

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

УДК 681.3

А. С. Васюра, к. т. н., доц.;

Л. М. Куперштейн

РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОГО ОБРОБЛЕННЯ ДАНИХ НА ПОРОГОВОМУ ПІДСУМОВУВАЛЬНОМУ ПРИСТРОЇ

Вступ

На теперішній час велика увага надається підвищенню швидкості обчислень через розпаралелення процесів оброблення інформації за допомогою нейромережових технологій. Адаже на протиприроду звичайній фон-Неймановській архітектурі, яка є послідовною за своєю природою, штучні нейронні мережі (ШНМ) виграють за рахунок оброблення з масовим паралелізмом [1]. Отже, задачі оброблення та аналізу зображень та сигналів, розпізнавання образів, керування динамічними об'єктами, фінансовою діяльністю, високо енергетичної фізики тощо ефективно розв'язуються системами, розробленими на основі нейронних алгоритмів [2].

Водночас в процесі моделювання принципу функціонування найпростіших нейроподібних структур існує проблема адекватного відтворення основних функцій нейрона, а саме накопичення множини вхідних сигналів шляхом підсумовування масиву операндів у вигляді зважених величин і порівняння отриманої суми із порогом [3]. Операція підсумовування є однією з основних операцій у нейрообробленні і виконується послідовним додаванням кожного із операндів, тобто не має відповідного рівня паралелізму. Відсутність даної властивості значно збільшує час оброблення даних у нейромережі. До того ж, порівняння із граничним значенням здійснюється після отримання суми зважених величин, а не паралельно із підсумовуванням, що також суттєво уповільнює процес оброблення. Тому доцільно було б реалізувати суміщення операцій паралельного підсумовування та порівняння з порогом оброблення, яке дозволило б отримати вихідний сигнал нейрона, не виконуючи остаточного формування суми вхідних операндів.

Отже, метою даної роботи є розробка алгоритму паралельного порогового підсумовування масиву чисел, розробка конвеєрного пристрою, який працює за даним алгоритмом, а також оцінка його часових параметрів.

Постановка задачі

Серед відомих способів розпаралелювання операції підсумовування найпридатнішим з цієї точки зору є відомий метод паралельного підсумовування масиву чисел [4], суть якого полягає у виділенні загальної частини доданків і формуванні відповідних поточних сум і залишків різнице-вих зрізів у процесі циклічної реалізації алгоритму з поступовим зменшенням кількості вхідних даних до моменту отримання нульового залишку. Крім того, в [5] доведена збіжність даного методу, а також його ефективність в процесі оброблення великих масивів даних.

Одним з підходів до реалізації такого оброблення є конвеєрний процес підсумовування, описаний у роботі [6], в основі якого також лежить принцип оброблення з використанням різнице-вих зрізів. Крім того, у роботі [6] розглядається структура пристрою, який реалізує саме метод паралельного додавання масиву чисел [4], але оброблення здійснюється у конвеєрному режимі. В даній роботі *пропонується* оброблення даних також у конвеєрному режимі, але воно являє собою порогове паралельне підсумовування масиву чисел, у якому порогове оброблення суміщене з підсумовуванням. Це дозволяє збільшити функціональні можливості пристрою, а також значно розширити

галузі його використання у порівнянні з відомим [6].

Удосконалюючи та модифікуючи паралельний пороговий пристрій [7] та враховуючи усі особливості методу паралельного підсумовування, а саме виконання обчислень, яке базується на використанні принципу різницевого зрізів [8], можна організувати паралельне порогове підсумовування векторного масиву даних, виконане у конвеєрному режимі [9].

Крім того, представляє інтерес моделювання розробленого пристрою, який виконує порогове паралельне підсумовування, на перспективній елементній базі — програмованих логічних інтегральних схемах (ПЛІС), що актуально під час реалізації нейрочіпів [10].

Паралельне порогове оброблення даних у конвеєрному режимі

Розглянемо докладніше процес конвеєрного підсумовування, який базується на алгоритмі порогового паралельного підсумовування масиву чисел. Його суть полягає в тому, що порогове оброблення n чисел зводиться до обчислення і підсумовування N часткових сум, де N — кількість різноманітних вхідних величин a_i , порівняння цих сум із порогом p і формування підсумкового сигналу вигляду

$$y = \begin{cases} 1, & \text{якщо } S = \sum_{i=1}^n a_i = \sum_{j=1}^N q_j d_j \geq p, \\ 0, & \text{у протилежному випадку,} \end{cases} \quad (1)$$

де y — вихідний сигнал пристрою, q_j — загальна значуща частина, d_j — кратність загальної частини, p — значення порогу оброблення, j — цикл оброблення.

