Vol. 1, No. 1, 2019

https://doi.org/10.23939/cds2019.01.001

П. Тимощук, С. Шатний
Національний університет "Львівська політехніка"

СХЕМОТЕХНІЧНА РЕАЛІЗАЦІЯ МОДЕЛІ РОЗПАРАЛЕЛЕНОЇ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ НЕЧІТКОЇ ТЕОРІЇ АДАПТИВНОГО РЕЗОНАНСУ

У статті описана і змодельована схемотехнічна реалізація розпаралеленої штучної нейронної мережі нечіткої теорії адаптивного резонансу. У мережі реалізовані паралельний вибір категорії та резонансу. Нейронні схеми типу "winner-take-all" неперервного та дискретного часу забезпечують ідентифікацію найбільших з М-входів. Схеми неперервного часу описані рівняннями стану з розривною правою частиною. Дискретний аналог описано різницевим рівнянням. Відповідні функціональні блок-діаграми схем містять М жорсткообмежувальних нейронів прямого зв'язку та один нейрон зворотного зв'язку, який використовують для обчислення динамічного зсуву входів. Схеми поєднують у собі такі переваги, як довільна скінченна роздільна здатність входів, висока швидкість збіжності операції "winner-take-all", низька обчислювальна складність і складність апаратної реалізації та незалежність від початкових умов. Схеми також використовують для знаходження елементів вхідного вектора з мінімальними/максимальними значеннями для його нормування у діапазоні [0,1].

Ключові слова: функціональна блок-схема, нечітка теорія адаптивного резонансу, нейронна мережа, вибір категорії, переможець-забирає-все, рівняння стану з розривною правою частиною.

Вступ

Теорія адаптивного резонасу (ТАР) дає можливість класифікувати головні операції мозку і являє собою швидкий, масштабований і зручний для паралельної реалізації інструмент. Суттєвою перевагою ТАР є висока швидкість і стабільність навчання [1]–[3].

Розроблено різні методи навчання ТАР без вчителя, зокрема ART 1, ART 2, ART 2-A, ART 3 і нечітка ТАР, а також алгоритми навчання з вчителем, а саме ARTMAP і нечітка ARTMAP. Навчанння в ТАР реалізується як в офлайн-режимі, так і в режимі реального часу. Методи ТАР передбачають масштабування для їх використанння для обробки великих наборів даних. ТАР дає можливість коректно обробляти зашумлені і спотворені дані [4], [5].

Велика кількість алгоритмів ТАР були реалізовані у таких програмних засобах як Matlab, C ++ і CUDA за допомогою комп'ютерного моделювання [4], [6], [7]. Алгоритми ТАР також реалізуються з використанням аналогового, цифрового і гібридного (аналого-цифрового) апаратного забезпечення [8]. Наприклад, апаратна реалізація в PSPICE довготривалої і короткотривалої пам'яті для алгоритму ART 1 з цифровими входами описана в [9]. У [10] представлено оптоелектронну реалізацію нейронної мережі (НМ) ART. Адаптивно-резонансна штучна нейронна мережа (ШНМ), заснована на обробці голографічної інформації в фоторефрактивному кристалі, описана в [11]. В [12], [13] алгоритм ART1, представлений у [14], реалізований з використанням аналогових схемотехнічних елементів VLSI. Алгоритм реалізовано на високоефективному кристалі, що реалізує стандартний CMOS-процес, який дозволяє функціонування у реальному часі.

Важливою й актуальною задачею є обробка великого об'єму даних, а також багатовимірних функцій протягом заданого часу [16]. Зокрема, обробка даних у реальному часі необхідна у таких застосуваннях, як медична діагностика, складні електромагнітні середовища, сильні фонові

зашумлення тощо [17]. Можливість обробки в реальному часі є однією з найважливіших переваг ШНМ [18]. Тому необхідні швидкодіючі ШНМ, які придатні для розв'язання таких задач. Крім того, необхідно розробляти паралельні та розподілені методи, здатні знаходити розв'язки таких задач [19]. Наприклад, класифікація невизначених і неповних даних у реальному часі є серйозною проблемою. Тому розробка швидких і точних ШНМ, здатних розв'язувати такі задачі, є дуже важливою для адаптивної обробки змінних у часі даних таких, наприклад, як ЕКГ і ЕМГ [20].

Підходи, які основані на ТАР, спроможні успішно розв'язувати такі задачі. Однак час обробки даних за допомогою ТАР може бути занадто великим для їх використанння у режимі реального часу. Наприклад, час обробки 50000 наборів даних за допомогою методу ТАР, реалізованого у програмному забезпеченні, може становити понад 30 секунд, що може бути неприйнятним для застосувань у режимі реального часу [21].

Значна частина часу виконання алгоритму ТАР може витрачатись на вибір категорії. Наприклад, вибір категорії займає понад 80 % часу виконання за оптоелектроної реалізації алгоритму ТАР [10]. Майже 90 відсотків часу виконання необхідно для вибору категорії за допомогою нечіткої ШНМ ТАР, реалізованої у VLSI [13]. Метод ТАР, реалізований у гібридному апаратному забезпеченні, для оновлення синаптичних ваг потребує більшої кількості операцій вибору максимального елемента з множини даних порівняно з числом операцій додавання, множення, логічного І та дефазифікації [15]. У послідовній нечіткій ТАР вибір категорії здійснюється за допомогою послідовних обчислень функцій вибору кластера. Крім цього, цей крок вимагає послідовної рекурсивної ідентифікації за допомогою модуля Winner-Take-All (WTA) функцій, які мають перше найбільше значення, друге найбільше значення і так далі аж до функції з найменшим значенням, яке задовольняє так звану умову слідкування [1]. Навіть більше, також послідовнно виконується перевірка критерію слідкування. Це може призводити до часозатратного процесу кластеризації, особливо для великих наборів даних.

