НОВІ ЗАСОБИ
КІБЕРНЕТИКИ, ІНФОРМАТИКИ,
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
ТА СИСТЕМНОГО АНАЛІЗУ

С.Л. КРЫВЫЙ, В.Н. ОПАНАСЕНКО

УДК 51.681.3

РАЗБИЕНИЕ МНОЖЕСТВА ВЕКТОРОВ С ЦЕЛЫМИ НЕОТРИЦАТЕЛЬНЫМИ КООРДИНАТАМИ С ИСПОЛЬЗОВАНИЕМ ЛОГИЧЕСКИХ АППАРАТНЫХ СРЕДСТВ

Аннотация. Рассмотрены задачи разбиения множества векторов с целыми неотрицательными координатами относительно порогового значения и порогового отношения на два класса средствами адаптируемых логических сетей. Обоснована корректность алгоритмов реализации такого разбиения для произвольного порогового значения и размерности векторов.

Ключевые слова: булева функция, классификация, целочисленные векторы, пороговые отношения.

ВВЕДЕНИЕ

Адаптация аппаратных средств к решению задачи разбиения полного множества значений булевых функций на основе порогового значения и отношения рассматривалась в работе [1]. Такой подход известен как «технология реконфигурируемого компьютинга» [2, 3], и его воплощение в реальные проекты стало возможным благодаря появлению программируемых логических интегральных схем (ПЛИС). В работах [4–8] рассматривалась задача адаптации аппаратных средств с формализованным обоснованием алгоритмов адаптации структуры адаптивных логических сетей (АЛС) на реализацию алгоритмов классификации.

В настоящей статье рассматривается задача классификации заданного множества n-мерных векторов с целыми неотрицательными координатами на основе известного метода разбиения заданного множества векторов на подмножества [7–10] с реализацией на структурах типа АЛС.

ПОСТАНОВКА ЗАДАЧИ

Пусть $V=\{v_1=(u_{11},\dots,u_{1n}), v_2=(u_{21},\dots,u_{2n}),\dots,v_m=(u_{m1},\dots,u_{mn})\}$ — множество n-мерных векторов, где $u_{ij}\in N$, N — множество натуральных чисел. Рассмотрим следующие задачи классификации.

Задача 1. Найти разбиение множества V на два подмножества: V_1 и V_2 согласно следующему условию: если для заданных j, $1\le j\le n$, и фиксированного $k\in N$ координата u_{ij} меньше или равна k, то вектор $v_i\in V_1$, иначе $v_i\in V_2$ для всех i, где $1\le i\le m$.

Задача 2. Найти разбиение множества V на два подмножества: V_1 и V_2 согласно такому условию: если для всех $1\le i\le m$, $1\le j\le n$ и фиксированного $k\in N$ значение u_{ij} меньше или равно k (по всем координатам векторов из V), то $v_i\in V_1$, иначе $v_i\in V_2$.

Задача 3. Найти разбиение множества V на два подмножества: V_1 и V_2 согласно такому условию: для заданного вектора $a = (a_1, \dots, a_n)$, который может как принадлежать множеству V, так и не принадлежать ему, и фиксированного $k \in N$ отнести $v_i \in V$ к множеству V_1, если вектор $v = v_i - a = (u_{i1} - a_1, \dots, u_{in} - a_n)$ имеет все координаты, меньшие или равные k, иначе отнести v_i к V_2.

Заметим, что задачи 1 и 2 являются частными случаями задачи 3. Действительно, задачи 1, 2 являются задачей 3, если $a = (0, \dots, 0)$ — нулевой вектор.

Приведенная постановка задачи классификации позволяет свести решение задач 1–3 к задаче классификации булевых функций на основе пороговых значения и отношения [1].

