
СИСТЕМИ ТЕХНІЧНОГО ЗОРУ І ШТУЧНОГО ІНТЕЛЕКТУ З ОБРОБКОЮ ТА РОЗПІЗНАВАННЯМ ЗОБРАЖЕНЬ

УДК 004.032.26

В. П. КОЖЕМЯКО¹, Т. Б. МАРТЫНЮК¹, Л. М. КУПЕРШТЕЙН²

МОДЕЛЬ "БЫСТРОГО НЕЙРОНА" С ОБРАБОТКОЙ ДАННЫХ ПО ПРИНЦИПУ РАЗНОСТНЫХ СРЕЗОВ

¹*Винницкий национальный технический университет,
Хмельницкое шоссе, 95, г. Винница, Украина*

²*Винницкий финансово-экономический университет,
Пирогова, 71А, г. Винница, Украина,
E-mail: kuperok@mail.ru*

Аннотация. В статье исследованы способы пороговой параллельной обработки векторных массивов данных с использованием принципа разностных срезов для модели "быстрого нейрона". Выполнен их сравнительный анализ по временным параметрам

Ключевые слова: нейрон, разностный срез, нейрочип.

Анотація. У статті досліджено способи порогового паралельного оброблення векторних масивів даних з використанням принципу різницевого зрізу для моделі "швидкого нейрона". Виконано їх порівняльний аналіз за часовими параметрами

Ключові слова: нейрон, різницевий зріз, нейрочип.

Abstract. In the article the methods of the threshold parallel processing of vector data arrays on a base of difference cuts principal for the model of "rapid neuron" are researched. Their comparative analysis on temporal parameters is executed.

Keyword: neuron, difference cut, neuralchip.

ВВЕДЕНИЕ

В последнее время количество публикаций по тематике, связанной с созданием, исследованием и применением нейронных сетей и нейрокомпьютеров, растет лавинообразно. В основном все публикации так или иначе касаются проблем усовершенствования, модернизации или развития известных типов и моделей нейросетей [1-6], при этом формальный нейрон (ФН) воспринимается в виде его классической модели, основу которой составляет последовательное выполнение сумматорной и активаторной (активационной) функций вида [2, 3, 7]:

$$S = \sum_{i=1}^n w_i \cdot x_i, \quad (1)$$

$$Y = f(S), \quad (2)$$

где x_i – i -й входной сигнал; w_i – i -й весовой коэффициент; S – сумматорная функция; $f(S)$ – активационная функция; Y – выходной сигнал нейрона.

Если для программной реализации такая модель является единственно возможной, то при аппаратной реализации она вносит большие временные задержки. При всем разнообразии имеющихся функций активации [1-3, 7] для ускорения реализации функции (1) единственно используется многоходовое суммирующее устройство в виде древовидной (пирамидальной) схемы сумматоров. Это позволяет уменьшить временные затраты при суммировании n операндов в $\frac{n}{\log_2 n}$ раз [8]. При этом

вопрос о совмещении выполнения обеих функций (1) и (2) остается открытым. Не в последнюю очередь это связано с классическим методом суммирования чисел. Отсюда напрашивается вывод: если мы хотим не только ускорить выполнение функции вида (1), но и вычислять одновременно обе функции (1) и (2), то необходимо пересмотреть метод выполнения функции (1).

ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

В работе [9] рассмотрены все возможные способы вычисления оператора группового суммирования (ГС) вида [10]:

$$X_n = \sum_{i=1}^n a_i, \quad (3)$$

который используется для реализации функции (1) при условии, что $a_i = w_i \cdot x_i$. Анализ эффективности известных способов вычисления этого оператора [10] показал, что оптимальным является способ, известный как алгоритм рекурсивного сдваивания [11], который и реализуется в виде упомянутой древовидной структуры. В то же время известен метод параллельного суммирования массива чисел по принципу разностных срезов (ПРС) [9, 12]. Впервые этот подход был описан в [13, 14] как параллельный способ сложения группы длительностей временных интервалов. Дальнейшие исследования показали универсальность такого подхода, в результате чего он был использован для параллельной обработки векторных массивов числовых данных, какими являются, по сути, взвешенные входы нейронов [15].

ПРС дает возможность создать модель формального нейрона – „быстрого нейрона”, в которой реализуется распараллеливание процесса суммирования, а главное, суммирование совмещено с вычислением пороговой функции активации нейрона. Это намного ускоряет процесс обработки, поскольку момент превышения суммы над порогом можно зафиксировать, не формируя конечную взвешенную сумму, в противоположность классической модели [16]. Выход Y такого нейрона, являясь функцией знака вида:

$$Y = f(S) = \text{sign}(\theta - S), \quad (4)$$

где θ - внешний порог обработки, принимает следующие значения:

$$Y = \begin{cases} 1, & \text{если } S = \sum_{i=1}^n a_i \geq \theta, \\ 0 & \text{в противоположном случае,} \end{cases} \quad (5)$$

т.е. реализует простейшую пороговую функцию – функцию единичного скачка [4, 7]. Это, в свою очередь, не уменьшает практической ценности модели, поскольку с использованием таких ФН эффективно решаются задачи идентификации динамических систем и адаптивное управление ними, классификация образов, распознавание изображений, обработка двоичных сигналов [4, 17-22].

Рассмотрим подробнее способы пороговой параллельной обработки по разностным срезам (РС), которые являются базовыми для модели «быстрого нейрона».

СПОСОБЫ ПОРОГОВОЙ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ВЕКТОРНЫХ ДАННЫХ ПО РС

Базовыми для обработки данных по РС являются понятия разностного среза A_j , текущей частичной суммы S_j элементов векторного массива и текущего внутреннего порога обработки q_j , формируемых в j -м цикле, где $j=1, \overline{N}$, N количество циклов [9, 16].

