НОВЫЕ СРЕДСТВА
КИБЕРНЕТИКИ, ИНФОРМАТИКИ,
ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И СИСТЕМНОГО АНАЛИЗА

В.Н. ОПАНАСЕНКО, С.Л. КРЫВЫЙ

УДК 51.681.3

РАЗДЕЛЕНИЕ ПОЛНОГО МНОЖЕСТВА ЗНАЧЕНИЙ БУЛЕВЫХ ФУНКЦИЙ НА ОСНОВЕ ЗАДАННОГО ПОРОГА И ПОРОГОВОГО ОТНОШЕНИЯ

Ключевые слова: *адаптация, булева функция, пороговое отношение, задача разделения.*

ВВЕДЕНИЕ

Адаптация аппаратных средств на решение заданного алгоритма была предложена в работе [1]. Такой подход получил название «технология реконфигурируемого компьютинга» [2], воплощение которой в реальные проекты стало возможным только с появлением современных программируемых логических интегральных схем (ПЛИС). В работе [3] рассмотрены основы теории адаптивных логических сетей (АЛС) на основе ПЛИС, однако не приведено формализованное обоснование алгоритмов адаптации структуры АЛС на реализацию алгоритмов классификации [4], т.е. разделение множества входных векторов булевых переменных на основе заданных порогового значения и порогового отношения [5]. Настоящая статья посвящена доказательству корректности реализации таких алгоритмов на структурах типа АЛС, а также применению этих алгоритмов к нейронным сетям.

ПОСТАНОВКА ЗАДАЧИ

Пусть $X=\{x_1, x_2, \ldots, x_n\}$ — алфавит булевых переменных, т.е. переменных, принимающих свои значения в множестве $X=\{0,1\}$. Пусть также $x=\{y_1, y_2, \ldots, y_n\}$ — булев вектор, множество всевозможных значений которого будем обозначать A и называть полным множеством значений, где $y_i \in X$, $i=1,2, \ldots, n$. Если $A=\{a_1, a_2, \ldots, a_{2^n}\}$ — полное множество значений булевого вектора $x=\{y_1, y_2, \ldots, y_n\}$, которые будем изображать строками, состоящих из нулей и единиц, то зафиксируем некоторое значение $a \in A$ вектора x и бинарное отношение R на множестве A. Значение a назовем пороговым значением, а отношение R — пороговым отношением.

Задача разделения множества A на два подмножества: относительно порогового значения a и порогового отношения R состоит в том, чтобы разделить множество A на подмножества A_1 и A_2 таких, что

$$A_1=\{x \in A: (a, x) \in R\}, \ A_2=\{x \in A: (a, x) \notin R\}.$$

Так, если $R = \leq$, то $A_1=\{x \in A: x \leq a\}$, $A_2=\{x \in A: a < x\}$.

Сформулированная задача обобщается на случай, когда на множестве A задано несколько бинарных отношений: $\{R_1, R_2, \ldots, R_s\}$.

В настоящей статье рассматриваются четыре бинарных отношения, заданные на множестве A: $R_1(<)$, $R_2(>)$, $R_3(\le)$, $R_4(\ge)$. На практике нет необходимости явно строить разбиение множества A, а следует вычислять значение функции

$$\varphi_R(a, x) = \begin{cases} 1, & \text{если } (a, x) \in R; \\ 0, & \text{если } (a, x) \notin R. \end{cases}$$

Из определения этой функции вытекает, что она принимает значение 1 на множестве A_1, а на множестве A_2 — значение 0.

НЕПОСРЕДСТВЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Поскольку множество значений состоит из булевых слов, т.е. слов в алфавите $X' = \{0,1\}$, то на эти слова естественным образом вводятся булевы операции, выполняемые поразрядно: + (конъюнкция), &(дизъюнкция), − (отрицание), $\oplus$ (сложение по модулю 2). С помощью этих операций строятся функции, называемые логическими:

$$a+b,\ \ \bar{a}+b,\ \ a+\bar{b},\ \ \bar{a}+\bar{b},\ \ a\& b,\ \ \bar{a}\& b,\ \ a\& \bar{b},\ \ \bar{a}\& \bar{b},$$
$$a \oplus b,\ \ a \oplus \bar{b},\ \ \bar{a},\ \ \bar{b}.$$

С математической точки зрения задача вычисления значения функции $\varphi_R(a, x)$ имеет очевидное решение благодаря тому, что пороговое отношение R можно перенести на разряды. Тогда, сравнивая пороговое значение a со значениями текущего вектора x, начиная со старших разрядов, получаем $(x, a) \in R$, где k-е разряды слов разные, а все предыдущие разряды одинаковые, и $(x_k, a_k) \in R$, где x_k и a_k обозначают k-е разряды (k-е символы) слов x и a соответственно. Если все разряды в словах a и x одинаковые, то $(x, a) \in R \Leftrightarrow R$ является одним из пороговых отношений $\{\le, \ge\}$.