РЕДУКЦИЯ ЗАДАЧ К БУЛЕВЫМ ФУНКЦИЯМ

Представим координаты векторов из V в двоичной системе счисления, т.е. в виде слов в алфавите $X = \{0, 1\}$. Пусть $t = \max\limits_{1 \le i \le m,\, 1 \le j \le n} (u_{ij} + a_i)$ — наибольшее значение координат векторов из V. Тогда длина каждого вектора не превосходит величины $m \cdot \log t$, а длины координат не превосходят $\log t$. В результате векторы из V принимают вид $v_i = (p_{i1}, p_{i2}, \dots, p_{in})$, где p_{ij} — слово в двоичном алфавите X длины $\le \log t$. Уточним задачу разбиения для булевых векторов и опишем вычислительную структуру, с помощью которой выполняется разбиение.

Пусть $X' = \{x_1, x_2, \dots, x_n\}$ — алфавит булевых переменных, принимающих свои значения в множестве $X = \{0, 1\}$. Пусть также $v_i = (p_{i1}, p_{i2}, \dots, p_{in})$ — булев вектор, A — множество всех таких векторов, которые соответствуют векторам из V. Множество A назовем полным множеством, соответствующим множеству векторов V. Зафиксируем некоторый вектор a, имеющий ту же размерность, что и векторы из V (вектор a может принадлежать V, а может ему не принадлежать), и бинарное отношение R на множестве A. Вектор a назовем пороговым вектором, а отношение R — пороговым отношением. В настоящей статье рассматривается случай четырех бинарных пороговых отношений, заданных на множестве $A: R_1(<), R_2(>), R_3(\le), R_4(\ge)$.

Задача разбиения множества A на два подмножества относительно порогового значения a и порогового отношения R состоит в том, чтобы разбить множество A на два подмножества: A_1 и A_2 таких, что $A_1 = \{x \in A : (x, a) \in R\}$, $A_2 = \{x \in A : (x, a) \notin R\}$. Например, если $R = \le$, то

$$A_1 = \{x \in A : x \le a\}, \ A_2 = \{x \in A : a > x\}.$$

На практике нет необходимости явно строить разбиение множества A, а следует вычислять только значение функции, соответствующей отношению R:

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

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

(В данной статье используем некоторые выкладки из работы [1].)

НЕПОСРЕДСТВЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

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

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

Вычисление значения функции $\varphi_R(x, a)$ имеет очевидное решение, поскольку пороговое отношение R можно перенести на разряды. Тогда, сравнивая пороговое значение вектора a со значениями текущего вектора x, начиная со старших разрядов, получаем $(x, a) \in R$, k-е разряды которых различные, а все предыдущие разряды одинаковые, и $(x_k, a_k) \in R$, где x_k и a_k — k-е разряды (k-е символы) слов x и a соответственно. Если все разряды слов a и x одинаковые, то $(x, a) \in R \Leftrightarrow R$ является одним из отношений $\{\leq, \geq\}$.

На основании этого описания получаем очевидный алгоритм решения задачи разбиения. 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. При этих обозначениях алгоритм принимает следующий вид.

РАЗБИЕНИЕ (*a*, *R*, *x*)

Вход: a — пороговый вектор, R — пороговое отношение, x — входной вектор.

Выход: значение функции $\varphi_R(x, a)$.

Метод:

begin
  $i = n$;
  while $(x_i = a_i \wedge i \neq 0)$ do $i = i - 1$ od
  if $i = 0$ then if $R \in \{\leq, \geq\}$ then return (1) else return (0)
  else if $(x_i, a_i) \in R$ then return (1) else return (0)
end

Правильность данного алгоритма очевидна и не требует обоснования. Назовем этот алгоритм 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 другие разряды различны, т.е. $a_2 \neq x_2$, $a_2 < x_2$ и поэтому $a < x$ и $(x, a) \notin R_1$; следовательно, окончательное значение предиката $\varphi_R(x, a)$ равно нулю;

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

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

АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ НА АЛС

Ситуация с обоснованием алгоритма изменяется, если имеется фиксированная АЛС, состоящая из универсальных логических элементов (реализующих произвольную логическую функцию), на которой необходимо реализовать функцию $\varphi_R(x, a)$, вычисляемую приведенным алгоритмом РАЗБИЕНИЕ. Особенность структуры элементов АЛС состоит в том, что они способны настраиваться на реализацию любой логической функции: $+$, $\&$, $\oplus$, $-$ [11]. Следует отметить, что на вход структуры подаются разряды текущего слова x, причем старший разряд подается на каждый узел нижнего уровня и на все последующие уровни в качестве одного из аргументов.