У цій роботі для підвищення швидкості процесу кластеризації в нечіткій ТАР вибір категорії і резонанс виконуються паралельно у режимі навчання. У режимі навчання застосовують паралельну обробку входів. Модулі WTA реалізовані за допомогою WTA нейронних схем (НС) і неперервного часу, і дискретних, що визначають найбільший/найменший серед N вхідних даних [23], [24]. Крім того, НС WTA використовують також для ідентифікації мінімальних і максимальних значень елементів початкового вхідного вектора на етапі його нормалізації [0,1]. WTA НС неперервного часу використовується для обробки вхідних даних неперервного часу. Для обробки вхідних даних дискретного часу застосовується дискретний аналог.

Як свідчать результати моделювання, швидкість функціонування розпаралеленої нечіткої ШНМ ТАР, реалізованої в апаратному забезпеченні, набагато вища, ніж реалізованої у програмному забезпеченні. Після розпаралелення точність функціонування мережі зберігається. Розпаралелена мережа здатна коректно обробляти не тільки стаціонарні вхідні дані, але й змінні у часі вхідні дані.

Нечітка теорія адаптивного резонансу

Метод нечіткої ТАР придатний до швидкого стабільного навчання без вчителя для розпізнавання категорій будь-яких скінченних послідовностей аналогових або дискретних входів. Цей метод послідовно обробляє вхідні вектори, здійснюючи пошук і узгодження доступних кластерів категорій у режимі реального часу [1]. Нечітка ТАР поєднує основні переваги всіх алгоритмів ТАР [26]. Метод широко використовують у різних сферах таких, як медицина, економіка, інженерія, інформатика, розпізнавання образів, класифікація тощо [27], [28]. Реалізація нечіткої ТАР у ШНМ представлена в [29]. Аналогову апаратну реалізацію підходу можна знайти в [30].

Нечітка ШНМ ТАР складається з двох шарів нейронів і слідкуючої підмережі, яка керується слідкуючим параметром $r \in [0,1]$ [31], [32]. Вхідний шар L_1 містить M вхідних нейронів. На кожен

вхідний нейрон надходять вхідні дані $I_i \in [0,1]$, *i=1,…, M* вхідного вектора $I=(I_1,...,I_M)$. Шар категорій L_2 містить *N* нейронів, які представляють можливі категорії. Кожен *i*-тий нейрон L_1-го шару з'єднаний з кожним *j*-тим нейроном L_2-го шару синаптичною вагою w_{ij}. Кожен нейрон L_2-го шару отримує вхід T_j, *j=1,…,N,* який визначає степінь подібності між вхідним вектором I і вектором ваг $w_j=(w_{1j},...,w_{Nj})$. кожен *j*-й нейрон L_2-го шару з'єднаний з кожним *i*-м нейроном L_1-го шару синаптичними вагами w_{ji}, де $w_{ij}=w_{ji}$. Початково значення всіх ваг дорівнюють одиниці, тобто

$$w_{i1}(0)=...=w_{iM}(0)=1 \tag{1}$$

або $w_{ij}(0)>1$ для глибших пошуків.

Процес кластеризації за допомогою методу ТАР складається з таких трьох основних етапів.

Вибір категорії: Функція вибору T_j для кожного *j*-го кластеру описується як

$$T_j=\frac{|I \wedge w_j|}{\alpha+|w_j|}, \tag{2}$$

де $\alpha>0$ – параметр вибору, $\wedge$ – нечіткий оператор, що визначається як

$$(x \wedge y)_j=\min(x_j,y_j). \tag{3}$$

Функція нормування описується як

$$|z| \equiv \sum_{j=1}^{N} z_j. \tag{4}$$

Вибір категорії описується так:

$$T_J=\max\{T_j: j=1,...,N\}. \tag{5}$$

Вираз (5) характеризує функціонування WTA-блоку мережі. Якщо більше ніж одне значення T_j є максимальним, вибирається категорія *j* з найменшим значенням індекса.

Вихід *J*-го нейрону шару L_2 :

$$y_J=\begin{cases}1, & \text{if } T_J=\max\{T_j\};\\ 0 & \text{otherwise.}\end{cases} \tag{6}$$

Резонанс: Для так званої категорії переможця *J* перевіряють критерій близькості

$$\rho \le |I|/|I \wedge w_J|, \tag{7}$$

де ρ – параметр близькості, який використовують для встановлення мінімального показника подібності для входів одного і того самого кластеру [4], [33]. Якщо умова (7) не задовольняється, категорія *J* не вибирається і встановлюється $T_J=0$. Після цього вибирається категорія з другим максимальним значенням T_J і знову перевіряється умова (7) і т. д. аж до того моменту, поки не задовільниться умова (7). Інакше, якщо умова (7) не задовольняється, генерується новий кластер.