Из этого описания вытекает очевидный алгоритм решения задачи разделения. Пусть $a = \{a_n\ a_{n-1} \ldots a_2\ a_1\}$ — значение порогового вектора, $x = \{x_n\ x_{n-1} \ldots x_2 x_1\}$ — текущий вектор, значения координат которого подаются на вход алгоритма, где a_k — k-й разряд порогового вектора a. В этих обозначениях алгоритм принимает следующий вид.

РАЗДЕЛЕНИЕ (a, R, x)

Вход: a — пороговое значение, R — пороговое отношение, x — текущее значение входного вектора.

Выход: значение логической функции

$$\varphi_R(a, x) = \begin{cases} 1, & \text{если } (a, x) \in R; \\ 0, & \text{если } (a, x) \notin R. \end{cases}$$

Метод:
```
begin
   i = n;
   while (x_i = a_i ∧ i ≠ 0) do
       i = i − 1
   od
   if i = 0 then
       if R ∈ {≤,≥} then return (1)
       else return (0)
   else
       if (x_i , a_i ) ∈ R then return (1)
       else return (0)
end
```

Правильность данного алгоритма очевидна и не требует обоснования. Этот алгоритм называется непосредственным.

Пример 1. Пусть $a = 11001$, $x = 11010$, $y = 01110$, $z = 11001$, $R_1 =<$. Тогда для R_1 и x, y, z получаем следующие значения, генерируемые алгоритмом:

— для x первые три разряда одинаковы, а остальные разряды различные, т.е. $a_2 \neq x_2$, $a_2 < x_2$ и поэтому $a < x$ и $(a, x) \notin R_1$; следовательно, окончательное значение функции $\varphi_R(a, x)$ равно 0;

— для y имеем $a_5 \neq y_5$, $a_5 > y_5$, поэтому $y < a$ и окончательное значение функции $\varphi_R(a, x)$ равно 1;

— для z все разряды равны соответствующим разрядам a и поскольку $R_1 \notin \{\leq, \geq\}$, то окончательное значение функции $\varphi_R(a, x)$ равно 0.

РЕШЕНИЕ ЗАДАЧИ РАЗДЕЛЕНИЯ НА ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЕ

Ситуация с обоснованием алгоритма изменяется, если на фиксированной структуре АЛС, состоящей из универсальных логических элементов (реализующих произвольную логическую функцию), необходимо реализовать функцию $\varphi_R(a, x)$, вычисленную алгоритмом РАЗДЕЛЕНИЕ.

Особенность структуры элементов АЛС (рис. 1) состоит в том, что они способны настраиваться на реализацию любой логической функции: $+$, $\&$, $\oplus$, $-$. Отметим, что на вход структуры подаются разряды текущего слова x, причем старший разряд подается на каждый узел нижнего уровня.

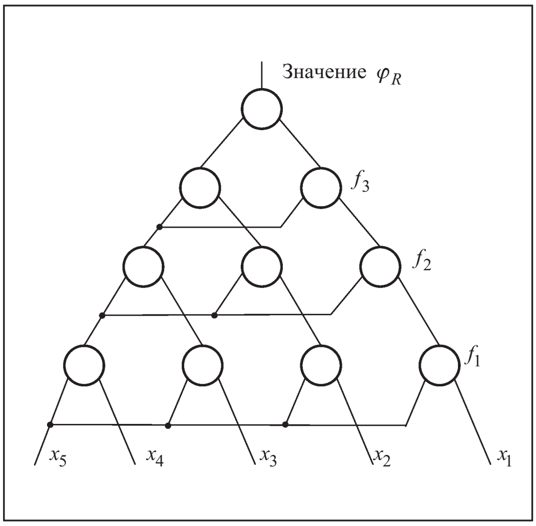

Рис. 1. Структура вычислительной среды ($n = 5$)

Рассмотрим решение задачи разделения на вычислительной структуре, показанной на рис. 1. Формирование среды вычисления на этой структуре происходит путем настраивания каждого ее уровня на заданную логическую функцию в зависимости от значения порога a и порогового отношения R . Тип логической функции для i-го уровня ($i \in [(n-1), 2]$) определяется согласно правилу

$$F_i^R = a + b\text{, если } a_i = 0; \quad F_i^R := a \& b\text{, если } a_i = 1.$$

От порогового отношения зависят значения порога, для которых вычисление значения отличается от принятого правила. Такие значения порога будем называть особыми точками порогового отношения.

Прежде чем рассмотреть вычисления значений пороговых отношений в особых точках, рассмотрим пример, который иллюстрируется вычислительной структурой из рис. 1. Пусть $n = 5$, $R =<$, $a = 10101$, $x = 10111$, $y = 01010$. Тогда на каждом уровне структуры для a и x имеем такие значения: $0111 \to 111 \to \to 11$, что определяет значение 0 для порогового отношения, а для a и y — значения $0000 \to 000 \to 00$, что определяет значение 1 для порогового отношения. Это значит, что $\varphi_R(a, x) = 0$ и $\varphi_R(a, y) = 1$, поскольку $a < x$ и $y < a$.