Решение задачи разбиения на вычислительной структуре, показанной на рис. 1, неформально представлено следующим образом. Формирование среды вычисления на этой структуре происходит настраиванием каждого ее уровня на заданную логическую функцию в зависимости от значения порога вектора a и порогового отношения R. Тип логической функции для i-го уровня ($i \in [n-1, 2]$) определяется согласно правилу

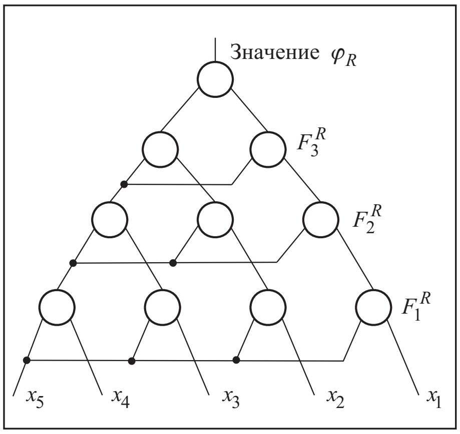

Рис. 1. Структура вычислительной среды (n = 5)

$$F_i^R = a + b, \text{ если } a_i = 0;$$

$$F_i^R = a \& b, \text{ если } a_i = 1. \qquad (1)$$

Для порогового отношения и некоторого порогового вектора a вычисление значения функции $\varphi_R(x, a)$ отличается от принятого правила. Такие векторы будем называть особыми точками порогового отношения.

Прежде чем определить значения пороговых отношений в особых точках, рассмотрим пример, который иллюстрируется вычислительной структурой из рис. 1.

Пример 2. Пусть $n = 5$, $R = <$, $a = 10101$, $x = 10111$, $y = 01010$. Тогда на каждом уровне структуры для a и x имеем такие значения: $0111 \to 111 \to 11$, что определяет значение 0 для порогового отношения R, а для a и y — значения $0000 \to 000 \to 00$, что определяет значение 1 для порогового отношения R. Это значит, что $\varphi_R(x, a) = 0$ и $\varphi_R(y, a) = 1$, поскольку $a < x$ и $y < a$.

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

$$\begin{aligned} &a_2 = 0,\ a_1 = 0, \text{ тогда } F_1^{R_2} := a + b,\ F_1^{R_3} := \bar{a} \& \bar{b}; \\ &a_2 = 0,\ a_1 = 1, \text{ тогда } F_1^{R_1} := \bar{a} \& \bar{b},\ F_1^{R_2} := a,\ F_1^{R_3} := \bar{a},\ F_1^{R_4} := a + b; \\ &a_2 = 1,\ a_1 = 0, \text{ тогда } F_1^{R_1} := \bar{a},\ F_1^{R_2} := a \& b,\ F_1^{R_3} := \bar{a} + \bar{b},\ F_1^{R_4} := a; \\ &a_2 = 1,\ a_1 = 1, \text{ тогда } F_1^{R_1} := \bar{a} + \bar{b},\ F_1^{R_4} := a \& b. \end{aligned} \qquad (2)$$

Если эти функции использовать в примере 2, то получим требуемые значения.

Рассмотрим теперь пороговые отношения и их особые точки. Настройка на функцию $\varphi_R(x, a)$ вычисления значения порогового отношения выполняется путем анализа первого бита порогового вектора и 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 нуль, и равно единице, если на нижнем уровне получена единица. Так, пусть $a = 10000$, $x = 11001$, $y = 01101$. Тогда имеем следующие значения на уровнях для a и x:

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

для a и y получаем

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

Пороговое отношение R_2 =>. Особой точкой этого отношения является значение порога $a = 011\ldots1$. Тогда вычисление значения R_1 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 нуль, и равно единице, если на нижнем уровне получена единица. Например, пусть $a = 01111$, $x = 11001$, $y = 01101$. Тогда имеем такие значения на уровнях для a и x:

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

для a и y получаем значение

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

Пороговое отношение $R_3 = \leq$. 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 нуль, и равно единице, если на нижнем уровне получена единица. Например, пусть $a = 01111$, $x = 11001$, $y = 01101$. Тогда имеем такие значения на уровнях для a и x:

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

для a и y получаем

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

Пороговое отношение $R_4 = \geq$. Особой точкой этого отношения является значение порога $a = 100\ldots0$. Тогда значение R_4 определяем 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 нуль, и равно единице, если на нижнем уровне получена единица. Например, пусть $a = 01111$, $x = 11001$, $y = 01101$. Тогда имеем такие значения на уровнях для a и x:

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

для a и y получаем

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

ОБОСНОВАНИЕ АЛГОРИТМА

Алгоритм, реализуемый на вычислительной структуре, приведенной на рис. 1, назовем РАЗБ-АЛС и будем считать его корректным, если значение, найденное этим алгоритмом для любых a, x, R, совпадает со значением функции $\varphi_R(x, a)$, вычисленным алгоритмом РАЗБИЕНИЕ.

Обоснование правильности работы этого алгоритма полностью приведено в статье [1], где доказано следующее утверждение.

Теорема 1. Алгоритм РАЗБ-АЛС корректен.

Доказательство теоремы основано на ряде однотипных лемм для каждого типа порогового отношения. Для демонстрации способа доказательства ограничимся доказательством только леммы 1, а с формулировкой остальных лемм можно ознакомиться в работе [1].

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

Доказательство. Пусть $a = a_n a_{n-1} \ldots a_2 a_1$ — пороговое значение, $x = x_n x_{n-1} \ldots$ $\ldots x_2 x_1$ — входной вектор. Пусть также $a_j = x_j$ 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 особой точки 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 РАЗБ-АЛС до появления особой точки получаем векторы, состоящие из нулевых строк.

Если в конце порогового значения 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 $\varphi_{R_1}(x, a) = 1$, которое вычисляет алгоритм РАЗБИЕНИЕ.

Если в конце порогового значения 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 при остальных значениях вектора $x_2 x_1$ выходные значения будут равны нулю. Полученное значение совпадает со значением функции $\varphi_{R_1}(x, a)$, которое вычисляет алгоритм РАЗБИЕНИЕ.

Если в конце порогового значения получаем $a_2 a_1 = 10$, то $F_1^{R_1} = \bar{a}$ и снова при значениях $x_2 x_1 = 00$ и $x_2 x_1 = 01$ на выходе получаем $F_1^{R_1} = \bar{a} = 1$, а при остальных значениях вектора $x_2 x_1$ выходные значения равны нулю. Полученные значения совпадают со значениями функции $\varphi_{R_1}(x, a)$, которые вычисляются алгоритмом РАЗБИЕНИЕ.

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

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

Если в конце порогового значения 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 нулевыми строками. Поэтому на выходе вычислительной структуры получаем $F_1^{R_1} = a \,\&\, b = 0$. Это значение совпадает со значением функции $\varphi_{R_1}(x, a) = 0$, которое вычисляет алгоритм РАЗБИЕНИЕ.

Если в конце порогового значения получаем $a_2 a_1 = 01$, то $F_1^{R_1} = \bar{a} \,\&\, \bar{b}$, и вновь при значениях $x_2 x_1 = 00$ на выходе вычислительной структуры получаем $F_1^{R_1} = \bar{a} \,\&\, \bar{b} = 1$, а при остальных значениях вектора $x_2 x_1$ выходные значения будут нулевыми. Полученное значение совпадает со значением функции $\varphi_{R_1}(x, a)$, которое вычисляет алгоритм РАЗБИЕНИЕ.

Если в конце порогового значения получаем $a_2 a_1 = 10$, то $F_1^{R_1} = \bar{a}$, и тогда при значениях $x_2 x_1 = 00$ и $x_2 x_1 = 01$ на выходе получаем $F_1^{R_1} = \bar{a} = 1$, а при остальных значениях вектора $x_2 x_1$ выходные значения будут нулевыми. Полученные значения совпадают со значениями функции $\varphi_{R_1}(x, a)$, которые вычисляются алгоритмом РАЗБИЕНИЕ.

