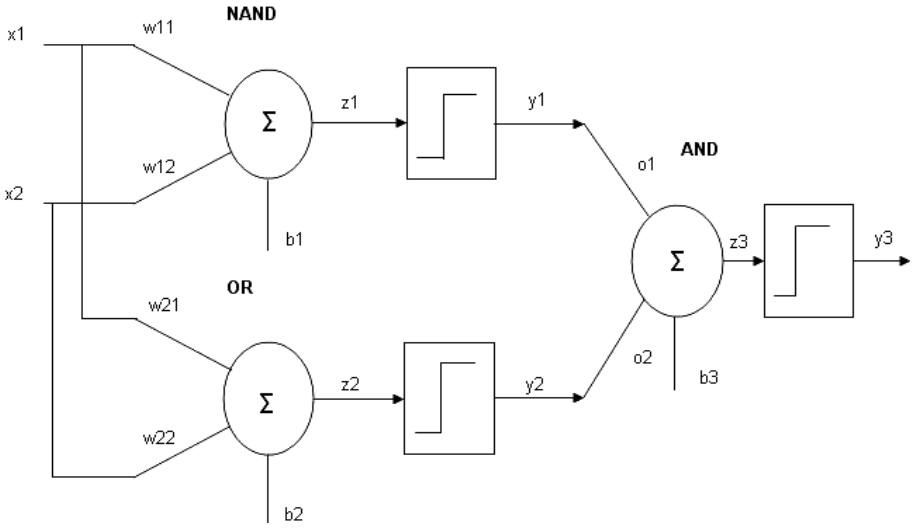

Рис. 7. Пример для нейрона с двумя входами – решение логической операции ***XOR*** [9]

Таблица 5. **Входные векторы и ожидаемые выходящие значения функции ИЛИ/*OR***

X_2	X_1	O_2
0	0	0
0	1	1
1	0	1
1	1	1

Таблица 6. **Входные векторы и ожидаемые выходящие значения функции И/*AND***

O_2	O_1	O
0	1	0
1	1	1
1	1	1
1	0	0

Можно видеть, что в результате комбинации функций ***NAND***, **ИЛИ/*OR*** и **И/*AND*** получена функция ***XOR*** (табл. 5, 6).

Аппаратные средства (хардвер)

Комплект PSoC 4 – Pioneer kit фирмы Cypress

На рис. 8 представлена электронная плата комплекта *PSoC 4* фирмы Cypress.

В состав комплекта PSoC 4 входят:

– 32-битовый MCU;
– 48 МГц ARM Cortex-M0 CPU;
– 32 Кб Flash;
– 4 Кб SRAM;
– усилители и компараторы;
– 12-битовый 1-Msps аналого-цифровой преобразователь (АЦП);
– 8-битовый цифро-аналоговый преобразователь (ЦАП);
– серийные коммуникационные модули I2C, SPI или UART;
– 16-битовые ШИМ (PWM) модули;
– 32-битовые таймер/счетные модули (каналы).