Навчання: Якщо умова (7) задовольняється, пошук припиняється і значення w_j оновлюється за різницевим рівнянням

$$w_j(new)=\beta(I \wedge w_j(old))+(1-\beta)w_j(old), \tag{8}$$

де $\beta \in [0,1]$ – параметр швидкості навчання, значення якого для швидкого навчання встановлюється $\beta=1$. Для ефективної обробки зашумлених вхідних даних встановлюється $\beta=1$ для незафіксованної категорії *J* і $\beta<1$ після того, як *J* стає зафіксованою.

Перед початком обробки методом нечіткої ТАР початкові входи повинні бути пронормовані у діапазоні [*0,1*]. Нехай $X = (X_1,...,X_M)$ – вектор початкових входів. Ми нормуємо ці входи у діапазоні [*0, 1*], використовуючи вираз:

$$I_i = (X_i - X_{\min}) / (X_{\max} - X_{\min}), \tag{9}$$

де $X_{\min} = \lambda \min\{X_1,...,X_M\}$, $X_{\max} = \max\{X_1,...,X_M\}$, $i=1,...,M$, $0 < \lambda < 1$.

Вибір категорії у випадку входів неперервного часу

Використаємо KWTA НС неперервного часу, описану в [23], з $K = 1$ для схемотехнічної реалізації вибору категорії (5) у випадку входів неперервного часу. Для цього модифікуємо таку схему, описавши її рівнянням стану:

$$du / dt = \gamma(|u| + p)\left(\sum_{j=1}^{N} y_j(u) - 1\right), \tag{10}$$

де

$$y_j(u) = \begin{cases} 1, & if \quad T_j' - u > 0; \\ 0 & otherwise \end{cases} \tag{11}$$

–j-тий вихід схеми, $j = 1,...,N$ $T_j' = T_j + \varepsilon$, T_j – j-тий вхід схеми, ε – невеликий аддитивний шум, $\sum_{j=1}^{N} y_j(u)$ визначає число додатних виходів, u – змінна стану з початковою умовою $-\infty < u_0 < \infty$, $p>0$, γ – коефіцієнт підсилення, який використовується для керування швидкістю збіжності траєкторії змінної стану до 1WTA операції. Схема вибирає найбільший/найменший серед N невідомих входів, які перебувають у невідомому діапазоні. Для уникнення отримання виходів, які не мають властивості 1WTA [34], до входів схеми T_j додається шум ε. Це гарантує виконання для кожного $i \neq l$ $l = 1,...,N$ нерівності $T_j \neq T_l$. Використаємо 1WTA НС, яка описується рівнянням стану (10) і ступінчастою функцією (11) для вибору категорії у випадку входів неперервного часу.

Схема містить N жорсткообмежувальних нейронів і два нейрони зворотного зв'язку, які використовують для визначення динамічного зсуву входів. Схема поєднує в собі здатність досягати будь-якої скінченної високої точності і швидкості збіжності до операції 1WTA, низьку складність апаратної реалізації і незалежність початкових умов. Зокрема, її траєкторії змінної стану є глобально стабільними і збіжними протягом скінченного часу до операції 1WTA. Час збіжності не залежить від кількості входів. Функціонування схеми не залежить від початкових станів. Тому така схема не потребує періодичного скидання, додаткової керуючої схеми і витрачання додаткового часу обробки на цей режим, що є важливим для обробки даних у режимі реального часу. Схема може бути реалізована в аналоговому апаратному забезпеченні на суматорах, перемножувачах, інтеграторі, перемикачах і зовнішніх джерелах напруги або струму, які придатні для обробки входів у реальному часі з використанням технологій VLSI.

Схема може визначити не тільки найменші/найбільші за значеннями стаціонарні входи T_j, $j=1,...,N$, але і найбільші/найменші за значенннями змінні у часі входи $T_j(t)$. У випадку змінних у часі входів повинна виконуватись умова

$$|dT_j(t) / dt| << |du / dt|, j=1,...,N \tag{12}$$

до досягненння кожною $u \neq u^*$ встановленого режиму u^*. З (10) неважко побачити, що у перехідних режимах $|du / dt|_{\min} = \gamma p$. Тому умова (12) може бути записана у вигляді:

$$|dT_j(t) / dt| << \gamma p, \; j=1,...,N. \tag{13}$$

Вибір категорії у випадку дискретного входу

Використаємо KWTA НС, описану в [24], при *K=1* для вибору категорії апаратної реалізації (5) у випадку дискретних входів. Для цього змодифікуємо схему описану для наступного диференційного рівняння:

$$v(k+1) = v(k) - W\left(1 - \sum_{j=1}^{N} z_j(k)\right)\gamma^k, \tag{14}$$

де $v(k)$ – k-те дискретне значення змінної стану траекторії, $0.5 < \gamma < 1$ ітеративно оновлюваний параметр, який гарантує збіжність $v(k)$ для 1WTA операції,

$$z_j(k) = \begin{cases} 1, & \textit{if} \quad T_j^* - v(k) > 0; \\ 0 & \textit{otherwise} \end{cases} \tag{15}$$

Застосовуємо 1WTA НС описану диференційним рівнянням (14) і кроковою функцією (15) для категорії вибору у випадку дискретних входів.