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

Лемма 1 доказана.

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

Лемма 3. Алгоритм РАЗБ-АЛС корректно вычисляет значение функции $\varphi_R(x, a)$ в особых точках пороговых отношений.

РЕШЕНИЕ ЗАДАЧ 1–3

Из приведенной выше структуры АЛС (см. рис. 1) следует, что она не зависит от разрядности входного слова и настраивается на любой пороговый вектор. Эта особенность позволяет решить сформулированные выше задачи 1–3, используя следующую структуру.

Решение задачи 1. Значение параметра k, фигурирущего в постановке задачи, представляется в виде двоичного слова, которое является пороговым вектором a в задаче разбиения множества A. Задавая пороговое отношение R, получаем все необходимые значения для настройки АЛС.

Пример 3. Пусть $k = 14$ и множество векторов V включает векторы

$$v_1 = (7, 16, 5),\ v_2 = (11, 10, 3),\ v_3 = (17, 22, 8).$$

Максимальные значения координат векторов не превышают 22, тогда для представления этих координат двоичными словами достаточно пяти битов (поскольку $\log 22 < 5$). В таком представлении векторам множества V будут соответствовать булевы векторы множества A вида

$$p_1 = (00111, 10000, 00101),\ \ p_2 = (01011, 01010, 00011),\ \ p_3 = (10001, 10110, 01000).$$

Пусть разбиение проводится по второй координате, тогда множество A разделится на подмножества $A_1 = \{v_2\}$, $A_2 = \{v_1, v_3\}$. В этом легко убедиться непосредственно. Значению параметра k будет соответствовать пороговый вектор $p = (01110)$ (т.е. вектор a), и если пороговое отношение R задано (например, $R = <$), то все необходимые значения для настройки АЛС имеются (разрядность АЛС по входу определена — 5 бит, т.е. четыре уровня). В соответствии с приведенным выше алгоритмом РАЗБ-АЛС для структуры АЛС (см. рис. 1) определим типы функций для всех четырех уровней АЛС на основе анализа бит (начиная со старшего) порогового вектора $p = (01110)$ для $R = <$ с учетом правил (1) и (2):

1) старший бит равен нулю, поэтому тип логической функции для первого уровня определен как $f_1 = a + b$;

2) следующий бит равен единице, поэтому тип логической функции для второго уровня определен как $f_2 = a \& b$;

Таблица 1

Булевы векторы множества A	Значение вектора	Полученные типы функций АЛС			
		1-го уровня ($f_1 = a + b$)	2-го уровня ($f_2 = a\&b$)	3-го уровня ($f_3 = a\&b$)	4-го уровня ($f_4 := \overline{a}$)
p_1	(00111)	(0111)	(000)	(00)	1
	(10000)	(1111)	(111)	(11)	0
	(00101)	(0101)	(000)	(00)	1
p_2	(01011)	(1011)	(011)	(00)	1
	(01010)	(1010)	(010)	(00)	1
	(00011)	(0011)	(000)	(00)	1
p_3	(10001)	(1111)	(111)	(11)	0
	(10110)	(1111)	(111)	(11)	0
	(01000)	(1000)	(000)	(00)	1

3) следующий бит равен единице, поэтому тип логической функции для третьего уровня определен как $f_3 = a \& b$;

4) на основе анализа двух младших битов (01) для порогового отношения $R =<$ тип логической функции для последнего (четвертого) уровня определен как $f_4 = \overline{a}$.

Результаты преобразования, сведенные в табл. 1, подтверждают, что разбиение проводится по второй координате, а множество A разделится на подмножества $A_1 = \{v_2\}$, $A_2 = \{v_1, v_3\}$.