Алгоритм має такий вигляд.

Крок 1. Визначається загальна значуща частина всіх доданків у j -му циклі, тобто

$$q_j = \min\{a_{i,j-1}\}_{i=1}^n, \quad j = \overline{1, N}, \quad (2)$$

де $a_{i,0}$ — i -й доданок на вході пристрою, тобто i -й елемент вхідної групи A_0 чисел.

Крок 2. Виділяється зріз різниць A_j , тобто сукупність величин різниці всіх доданків j -го циклу з їхньою загальною частиною q_j , тобто

$$A_j = \{a_{i,j}\}_{i=1}^n = \{a_{i,j-1} - q_j\}_{i=1}^n. \quad (3)$$

В подальшому отриманий зріз різниць A_j є вхідною множиною доданків для наступного $(j + 1)$ -го циклу.

Крок 3. Формується кратне загальної частини, тобто часткова сума S_j , де кратність d_j визначається кількістю додатних ненульових доданків j -го циклу:

$$S_j = q_j d_j. \quad (4)$$

Крок 4. В першому циклі формується різниця між заданим порогом p та частковою сумою S_1 , отриманою на попередньому кроці, тобто

$$\Delta_1 = p - S_1 \quad (5)$$

і перевіряється умова

$$\Delta_1 \leq 0. \quad (6)$$

В подальшому поточна різниця Δ_{j-1} є поточним значенням порогу на наступному j -му циклі оброблення, тобто

$$\Delta_j = \Delta_{j-1} - S_j, \quad j = 2, 3, \dots \quad (7)$$

На цьому ж кроці перевіряється умова

$$\Delta_j \leq 0. \quad (8)$$

У разі її виконання формується вихідний сигнал пристрою Y , який дорівнює 1, і порогове оброблення припиняється, а в протилежному випадку виконуються кроки 1—4 для

$j = 2, 3, \dots$ до виконання умови (8) або до визначення нульової величини q_j (2).