Из этого примера следует, что окончательное значение порогового отношения $\varphi_R(a, x)$ для заданного порога и заданного отношения определяется, исходя из значений двух последних битов, получаемых на уровне $i=1$. Пусть a_2, a_1 — два последних бита, получаемых в результате вычисления. Тогда значение порогового отношения на этом уровне вычислительной структуры определяются следующими функциями:

$$a_2=0,\ a_1=0,\ \text{тогда}\ \ F_1^{R_2}=a+b,\ F_1^{R_3}=\bar{a}\,\&\,\bar{b};$$

$$a_2=0,\ a_1=1,\ \text{тогда}\ \ F_1^{R_1}=\bar{a}\,\&\,\bar{b},\ \ F_1^{R_2}=a,\ F_1^{R_3}=\bar{a},\ \ F_1^{R_4}=a+b;$$

$$a_2=1,\ a_1=0,\ \text{тогда}\ \ F_1^{R_1}=\bar{a},\ \ F_1^{R_2}=a\&\,b,\ F_1^{R_3}=\bar{a}+\bar{b},\ \ F_1^{R_4}=a;$$

$$a_2=1,\ a_1=1,\ \text{тогда}\ \ F_1^{R_1}=\bar{a}+\bar{b},\ F_1^{R_4}=a\&\,b.$$

Если эти функции использовать в рассмотренном выше примере, то получим требуемые значения.

Далее рассмотрим пороговые отношения и их особые точки. Настройка на функцию вычисления значения отношения выполняется путем анализа первого бита порогового значения, а также порогового отношения следующих типов.

— Пороговое отношение R_1 =< . Особой точкой этого отношения является значение порога $a=100\ldots0$. Тогда вычисление значения R_1 выполняется с помощью функций

$$F_i^{R_1}=\bar{a}\ ,\ F_{i-1}^{R_1}=a\&\,b.$$

Для уровня $i=1$ значение отношения равно нулю, если на нижнем уровне получен 0, и равно значению 1, если на нижнем уровне имеем 1. Например, пусть $a=10000$, $x=11001$, $y=01101$. Тогда на уровнях для a и x получаем значения

$$0000\ \to\ 000\ \to\ 00\ \to\ 0,\ \text{так как}\ a<x,$$

а для a и y — значения

$$1111\ \to\ 111\ \to\ 11\ \to\ 1,\ \text{так как}\ y<a.$$

— Пороговое отношение R_2 =>. Особой точкой этого отношения является значение порога $a=011\ldots1$. Тогда вычисление значения R_2 выполняется с помощью функций

$$F_i^{R_2}=a\ ,\ F_{i-1}^{R_2}=a\&\,b.$$

Для уровня $i=1$ значение отношения равно нулю, если на нижнем уровне получен 0, и равно значению 1, если на нижнем уровне имеем 1. Например, пусть $a=01111$, $x=11001$, $y=01101$. Тогда на уровнях для a и x получаем значения

$$1111\ \to\ 111\ \to\ 11\ \to\ 1,\ \text{так как}\ a<x,$$

а для a и y — значения

$$0000\ \to\ 000\ \to\ 00\ \to\ 0,\ \text{так как}\ y<a.$$

— Пороговое отношение R_3 $=\le$. Особой точкой этого отношения является значение порога $a=011\ldots1$. Тогда вычисление значения R_3 выполняется с помощью функций

$$F_i^{R_3}=\bar{a}\ ,\ F_{i-1}^{R_3}=a\&\,b.$$

Для уровня $i=1$ значение отношения равно нулю, если на нижнем уровне получен 0, и равно значению 1, если на нижнем уровне имеем 1. Например, пусть $a=01111$, $x=11001$, $y=01101$. Тогда на уровнях для a и x получаем значения

$$0000\ \to\ 000\ \to\ 00\ \to\ 0,\ \text{так как}\ a<x,$$

а для a и y — значения $1111\ \to\ 111\ \to\ 11\ \to\ 1$, так как $y<a$.

— Пороговое отношение $R_4 = \geq$. Особой точкой этого отношения является значение порога $a = 100\ldots0$. Тогда вычисление значения R_4 выполняется с помощью функций

$$F_i^{R_4} = a\ ,\ F_{i-1}^{R_4} = a\&\ b\ .$$

Для уровня $i = 1$ значение отношения равно нулю, если на нижнем уровне получен 0, и равно значению 1, если на нижнем уровне имеем 1. Например, пусть $a = 01111$, $x = 11001$, $y = 01101$. Тогда на уровнях для a и x получаем значения

$$1111 \to 111 \to 11 \to 1,\ \text{так как}\ a < x$$

а для a и y — значения

$$0000 \to 000 \to 00 \to 0,\ \text{так как}\ y < a\ .$$

ОБОСНОВАНИЕ АЛГОРИТМА

Рассмотрим вопрос о корректности представленного алгоритма на основе вычислительной среды (см. рис. 1). Назовем этот алгоритм РАЗД–АЛС и будем считать его корректным, если значение, найденное с помощью этого алгоритмом для любых a, R, x , совпадает со значением функции $\varphi_R(a, x)$, вычисленным алгоритмом РАЗДЕЛЕНИЕ.

