

НЕЙРОІНФОРМАТИКА ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

НЕЙРОИНФОРМАТИКА И ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

NEUROINFORMATICS AND INTELLIGENT SYSTEMS

УДК 004.93

Олійник А. О.

Канд. техн. наук, доцент, доцент кафедри програмних засобів, Запорізький національний технічний університет, Запоріжжя, Україна

СИНТЕЗ НЕЙРО-НЕЧІТКИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я ХВОРОГО НА ОСНОВІ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Вирішено завдання розробки математичного забезпечення для автоматизації процесу індивідуального прогнозування стану здоров'я хворого гіпертонічною хворобою. Об'єктом дослідження є процес синтезу моделей для медичного діагностування. Предмет дослідження становлять методи синтезу нейро-нечітких моделей для медичного діагностування. Мета роботи: підвищити ефективність процесу синтезу нейро-нечітких мереж для побудови діагностичних моделей на основі навчальних вибірок. Запропоновано стохастичний метод синтезу нейро-нечітких моделей на основі паралельних обчислень, що ґрунтується на стохастичному підході при пошуку значень параметрів, що настраюються, і полягає в розподілі найбільш ресурсномістких етапів по вузлах паралельної обчислювальної системи. Запропонований метод дозволяє скоротити час настроювання параметрів (значень вагових коефіцієнтів і параметрів функцій належності нейроелементів) синтезованих нейромоделей. Розроблено програмне забезпечення, що реалізує запропонований метод. Проведено експерименти з дослідження властивостей запропонованого методу. Результати експериментів дозволяють рекомендувати запропонований метод для використання на практиці.

Ключові слова: множина рішень, нейронна мережа, ознака, паралельне програмування, прогнозування, синтез моделі.

НОМЕНКЛАТУРА

$D(\chi_k^{(i)}, C_j)$ – відстань між рішенням $\chi_k^{(i)}$ і центром j -го кластера C_j ;

$Entr(\Delta p_{mn})$ – ентропія інтервалу Δp_{mn} ;

$g_{lk}^{(0)}$ – значення l -го елемента (параметра) в k -му рішенні;

$G_{k,norm}^{(i,j)}$ – нормоване значення цільової функції $G_k^{(i,j)}$ для множини $R^{(i,j)}$;

$Q|_{p_m \in \Delta p_{mn}}$ – кількість спостережень вибірки S , значення m -х ознак p_m яких потрапили в інтервал Δp_{mn} ;

M – кількість атрибутів;

N_g – кількість параметрів в рішенні $\chi_k^{(i)}$;

N_{pr} – кількість процесів, одночасно виконуваних в паралельній комп'ютерній системі;

$N_{Q,p_m \in \Delta p_{mn}}$ – кількість інтервалів розбиття (класів) вихідного параметра T фрагменту вибірки $S_{p_m \in \Delta p_{mn}}$, в якому значення m -х ознак p_m потрапили в інтервал Δp_{mn} ;

N_χ – кількість елементів множини $R^{(0)}$ (число випадково згенерованих рішень при ініціалізації);

p_{qm} – значення m -го атрибуту q -го об'єкта;

$P = (p_{qm})_{QM}$ – множина значень характеристик екземплярів вибірки;

Q – кількість об'єктів;

$Q|_{T=q; S_{p_m \in \Delta p_{mn}}}$ – кількість спостережень у множині $S_{p_m \in \Delta p_{mn}}$, в яких значення вихідного параметру T дорівнює q ($T = q$);

$R^{(i,j)}$ – j -та підмножина i -ї множини $R^{(i)}$;

t_q – значення відгуку q -го об'єкта;

T – множина значень відгуку;

$\chi_k^{(0)} = \{g_{1k}^{(0)}, g_{2k}^{(0)}, \dots, g_{N_g k}^{(0)}\}$ – k -те рішення у множині $R^{(0)}$.

ВСТУП

Артеріальна гіпертензія є широко поширеним захворюванням, яке може загрожувати життю та здоров'ю пацієнта [1]. На характер перебігу артеріальної гіпертензії

впливають різні фактори (погодні і кліматичні умови, супутні захворювання, а також стан здоров'я в попередні моменти часу) [2].

Для запобігання істотних стрибків тиску пацієнта, які можуть викликати погіршення стану пацієнта, а можливо і привести до летального наслідку, необхідно своєчасно прогнозувати розвиток гіпертензії в короткостроковому періоді часу (на найближчі половину доби або добу). Це дозволить своєчасно реалізувати профілактичні заходи, пов'язані з прийомом необхідних препаратів для запобігання очікуваних негативних наслідків.

Для прогнозування стану здоров'я хворого гіпертонічною хворобою потрібно мати модель, яка буде унікальною для кожного конкретного пацієнта. Побудова такої моделі потребує обробки великого масиву спостережень, розподілених у часі.