Подібно до аналогової схеми, дискретні схеми з N нейронів прямого зв'язку і одного жорсткообмежувального нейрону зворотнього зв'язку, застосовуються для знаходження базового зсуву входів. 1WTA НС може визначити найбільші/найменші входи T_j, $j=1,\ldots,N$ як максимальний/мінімальний $T_j(k)$, $k=1,2,\ldots$ скінченної зміни швидкості. В останньому випадку, модуль зсуву змінної стану повинен бути значно більшим ніж входи у перехідних режимах, тобто нерівність

$$\left|T_j(k+1) - T_j(k)\right| << \left|v(k+1) - v(k)\right|, \tag{16}$$

$j=1,\ldots,N$ повинна задовільняти поки не буде досягнуто стану готовності $\bar{v}(k)$ для кожного $v(k) \neq \bar{v}(k)$. Оскільки відповідно до (14) $\left|v(k+1) - v(k)\right| = W\gamma^k \left|1 - \sum_{j=1}^{N} z_j\left(v(k)\right)\right|$, слідує що $\left|v(k+1) - v(k)\right|_{\min} = W\gamma^n$ для будь-якого $0,5 < \gamma < 1$, де n це кількість ітерацій для досягнення 1WTA операції. Тому умова (16) може бути записана як

$$\left|T_j(k+1) - T_j(k)\right|_{\max} << W\gamma^n, \tag{17}$$

Якщо нерівність (17) зберігається, тоді дискретна 1WTA схема може бути використана для обробки дискретних змінних входів $T_j(k)$, $j=1,\ldots,N$. Ми також використовуємо схему для визначення мінімальних і максимальних дискретних стаціонарних входів T_j, $j=1,\ldots,N$ і змінних по часу входів $T_j(k)$, $k=1,2,\ldots$ для їхньої нормалізації в діапазоні *[0, 1]* за умовою (9)

Паралелізація мережі

Функція вибору T_j для кожного j-го кластеру (2) нечіткої ТАР ШНМ обчислюється послідовно рекурсивно. Крім того, вибір категорії також виконується послідовно для кожного WTA блоку шару L_2 за рахунок повторення пошуку максимального елекмента T_j, $j=1,\ldots,N$ (5) та визначення виходів (6) j-го нейрону шару L_2. Більше того, параметр слідкування (7) також визначається рекурсивно для кожної переможної категорії J. Тому процес кластеризації може зайняти багато часу, особливо якщо нечітка ТАР застосовується для вирішення складних задач. Додатково попередня обробка входів шляхом їх нормалізації в діапазоні *[0, 1]* також потребує додаткового часу обробки. В результаті цей спосіб, може мати надто високу обчислювальну

складність, що не може бути застосуване для розв'язання складних задач, та/або задач реального часу [21].

Для збільшення швидкості процесу кластеризації використовуючи нечітку ТАР, використаємо паралельне обчислення функції вибору T_j, j=1,…,N (2) для кожного j-го кластеру. Більше того, вирифікація параметру слідкування (7) також проводиться паралельно.

На рис. 1 представлений алгоритм роботи паралелізованої нечіткої ТАР в режимі тренування, побудованого відповідно до паралелізованих псевдокодів, наведених у [7]. Алгоритм складається з наступних етапів:

1. Початкових вхідний вектор $X=(X_1,...,X_M)$ нормується в паралельному режимі до [*0,1*] за умовою (9).

2. Вхідний набір даних $I=(I_1,...,I_M)$ подається до мережі в послідовному режимі.

3.Параметр (7) перевіряється в паралельному режимі для переможної категорії J: якщо r $|I| \le |I \wedge w_J|$, тоді крок 4 виконаний; інакше, якщо r $|I| > |I \wedge w_J|$, тоді T_J прирівнюється до ‘0’.

4. Функція вибору T_j, j=1,…,N (2) обчислюється паралельно для кожної категорії j.

5. Категорія J, для якої T_J максимальне, вибирається за умовою (5).

6. Перевіряється нерівність $T_J \neq 0$: якщо $T_J \neq 0$, ваги оновлюються відповідно до (8); інакше, якщо $T_J = 0$, вибирається новий нейрон у шарі L_2.

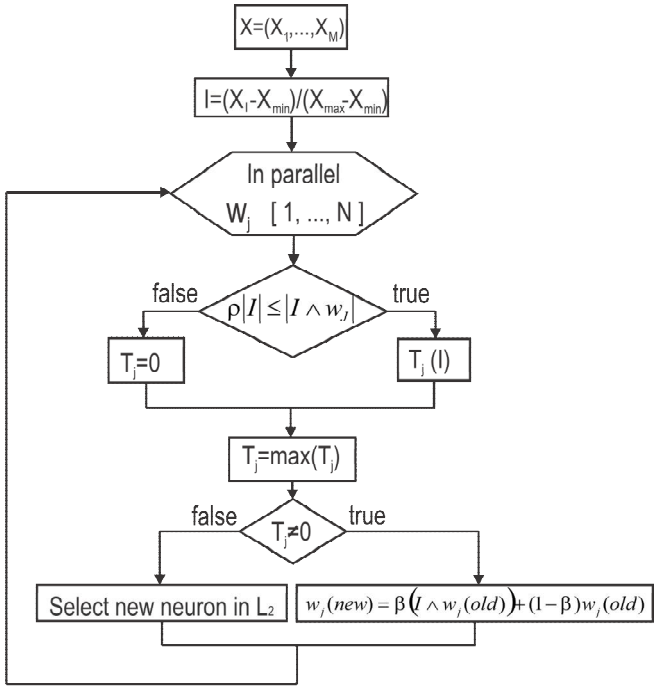

Рис. 1. Алгоритм паралелізованого тренування нечіткої ТАР ШНМ.