Теорема 1. Алгоритм РАЗД–АЛС корректен.

Доказательство теоремы опирается на ряд однотипных лемм для каждого порогового отношения. Докажем это для первых двух отношений, а для пороговых отношений $R_3 = \leq$, $R_4 = \geq$ к доказательству корректности алгоритма РАЗД–АЛС можно прийти самостоятельно.

Лемма 1. Алгоритм РАЗД–АЛС правильно вычисляет значение $\varphi_{R_1}(a, x)$ для порогового отношения $R_1 = <$.

Доказательство. Пусть $a = a_n a_{n-1} \ldots a_2 a_1$ — пороговое значение и $x = x_n x_{n-1} \ldots x_2 x_1$ — входной вектор. Пусть $a_j = x_j$ для всех $j = n, \ldots, (i+1)$ и $a_i \neq x_i$, причем $a = a_j a_{j-1} \ldots a_2 a_1$ не содержит особых точек отношения R_1 для $j = n, \ldots, (i+1)$. Рассмотрим возможные случаи.

1. Пусть $a_i = 1$, $x_i = 0$. Тогда на всех последующих уровнях согласно алгоритму РАЗД–АЛС до возникновения особой точки получаем векторы, состоящие из нулевых строк.

Если в конце последовательности получаем $a_2 a_1 = 00$, то это значит, что на k-м уровне ($i \leq k < 3$) имеется особая точка отношения R_1 . Тогда на этом уровне вычислительная структура настраивается на функции $F_i^{R_1} = \bar{a}$, $F_{i-1}^{R_1} = a\&\ b$ и, начиная с этого уровня, все входные значения становятся равными единице. Поэтому на выходе вычислительной структуры получаем $F_1^{R_1} = a\&\ b = 1$. Это значение совпадает со значением функции $\varphi_{R_1}(a, x) = 1$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 01$, то $F_1^{R_1} = \bar{a}\ \&\ \bar{b}$ и при значениях $x_2 x_1 = 00$ на выходе вычислительной структуры получаем $F_1^{R_1} = a\&\ b = 1$, а на остальных векторах $x_2 x_1$ выходные значения равны нулю. Это значение совпадает со значением функции $\varphi_{R_1}(a, x)$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 10$, то $F_i^{R_1} = \bar{a}$ и при значениях $x_2\ x_1 = 00$ и $x_2\ x_1 = 01$ на выходе вычислительной структуры получаем $F_i^{R_1} = \bar{a} = 1$, а на остальных векторах $x_2 x_1$ выходные значения равны нулю. Эти значения совпадают со значениями функции $\varphi_{R_1}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 11$, то $F_1^{R_1} = \bar{a} + \bar{b}$ и при всех значениях $x_2 x_1$, кроме значения $x_2 x_1 = 11$, на выходе вычислительной структуры получаем $F_1^{R_1} = \bar{a} + \bar{b} = 1$. Эти значения совпадают со значениями функции $\varphi_{R_1}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

2. Пусть $a_i = 0$, $x_i = 1$. Тогда на всех последующих уровнях согласно алгоритму РАЗД–АЛС до возникновения особой точки (если таковая имеется) получаем векторы, состоящие из единиц.

Если в конце последовательности получаем $a_2 a_1 = 00$, то это значит, что на k-м уровне ($i \le k < 3$), встретилась особая точка отношения R_1. Тогда на этом уровне вычислительная структура настраивается на функции $F_i^{R_1} = \bar{a}$, $F_{i-1}^{R_1} = a \& b$ и, начиная с этого уровня, все входные значения становятся нулевыми строками. Поэтому на выходе вычислительной структуры получаем $F_1^{R_1} = a \& b = 0$. Это значение совпадает со значением функции $\varphi_{R_1}(a, x) = 0$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 01$, то $F_1^{R_1} = \bar{a} \& \bar{b}$ и при значениях $x_2 x_1 = 00$ на выходе вычислительной структуры получаем $F_1^{R_1} = \bar{a} \& \bar{b} = 1$, а на остальных векторах $x_2 x_1$ выходные значения равны нулю. Это значение совпадает со значением функции $\varphi_{R_1}(a, x)$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 10$, то $F_i^{R_1} = \bar{a}$ и при значениях $x_2 x_1 = 00$ и $x_2 x_1 = 01$ на выходе вычислительной структуры получаем $F_i^{R_1} = \bar{a} = 1$, а на остальных векторах $x_2 x_1$ выходные значения равны нулю. Эти значения совпадают со значениями функции $\varphi_{R_1}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 11$, то $F_1^{R_1} = \bar{a} + \bar{b}$ и при всех значениях $x_2 x_1$, кроме значения $x_2 x_1 = 11$, на выходе вычислительной структуры получаем $F_1^{R_1} = \bar{a} + \bar{b} = 1$. Эти значения совпадают со значениями функции $\varphi_{R_1}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

Лемма доказана.