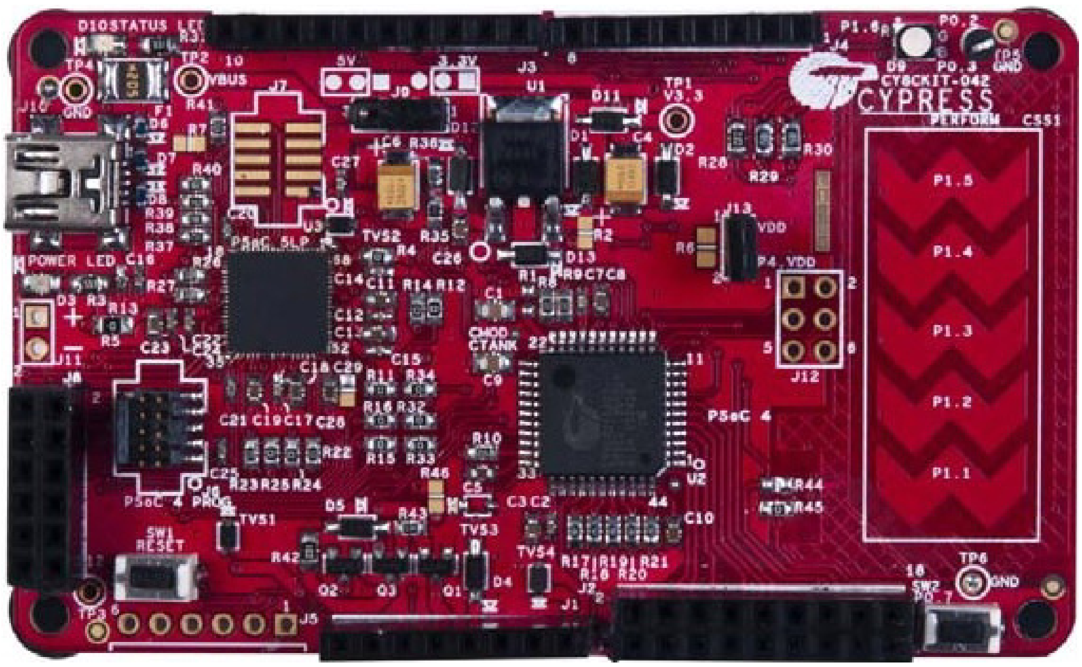

Рис. 8. Электронная плата комплекта *PSoC 4* [10]

Сумматор

На рис. 9 представлена схема сумматора.

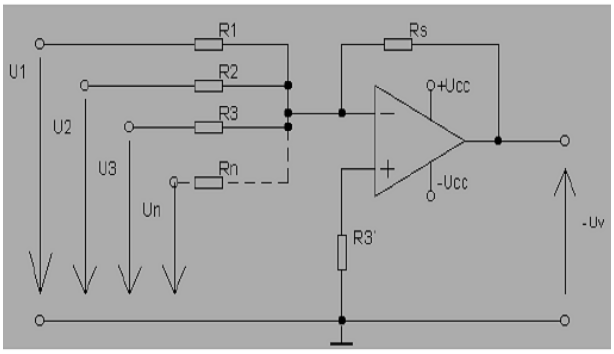

Рис. 9. Схема сумматора [11]

Операционный усилитель (далее ОУ) служит для формирования на выходе напряжения, пропорционального сумме всех входных напряжений *N*. На суммирующий узел подается ток *I1, I2, ... In*, проходящий через резисторы *R1, R2, ... Rn*. В результате подачи точных **напряжений** на **инвертирующий вход** получаем обратное (инверсное) выходное напряжение *Uv* в соответствии с выражением:

$$Uv = -Rs\left(U1/R1 + U2/R2 ... + Un/Rn\right). \quad (5)$$

Резистор *R3*' служит для минимизации колебания нуля, и его расчет производится следующим образом:

$$1/R3' = 1/R1 + 1/R2 + ... + 1/Rn . \quad (6)$$

Усиление отдельных входов ОУ определяется из выражения:

$$A_i = R/R_i . \quad (7)$$

Общее (суммирующее) усиление ОУ дано выражением:

$$A = R/R0 + R/R1 + ... + R/Rn. \quad (8)$$

Реализация математической модели нейрона с помощью операционного усилителя

При аппаратной реализации математической модели нейрона с использованием аппаратных средств исходим из предположения, что усиление отдельных входов ОУ равняется абсолютному значению весов отдельных входных сигналов нейрона.

$$A_i = abs\left(W_i\right). \quad (9)$$

При этом программный сумматор был заменен сумматором, реализованным с помощью ОУ, а функция сигма была заменена компаратором.

Питающее напряжение ОУ и компаратора равняется +5 В. Используется входное напряжение с уровнем +/– 1 В или +/– 2,5 В. Входное напряжение создается стабилизатором. Входное напряжение +/– 2,5 В получаем с помощью двух последовательно соединенных сопротивлений (рис. 10) напротив точки взаимного соединения этих сопротивлений.