Решение задачи 2. Решение вытекает из способа решения задачи 1. Действительно, представляя координаты порогового вектора $a = (k, k, \ldots, k)$ в виде двоичных слов, выполняем n итераций по каждой координате. Следует заметить, что такое разбиение можно выполнять параллельно, настраивая каждую из n АЛС на одно и то же пороговое значение и одно и то же отношение. В этом случае возможно дополнительное ускорение выполнения классификации: если по одной координате какого-либо вектора значение функции $\varphi_R(x, a) = 0$, то дальнейшие вычисления с этим вектором проводить не следует, поскольку он будет отнесен ко второму подмножеству V_2.

В случае необходимости настройку АЛС можно выполнять для разных значений координат порогового вектора и разных отношений по каждой координате в отдельности.

Решение задачи 3. Для решения задачи необходимо вектор a просуммировать с вектором $K = (k, k, \ldots, k)$ и полученный вектор $a + K$ считать пороговым. Далее выполняется проверка способом решения задачи 2. Если вектор v_i удовлетворяет пороговому отношению R, то он принадлежит множеству A_1, иначе он принадлежит множеству A_2.

Из приведенных способов решения задач вытекает необходимость выполнения операции сложения. Для реализации этой операции рассмотрим реализацию сумматора на основе АЛС, причем в отличие от структуры порогового устройства [12] в этом случае используется структура АЛС трапецеидального типа [3], т.е. структура, имеющая $2n+1$ вход и α выходов ($\alpha = \log_2 n + 1$). Сумматор реализует следующую операцию [13]:

$$C = A + B + c_{\text{int}} = \sum_{\lambda=0}^{n-1} 2^{\lambda} a_{\lambda} + \sum_{\lambda=0}^{n-1} 2^{\lambda} b_{\lambda} + c_{\text{int}},$$

где a_{λ}, b_{λ} — компоненты, которые содержат значение λ-го бита двоичного

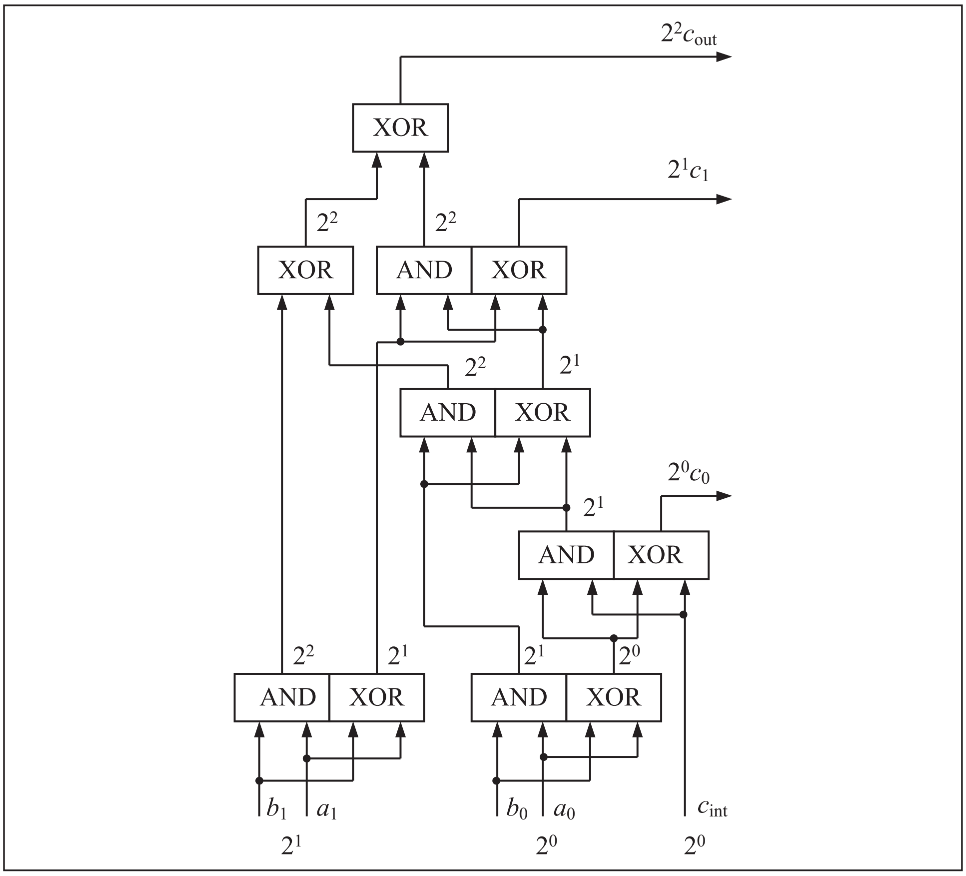

Рис. 2. Структурная схема сумматора на основе базовых логических операций XOR и AND