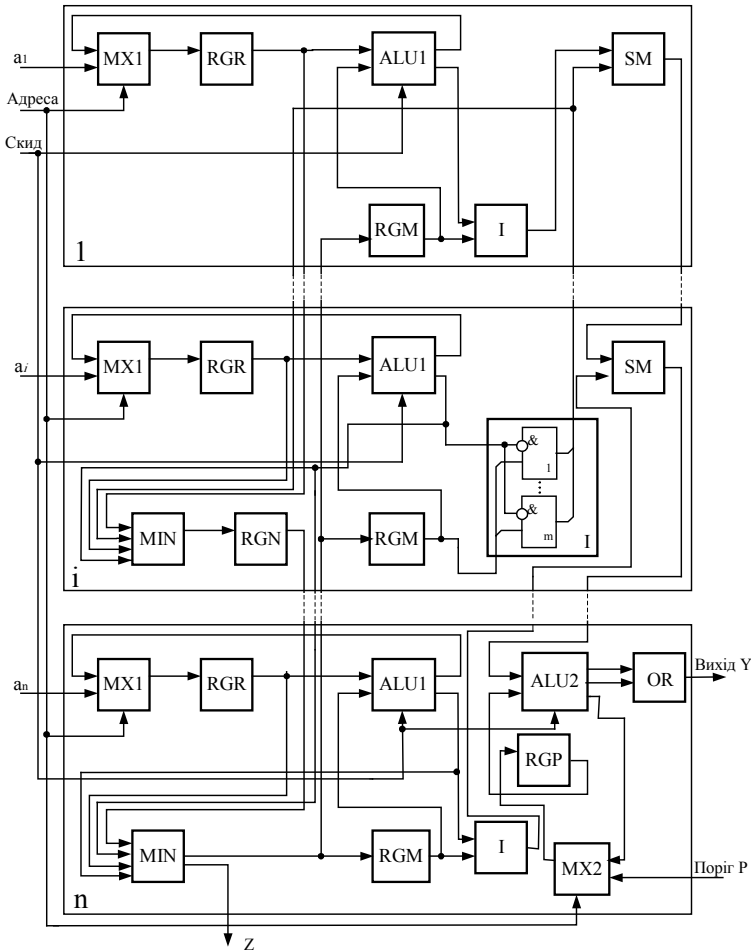


Рис. 1. Функціональна схема порогового підсумовувального пристрою

типлексор $MX1$ i -й операнд надходить в регістр RGR часткового результату i -го ярусу. Після цього виконується послідовне виділення загальної (мінімальної значущої) частини двох операндів — $(i-1)$ -го та i -го мініматором MIN і запис результату в регістр RGN i -го ярусу 1. Згідно із виразом (2) на кроці 1 алгоритму ця операція виконується послідовно, починаючи з другого ярусу. Кінцевий результат q_1 виділення загальної частини всіх n операндів формується в n -му ярусі і з виходу мініматора MIN цього ярусу записується в регістри RGM всіх n ярусів паралельно. Після цього в i -му ярусі відбувається порівняння в $ALU1$ i -го операнда $a_{i,0}$, що надходить з регістра RGR , та загальної частини q_1 всіх операндів, що надходить з регістра RGM . Порівняння відбувається в процесі віднімання згідно із виразом (3) на кроці 2 алгоритму, і ця різниця $a_{i,1}$ записується з виходу $ALU1$ через мультиплексор $MX1$ в регістр RGR . При цьому на другому виході $ALU1$ формується ознака від'ємного результату різниці. З відсутністю одиничного сигналу ознаки від'ємного результату нульовий сигнал на інверсних входах блока I дозволяє проходження в i -му ярусі через блок I загальної частини q_1 операндів з регістра RGM в суматор SM , де відбувається накопичування величини q_1 .

У випадку, якщо на другому виході $ALU1$ i -го ярусу 1 з'явиться одиничний сигнал ознаки від'ємного результату, то його наявність на інверсних входах блока I заборонить надходження величини q_1 в даному циклі оброблення через блок I на вхід суматора SM . В суматорах SM кожного ярусу відбувається послідовне підсумовування величин q_1 , а в результаті на виході суматора SM $(n-1)$ -го ярусу формується часткова сума S_1 згідно із виразом (4) на кроці 3 алгоритму. З виходу регістра RGP значення порогового сигналу p одночасно із частковою сумою S_1 надходить до $ALU2$, в результаті на інформаційному виході $ALU2$ формується поточна різниця Δ_1 згідно із виразом (5) на кроці 4 алгоритму, яка потім через мультиплексор $MX2$ знову записується у регістр

На рис. 1 зображена функційна схема пристрою, який реалізує вищенаведений алгоритм [11]. Пристрій містить n ярусів, кожний з яких складається з блоку $ALU1$ для формування зрізів різниць, мініматора MIN для визначення мінімальної величини серед двох, які подаються на його входи, суматора SM для підсумовування двох доданків, що надходять на його входи. Мультиплексор $MX1$ і блок елементів I необхідні у кожному ярусі для комутації входів, а регістри RGR , RGM , RGN — для тимчасового збереження інформації. Крім того, останній n -й ярус додатково містить блок $ALU2$, мультиплексор $MX2$ та регістр RGP .

Розглянемо детальніше принцип роботи вищезазначеного пристрою. На вхід i -го ярусу надходить i -й операнд $a_{i,0}$ з групи A_0 операндів, кількість яких n , а на вхід мультиплексора $MX2$ n -го ярусу надходить значення порогового сигналу p , який з його виходу записується у регістр RGP . Запис операндів в яруси 1 виконується паралельно. Через муль-