В случае, если в качестве внутреннего порога q_j выбирается минимальный элемент текущего массива данных A_{j-1} , т.е.

$$q_j = \min_i A_{j-1} = \min_i \{a_{i,j-1}\}_1^n, \quad (6)$$

то РС A_j и текущая частичная сумма S_j следующего j -го цикла обработки формируются следующим образом [9, 16]:

$$A_j = \{a_{i,j}\} = \{a_{i,j-1} - q_j\}_1^n, \quad (7)$$

$$S_j = q_j \cdot d_j, \quad (8)$$

где d_j – количество положительных элементов $a_{i,j}$ в РС A_j .

Вариантность пороговой параллельной обработки по РС зависит от способа формирования текущего внешнего порога θ обработки, т.е. от способа сравнения либо накопленной суммы S , либо текущей частичной суммы S_j , либо текущего внутреннего порога q_j с внешним порогом θ обработки с целью определения результирующего сигнала Y (4).

В основе первого варианта порогового параллельного суммирования элементов векторного массива данных (целых неотрицательных чисел) лежит параллельное формирование и накопление текущих частичных сумм этих элементов с постепенным уменьшением количества значащих элементов к моменту получения нулевого остатка, а также сравнение окончательной суммы с заданным внешним порогом [23].

В математической модели предложенного способа пороговой параллельной обработки сначала накапливается сумма текущих частичных сумм вида

$$\sum_{i=1}^n a_i = S = S_1 + S_2 + \dots + S_N = \sum_{j=1}^N S_j, \quad (9)$$

а на последнем этапе проверяется условие

$$\Delta = \theta - S = \theta - (S_1 + S_2 + \dots + S_N) \leq 0. \quad (10)$$

Рассмотрим пример такого порогового параллельного суммирования элементов векторного массива данных при $n=5$ (рис. 1). Суммируются элементы массива $\{11, 3, 5, 8, 15\}$, а значение порога θ равно 28.

На рис. 1 показаны следующие временные характеристики:

$$\begin{cases} T_{\min}^j = t_{\text{com}} \cdot (n-1) = \text{const}, j=\overline{1,N}, \\ T_{\text{SM}}^j = t_{\text{sm}} \cdot d_j \neq \text{const}, j=\overline{1,N}, \\ T_1 = T_{\min} \cdot 6 + t_{\text{sm}} \cdot 4 + t_{\text{sm}} + t_{\text{sub}} = 6 \cdot T_{\min} + 6 \cdot t_{\text{sm}}, \end{cases} \quad (11)$$

$$t_{\text{sub}} = t_{\text{sm}} = \text{const}, \quad (12)$$

где T_{\min}^j – время определения минимального элемента массива в j -м цикле; t_{com} – время сравнения двух чисел; T_{SM}^j – время формирования текущей частичной суммы в j -м цикле; t_{sm} – время суммирования двух чисел; t_{sub} – время вычитания двух чисел для формирования РС, а также разности Δ ; T_1 – общее время обработки для 1-го варианта; n – количество элементов в массиве; N – количество циклов обработки.

В общем виде максимальное и минимальное значение времени обработки для первого варианта можно определить таким образом:

$$T_1^{\max} = (n-1)t_{\text{com}} \cdot N + t_{\text{sub}}(N-1) + t_{\text{sub}}, \quad (13)$$

$$T_1^{\min} = 2T_{\min} + t_{\text{sm}} = 2t_{\text{com}}(n-1) + t_{\text{sm}} + t_{\text{sub}}, \quad (14)$$

где $N=n+1$ при условии, что в векторном массиве нет одинаковых и нулевых элементов, а время выполнения одного цикла всегда постоянно. С учетом того, что

$$t_{\text{com}} \approx t_{\text{sm}} = t_{\text{sub}} = t, \quad (15)$$

получим

$$T_1^{\max} = n \cdot N \cdot t = n^2 \cdot t + n \cdot t = (n^2 + n) \cdot t, \quad (16)$$

$$T_1^{\min} = 2n \cdot t, \quad (17)$$

а общее время для приведенного примера обработки соответственно равно $T_1 = 30t$ (рис.1).

Вторым вариантом порогового параллельного суммирования элементов векторного массива данных является способ, в котором в каждом цикле обработки выполняется формирование текущей частичной суммы элементов, сравнение ее с текущим порогом, а также формирование порога для следующего цикла обработки путем вычитания текущей частичной суммы из текущего порога и формирование результирующего сигнала. Это, в свою очередь, позволяет завершить процесс обработки при наличии единичного результирующего сигнала без формирования конечной суммы элементов массива, что приводит к ускорению параллельной пороговой обработки [24].

Математическую модель второго способа порогового параллельного суммирования можно представить следующим образом. Выражение (10) можно переписать в виде

$$\Delta = (\dots((\theta - S_1) - S_2) - \dots - S_N). \quad (18)$$

Тогда в каждом j -м цикле обработки необходимо проверить правильность условия

$$\Delta_j = \Delta_{j-1} - S_j, \quad j = \overline{1, N}, \quad (19)$$

где $\Delta_0 = \theta$.

Рассмотрим пример (рис. 2) для второго варианта порогового параллельного суммирования элементов на одном и том же векторном массиве и с тем самым порогом θ , что и в первом варианте. На рис. 2 приняты такие же обозначения, что и на рис. 1, лишь t_{sub} – это время формирования порога для следующего цикла обработки, а также проверка его на отрицательность или равенство нулю.

Для второго варианта порогового параллельного суммирования общее, максимальное и минимальное время обработки соответственно составляют:

$$T_2 = T_{\min} \cdot 3 + t_{\text{sm}} \cdot 4 + t_{\text{sub}} = 3 \cdot T_{\min} + 5 \cdot t_{\text{sm}},$$

$$T_2^{\max} = (n-1)t_{\text{com}} + \sum_{j=1}^N (t_{\text{sm}}(n-j) + t_{\text{sub}}), \quad (20)$$

$$T_2^{\min} = 2T_{\min} + t_{\text{sm}} = 2t_{\text{com}}(n-1) + t_{\text{sm}},$$

а при условии (15)

$$T_2^{\max} = -0.5N^2 \cdot t + 0.5N \cdot t + N \cdot n \cdot t + n \cdot t - t, \quad (21)$$

$$T_2^{\min} = 2n \cdot t - t, \quad (22)$$

и общее время обработки соответственно равно $T_2 = 17t$ (рис. 2).