На етапі тестування входи можуть оброблятися паралельно. Алгоритм роботи паралелізованого тестування нечіткої ТАР, побудованого відповідно до паралелізованих псевдокодів, представлених у [7], показаний на рис. 2. Алгоритм складається з наступних етапів:

1. Вхідний набір даних $I=(I_1,...,I_M)$ подається до мережі в паралельному режимі.

2. Функція вибору T_J прирівнюються до ‘0’.

3. Критерій слідування (7) послідовно перевіряється для виграшної категорії *J*: якщо $r|I|£|I Ù w_J|$, тоді функція вибору T_j, *j=1,…,N* (2) обчислюється послідовно для кожної категорії *j*; перевіряється нерівність $T_j > T_J$: якщо $T_j > T_J$, тоді $T_J = T_j$ і $I Î j$.

4. Перевіряється рівність $T_J = 0$: якщо $T_J = 0$, тоді $I Ï [1,N]$.

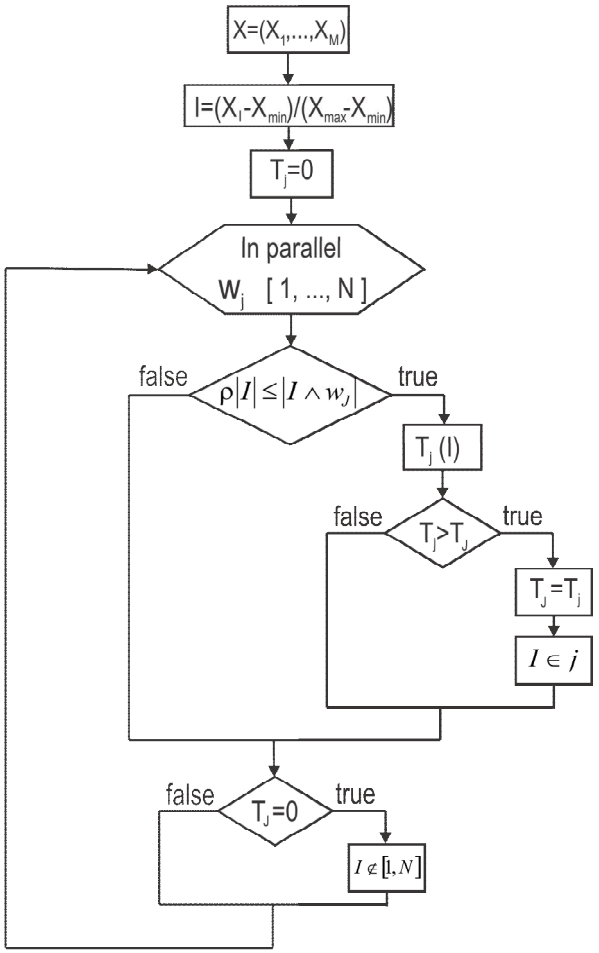

Рис. 2. Алгоритм паралелізованого тестування нечіткої ТАР ШНМ

Технічна реалізація паралелізованої мережі

Реалізовані із використанням паралельного апаратного забезпечення алгоритми роботи представлені на рис. 1 та рис. 2, забезпечують значне пришвидшення режиму навчання і кластеризації на етапі тестування. Крім того, апаратне забезпечення має набагато вищу ефективність і низький рівень енергоспоживання порівняно з алгоритмами програмної реалізації [35]. Це може бути надзвичайно важливо для нечіткої ТАР високошвидкісного навчання [9], [12], [13].

Для обробки неперервних вхідних даних, паралелізована нечітка ТАР ШНМ може бути реалізована із використанням аналогового обладнання. Таке обладнання безпосередньо взаємодіє з входами, тоді як цифрова реалізація вимагає швидкого аналого-цифрового перетворювача для введення даних і цифро-аналогових перетворювачів для виведення даних. Аналогове обладнання забезпечує компактність мережі і високу швидкодію при низькому енергоспоживанні. Практична точність та швидкість роботи мережі обмежені апаратною реалізацією. Найбільш інерційним елементом аналогової апаратної реалізації мережі є аналоговий інтегратор, який, однак, здатний

працювати в широкому діапазоні частот до МГц. Тому очікується, що аналогова апаратна реалізація мережі буде швидкою, компактною та енергоефективною.

Для реалізації мережі в цифровому апаратному забезпеченні архітектура на основі реконфігурованої обчислювальної системи FPGA цілком підходить, оскільки паралельна структура FPGA відповідає топології мережі і придатна до реконфігурації. Мережева архітектура може бути реалізована на чіпі FPGA, що виконує навчання в режимі реального часу. Завдяки своїй гнучкості, FPGA придатна до реалізації надскладної мережі.

Як видно з алгоритмів, вони можуть бути реалізовані в апаратних засобах із використанням аналогових і цифрових засобів, мультиплексорів, подільників, інтеграторів, керованих перемикачів, інверторів і зовнішніх джерел напруги або струму, які підходять для роботи в режимі реального часу за допомогою VLSI технології. Зокрема, множення (8), ділення (2) і (9), а також аналого-цифрові перетворювачі можуть бути реалізовані за допомогою логарифмічних схем, реалізованих на комутованих конденсаторах. Такі контури дають змогу досягти точності операції до 0,01 % і часу до 10–20 нс. Це дає можливість отримати високу продуктивність операції нечіткої ТАР ШНМ. Для цього не потрібно змінювати функцію вибору (2), щоб уникнути операції ділення, як це зроблено в [12].