RGP, оскільки вона буде поточним значенням порогу у наступному циклі оброблення згідно із виразом (7). При цьому сигнали з виходів знаку і ознаки нульового результату ALU2 надходять на входи елемента OR. Це необхідно для перевірки умови (6) на кроці 4 алгоритму. Вихід елемента OR формує вихідний сигнал у на виході Y n-го ярусу пристрою, який буде одиничним в разі виконання умови (6) алгоритму і нульовим в протилежному випадку. При цьому присутність одиничного сигналу хоча б на одному з виходів ALU2, тобто наявність нульової чи від'ємної різниці Δ_1 , призведе до проходження його на вихід Y n-го ярусу пристрою, в результаті чого порогове оброблення припиняється. Так виконується 1-ий цикл обробки. У випадку нульового сигналу на виході Y n-го ярусу пристрою виконуються кроки 1—4 наступного $j = 2$ циклу обробки. У всіх циклах, починаючи з другого, на кроці 4 алгоритму виконується операція (7) і перевіряється умова (8). У випадку, якщо умова (8) на кроці 4 алгоритму у всіх циклах не виконується, тобто на виході Y пристрою присутній нульовий сигнал, процес обробки закінчується тоді, коли загальна частина всіх поточних доданків дорівнює нулю, тобто присутній одиничний сигнал Z на виході n-го ярусу.

Наочність роботи конвеєрного пристрою для конкретного масиву цілих додатних чисел: 13, 8, 3, 11, і порогового значення $p = 30$ показано у вигляді діаграм на рис. 2, де прийнято такі позначення: Sign ALU1, Sign ALU2 — відповідно знаковий вихід ALU1 і знаковий вихід ALU2; Z — вихід сигналу «Кінець» пристрою; Y — вихід пристрою.

Часова діаграма (рис. 2) визначається в такий спосіб:

t_1, t_2 — час запису відповідно у регістр (t_{WR}) через мультиплексор (t_{MX}), тобто $t_1 = t_2 = t_{MX} + t_{WR}$; t_3 — час послідовного формування мінімальної додатної величини у мініматорах ярусів (t_{COM}) і запис її у регістр (t_{WR}), тобто $t_3 = (n - 1)(t_{COM} + t_{WR})$; t_4 — час паралельного формування зрізу різниць в АЛП (t_{SUB}), тобто $t_4 = t_{SUB}$, t_5 — час послідовного формування поточної часткової суми у суматорах (t_{SM}) з урахуванням проходження через блок елементів I (t_{MX}), тобто $t_5 = (n - 1)t_{SM} + t_{MX}$; t_6 — час формування поточного значення порогу у АЛП (t_{SUB}) та вихідного сигналу Y (t_Y), тобто $t_6 = t_{SUB} + t_Y$.

Після запису операндів через мультиплексори MX1 та MX2 у регістри RGR та RGP ($t_1 + t_2$) та послідовного формування мінімальної додатної величини у мініматорах MIN ярусів і запису її у