Лемма 2. Алгоритм РАЗД–АЛС правильно вычисляет значение $\varphi_{R_2}(a, x)$ для порогового отношения $R_2 = >$.

Доказательство. Пусть $a = a_n a_{n-1} \ldots a_2 a_1$ — пороговое значение и $x = x_n x_{n-1} \ldots x_2 x_1$ — входной вектор. Пусть $a_j = x_j$ для всех $j = n, \ldots, (i+1)$ и $a_i \ne x_i$, причем $a = a_j a_{j-1} \ldots a_2 a_1$ не содержит особых точек отношения R_2 для $j = n, \ldots, (i+1)$. Рассмотрим возможные случаи.

1. Пусть $a_i = 1$, $x_i = 0$. Тогда на всех последующих уровнях согласно алгоритму РАЗД–АЛС до возникновения особой точки получаем векторы, состоящие из нулевых строк.

Если в конце последовательности получаем $a_2 a_1 = 11$, то это значит, что на k-м уровне ($i \le k < 3$) имеется особая точка отношения R_2. Тогда на этом уровне вычислительная структура настраивается на функции $F_i^{R_2} = a$, $F_{i-1}^{R_2} = a \& b$ и, начиная с этого уровня, все входные значения становятся равными нулю. Поэтому на выходе вычислительной структуры для $x_2 x_1 = 00$ получаем $F_1^{R_2} = a \& b = 0$. Это значение совпадает со значением функции $\varphi_{R_2}(a, x) = 0$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $a < x$.

Если в конце последовательности получаем $a_2 a_1 = 01$, то $F_1^{R_2} = a$ и при значениях $x_2 x_1 = 00$ на выходе вычислительной структуры получаем значение 0, которое совпадает со значением функции $\varphi_{R_2}(a, x)$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 10$, то $F_1^{R_2} = a \& b$ и при значениях $x_2 x_1 = 00$ получаем $F_1^{R_2} = a \& b = 0$. Эти значения совпадают со значениями функции $\varphi_{R_2}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 00$, то $F_1^{R_2} = a + b$ и при значениях $x_2 x_1 = 00$ на выходе вычислительной структуры получаем $F_1^{R_2} = a + b = 0$. Эти значения совпадают со значениями функции $\varphi_{R_2}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

2. Пусть $a_i = 0$, $x_i = 1$. Тогда на всех последующих уровнях согласно алгоритму РАЗД–АЛС до возникновения особой точки (если таковая имеется) получаем векторы, состоящие из единиц.

Если в конце последовательности получаем $a_2 a_1 = 00$, то $F_1^{R_2} = a + b$ и при значениях $x_2\, x_1 = 11$ получаем на выходе вычислительной структуры значение $F_1^{R_2} = a + b = 1$, которое совпадает со значением функции $\varphi_{R_2}(a, x) = 0$, вычисляемое алгоритмом РАЗДЕЛЕНИЕ, поскольку $x > a$.

Если в конце последовательности получаем $a_2 a_1 = 01$, то при значениях $x_2 x_1 = 11$ на выходе вычислительной структуры получаем значение $F_1^{R_2} = a = 1$, которое совпадает со значением функции $\varphi_{R_2}(a, x)$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 10$, то $F_1^{R_2} = a \& b$, тогда при значениях $x_2\, x_1 = 11$ на выходе вычислительной структуры получаем $F_1^{R_2} = a \& b = 1$. Эти значения совпадают со значениями функции $\varphi_{R_2}(a, x)$, которые вычисляются алгоритмом РАЗДЕЛЕНИЕ.

Если в конце последовательности получаем $a_2 a_1 = 11$, то это значит, что на k-м уровне $(i \le k < 3)$ встретилась особая точка отношения R_2. Тогда на этом уровне вычислительная структура настраивается на функции $F_i^{R_2} = a$, $F_{i-1}^{R_2} = a \& b$ и, начиная с этого уровня, все входные значения становятся строками, состоящими из единиц, т.е. $x_2\, x_1 = 11$. Поэтому на выходе получаем $F_1^{R_2} = a \& b = 1$. Это значение совпадает со значением функции $\varphi_{R_2}(a, x) = 1$, которое вычисляет алгоритм РАЗДЕЛЕНИЕ.

Лемма доказана.

Остается доказать корректность алгоритма РАЗД–АЛС для случая особой точки.

Лемма 3. Алгоритм РАЗД–АЛС правильно вычисляет значение функции $\varphi_R(x)$ в особых точках пороговых отношений.

Доказательство. Рассмотрим возможные случаи.

1. Пороговое отношение $R_1 =<$ и его особая точка на i-м уровне вычислений $a(i) = 100...00$. Тогда согласно определению имеем $F_i^{R_1} = \bar{a}$, $F_{i-1}^{R_1} = a \& b$.

