

УДК 004.93

**А. О. Олійник, С. Ю. Скрупський, С. О. Субботін**  
Запорізький національний технічний університет, м. Запоріжжя  
кафедра програмних засобів

## ДОСЛІДЖЕННЯ ПРОЦЕСУ СИНТЕЗУ НЕЙРО-НЕЧІТКИХ МЕРЕЖ НА ВЕКТОРНОМУ ПРОЦЕСОРІ

### *Анотація*

*Олійник А. О., Скрупський С. Ю., Субботін С. О.. Дослідження процесу синтезу нейро-нечітких мереж на векторному процесорі. Виконано дослідження процесу побудови нейро-нечітких мереж. Проведено експерименти з використання паралельного методу синтезу нейро-нечітких моделей на графічному процесорі.*

**Ключові слова:** *графічний процесор, нейро-нечітка модель, прискорення.*

**Постановка проблеми.** Останнім часом широкого поширення набули нейромережеві технології штучного інтелекту [1–3]. Проте процес синтезу нейромережевих та нейро-нечітких моделей пов'язаний з проблемами вибору початкової точки пошуку в просторі ваг, необхідністю обчислення похідних цільової функції та низькою швидкістю збіжності. Тому актуальною є розробка нових методів синтезу нейромоделей, вільних від зазначених недоліків.

**Аналіз літератури.** Проведений аналіз методів синтезу нейро-нечітких моделей [1–3] показав, що методи градієнтної оптимізації [4], які традиційно використовуються для побудови нейромереж, вимагають диференційованості цільової функції та схильні до зациклювання у точках локальних екстремумів, що в деяких випадках ускладнює їх застосування на практиці. Тому для синтезу нейро-нечітких моделей доцільним є застосування методів стохастичної оптимізації [5–7], які не висувають додаткових вимог до виду та параметрів цільової функції. У роботі [8] запропоновано модель параметричного синтезу нейро-нечітких мереж в ярусно-паралельній формі.

**Мета статті** – дослідити ефективність застосування векторних процесорів для синтезу нейро-нечітких мереж на основі паралельних обчислень.

**Постановка задачі дослідження.** Нехай задано набір об'єктів  $S = \langle P, T \rangle$ , де  $P$  – множина характеристик об'єктів;  $T$  – множина значень відгуку. Множини значень  $P$  і  $T$  представляються, відповідно, у вигляді

матриці  $P = (p_{qm})_{QM}$  і вектора  $T = (t_q)_Q$ , де  $p_{qm}$  – значення  $m$ -го атрибута  $q$ -го об'єкта ( $m = 1, 2, \dots, M$ ,  $q = 1, 2, \dots, Q$ );  $t_q$  – значення відгуку  $q$ -го об'єкта;  $M$  – кількість атрибутів;  $Q$  – кількість об'єктів.

Тоді завдання параметричного синтезу нейро-нечіткої моделі  $NFN$  полягає в ідентифікації її параметрів таким чином, щоб забезпечувалося прийнятне значення заданого критерію якості  $G$ . Таким критерієм при навчанні нейро-нечітких моделей може бути використана, наприклад, помилка розпізнавання або середньоквадратична помилка.

**Розв'язання задачі та результати дослідження.** Перевага векторного процесора над центральним процесором полягає в тому, що більшу частину кристала центрального процесора займає керуючий пристрій, в той час як кристал векторного процесора практично повністю заповнений безпосередніми виконавцями обчислень. З результатів дослідження [8] видно, що запропонований метод, розроблений на основі моделі [8] добре розпаралелюється на SIMD-архітектурі, коли один потік інструкцій виконується одночасно над вектором даних. Тому доцільно виконати експерименти на векторному процесорі і порівняти ефективність його застосування з ефективністю застосування кластеру для реалізації запропонованого методу.

В якості представника SIMD-архітектури будемо використовувати графічний процесор (GPU) NVIDIA GTX285, програмований за технологією Nvidia CUDA [9]. Даний GPU містить 240 потокових процесорів, кожен з яких може виконувати окремих обчислювальний потік.