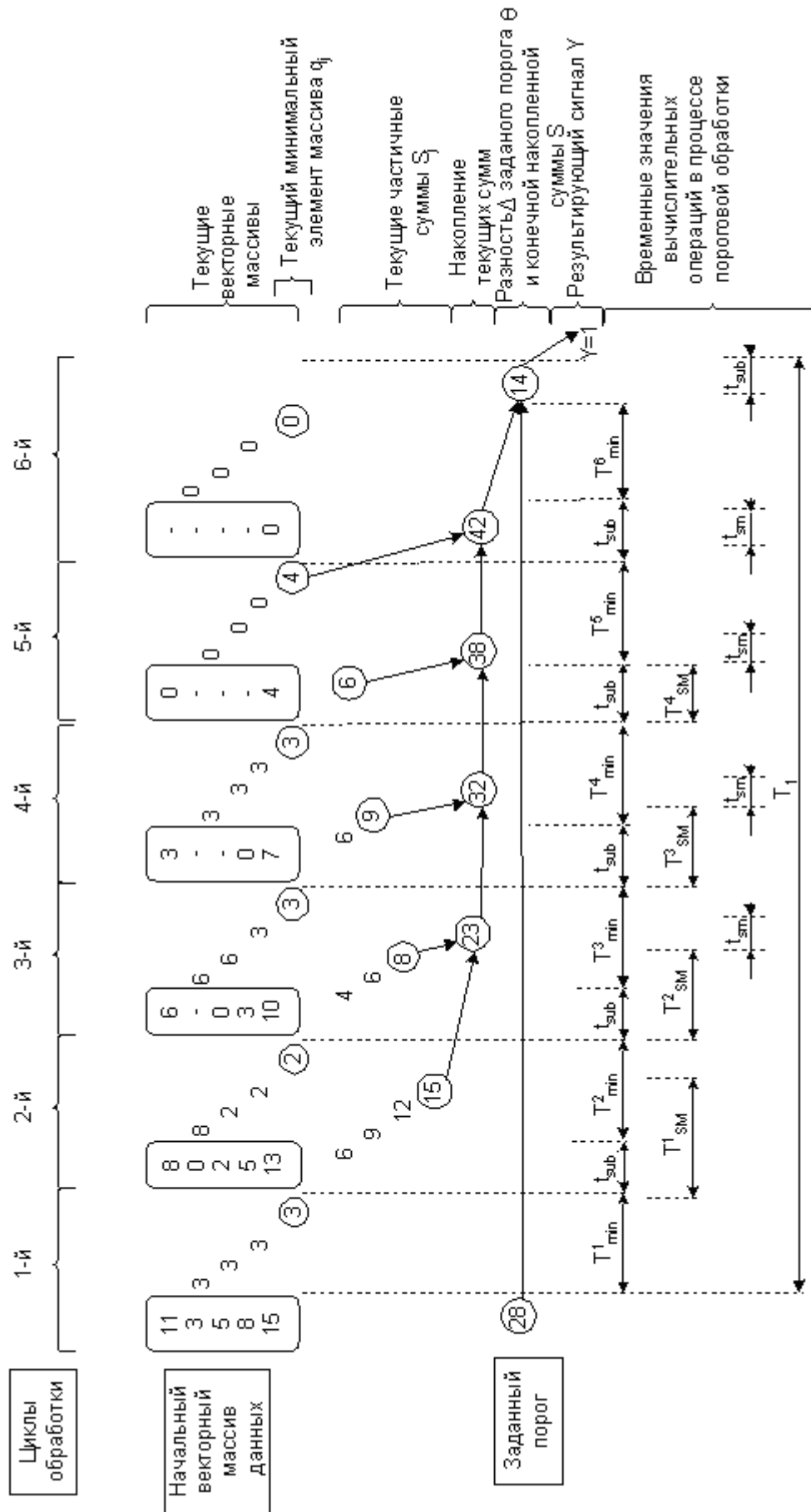


Рис. 1. Первый вариант порогового параллельного суммирования элементов векторного массива данных