Если $x_i = 0$, то на следующих уровнях все значения будут единичными строками и, следовательно, $x_2\, x_1 = 11$. На этом уровне выходным значением будет единица, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x < a$.

Если $x_i = 1$, то на всех последующих уровнях значения будут нулевыми строками и, следовательно, $x_2\, x_1 = 00$. На этом уровне выходным значением алгоритма будет нуль, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x \ge a$.

2. Пороговое отношение $R_2 =>$ и его особая точка на *i*-м уровне вычислений $a(i)=011\ldots11$. Тогда согласно определению имеем $F_i^{R_2}=a$, $F_{i-1}^{R_2}=a\&\,b$.

Если $x_i=0$, то на следующих уровнях все значения будут нулевыми строками и, следовательно, $x_2\,x_1=00$. На этом уровне выходным значением будет нуль, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x\le a$.

Если $x_i=1$, то на всех последующих уровнях значения будут единичными строками и, следовательно, $x_2\,x_1=11$. На этом уровне выходным значением алгоритма будет единица, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x>a$.

3. Пороговое отношение $R_3 =\le$ и его особая точка на *i*-м уровне вычислений $a(i)=011\ldots11$. Тогда согласно определению имеем $F_i^{R_3}=\bar{a}$, $F_{i-1}^{R_3}=a\&\,b$.

Если $x_i=0$, то на следующих уровнях все значения будут единичными строками и, следовательно, $x_2\,x_1=11$. На этом уровне выходным значением будет единица, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x\le a$.

Если $x_i=1$, то на всех последующих уровнях значения будут нулевыми строками и, следовательно, $x_2\,x_1=00$. На этом уровне выходным значением алгоритма будет нуль, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x>a$.

4. Пороговое отношение $R_4 =\ge$ и его особая точка на *i*-м уровне вычислений $a(i)=100\ldots00$. Тогда согласно определению имеем $F_i^{R_4}=a$, $F_{i-1}^{R_4}=a\&\,b$.

Если $x_i=0$, то на следующих уровнях все значения будут нулевыми строками и, следовательно, $x_2\,x_1=00$. При этих значениях выходным значением будет нуль, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x<a$.

Если $x_i=1$, то на всех последующих уровнях значения будут единичными строками и, следовательно, $x_2\,x_1=11$. При этих значениях выходным значением алгоритма будет единица, что совпадает со значением, которое вычисляет алгоритм РАЗДЕЛЕНИЕ, поскольку $x\ge a$.

Лемма доказана.

Доказательство теоремы вытекает из доказанных лемм.

ПРИМЕНЕНИЕ АЛС

Одним из применений АЛС — нейронные сети на основе персептронов. В [6,7] описана проблема функции XOR, согласно которой однослойный персептрон не может воспроизвести такую простую функцию, как XOR. Проблема иллюстрируется однослойной однонейронной системой с двумя входами (x, y) (рис. 2) с соответствующими весовыми коэффициентами (w_1, w_2). Нейрон вычисляет сумму взвешенных входов: $NET=xw_1+xw_2$. Это уравнение линейно по x и y. Функция G реализует пороговое отношение $R_4 =\ge$, поэтому выход персептрона OUT принимает значение нуль, когда значение взвешенной суммы NET менее значения порога (вектора a), и значение единицы в случае равенства или превышения значения порога.

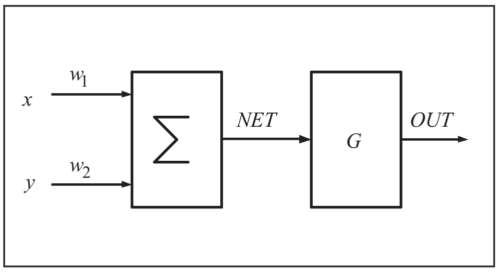

Рис. 2. Однонейронная система

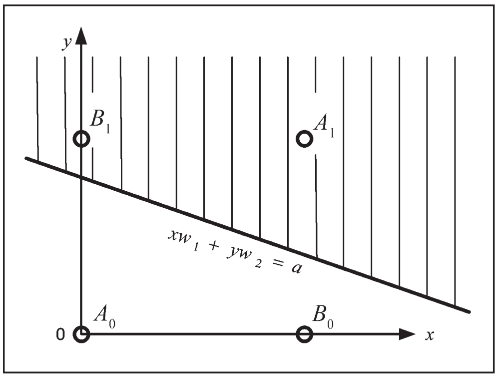

Рис. 3. Графическое представление проблемы XOR

Таблица 1

Точка	Значение x	Значение y	Значение функции XOR
A_0	0	0	0
B_0	1	0	***1***
B_1	0	1	***1***
A_1	1	1	0

Все возможные комбинации функций переменных состоят из четырех точек на плоскости (рис. 3). В зависимости от переменных значений w_1, w_2 и порога a $(a(1)-a(3))$ изменяется угол наклона и смещение прямой относительно начала координат. Для того чтобы сеть реализовала функцию XOR, необходимо расположить прямую так, чтобы точки A были с одной стороны прямой, а точки B — с другой. Но это невозможно. Никакая комбинация значений двух весовых коэффициентов и порога не сможет дать соотношения между входом и выходом, которое задано табл. 1.