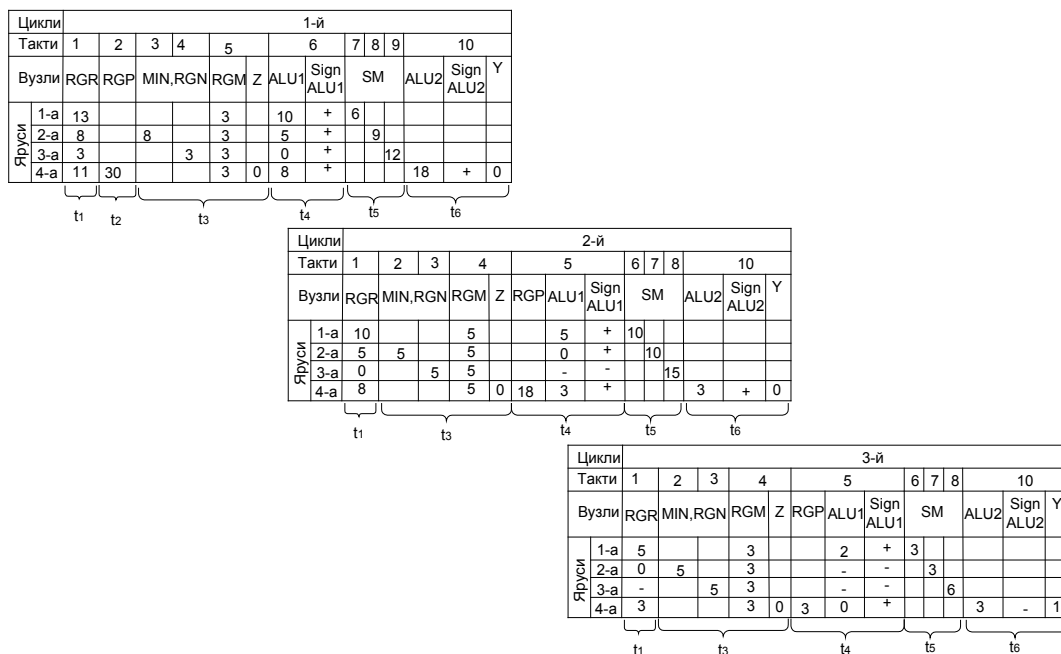


Рис. 2. Часова діаграма обчислень у конвеєрному пристрої

регістри RGM (t_3) формуються зрізи різниць в ALU1 (t_4), які є початковими даними для наступного циклу оброблення. Тобто, не очікуючи накопичення кратних загальної частини та порівняння отриманої часткової суми із порогом, починається наступний цикл оброблення. Отже, час розгону даного конвеєрного пристрою складає

$$t_p = t_1 + t_2 + t_3 + t_4.$$

Тоді загальний час оброблення визначається таким чином:

$$T = t_p + (N - 1)t_{ц} + t_{к},$$

де $t_{ц}$ — час тіла циклу, причому $t_{ц} = t_1 + t_3 + t_4$, $t_{к}$ — час закінчення циклу, $t_{к} = t_5 + t_6$.

Враховуючи, що $t_{SUB} = t_{SM}$, $t_{MX} \approx t_{WR}$, $t_Y \ll t_{SUB}$, загальний час оброблення можна записати $T = N(n - 1)t_{COM} + (N + n)t_{SM} + [N(n + 1) + 3]t_{WR}$.

Для роботи конвеєрного пристрою без простоїв, враховуючи, що $t_{WR} < t_{COM} < t_{SM}$ [12], необхідне виконання умови

$$t_1 + t_3 \geq t_5 + t_6,$$

тобто

$$(n - 1)t_{COM} + (n - 1)t_{WR} \geq nt_{SM} + t_{WR},$$

або

$$(n - 1)t_{COM} + nt_{WR} \geq nt_{SM}.$$

Отже, для забезпечення ритмічної роботи конвеєрного пристрою необхідно використовувати швидкодійні суматори і АЛП з такими часовими співвідношеннями:

$$t_{SM} \leq t_{COM} + t_{WR}.$$

Перспективи застосування

Аналізуючи принцип роботи даного пристрою, не важко побачити, що він працює як звичайний перцептронний нейрон Маккалокка і Піттса з пороговою функцією активації [13, 14], але оперує вже зі зваженими вхідними операндами. Тому даний пристрій безпосередньо можна використати у нейронному обробленні даних, якій на теперішній час приділяється значна увага. Інтерес до цієї області науки та техніки викликаний тим, що багато сучасних задач, таких як класифікація образів, кластеризація, апроксимація функцій, прогноз, оптимізація та ін., розв'язуються найкращим чином за допомогою нейронних та нейроподібних структур через їх високу продуктивність, високу швидкодію, надійність, придатність до навчання у порівнянні з обчислювальними структурами фон-Неймановського типу [2]. Хоча один нейрон здатен виконувати найпростіші операції розпізнавання, потужність нейронних обчислень збільшується від з'єднання нейронів у мережах. Тому планується реалізація нейрочіпу на основі розробленого конвеєрного порогового паралельного пристрою для задач класифікації та оброблення сигналів. В цьому нейрочіпі буде розташовано від одного до декількох десятків нейронів в залежності від ємності кристалу чіпа про що можна говорити виходячи із результатів нижченаведеного моделювання. Потім методом каскадування таких чіпів можна отримати потужну та швидкодійну нейромережу. При цьому за елементну базу пропонується використання саме прогамованих логічних інтегральних схем через їх дешевизну, високу швидкість обчислень, велику ступінь інтеграції, малий час розроблення проекту, можливість перепрограмування, а не надвеликі інтегральні схеми (НВІС), які для розроблення нової моделі нейрочіпу потребують проектування нового кристалу [10, 15]. Крім того, ПЛІС дають можливість реалізувати не тільки нові алгоритми оброблення інформації, але й повернутись до розгляду і вдосконалення вже відомих методів, які в свій час не знайшли застосування через обмеженість елементної бази.

Отже, була промодельована структура даного пристрою на три інформаційні входи, тобто три яруси. Як елементну базу використано ПЛІС XC95288XL-6-BG256 фірми Xilinx [16], а програмного забезпечення — пакет проектування цифрових пристроїв на базі ПЛІС — WebPack ISE. На рис. 3 показано фрагмент файлу повідомлення програми про результати імплементації вищезгаданої ПЛІС.