Перспективним засобом побудови моделей для індивідуального прогнозування є методи обчислювального інтелекту (нейро-нечіткі технології та методи стохастичною оптимізації) [3–7]. Штучні нейро-нечіткі мережі характеризуються здатністю до навчання за прецедентами, високими узагальнюючими властивостями, що обумовлює їх здатність отримувати знання з даних. Стохастичні методи, на відміну від широко використовуваних градієнтних методів оптимізації, не вимагають обчислень значень похідних цільової функції, дозволяють виходити з локальних екстремумів і синтезувати моделі на основі заданих вибірок даних.

Для врахування динаміки стану пацієнта необхідно з плином часу синтезувати нові моделі (періодично перебудовувати вже існуючі моделі) на основі постійно зростаючих масивів спостережень. Це в свою чергу вимагає використання великих обчислювальних ресурсів. Для подолання цих обмежень пропонується розпаралелити процес синтезу діагностичної моделі.

Метою даної роботи є розробка паралельного методу синтезу прогнозуючих моделей на основі нейро-нечітких мереж і стохастичного пошуку.

1 ПОСТАНОВА ЗАДАЧІ

Нехай є вибірка даних $S = \langle P, T \rangle$. Тоді задача синтезу нейро-нечіткої моделі NFN полягає в ідентифікації її параметрів таким чином, щоб забезпечувалося прийнятне значення заданого критерію якості G . Як цільовий критерій G при навчанні нейро-нечітких моделей можуть бути використані, наприклад, похибка розпізнання або середньоквадратична помилка.

Вихідна вибірка даних про стан здоров'я хворого гіпертонічною хворобою була отримана в м. Запоріжжя (Україна). Вибірка $S = \langle P, T \rangle$ містила спостереження за період з 2004 р. по 2014 р., де кожен екземпляр являв собою набір даних, що характеризують стан пацієнта в певну частину дня.

В якості об'єктивних клініко-лабораторних показників використовувалися: p_1 – спостережуваний артеріальний тиск (систолический та діастолічний, мм. рт. ст.); p_2 – пульс (ударів на хвилину); відомості про прийом ліків (p_3 – Амло (0 – немає, 1 – є), p_4 – Егілок (0 – немає, 1 – є); p_5 – Берліпріл (0 – немає, 1 – є)). В якості суб'єктивних показників використовувались характеристики самопо-

чуття (p_6 – наявність екстросистолю (0 – немає, 1 – є), p_7 – наявність болю в голові (0 – немає, 1 – є), p_8 – наявність болю в потилиці (0 – немає, 1 – є), p_9 – наявність пульсації (0 – немає, 1 – є), p_{10} – наявність болю в лівому боці (0 – немає, 1 – є), p_{11} – наявність болю в області серця (0 – немає, 1 – є), p_{12} – нестача повітря (0 – немає, 1 – є), p_{13} – наявність болю в животі (0 – немає, 1 – є), p_{14} – загальна слабкість (0 – немає, 1 – є)). В якості метеорологічних характеристик [8] використовувалися (p_{15} – температура повітря (°C), p_{16} – атмосферний тиск (мм. рт. ст.), p_{17} – тип хмарності (0 – немає, 1 – мала, 2 – хмарно, 3 – похмуро), p_{18} – наявність грози (0 – немає, 1 – є), p_{19} – напрямок вітру (0 – штиль, 1 – північний, 2 – північно-східний, 3 – східний, 4 – південно-східний, 5 – південний, 6 – південно-західний, 7 – західний, 8 – північно-західний), p_{20} – швидкість вітру (м/с), p_{21} – дані сонячної активності (Mg II індекс [9]). В якості тимчасових характеристик використовувалися: дата (рік, місяць, день), код дня тижня (p_{22}), час (година) спостереження (p_{23}), код частини дня (0 – ранок, 1 – вечір) (p_{24}).

Фрагмент вихідних даних в графічному виді представлений на рис. 1.

Отримані спостереження методом «вікон» були використані для формування вибірки для вирішення задачі якісного прогнозування стану пацієнта на найближчу другу половину доби за даними попередніх спостережень: в якості вхідних ознак використовувалися дані за попередню (ранок та вечір) і поточну добу (ранок), а в якості виходу – стан пацієнта ввечері в поточну добу (0 – нормальний, 1 – погіршення стану, що супроводжується підвищенням артеріального тиску).

2 ОГЛЯД ЛІТЕРАТУРИ

У найпростішому випадку нейро-нечітка модель може розглядатися як передоброблювач, де механізм навчання штучної нейронної мережі визначає функції належності або нечіткі правила системи нечіткого виведення за навчальними даними. Як тільки параметри системи нечіткого виведення визначено, нейронна мережа припиняє використовуватися. База правил нечіткої системи звичайно визначається на основі карт, що самоорганізуються, або методів нечіткої кластеризації. Функції належності звичайно апроксимуються нейронною мережею за навчальними даними. Найбільш відомими прикладами паралельних нейро-нечітких систем є: нечітка асоціативна пам'ять Коско (В. Kosko), виділення нечітких правил на основі карт Педрича (W. Pedrycz), що самоорганізуються, та системи Номури (Н. Nomura), здатні до навчання параметрів нечітких множин [3–5].