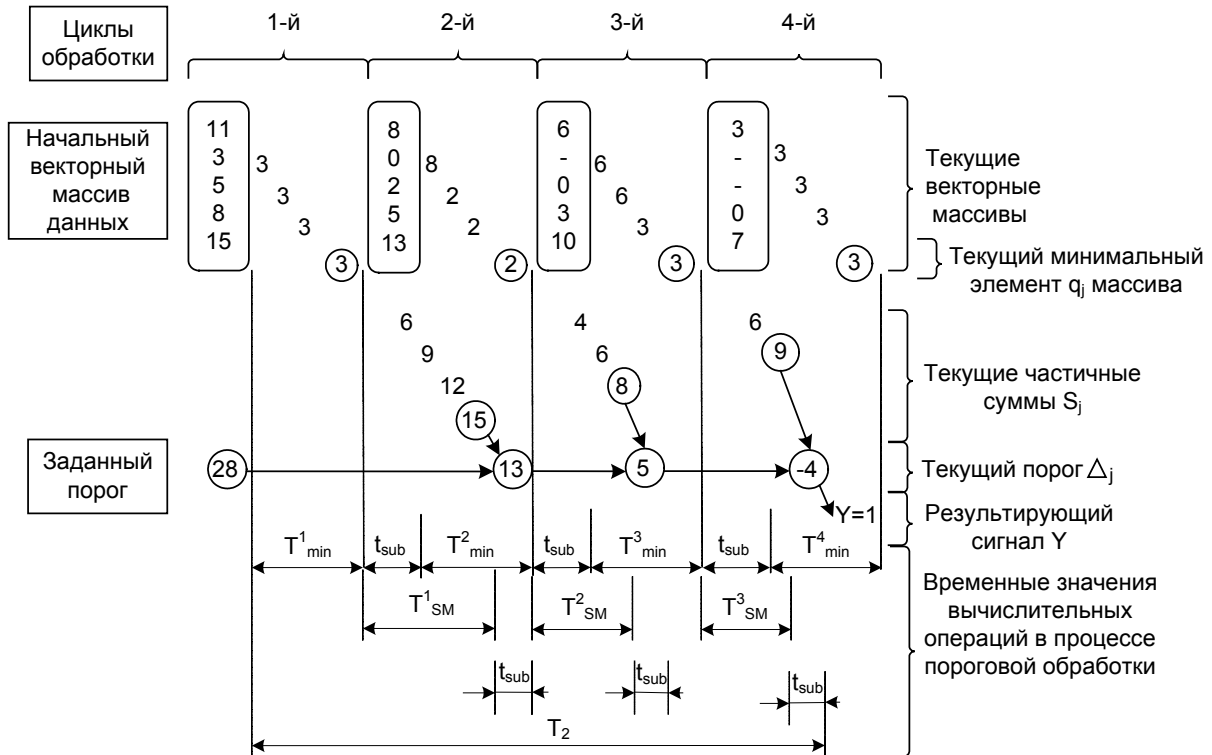


Рис.2. Второй вариант порогового параллельного суммирования элементов векторного массива данных

Третьим вариантом порогового параллельного суммирования элементов векторного массива данных является способ, в котором в каждом цикле обработки выполняется формирование текущих частичных разностей, а также формирование текущего порога для следующего цикла обработки путем последовательного вычитания из текущего порога текущего минимального элемента такое количество раз, которое равно количеству положительных ненулевых элементов в текущем векторном массиве, и формирование результирующего сигнала, что приводит к ускорению способа за счет сокращения времени определения отрицательной текущей частичной разности [25].

Математическую модель предложенного способа порогового параллельного суммирования можно представить таким образом: если сумму S элементов первоначального массива можно представить в виде (9), то величину $\Delta = \theta - S$ можно определить таким образом:

$$\Delta = \theta - S = \theta - (S_1 + S_2 + \dots + S_N) = \theta - S_1 - S_2 - \dots - S_N = (\dots ((\theta - S_1) - S_2) - \dots - S_N) \quad (23)$$

Используем обозначение для текущего порога вида (19). Тогда выражение (23) можно записать так:

$$\Delta = [\dots [[\theta - \underbrace{(q_1 + q_1 + \dots + q_1)}_{d_1}] - \underbrace{(q_2 + q_2 + \dots + q_2)}_{d_2}] - \dots - \underbrace{(q_N + q_N + \dots + q_N)}_{d_N}] =$$

$$= \left[\left[\left[\left[\dots (\theta - q_1) - q_1 \right] - \dots - q_1 \right] - q_2 - q_2 - \dots - q_2 \right] - \dots - q_N - q_N - \dots - q_N \right] \quad (24)$$

или

$$\Delta = [\dots [[(\dots ((\theta - q_1^{(1)})) - q_1^{(2)}) - \dots - q_1^{(d_1)}] - q_2^{(1)}] - q_2^{(2)}] - \dots - q_2^{(d_2)}] - \dots - q_N^{(1)} - q_N^{(2)}] - \dots - q_N^{(d_N)}]. \quad (25)$$

Если выражение в круглых скобках обозначить как

$$\delta_j^{(k)} = \delta_j^{(k-1)} - q_j^{(k)}, \quad k = \overline{1, d_j}, \quad \delta_1^0 = \theta, \quad (26)$$

то текущий порог (19) будет иметь следующий вид

$$\Delta_j = \delta_{j-1}^{(d_j)} \leq 0, \quad \delta_j^0 = \Delta_{j-1}. \quad (27)$$

Итак, в каждом цикле обработки можно проверять выполнение условия $\delta_j^{(k)} \leq 0$, а не только $\Delta_j \leq 0$, что позволяет ускорить процесс пороговой обработки.

Таким образом, на каждом шаге обработки, кроме первого, выполняется сравнение с заданным порогом, который задается в начале процесса суммирования. При этом, если значение текущей частичной разности между текущим порогом и текущим минимальным элементом массива отрицательное или равно нулю, то процесс обработки прекращается. Начиная со второго шага, текущее значение порога будет принимать значение последней текущей разности между текущим порогом и текущим минимальным элементом, полученными на предыдущем шаге.

Рассмотрим третий вариант порогового параллельного суммирования элементов (рис. 3) на том же векторном массиве и с тем самым порогом θ , что и в первом и втором вариантах. На рис. 3 принятые те же обозначения, что и на рис. 1, лишь $T_{SUB}^j = t_{sub} \cdot d_j \neq const$ – это время формирования текущих частичных разностей.

Общее, максимальное и минимальное время обработки для третьего варианта имеет вид:

$$T_3 = T_{min} \cdot 3 + t_{sub} \cdot 4 = 3 \cdot T_{min} + 4 \cdot t_{sm}, \quad (28)$$

$$T_3^{max} = (n-1)t_{com} + \sum_{j=1}^N t_{sm}((n+1)-j), \quad (29)$$

$$T_3^{min} = T_{min} + t_{sub} = t_{com}(n-1) + t_{sm}, \quad (30)$$

а при условии (15)

$$T_3^{max} = -0.5N^2 \cdot t + 0.5N \cdot t + N \cdot n \cdot t + n \cdot t - t, \quad (31)$$

$$T_3^{min} = n \cdot t, \quad (32)$$

а общее время обработки соответственно равно $T_3 = 16t$ (рис. 3).