Для проведення експериментів було розроблено програмне забезпечення на мові C++ із застосуванням бібліотеки CUDA. На рис. 1 наведено усереднені часові витрати на побудову нейро-нечітких моделей за допомогою запропонованого методу на векторному процесорі.

На рис. 2 зображено графік прискорення обчислювального процесу на GPU. В якості часу виконання алгоритму на одному процесі взято час виконання на одному вузлі кластера. Це зроблено для забезпечення можливості зіставлення прискорення на кластері і на GPU.

З рис. 2 видно, що з ростом кількості потоків понад 160 прискорення починає зростати значно повільніше, ніж раніше. Це пов'язано з тим, що кількість потоків, які реалізують ітерації алгоритму насичують приблизно 120 потоків GPU, інші потоки векторного процесора залишаються незадіяними.

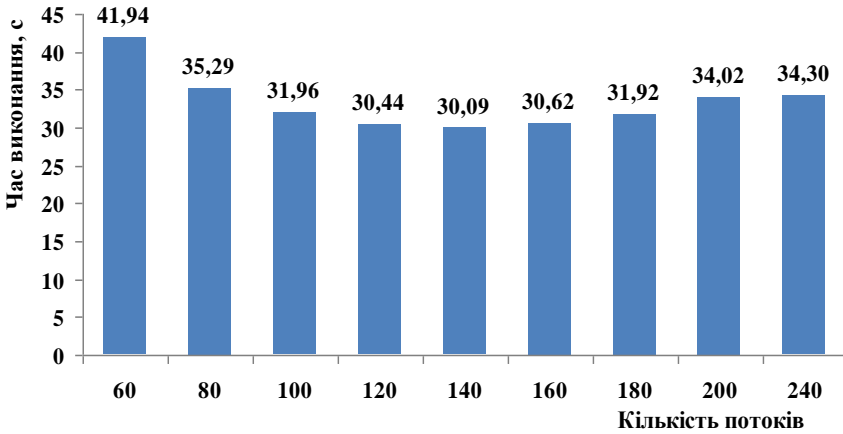


Рисунок 1 – Усереднені часові витрати на побудову нейро-нечітких моделей на векторному процесорі

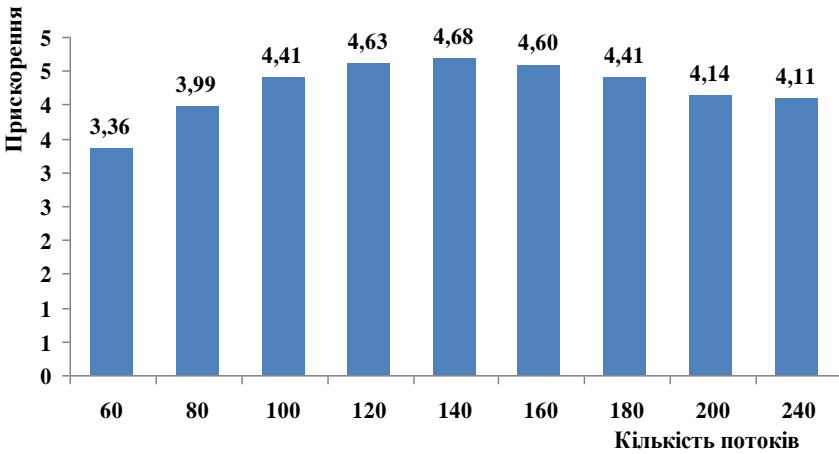


Рисунок 2 – Графік прискорення обчислювального процесу на GPU

В експериментах на GPU також була оцінена частка пересилок залежно від частоти міграції та відсотка мігруючих особин.

Результати оцінки частки пересилок при частоті міграцій 1/25 та 2% мігруючих особин [8] наведено в табл. 1 та на рис. 3.