представления двоичных векторов A и B соответственно; 2^{λ} — вес λ-го бита двоичных векторов; c_{int} — значения входа переноса. Результат сумматора представлен $(n+1)$ битом (с учетом выхода переноса c_{out}).

Основой реализации сумматора является битуровневая (h — число уровней обработки, $s = 1 \div h$) схема преобразований, использующая сложение и умножение по модулю 2 (логические операции XOR и AND) как базовые битовые операции.

Преобразования выполняются попарной обработкой входных векторов. Обработка групп взвешенных компонент векторов на каждом уровне предполагает формирование информационных битов с текущим весом (операция сложения по модулю 2) и битов переноса, вес которых на единицу больше (операция умножения по модулю 2). Таким образом, на уровне $s = 1$ формируются группы с весом 2^1 и 2^0. Группа с весом 2^1 представляет бит переноса, сформированный с помощью операции AND. Группа с весом 2^0 — это информационный бит, сформированный с помощью операции XOR, и т.д. Группа с весом 2^{γ} обрабатывается до тех пор, пока мощность взвешенной группы ни станет равной единице, т.е. ни будет представляться одним информационным битом. Последним будет сформирован информационный бит с наибольшим весом 2^{γ} $(\gamma = n)$.

Рассмотрим пример синтеза сумматора для двоичных векторов A и B $(n = 1)$ на основе базовых логических операций XOR и AND, структурная схема которого представлена на рис. 2.

Пример 4. Пусть векторы из множества V такие, как в примере 2, параметр $k = 8$, пороговое отношение $R = \leq$, пороговый вектор $a = (7, 6, 3)$. Тогда вектор $a + K = (7, 6, 3) + (8, 8, 8) = (15, 14, 11)$ будет пороговым, а векторы v_i разбиваются структурой АЛС (см. рис. 1) на два подмножества: $V_1 = \{v_2\}$, $V_2 = \{v_1, v_3\}$.

ЗАКЛЮЧЕНИЕ

В настоящей статье рассмотрено решение задачи классификации множества векторов с целыми неотрицательными координатами с помощью адаптируемых логических сетей. Приведенное решение не зависит от разрядности координат векторов и применимо к произвольным величинам координат векторов и пороговым значениям. Данное решение требует выполнения операции суммирования, и с этой целью построен сумматор, использующий только операции AND и XOR, на базе которых строится АЛС для решения рассматриваемых задач классификации. Приведенное решение применимо и для задачи классификации множества векторов с целочисленными отрицательными координатами. Для этого необходимо выполнить смещение, в результате которого отрицательные величины переводятся в неотрицательные, и в конце преобразований выполняется обратное смещение для получения истинных значений.