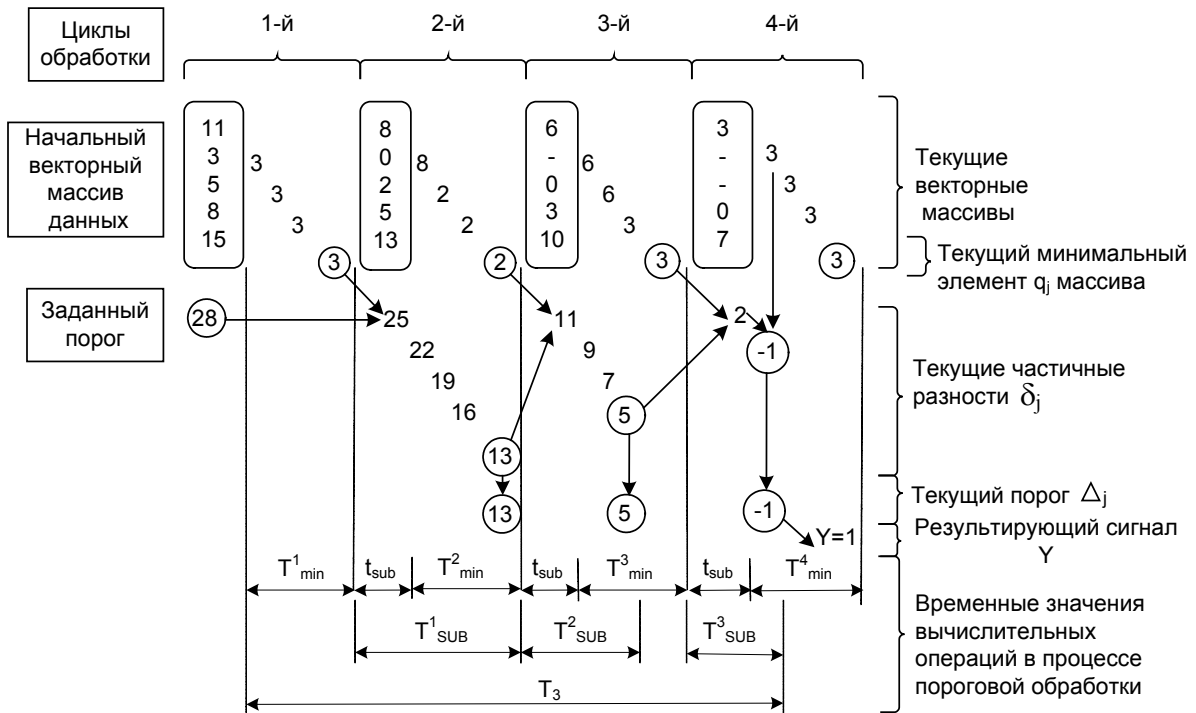


Рис. 3. Третий вариант порогового параллельного суммирования элементов векторного массива данных

Предложенные способы пороговой параллельной обработки являются более быстродействующими по сравнению с аналогом [26] (рис. 4).

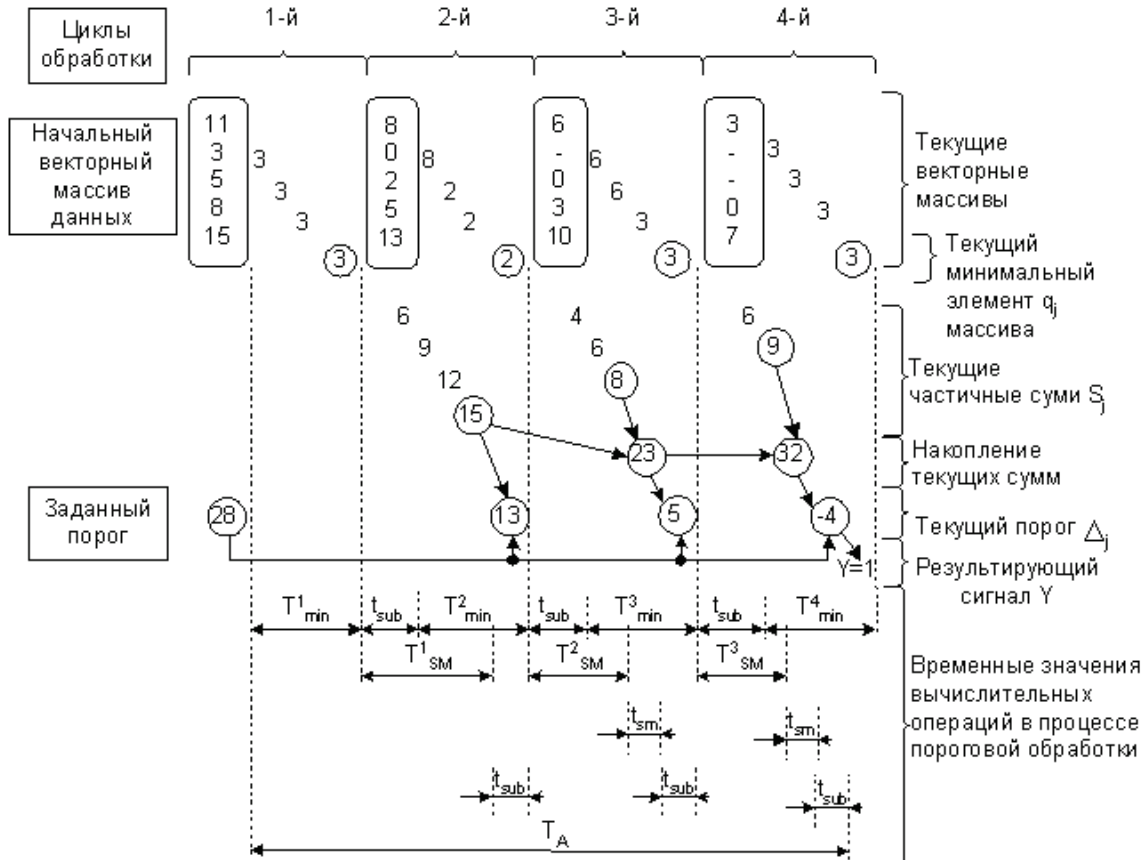


Рис. 4. Аналог порогового параллельного суммирования по ПРС

Чтобы это доказать, необходимо рассмотреть зависимость максимального времени обработки от размерности первоначального векторного массива для всех способов и сравнить их. На рис. 4 приняты такие же обозначения, что и на рис. 1. Суммируются элементы, которые составляют массив $\{11, 3, 5, 8, 15\}$, а значение порога равно 28.