Для реализации функции XOR рассмотрим двухслойную сеть с двумя двухвходовыми нейронами первого слоя, которые соединены с одним нейроном во втором слое (рис. 4). Каждый нейрон первого слоя разбивает плоскость $x-y$ на две полуплоскости. Первая обеспечивает «единичный» выход (точка A_1) выше верхней линии (логическая функция AND), вторая — «единичный» выход (точка A_0) ниже нижней линии (логическая функция NOR1).

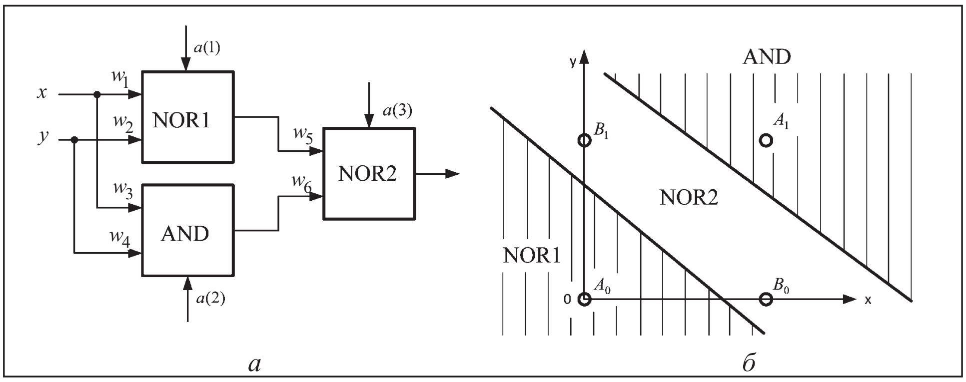

Рис. 4. Структура двухслойной сети для решения проблемы XOR (*а*) и ее графическое представление (*б*)

Выходной сигнал нейрона второго слоя равен единице только в случае, когда он выше верхней и ниже нижней линий (логическая функция NOR2). Таким образом, проблема XOR разрешима (табл. 2).

Таблица 2

Точка	Значение x	Значение y	NOR1	AND	Выходное значение (NOR2)
A_0	0	0	1	0	0
B_0	1	0	***0***	***0***	***1***
B_1	0	1	***0***	***0***	***1***
A_1	1	1	0	1	0

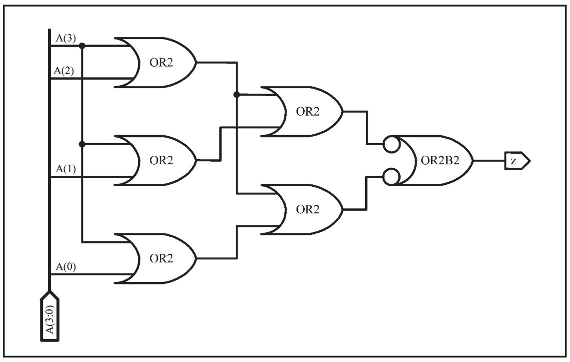

Рис. 5. Синтезированная структурная схема порогового устройства

Рассмотрим пример разрешения проблемы XOR в нейронных сетях с помощью двухслойной сети, где в качестве порогового устройства в нейронах используются не компараторы, а синтезированные с помощью рассмотренного выше алгоритма адаптации (синтеза) структур пороговые устройства для заданных параметров: значение порога $a = 0010$, пороговое отношение $R_3 = \leq$. Начинаем анализировать значения порога (вектора a) со старших битов. При $a_4 = 0$ тип логической функции для этого уровня определяется как $F_3^3 := a + b$, при $a_3 = 0$ имеем $F_2^3 := a + b$, при $a_2 = 1$, $a_1 = 0$ имеем $F_1^3 := \bar{a} + \bar{b}$.

Синтезированная структурная схема порогового устройства с использованием инструментальной системы ISE Foundation представлена на рис. 5, а его временная диаграмма работы — на рис. 6.

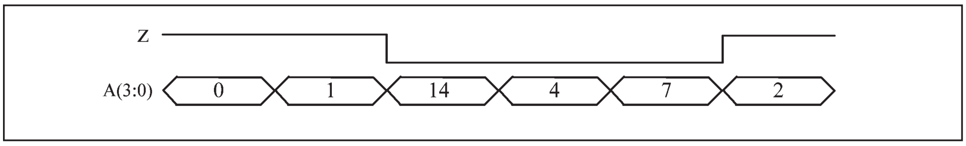

Рис. 6. Временная диаграмма работы порогового устройства

Таблица 3

Значение параметра	Первый нейрон (neur_1) первого слоя (NOR1)	Второй нейрон (neur_2) первого слоя (AND)	Третий нейрон (neur_3) второго слоя (NOR2)
R	$R_1 =<$	$R_4 =\geq$	$R_1 =<$
a	00011	11000	00010
w	$w_1 = 4$, $w_2 = 7$	$w_3 = 14$, $w_4 = 11$	$w_5 = 5$, $w_6 = 6$

Исходные данные для синтеза двухслойной нейронной сети приведены в табл. 3. Синтезированная структура двухслойной нейронной сети с использованием инструментальной системы ISE Foundation представлена на рис. 7, а ее временная диаграмма работы — на рис. 8. Если в аналогичной нейронной сети пороговые устройства реализовать с помощью стандартных компараторов, то получим результаты синтеза структуры (табл. 4).