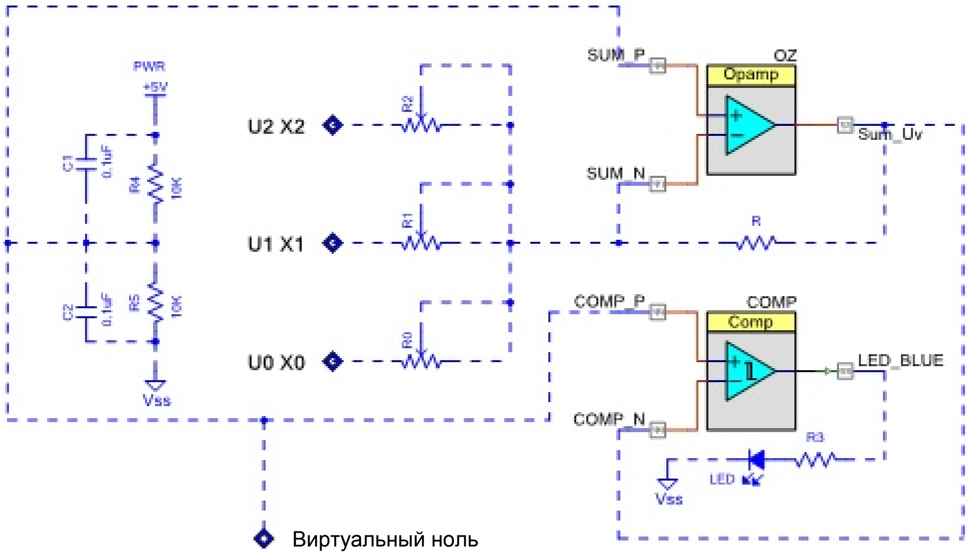

Рис. 10. Пример аппаратного решения нейрона с передаточной функцией

Эта точка принята за «виртуальный нуль» и служит в качестве опорного напряжения. «Виртуальный нуль» соединен с плюсом на входе операционного усилителя. К минусу на входе операционного усилителя подключены отдельные входные сопротивления *R0, R1* и *R2*. Между отрицательным входом и выходом операционного усилителя подключено сопротивление *R* с обратной связью. Общее (суммирующее) усиление операционного усилителя определено выражением (8). Суммирующее выходное напряжение SUM_Uv дано в выражении (5). Усиление *R/Ri* соответствует абсолютному значению весов W_i нейрона (выражение (9)). С учетом этого следует, что:

$$Z = \sum X_i \cdot W_i = \sum \pm U_i \cdot A_i = \sum \pm U_i \cdot R/R_i \cdot i = 0 - n , \quad (10)$$

где *n* – количество входов нейрона.

Следует подчеркнуть, что если значение некоторого веса W_i является отрицательным, напряжение U_i на соответствующем входе X_i тоже будет отрицательным, поскольку сопротивление не может принимать отрицательные значения.

Таблица 7. **Логическая операция И/*AND* с входящим напряжением ±1 В (*ní*=0,5)**

AND	X_2 = 1 В	X_1 = 1 В	X_0= -1 В	*Z*	*Y*	*SUM_Uv*	*COMP*
1	0	0	1	–0,9	0	0,89	1
2	0	1	1	–0,2	0	0,2	1
3	1	0	1	–0,6	0	0,6	1
4	1	1	1	0,1	1	–0,09	0
	W_2= 0,3	W_1= 0,7	W_0= = –0,9				

$A_2 = W_2 = R/R2,$
$A_1 = W_1 = R/R1,$
$A_0 = W_0 = R/R0.$
R = 1000 Ом, Ui = ±1 В,
$R2 = R/W_2$ = 1000/0,3 = 3333 Ом,
$R1 = R/W_1$ = 1000/0,7 = 1429 Ом,
$R0$ = R/W_0 = 1000/0,9 = 1111 Ом.

Таблица 8. **Логическая операция *NAND* с входящим напряжением ± 2,5 В (*ní*=0,5)**

NAND	X_2=–2,5 В	X_1=–2,5 В	X_0 = 2,5 В	*Z*	*Y*	*SUM_Uv*	*COMP*
1	0	0	1	1,6	1	–1,58	0
2	0	1	1	0,3	1	–0,29	0
3	1	0	1	0,9	1	–0,88	0
4	1	1	1	–0,4	0	0,38	1
	W_2= –0,7	W_1= –1.3	W_0=1,6				

R = 1000 Ом, Ui = ±2,5 В,
$R2 = R/W_2$ = (1000/0,7) · 2,5 = 3571 Ом,
$R1 = R/W_1$ = (1000/1,3) · 2,5 = 1923 Ом,
$R0 = R/W_0$ = (1000/1,6)· 2,5 = 1563 Ом.

Таблица 9. **Логическая операция *ИЛИ/OR* с входящим напряжением ±2,5 В (*ní* = 0,5)**

OR	X_2=2,5 В	X_1=2,5 В	X_0=–2,5 В	*Z*	*Y*	*SUM_Uv*	*COMP*
1	0	0	1	–0,4	0	0,4	1
2	0	1	1	0,3	1	–0,29	0
3	1	0	1	0,4	1	–0,39	0
4	1	1	1	1,1	1	–1,09	0
	W_2= 0,8	W_1= 0,7	W_0=–0,4				

R=1000 Ом, Ui=±2,5 В,
$R2=R/W_2$=(1000/0,8) · 2,5=3125 Ом,
$R1=R/W_1$=(1000/0,7) · 2,5=3571 Ом,
$R0=R/W_0$=(1000/0,4) · 2,5=6250 Ом.