Как видно из рис. 4, процесс обработки прекращается уже на четвертом цикле, поскольку зафиксировано превышения накопленной текущей частичной суммы над порогом.

Выражения для вычисления общего, максимального и минимального времени пороговой обработки аналога имеют вид:

$$T_{\dot{A}} = T_{\min} \cdot 3 + t_{\text{sm}} \cdot 4 + t_{\text{sm}} + t_{\text{sub}} = 3 \cdot T_{\min} + 6 \cdot t_{\text{sm}}, \quad (33)$$

$$T_{\dot{A}}^{\max} = (n-1)t_{\text{com}} + \sum_{i=1}^N (t_{\text{sm}}(n-i) + t_{\text{sub}}) + t_{\text{sub}}(N-1), \quad (34)$$

$$T_{\dot{A}}^{\min} = 2T_{\min} + t_{\text{sm}} = 2t_{\text{com}}(n-1) + t_{\text{sm}}. \quad (35)$$

Учитывая выражение (15), получим:

$$T_{\dot{A}}^{\max} = -0.5N^2 \cdot t + 1.5N \cdot t + N \cdot n \cdot t + n \cdot t - t, \quad (36)$$

$$T_{\dot{A}}^{\min} = 2t \cdot n - t, \quad (37)$$

а общее время обработки соответственно равно $T_{\dot{A}} = 18t$ (рис. 4).

Итак, проанализируем полученные результаты с точки зрения длительности обработки (максимальное и минимальное время) всеми способами.

АНАЛИЗ СПОСОБОВ ПОРОГОВОЙ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ВЕКТОРНЫХ ДАННЫХ

Анализируя вышеприведенные способы, можно утверждать, что третий вариант порогового параллельного суммирования является наиболее быстродействующим. Это достигается за счет отказа от накопления текущих частичных сумм элементов векторного массива (первый вариант) и вычисления текущих частичных величин на каждом шаге обработки (второй вариант).

Подход, который предложен во всех вариантах, основан на использовании ПРС и обеспечивает максимальный параллелизм обработки массива чисел, поскольку в процессе задействованы одновременно все операнды, а это, в свою очередь, приводит к ускорению процесса обработки, поскольку в случае превышения суммы над порогом этот факт можно зафиксировать без формирования окончательного значения суммы всех чисел массива. Процесс порогового параллельного суммирования элементов векторного массива данных во втором варианте по сравнению с первым более ускоренный за счет отказа от формирования накопленных текущих частичных сумм и совмещения суммирования элементов массива с их пороговой обработкой. В свою очередь, третий вариант более быстродействующий по сравнению со вторым за счет последовательного формирования текущих частичных разностей на каждом шаге обработки и проверки их на неотрицательность. Тем не менее проверка на выполнение условия (19) для текущих порогов, которая также реализована во втором варианте, приводит к сокращению времени определения отрицательной текущей частичной разности.

Анализируя временные зависимости рассмотренных способов порогового параллельного суммирования, необходимо отметить, что общей и значительной частью продолжительности обработки по РС является время выделения минимального элемента в векторном массиве, которое принимается постоянным в каждом цикле обработки. Но это время можно рассматривать как “плавающую” величину, которая определяется как $t_{\text{com}} \cdot d_j$. Тогда аналитические зависимости максимального времени пороговой обработки от размерности векторного массива при переменной длительности цикла обработки для трех способов и аналога примут вид:

$$T_1^{*max} = t_{com} \sum_{j=1}^N (n-j) + N \cdot t_{sub} + t_{sub} \quad (38)$$

$$T_2^{*max} = T_3^{*max} = T_A^{*max} = t_{com} \sum_{j=1}^N (n-j) + N \cdot t_{sub} \quad (39)$$

Графики временных зависимостей пороговой параллельной обработки по РС от размерности n массива показаны на рис. 5.

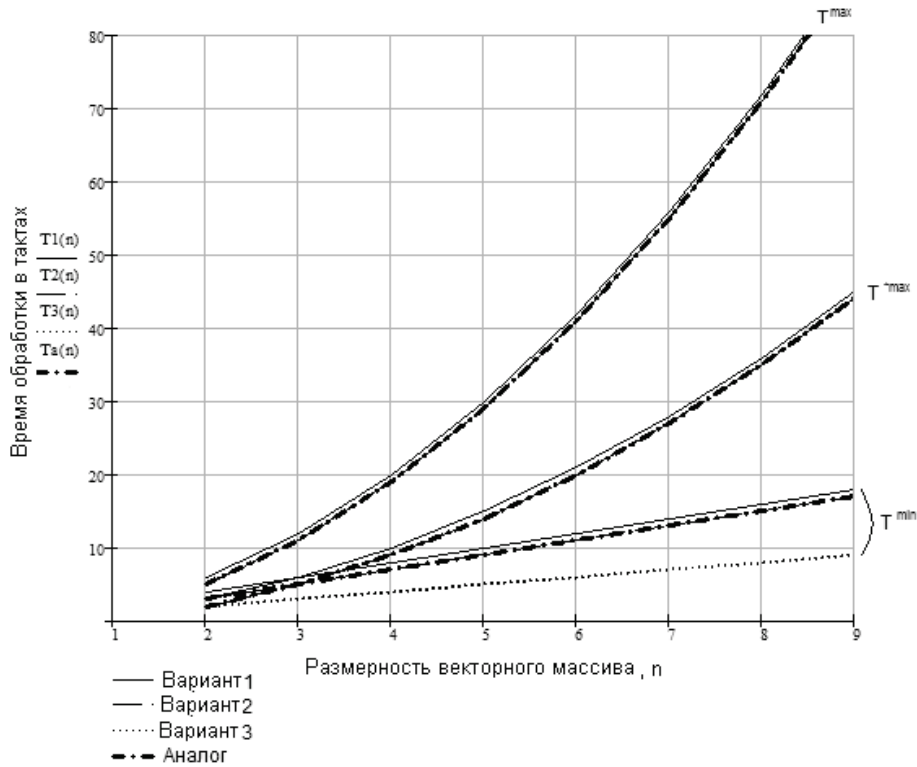


Рис. 5. Временные зависимости пороговой параллельной обработки от размерности n векторного массива