Симуляція технічної реалізації мережі

Розглянемо два приклади моделювання апаратної реалізації, які ілюструють продуктивність паралелізованої нечіткої ТАР ШНМ. Час моделювання порівнюється з часом виконання послідовної нечіткої ТАР ШНМ[1], [29], реалізованої на процесорі на мові C ++ і з часом виконання паралелізованої нечіткої ТАР ШНМ, реалізованої на GPU з використанням CUDA [6], [7].

Параметр вибору α=0.1, параметр навчання β=0.1, слідкуючий параметр ρ=0.9, що відповідає обчислювальній складності мережі ТАР. У цій мережі схеми описуються за (13), (14) та (17), (18), використовуються як аналогові та дискретні WTA модулі шару L_2, в яких $\delta = 10^6$, $p=1$, та z рівномірно розподілений в інтервалі $\left(0.10^{-6}\right)$ довільний шум. Ці схеми також використовуються для знаходження мінімального та максимального значення елементів вектора X при його нормуванні до [0,1].

Для режиму навчання та різної кількості вхідних наборів $P=N\times M$ довжиною 32, на рис. 3 представлено в напівлогарифмічній шкалі графіків часу t_{SH} необхідної для моделювання послідовної апаратної реалізації нечіткої ТАР, час t_{PH} необхідний для моделювання розпаралеленої апаратної реалізації мережі, час t_{CPU} r необхідний для послідовної програмної реалізації мережі на процесорі з використанням C ++, та час t_{GPU} необхідний для роботи паралельної програмноїреалізації мережі на GPU з використанням CUDA.

Для режиму тестування, відповідні графіки t_{SH}, t_{PH}, t_{CPU}, та t_{GPU} в залежності від P представлені на рис. 4. Відповідно до результатів симуляції, паралелізована технічна реалізація нечіткої ТАР показує мінімальне пришвидшення порівняно із програмною реалізацією мережі із використанням GPU і отримана для значення P=10×1000. Максимальне пришвидшення тестування порівняно з послідовною технічною реалізацією мережі й отримана для значення P=10000×10.

Як видно з результатів симуляції, представлених на рис. 4 та рис. 5, час t_{PH} менший від часу t_{SH}, t_{CPU}, та t_{GPU} для всіх заданих кількостях наборів даних. Пришвидшення зростає із зростанням значення P. Необхідна кількість паралельних каналів є помірною, а точність роботи схеми зберігається і після розпаралелення. Отже, апаратна реалізація паралелізованої нечіткої ТАР може бути рекомендованою для пришвидшення швидкодії процесу кластеризації.

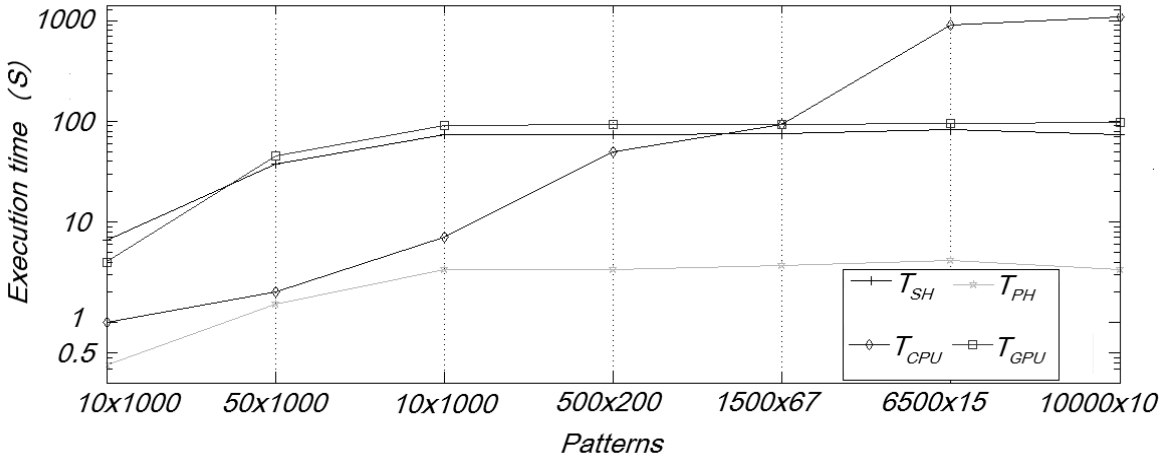

Рис. 3. Часові залежності необхідні для моделювання апаратного та програмного забезпечення послідовної і паралельної нечіткої ТАР в режимі навчання для різних номерів вхідних шаблонів.