XACT: version C.16		Xilinx Inc.		
Fitter Report				
Design Name: kupertst		Date: 5-05-2004, 10:39PM		
Fitting Status: Successful				
***** Resource Summary *****				
Design Name	Device Used	Macrocells Used	Product Terms Used	Pins Used
kupertst	XC95288XL-6-BG256	67/288 (23%)	471/1440 (32%)	16/192 (8%)

Рис. 3

Дане моделювання показало значні результати і ще раз підтвердило ефективність використання даної елементної бази. Використано лише 23 % ємності кристалу чіпу, а саме 67 із 288 макрокомірок, 32 % термів (471 із 1440) та 16 виводів із 192 можливих. Використана ПЛІС у порівнянні з іншими є малошвидкісною та малоємнісною, вибір якої пояснюється відсутністю бібліотек з високопродуктивними серіями ПЛІС, оскільки моделювання виконувалось за допомогою безкоштовної вільно розповсюджуваної версії програмного забезпечення через дуже високу ціну повноцінного продукту [16, 17]. Тому в подальшому доцільним буде використання ємнісніших ПЛІС, наприклад, серій CoolRunner, Spartan або Virtex, оскільки розроблений пристрій орієнтований на оброблення великих масивів даних. Отже, використавши мікросхему одної з цих серій та організувавши декілька таких пристроїв з необхідною кількістю входів у структуру одношарової чи багатшарової мережі, можна отримати нейрочіп, на якому можна також реалізувати ще й пам'ять вагових коефіцієнтів.

Висновки

1. Розроблений пристрій виконує у конвексному режимі одночасно паралельне підсумовування векторного масиву із n чисел та порівняння поточних результатів із зовнішнім порогом за рахунок використання принципу різницевого зрізів.

2. Розширення функціональних можливостей процесу паралельного підсумовування досягається за рахунок порівняння на кожному кроці підсумовування поточної часткової суми чисел масиву з порогом оброблення і формування підсумкового сигналу, що дорівнює одиниці тільки тоді, коли поточна сума більше або дорівнює величині порогу оброблення. Такий підхід забезпечує максимальний паралелізм оброблення масиву чисел, оскільки в процесі задіяні одночасно всі операнди, а це, в свою чергу, призводить до прискорення процесу оброблення, оскільки у випадку перевищення суми над порогом, цей факт можна зафіксувати без формування остаточного значення суми всіх чисел масиву.

3. Використання можливостей сучасної технології робить перспективною реалізацію нейрочіпів на ПЛІС, оскільки темпи зростання їх обчислювальної потужності, принаймні, не поступаються аналогічній характеристиці універсальних процесорів, ПЛІС дешеві і мають короткий час проектування (до декількох тижнів). Крім цього, сучасні ПЛІС працюють на тактовій частоті до 300 МГц (це зовнішня частота без множення усередині процесора). При цьому ємність ПЛІС досягає 3 млн. еквівалентних логічних вентилів, що майже в 10 разів більше, ніж ємність усього Pentium II.

4. В результаті імітаційного моделювання отримано нейрочіп на базі ПЛІС XC95288XL-6-BG256. Нейрочіп містить всього один нейрон на три входи через малу ємність кристалу мікросхеми, чіп оперує з двійковими чотирирозрядними даними та має порогову функцію активації. Моделювання показало, що схема пристрою у мікросхемі займає 23 % макрокомірок та 16 % виводів.

СПИСОК ЛІТЕРАТУРИ

1. Liao Y. Neural Networks in Hardware: A Survey, Department of Computer Science, University of California, Davis (<http://wwwwscif.cs.ucdavis.edu/~liaoy/research/NNhardware.pdf>)
2. Галушкин А. И., Иванов В. В. Некоторые концептуальные вопросы развития нейрокомпьютеров // Зарубежная радиоэлектроника. — 1997. — № 2. — С. 3—10.
3. The neural and neural-like networks: synthesis, realization, application and future / V. V. Hrytsyk et al. // Інформаційні технології і системи. — 1998. — Т. 1. — № 1/2. — С. 15—55.