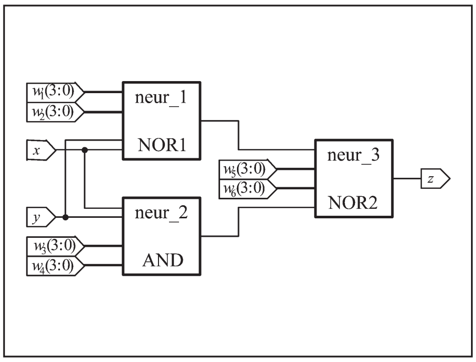

Рис. 7. Структура двухслойной нейронной сети на основе порогового устройства

Таблица 4

Параметры	Оценка вариантов нейронной сети	
	на основе компаратора	на основе порогового устройства
Аппаратные затраты (количество слайсов)	24	18
Быстродействие, нс	8,42	7,94

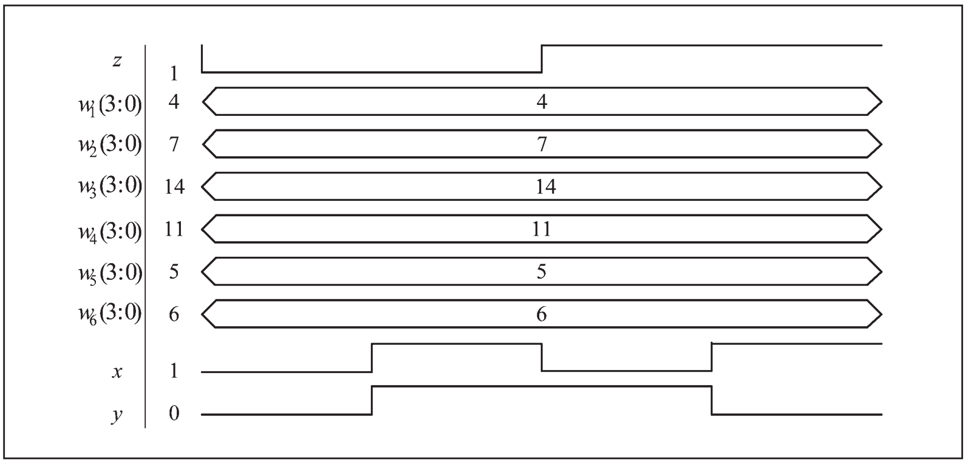

Рис. 8. Временная диаграмма работы двухслойной сети

ЗАКЛЮЧЕНИЕ

Рассмотренный в статье алгоритм адаптации структур типа АЛС на разделение множества значений булевых функций на основе заданных порогового значения и порогового отношения позволяет при фиксированной структуре связей определить типы логических функций для универсальных логических элементов. Применение АЛС для реализации порогового устройства в нейронных сетях на основе персептронов позволяет на треть сократить аппаратные затраты кристалла ПЛИС (выраженные в количестве слайсов), а также (хотя и незначительно) повысить быстродействие.

СПИСОК ЛИТЕРАТУРЫ

1. Estrin G. Organization of computer system: The fixed plus variable structure computer // Proc. Western Joint Computer Conf. — 1960. — N 5. — P. 33–40.
2. Палагин А.В., Опанасенко В.Н. Технология реконфигурируемого компьютинга // Кибернетика и системный анализ. — 2007. — **43**, № 5. — С. 72–86.
3. Палагин А.В., Опанасенко В.Н. Реконфигурируемые вычислительные системы. — К.: Просвіта, 2006. — 295 с.
4. Дискретная математика и математические вопросы кибернетики / Ю.Л. Васильев, Ф.Я. Ветухновский, В.В. Глаголев, Ю.И. Журавлёв и др. — М.: Наука, 1971. — 311 с.
5. Биркгоф Г., Барти Т. Современная прикладная алгебра. — М.: Мир, 1976. — 400 с.
6. Куссуль Н.Н. Обучение нейронных сетей с использованием метода нечетких эллипсоидальных оценок // Проблемы управления и информатики. — 2001. — № 1. — С. 14–21.
7. Куссуль Н.М., Шелестов А.Ю., Лавренюк А.М. Інтелектуальні обчислення. — К.: Наук. думка, 2006. — 186 с.

Поступила 21.02.2011