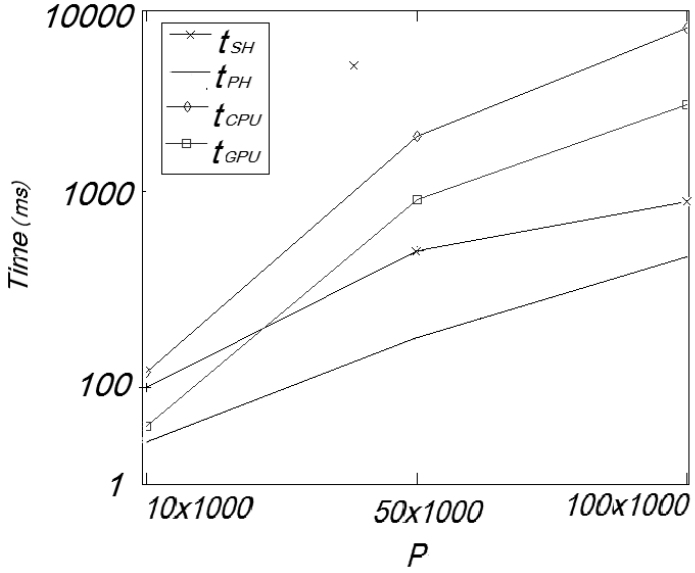

Рис. 4. Часові залежності необхідні для моделювання апаратного та програмного забезпечення послідовної і паралелізованої нечіткої ТАР в режимі тестування для різних номерів вхідних шаблонів.

Висновки

Описана розробка аналогової та цифрової апаратної реалізації паралелізованої нечіткої ТАР ШНМ. Зокрема, реалізовані паралельні обчислення функції вибору та резонансу. Високошвидкісні та точні безперервні і дискретні схеми WTA застосовуються в мережі для вибору категорії. WTA блоки також використовують для знаходження елементів мінімального і максимального значення початкового вхідного вектора на етапі нормалізації до $[0,1]$. Аналогову апаратну реалізацію мережі використовують для обробки неперервних входів. Для обробки дискретних входів використовують цифрову реалізацію схеми. Згідно з результатами моделювання, швидкість послідовної і паралелізованої операцій, реалізованих із використанням апаратних засобів, набагато вища, ніж у їхніх програмних аналогів. Пришвидшення зростає зі збільшенням значення P. Експлуатаційна точність нечіткого ТАР ШНМ зберігається після його розпаралелювання, а кількість паралельних каналів, необхідних для цієї задачі, не є високою.

Мережа може бути вбудована в інші системи і застосовуватися для обробки великих обсягів даних із високою швидкістю і точністю.

1. Carpenter G. A., Grossberg S., and Rosen D. B., "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System", Neural Networks, vol. 4, no. 6, pp. 759–771, Jun. 1991.

2. Grossberg S. and Levine D. S., "Neural dynamics of attentionally modulated Pavlovian conditioning: blocking, inter-stimulus interval, and secondary reinforcement", Applied Optics, vol. 26, no.23, pp. 5015–5030, Dec. 1987.

3. Wunsch II D. C., "ART properties of interest in engineering applications", in Proc. Int. Joint Conf. Neural Networks, 2009, pp. 3380–3383.

4. Grossberg S., "Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world", Neural Networks, vol. 37, pp. 1–47, Jan. 2013.

5. Tan A.-H., Carpenter G. A., and Grossberg S., "Intelligence through interaction: Towards a unified theory for learning", in Proc. 4th Int. Symp. Neural Networks, LNCS 4491, 2007, pp. 1094–1103.

6. Martínez-Zarzuela M., Pernas F., Díez Higuera J., and Antón-Rodríguez M., "Fuzzy ART neural network parallel computing on the GPU", in Proc. 9th Int. Work-Conf. Art. Neural Networks, LNCS 4507, 2007, pp. 463–470.

7. Martnez-Zarzuela M., Pernas F., Pablos A. de, Rodrguez M., Higuera J., Giralda D., and Ortega D., "Adaptive Resonance Theory fuzzy networks parallel computation using CUDA", in Proc. 10th Int. Work-Conf. Art. Neural Networks, LNCS 5517, 2009, pp. 149–156.

8. Ho C. S., Liou J. J., Georgiopoulos M., Heileman G. L., and Christodoulou C., "Analogue circuit design and implementation of an adaptive resonance theory (ART) neural network architecture", Int. J. Electronics, vol. 76, no 2, pp. 271–291, Apr. 1994.

9. Tsay S. W. and Newcomb R. W., "VLSI implementation of ART1 memories", IEEE Trans. Neural Networks, vol. 2, no. 2, pp. 214–221, March 1991.

10. Wunsch D. C., Caude U. T. P., Capps C. D., Marks R. J., and Falk R. A., "An optoelectronic implementation of the adaptive resonance fuzzy neural network", IEEE Trans. Neural Networks, vol. 4, no. 4, pp. 673–684, July 1993.

11. Wunsch II D. C., Morris D. J., McGann R. K., and Caudell T. P., "Photorefractive Adaptive Resonance Neural Network", Applied Optics, vol. 32, no. 8, pp. 1399–1407, Mar. 1993.

12. Serrano-Gotarredona T. and Linares-Bamanco B., "A modified ARTI algorithm more suitable for VLSI implementations", Neural Networks, vol. 9, no. 6, pp. 1025–1043, Aug. 1996.

13. Serrano-Gotarredona T. and Linares-Bamanco B., "A real-time clustering microchip neural engine", IEEE Trans. VLSI Systems, vol. 4 no. 2, pp. 195–209, June 1996.

14. Carpenter G. A. and Grossberg S., "A massively parallel architecture for a self-organizing neural pattern recognition machine", Computer Vision, Graphics, and Image Processing, vol. 37, no. 1, pp. 54–115, Jan. 1987.

15. Versace M., Kozma R. T., and Wunsch D. C., "Adaptive resonance theory design in mixed memristive-fuzzy hardware", in Advances in Neuromorphic Memristor Science and Applications, R. Kozma, R. Pino, and G. Pazienza, Eds. Netherlands: Springer, 2012, pp. 133–153.