Анализ этих зависимостей подтверждает теоретические предположения о том, что на продолжительность пороговой обработки в общем случае влияют размерность входного векторного массива, характер распределения элементов в массиве, величина порога и принцип формирования его текущего значения. А это, в свою очередь, свидетельствует о непостоянном времени пороговой обработки по РС в отличие от классической модели ФН, поскольку процесс обработки может закончиться в любом цикле, начиная со второго, в случае выполнения условий (19), (27). На рис. 5 приняты обозначения: T^{max} — максимальное время пороговой обработки с постоянной длительностью цикла обработки; T^{*max} — максимальное время пороговой обработки с переменной длительностью цикла обработки, которое объясняется постепенным уменьшением в каждом следующем цикле количества значащих элементов в текущему РС; T^{min} — минимальное время пороговой обработки.

Кроме того, анализируя аналитические и графические временные зависимости пороговой параллельной обработки по РС, видно, что максимальное значение времени обработки у всех вариантов на один такт меньше, чем у первого. Таким образом, налицо повышение быстродействия как минимум на 12% (в 1,14 раза) при условии выполнения всех циклов обработки до момента формирования нулевого остатка в РС. Кроме того, минимальное время обработки составляет для аналога и предложенного способа $(2n-t-t)$ и $2n-t$ соответственно. Таким образом, достигнуто повышение быстродействия максимум в $(2n-t-t)/(2n-t) = 1,9$ раза (на 47,4 %) [28].

Таким образом, при реализации нейронных сетей на базе нейронов с предложенными вариантами обработки данных повышается скорость вычислений и расширяется класс решаемых задач. Кроме того, особенности принципа разностных срезов и моделей ФН, построенных на его основе,

позволяют получить отсортированный входной векторный массив данных, а также реализовать его восстановление [9, 27], что значительно расширяет функциональные возможности моделей.

ВЫВОДЫ

1. Предложенный первый способ является усовершенствованием метода параллельного суммирования элементов векторного массива данных по РС. Он отличается от известного параллельным формированием и накоплением текущих частичных сумм элементов векторного массива данных с постепенным уменьшением количества первоначальных операндов к моменту получения нулевого остатка, а также сравнением окончательной суммы с заданным порогом. Это позволяет расширить функциональные возможности метода за счет возможности получения отсортированных элементов векторного массива данных.

2. Предложены второй и третий способы порогового параллельного суммирования элементов векторного массива данных по РС. Один из них включает формирование текущей частичной суммы элементов векторного массива, сравнение ее с текущим порогом, а также формирование текущего порога для следующего цикла обработки путем вычитания текущей частичной суммы от текущего порога. Это позволяет завершить процесс обработки при наличии единичного результирующего сигнала без формирования конечной суммы элементов массива, что приводит к ускорению параллельной пороговой обработки. Другой способ отличается формированием текущего порога для следующего шага обработки путем последовательного вычитания от предыдущего текущего порога текущего минимального элемента определенное количество раз, которое соответствует количеству положительных ненулевых элементов в текущем векторном массиве. Это позволяет завершить процесс обработки при наличии единичного результирующего сигнала без формирования текущего порога на каждом шаге обработки. Такая обработка ускоряет пороговое параллельное суммирование за счет сокращения времени определения отрицательной текущей частичной разности. Полученное ускорение обработки этими способами составляет от 12% до 47%.

3. Проведен анализ предложенных способов пороговой параллельной обработки по РС. Анализ показал, что длительность каждого следующего цикла обработки, как и сам процесс обработки, может быть как постоянной, так и переменной величиной в отличие от классической модели. Поэтому процесс обработки может закончиться в любом цикле и даже на любом шаге обработки (третий вариант), что приводит к повышению быстродействия (уменьшению максимального и минимального времени обработки) данной модели ФН.

СПИСОК ЛИТЕРАТУРЫ

1. Галушкин А.И. Нейрокомпьютеры : учеб. пособие для вузов / А.И. Галушкин. – М.: ИПРЖР, 2000. – 528 с. – ISBN 5-9310-8050-8.
2. Нейрокомпьютеры: учеб. пос. / Комарцова Л.Г., Максимов А.В. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с. – ISBN 5-7038-2554-7.
3. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая линия – Телеком, 2002. – 382 с. – ISBN 5-93517-031-0.
4. Каллан Р. Основные концепции нейронных сетей / Р. Каллан; пер. с англ. – М.: Издательский дом “Вильямс”, 2001. – 288 с. – ISBN 5-8459-0210-6.
5. Осовский С. Нейронные сети для обработки информации / С. Осовский. – М.: Финансы и статистика, 2002. – 289 с. – ISBN 5-279-02567-4.
6. Джонс М.Т. Программирование искусственного интеллекта в приложениях / М.Т. Джонс; пер. с англ. – М.: ДМК Пресс, 2004. – 312 с. – ISBN 5-94074-275-0.
7. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен; пер. с англ. – М.: Мир, 1992. – 240 с.
8. Хохлюк В.И. Параллельные алгоритмы целочисленной оптимизации / В.И. Хохлюк. – М.: Радио и связь, 1987. – 224 с.
9. Мартинюк Т.Б. Рекурсивні алгоритми багатооперандної обробки інформації / Т.Б. Мартинюк. – Вінниця: УНІВЕРСУМ-Вінниця, 2000. – 216 с. – ISBN 966-7199-98-3.
10. Справочник по цифровой вычислительной технике [под ред. В.Н. Малиновского]. – К.: Техника, 1980. – 320 с.
11. Системы параллельной обработки / пер с англ., под. ред. Д. Ивенса. – М.: Мир, 1985. – 416 с.
12. Мартинюк Т.Б. Реалізація концепції різницевих зрізів при обробленні зображень та розпізнаванні образів / Т.Б. Мартинюк, А.В. Кожем'яко // Оптико-електронні інформаційно-енергетичні