СПИСОК ЛИТЕРАТУРЫ

1. Opanasenko V.N., Kryvyi S.L. Partitioning the full range of boolean functions based on the threshold and threshold relation. *Cybernetics and Systems Analysis*. 2012. Vol. 48, N 3. P. 459–468.
2. Palagin A.V., Opanasenko V.N. Reconfigurable computing technology. *Cybernetics and Systems Analysis.* 2007. Vol. 43, N 5. P. 675–686.
3. Palagin A.V., Opanasenko V.N. Design and application of the PLD-based reconfigurable devices. *Design of Digital Systems and Devices. Lecture Notes in Electrical Engineering*. Adamski M., Barkalov A., Wegrzyn M. (Eds.). Berlin; Heidelberg: Springer-Verlag, 2011. Vol. 79. P. 59–91.
4. Opanasenko V.N., Kryvyi S.L. Synthesis of adaptive logical networks on the basis of Zhegalkin polynomials. *Cybernetics and Systems Analysis.* 2015. Vol. 51, N 6. P. 969–977.
5. Palagin A., Opanasenko V., Kryvyi S. The structure of FPGA-based cyclic-code converters. *Optical Memory & Neural Networks (Information Optics).* 2013. Vol. 22, N 4. P. 207–216.
6. Opanasenko V., Kryvyi S. Synthesis of multilevel structures with multiple outputs. *CEUR Workshop Proceeding of 10th International Conference of Programming, UkrPROG 2016*. Kyiv, Ukraine. 2016. Vol. 1631, Code 122904. P. 32–37.
7. Kondratenko Y.P., Gordienko E. Implementation of the neural networks for adaptive control system on FPGA. *Annals of DAAAM for 2012 & Proceeding of 23rd DAAAM International Symposium on Intelligent Manufacturing and Automation:* Katalinic B. (Ed.). Vienna, Austria: DAAAM International, 2012. 23 (1). P. 0389–0392.
8. Kondratenko Y.P., Sidenko Ie.V. Decision-making based on fuzzy estimation of quality level for cargo delivery. *Recent Developments and New Directions in Soft Computing. Studies in Fuzziness and Soft Computing.* Zadeh L.A. et al. (Eds.). Springer International Publishing Switzerland, 2014. P. 331–344.
9. Кривонос Ю.Г., Крак Ю.В., Кириченко М.Ф. Моделювання, аналіз і синтез маніпуляційних систем. Київ: Наук. думка, 2006. 202 с.
10. Стасюк О.І., Гончарова Л.Л. Математичні моделі комп'ютерної інтелектуалізації технологій синхронних векторних вимірювань параметрів електричних мереж. *Кибернетика и системный анализ*. 2016. Т. 52, № 5. С 186–192.
11. Брейтон Р.К., Хэтчел Г.Д., Санджованни-Винчентелли А.Л. Синтез многоуровневых комбинационных логических схем. *ТИИЭР*. 1990. Т. 78, № 2. С. 38–83.
12. Palagin A.V., Opanasenko V.N., Chigirik L.G. Synthesizing a Hamming adder of arbitrary word width. *Journal of Automation and Information Sciences.* 1995. Vol. 27, N 2. P. 39–42.

13. Drozd J., Drozd A., Antoshchuk S., Kushnerov A., Nikul V. Effectiveness of matrix and pipeline FPGA-based arithmetic components of safety-related systems. *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*: *Technology and Applications*. Warsaw, Poland. P. 785–789.

Надійшла до редакції 29.05.2017

С.Л. Кривий, В.М. Опанасенко

РОЗБИТТЯ МНОЖИНИ ВЕКТОРІВ З ЦІЛИМИ НЕВІД'ЄМНИМИ КООРДИНАТАМИ З ВИКОРИСТАННЯМ ЛОГІКОВИХ АПАРАТНИХ ЗАСОБІВ

Анотація. Розглянуто задачі розбиття множини векторів з цілими невід'ємними координатами відносно порогового значення і порогового відношення на два класи засобами логікових мереж зі здатністю адаптації. Обґрунтовано коректність алгоритмів реалізації такого розбиття для довільного порогового значення і розмірності векторів.

Ключові слова: булева функція, класифікація, цілочислові вектори, порогові відношення.

S.L. Kryvyi, V.M. Opanasenko

PARTITIONING OF A SET OF VECTORS WITH NONNEGATIVE INTEGER COORDINATES BY MEANS OF LOGICAL HARDWARE

Abstract. The problems of partition of a set of vectors with nonnegative items into two classes with respect to threshold values by using adaptive logical systems is considered. The correctness of the corresponding algorithms for implementation of such partition is proved for an arbitrary threshold value and vector dimension.

Keywords: Boolean function, classification, integer vectors, threshold value.

Крывый Сергей Лукьянович,
доктор физ.-мат. наук, профессор, профессор кафедры Киевского национального университета имени Тараса Шевченко, e-mail: sl.krivoi@gmail.com.

Опанасенко Владимир Николаевич,
доктор техн. наук, профессор, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Киев, e-mail: opanasenkovm@nas.gov.ua.