Из табл. 7–9 видно, что измеренные выходящие напряжения операционного усилителя практически равны ожидаемым значениям *Z*. Знаки в столбцах *Z* и *SUM_Uv* являются противоположенными, а значения в столбцах *Y* и *COMP* являются обратными (инверсными) вследствие использования инвертирующего сумматора. При использовании иного входного напряжения, чем ±1 В, сопротивления *Ri* необходимо умножить на величину входного напряжения, например 2,5 В.

В столбце *Z* находятся данные, полученные на выходе математического нейрона после завершения операции его обучения, а в столбце *Y* приведены данные, полученные на выходе передаточной функции сигма. Столбец *SUM_Uv* содержит данные, полученные на выходе сумматора, и столбец *COMP* содержит данные, полученные на выходе компаратора, который заменял передаточную функцию сигма.

Значение «1» в табл. 7–9 указывает на подачу напряжения (например, ±1 В или ±2,5 В) на входах X_i. Значение «0» в табл. 7–9 означает отсутствие напряжения на входах X_i.

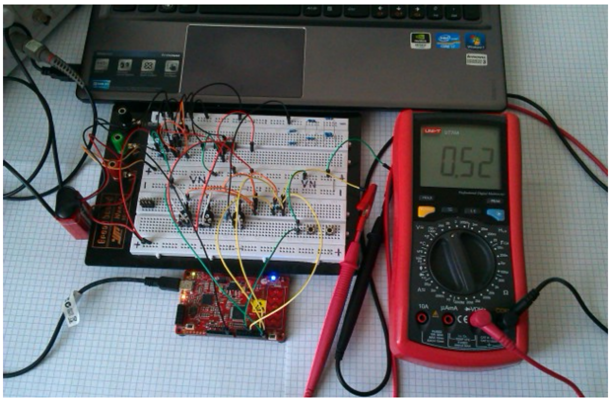

Рис. 11. Реализация логической операции ИЛИ/OR с помощью операционного усилителя комплекта *PSoC 4* (напряжение на входах ±2,5 В)

Заключение

Сопротивления были созданы на базе резисторов с номинальным сопротивлением 10 кОм. Точность результата зависит от точности получаемых при измерении значений сопротивлений. Лучше использовать цифровые потенциометры. Преимущество аппаратной реализации нейрона (нейронной сети) заключается в том, что при большом количестве нейронов аппаратная реализация является более быстродействующей по сравнению с программной реализацией, потому что в этом случае реализуются параллельные расчеты, а не последовательные.

Однослойная нейронная сеть, реализованная с использованием обучения Хебба, относится к наиболее простым задачам. В случае более сложных объектов используют иные типы сетей. Например, карты Кохонена, многослойные типы сетей и тому подобное. В многослойных сетях с большим числом входов нет возможности вручную определить параметры такой сети, поэтому в этих случаях пользуются некоторыми существующими вычислительными алгоритмами. Самым известным алгоритмом обучения многослойной нейронной сети является метод обратного распространения ошибки (Backpropagation). Искусственные нейронные сети используются в различных сферах деятельности – от распознавания различных типов бутылок в сортировочных автоматах стеклотары до управления космическими роботами.

Библиографические ссылки

1. Umělé neuronové sítě http://statnice.e-misa.info/C2-01.pdf.
2. Ibid.
3. Ibid.
4. *Petr Novák*: Mobilní roboty, Praha 2005, ISBN 80-7300-141-1.
5. Ibid.

6. Ibid.
7. Ibid.
8. Ibid.
9. Ibid.
10. Datasheet PSoC4 – PioneerKit http://www.cypress.com/?docID=47035.
11. Datasheet http://nika.informacie.sk/kp/2kp_oz1.pdf.

* * *

D. Horvath, PhD in Engineering, Institute of Applied Mechanics and Mechatronics, Faculty of Mechanical Engineering of the Slovak University of Technology in Bratislava

Mathematical model of neuron obtained using operational amplifier

The article deals with the software and hardware solution of the mathematical model of neuron.

Keywords: neuron net, Hebbian learning algorithm, operational amplifier.

Получено: 16.10.14