4. Свечников С. В., Кожемяко В. П., Тимченко Л. И. Квазиимпульсно-потенциальные оптоэлектронные элементы и устройства логико-временного типа. — К.: Наукова думка, 1987. — 256 с.
5. Мартинюк Т. Б., Хом'юк В. В., Куперштейн Л. М., Матвеев Є. С. Аналіз моделей паралельного підсумовування елементів числового масиву // Вісник ВПІ. — 2002. — № 6. — С. 51—56.
6. Кожем'яко А. В. Особливості конвеєрного процесу підсумовування масиву чисел // Вісник ВПІ. — 2000. — № 6. — С. 65—68
7. Патент №55862А, G066/14. Підсумовуючий пороговий пристрій / Т. Б. Мартинюк, О. В. Павлюк, Б. С. Барський, Л. М. Куперштейн, І. В. Мисловський. — № 2002075728; Заявлено 11.07.2002; Опубл. 15.04.2003. — Бюл. № 4. — 4с.
8. Мартинюк Т. Б. Рекурсивні алгоритми багатооперандної обробки інформації. Монографія. — Вінниця: «Універсум — Вінниця», 2000. — 216 с.
9. Куперштейн Л. М., Кожем'яко А. В. Модель формального нейрона з використанням принципу різницевого зрізів // Вісник ВПІ. — 2003. — № 6. — С. 284—287.
10. Логовский А. Технология ПЛИС и ее применение для создания нейрочипов // Открытые системы. — 2000. — № 10. — С. 20—25 (http://www.osp.ru/os/2000/10/019_print.htm).
11. Патент № 67004А, G06G7/14. Конвеєрний пристрій / Т. Б. Мартинюк, А. С. Васюра, Л. М. Куперштейн, А. В. Кожем'яко, І. В. Мисловський. — № 2003043663; Заявлено 22.04.2003; Опубл. 15.06.2004. — Бюл. № 6. — 22 с.
12. Вениаминов В. Н. Микросхемы и их применение. — М.: Радио и связь, 1989. — 240 с.
13. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Пер. с англ. — М.: Мир, 1992. — 240 с.
14. Перцептрон — система распознавания образов / Под ред. А. Г. Ивахненко. — К.: Наукова думка, 1975. — 431 с.
15. Хехт–Нильсен Р. Нейрокомпьютинг: история, состояние, перспективы // Открытые системы. — 1998. — № 4. — С. 32—37.
16. <http://www.plis.ru/catalog.html>
17. <http://www.xilinx.com/sxpresso/webpack.htm>

Рекомендована кафедрою автоматики та інформаційно-вимірювальної техніки

Надійшла до редакції 20.04.04
Рекомендована до друку 14.09.04

Васюра Анатолій Степанович — професор; **Куперштейн Леонід Михайлович** — здобувач.
Кафедра автоматики та інформаційно-вимірювальної техніки. Вінницький національний технічний університет

УДК 681.518.03

О. М. Роїк, д. т. н., проф.,

Ю. Я. Табачник, асп.

АНАЛІЗ МЕТОДІВ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ ПРОСТОРОВИХ ДАНИХ У ГЕОІНФОРМАЦІЙНИХ СИСТЕМАХ

Постановка задачі

З розвитком комп'ютерних технологій з'явилася можливість перенести величезну кількість картографічної інформації з паперових носіїв (топографічні плани, карти тощо) у електронний вигляд. Програмні засоби, що використовуються для зберігання, відображення та аналізу географічних даних, носять назву географічних інформаційних систем (ГІС).

Однією із задач, що виникають під час проведення географічних, екологічних, економічних та інших досліджень за допомогою ГІС, є задача класифікації певної інформації. Вихідною інформацією для методів класифікації є багатовимірні дані, що описують досліджувані складні об'єкти, які знаходяться у атрибутивному багатовимірному просторі ознак. Інформація про ознаки об'єктів та зв'язки між ними знаходиться у таблицях вигляду «об'єкт–ознака» та «об'єкт–об'єкт» відповідно. Метою класифікації є розбиття множини об'єктів на деяку (як заздалегідь задану, так і не задану) кількість класів таким чином, щоб об'єкти одного класу були максимально подібними за обраними ознаками, а об'єкти різних класів – максимально відрізнялись.

Особливістю досліджень, у яких використовується інформація про географічні об'єкти, є оперування просторовими даними, тобто даними, що мають атрибут місцезнаходження. Для позна-