16. Xu R. and Wunsch II D., "Survey of clustering algorithms", IEEE Trans. Neural Networks, vol. 13, no 3, pp. 645–678, May 2005.

17. Kalouptisidis N., Signal Processing Systems. Theory and Design. New York: Wiley, 1997.

18. Tank D. W. and Hopfield J. J., "Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit", IEEE Trans. Circuits Syst., vol. 33, no. 5, pp. 533–541, May 1986.

19. Xia Y. and Wang J., "A one-layer recurrent neural network for support vector machine learning", IEEE Trans. Systems, Man and Cybernetics – Part B: Cybernetics, *vol. 34, no. 2, pp. 1261–1269, 2004.*

20. Carpenter G. A., Grossberg S., and Reynolds J. H., "ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network", Neural Networks, vol. 4, no 5, pp. 565–588, Feb. 1991.

21. Liu L., Huang L., Lai M., and Ma C., "Projective ART with buffers for the high dimensional space clustering and application to discover stock associations", Neurocomputing, vol. 72, nos. 4-6, pp. 1283–1295, Jan. 2009.

22. Kim S. and Wunsch II D., "A GPU based parallel hierarchical fuzzy ART clustering", in Proc. Int. Joint Conf. Neural Networks, 2011, pp. 2778–2782.

23. Tymoshchuk P. V., "A dynamic K-winners take all analog neural circuit", in Proc. IVth Int. Conf. "Perspective technologies and methods in MEMS design", Lviv-Polyana, Ukraine, May 21–24, 2008, pp. 13–18.

24. Tymoshchuk P. V., "A discrete-time dynamic K-winners-take-all neural circuit", Neurocomputing, vol. 72, nos. 13–15, pp. 3191–3202, Aug. 2009.

25. Cai X., Prokhorov D. and Wunsch D., "Training winner-take-all simultaneous recurrent neural networks", IEEE Trans. Neural Networks, vol. 18, no 3, pp. 674–684, May 2007.

26. Carpenter G. A., Grossberg S., Markuzon N., Reynolds J. H., and Rosen D. B., "Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps", IEEE Trans. Neural Networks, vol. 3, no. 5, pp. 698–713, Sept. 1992.

27. Lopes M. L. M., Minussi C. R., and Lotufo A. D. P., "Electric load forecasting using a fuzzy ART and ARTMAP neural network", Applied Soft Computing, vol. 5, no. 2, pp. 235–244, Jan. 2005.

28. Meng L., Tan A.-H., and Xu D., "Semi-supervised heterogeneous fusion for multimedia data co-clustering", IEEE Trans. Knowledge Data Engineering, vol. 26, no. 9, pp. 2293–2306, 2014.

29. Carpenter G. A., Grossberg S., and Rosen D. B., "A neural network realization of fuzzy ART", Technical Report CAS/CNS-91-021. Boston, MA: Boston University, 1991.

30. Serrano-Gotarredona T., Linares-Barranco B., and Andreou A. G., Adaptive Resonance Theory Microchips: Circuit Design Techniques. Norwell, MA: Kluwer, 1998.

31. Xu R., Xu J., and Wunsch II D. C., "Using Default ARTMAP for cancer classification with MicroRNA expression signatures", in Proc. Int. Joint Conf. Neural Networks, 2009, pp. 3398–3404.

32. Xu R., Xu J., and Wunsch II D. C. "MicroRNA expression profile based cancer classification using Default ARTMAP", Neural Networks, vol. 22, pp. 774–780, June 2009.

33. Meng L., Tan A.-H., and Wunsch D. C., "Vigilance adaptation in adaptive resonance theory", in Proc. Int. Joint Conf. Neural Networks, 2013, pp. 1–7.

34. Tymoshchuk P. V., "A simplified continuous-time model of analogue K-winners-take-all neural circuit", in Proc. XI Int. Conf. "The Experience of Designing and Application of CAD Systems in Microelectronics", Polyana-Svalyava, Ukraine, February 23–25, 2011, pp. 121–125.

35. Cichocki A. and Unbehauen R., Neural networks for optimization and signal processing, Baffins Lane, Chichester: John Wiley & Sons, 1993.

P. Tymoshchuk, S. Shatny
Lviv Polytechnic National University

HARDWARE IMPLEMENTATION OF PARALLELIZED FUZZY ADAPTIVE RESONANCE THEORY NEURAL NETWORK

A hardware implementation design of parallelized fuzzy Adaptive Resonance Theory neural network is described and simulated. Parallel category choice and resonance are implemented in the network. Continuous-time and discrete-time winner-take-all neural circuits identifying the largest of M inputs are used as the winner-take-all units. The continuous-time circuit is described by a state equation with a discontinuous right-hand side. The discrete-time counterpart is governed by a difference equation. Corresponding functional block-diagrams of the circuits include M feed-forward hard-limiting neurons and one feedback neuron, which is used to compute the dynamic shift of inputs. The circuits combine arbitrary finite resolution of inputs, high convergence speed to the winner-take-all operation, low computational and hardware implementation complexity, and independence of initial conditions. The circuits are also used for finding elements of input vector with minimal/maximal values to normalize them in the range [0,1].

Key words: Functional block-diagram, fuzzy Adaptive Resonance Theory, neural network, category choice, winner-take-all, state equation with a discontinuous right-hand side.