- технології. – 2001. - №1. – С. 79-85.
13. Свечников С.В. Квазиимпульсно-потенциальные оптоэлектронные элементы и устройства логико-временного типа / С.В. Свечников, В.П. Кожемяко, Л.И. Тимченко. – К. : Наукова думка, 1987. – 256 с.
 14. А.с. 1119035 СССР, МКИ⁵ G06G 7/14 Способ параллельного сложения длительностей группы временных интервалов / В.П. Кожемяко, Л.И. Тимченко, Т.В. Головань, Н.Е. Фурдияк, Т.Б. Мартынюк (СССР). – № 3528309/18; заявлено 24.12.82; опубл. 15.10.84, Бюл. № 38.
 15. Мартинюк Т.Б. Методи та засоби паралельних перетворень векторних масивів даних / Т.Б. Мартинюк, В.В. Хом'юк. – Вінниця: УНІВЕРСУМ-Вінниця, 2005. – 203 с. – ISBN 966-641-114-8.
 16. Мартынюк Т.Б. Модель порогового нейрона на основе параллельной обработки по разностным срезам / Т.Б. Мартынюк // Кибернетика и системный анализ. – 2005. - №4. – С. 78-89.
 17. Кожем'яко В.П. Паралельно-ієрархічне перетворення як системна модель оптико-електронних засобів штучного інтелекту / [ред. рада: В.П. Кожем'яко. (голова) та ін.]. – Вінниця : УНІВЕРСУМ-Вінниця, 2003. – 324 с. – ISBN 966-641-072-9.
 18. Нейронные сети в системах автоматизации / Архангельский В.И., Богаенко И.Н., Грабовский Г.Г., Рюмшин Н.А. – К. : Техника, 1999. – 364 с. – ISBN 966-575-072-0.
 19. Минаев Ю.Н. Методы и алгоритмы идентификации и прогнозирования в условиях неопределенности в нейросетевом логическом базисе / Ю.Н. Минаев, О.Ю. Филимонова, Л. Бенаумер. – М. : Горячая линия – Телеком, 2003. – 205 с. – ISBN 5-93517-129-5.
 20. Омату С. Нейроуправление и его приложения / Омату С., Халид М.; пер. с. англ. – М. : ИПРЖР, 2000. – 272 с. – ISBN 5-93108-006-6.
 21. Стариков А.А. Практическое применение нейронных сетей для задач классификации (кластеризации) : – Режим доступа до публікацій компанії BaseGroup Labs: <http://www.basegroup.ru/neural/prectice.htm>.
 22. Неймарк Ю.И. Многомерная геометрия и распознавание образов / Ю.И. Неймарк // Математика. – 1996. - №7. – С. 119-123.
 23. Куперштейн Л.М. Модель формального нейрона з використанням принципу різницевиx зрізів / Л.М. Куперштейн, А.В. Кожем'яко // Вісник Вінницького політехнічного інституту. – 2003. – №6. – С. 284-288.
 24. Патент № 70884А України, МПК⁷ G 06 G 7/14. Спосіб порогового паралельного додавання тривалостей групи часових інтервалів / Т.Б.Мартинюк, А.С. Васюра, Л.М. Куперштейн, А.В. Кожем'яко. – № 20031213278; заявлено 31.12.2003; опубл. 15.10.2004, Бюл. № 10.
 25. Патент № 70886А України, МПК⁷ G 06 G 7/14. Спосіб порогового паралельного додавання тривалостей групи часових інтервалів / Т.Б.Мартинюк, Л.М. Куперштейн. – № 2003121328; заявлено 31.12.2003; опубл. 15.10.2004, Бюл. № 10.
 26. Патент № 40228 України, МПК⁷ G 06 G 7/14. Спосіб порогового паралельного додавання тривалостей групи часових інтервалів / Кожем'яко В.П., Мартинюк Т.Б., Вербицький І.А., Мартинюк О.Б. – № 2000105973; заявлено 23.07.2000; опубл. 16.07.2001, Бюл. № 6.
 27. Мартинюк Т.Б. Особливості багатооперандної обробки векторного масиву чисел за методом різницевиx зрізів / Т.Б. Мартинюк, А.Г. Буда, Л.М. Куперштейн, К.А. Балашов // Сучасні інформаційні технології. “Динаміка наукових досліджень 2004”: III Міжнар. наук.-практ. конф., 21-30 червня 2004р.: матеріали. – Дніпропетровськ: Наука і освіта. – 2004. – С. 8–12. – ISBN 966-7191-86-9.
 28. Куперштейн Л.М. Методи та засоби нейроподібної обробки даних для систем керування: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спец. 05.13.05 „Елементи та пристрої обчислювальної техніки та систем керування” / Л.М. Куперштейн. – Вінниця, 2007. – 20 с.

Надійшла до редакції 21.05.2009р.

КОЖЕМ'ЯКО ВЛАДИМИР ПРОКОФЬЕВИЧ – д.т.н., професор, завдующий кафедрой лазерной и оптоэлектронной техники, Винницкий национальный технический университет, Винница, Украина.

МАРТЫНЮК ТАТЬЯНА БОРИСОВНА – к.т.н., доцент кафедры лазерной и оптоэлектронной техники, Винницкий национальный технический университет, Винница, Украина.

КУПЕРШТЕЙН ЛЕОНИД МИХАЙЛОВИЧ – к.т.н., доцент кафедры экономической кибернетики Винницкий финансово-экономический университет, Винница, Украина.