Таблиця 1 – Частка пересилок між потоками GPU при частоті міграцій 1/25 та 2% мігруючих особин

Кількість потоків	Час пересилки, с	Частка пересилок
60	4,70	0,112
80	4,31	0,122
100	4,22	0,132
120	4,35	0,143
140	4,60	0,153
160	4,99	0,163
180	5,52	0,173
200	6,26	0,184
240	7,20	0,210

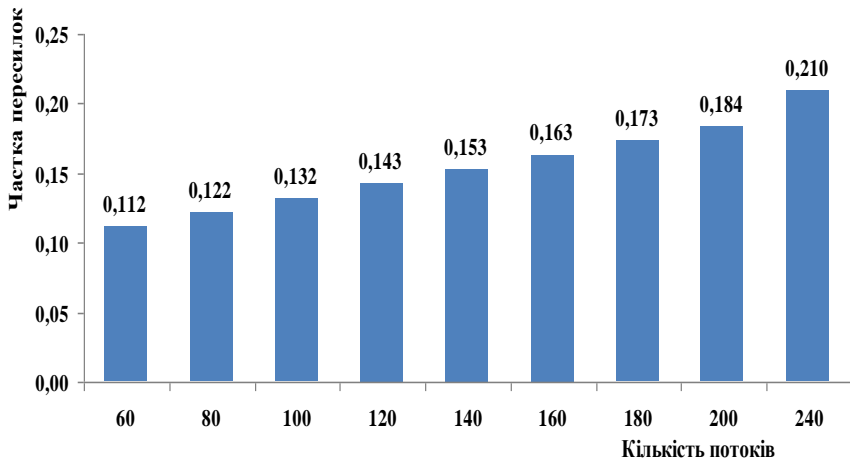


Рисунок 3 – Залежність частки пересилок між потоками при частоті міграцій 1/25 та 2% мігруючих особин

Результати експериментів показали, що час пересилок на GPU більше, ніж на цілому кластері, оскільки код, що містить пересилки, на GPU виконується послідовно, а не паралельно, що істотно знижує продуктивність системи в порівнянні з кластером.

**Висновки.** У роботі досліджено ефективність застосування векторних процесорів для синтезу нейро-нечітких мереж на основі паралельних обчислень. Проведені на векторному процесорі експерименти свідчать про прийнятні значення показників ефективності запропонованого паралельного методу параметричного синтезу нейро-нечітких моделей [8]. Результати експериментів з дослідження залежності показників ефективності та частки пересилок від кількості процесів показали, що на практиці даний метод доцільно застосовувати на векторних процесорах при задіяній кількості легковагих потоків до 120, що дозволяє отримати результати за прийнятний час.

### Список літератури

1. Rutkowski L. Flexible neuro-fuzzy systems : structures, learning and performance evaluation / L. Rutkowski. – Boston: Kluwer, 2004. – 276 p.
2. Бодяньський Є. В. Нейро-фаззі моделі в системах штучного інтелекту : навч. посібник / Є. В. Бодяньський, Є. І. Кучеренко. – Харків: ХНУРЕ, 2006. – 196 с.
3. Jang J. R. ANFIS: Adaptive-network-based fuzzy inference system / J. R. Jang // IEEE transactions on systems and cybernetics. – 1993. – Vol. 23. – P. 665–685.
4. Ravindran A. Engineering optimization: methods and applications / A. Ravindran, K. M. Ragsdell, G. V. Reklaitis. – New Jersey: John Wiley & Sons, 2006. – 688 p.
5. Скобцов Ю. А. Основы эволюционных вычислений / Ю. А. Скобцов. – Донецк: ДонНТУ, 2008. – 330 с.
6. Abraham A. Engineering Evolutionary Intelligent Systems / A. Abraham, C. Grosan, W. Pedrycz. – Berlin: Springer, 2008. – 444 p.
7. Tenne Y. Computational Intelligence in Expensive Optimization Problems / Y. Tenne, C.-K. Goh. – Berlin: Springer: 2010. – 800 p.
8. Олейник А. А. Модель параметрического синтеза нейро-нечетких сетей в ярусно-параллельной форме / А. А. Олейник, С. Ю. Скрупский, С. А. Субботин // Наукові праці Донецького національного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка". – 2014. – № 18 (206). – В печаті.
9. NVIDIA Corporation . NVIDIA CUDA Compute Unified Device Architecture 5.5 / NVIDIA Corporation . : NVIDIA , 2014 . – 117 p.