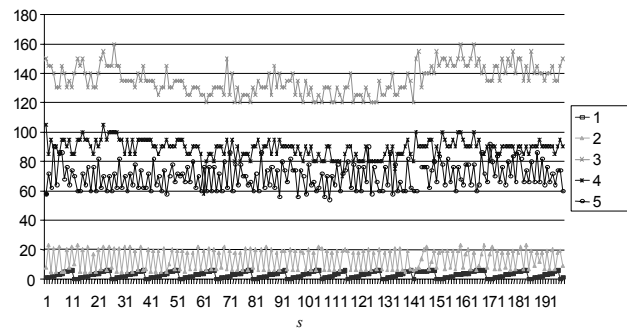


Рисунок 1 – Візуалізація фрагменту вибірки даних: 1 – день тижня, 2 – час, 3 – систолічний тиск, 4 – діастолічний тиск, 5 – пульс

В інтегрованій моделі нейро-нечітких мереж для визначення параметрів системи нечіткого виведення використовуються методи навчання нейронних мереж. Інтегровані нейро-нечіткі системи розподіляють структури даних і подання знань. Система нечіткого виведення може використовувати людські експертні знання, зберігаючи їхні істотні компоненти в базі правил і базі даних, та виконувати нечітке виведення для одержання вихідного значення. Формування правил і відповідних функцій належності сильно залежить від апріорного знання щодо розглянутої системи. Однак універсальний спосіб перетворення дослідних знань людських експертів у базу знань системи нечіткого виведення є невідомим. Є також потреба в адаптованості або деяких методах навчання для одержання виходів з необхідним рівнем точності. З іншого боку, механізм навчання нейронних мереж не покладається на людську експертизу. Через однорідну структуру нейронних мереж, з них складно витягти структуроване знання. Ваги нейронної мережі являють собою коефіцієнти гіперплощини, що розподіляє вхідний простір на області з різними значеннями виходу. Якщо ми можемо візуалізувати цю структуру гіперплощини за навчальними даними, тоді наступні процедури навчання в нейронній мережі можуть бути виключені.

Найбільш загальний спосіб застосування методу навчання до нечіткої системи полягає в тому, щоб подати її у вигляді архітектури, подібної нейронній мережі. Однак звичайні (градієнтні) методи навчання нейронних мереж не можуть безпосередньо застосовуватися до такої системи, оскільки функції, використовувані в процесі виведення звичайно є недиференційовані. Ця проблема може бути вирішена шляхом використання диференційованих функцій у системі виведення або використання стохастичних методів навчання нейромереж [5–7].

3 МАТЕРІАЛИ ТА МЕТОДИ

Оскільки навчальна вибірка характеризується високими значеннями кількості ознак і екземплярів, процес побудови моделі вимагає значних обчислювальних і часових ресурсів. Це обумовлює доцільність розпаралелювання обчислювальних операцій при синтезі розпізнавальної моделі.

Для синтезу моделі прогнозування стану здоров'я хворого гіпертонічною хворобою розроблений паралельний стохастичний метод. Запропонований метод заснований на стохастичному підході [6, 7, 10, 11], передбачає розпаралелювання основних обчислювальних операцій і використовує апріорну інформацію про значущість ознак. Це дозволяє використовувати можливості багатоядерних процесорів при синтезі нейро-нечітких моделей прогнозування стану здоров'я хворого гіпертонічною хворобою, а також скоротити час на побудову таких моделей.

У розробленому методі на етапі ініціалізація початкової множини рішень відбувається формування множини $R^{(0)} = \{\chi_1^{(0)}, \chi_2^{(0)}, \dots, \chi_{N_\chi}^{(0)}\}$ структур виду $\chi_k^{(0)} = \{g_{1k}^{(0)}, g_{2k}^{(0)}, \dots, g_{N_g k}^{(0)}\}$, що визначають основні параметри (значення вагових коефіцієнтів нейронів і параметрів функцій активації) синтезованих нейро-нечітких моделей.

Для наближення початкових точок пошуку $\chi_k^{(0)} = \{g_{1k}^{(0)}, g_{2k}^{(0)}, \dots, g_{N_g k}^{(0)}\}$ ($k = 1, 2, \dots, N_\chi$) до оптималь-

них множини $R^{(0)}$, на відміну до відомих методів стохастичного пошуку [7, 10, 11], будемо формувати з урахуванням апріорної інформації про навчальну вибірку $S = \langle P, T \rangle$. Для цього діапазон значень $\Delta p_m = [p_{m \min}; p_{m \max}]$ кожної m -ї ознаки p_m ($m = 1, 2, \dots, M$) розбивається на задану кількість інтервалів N_{int} . Таким чином формуються інтервали $\Delta p_{mn} = [p_{mn \min}; p_{mn \max}]$ ($n = 1, 2, \dots, N_{\text{int}}$), на підставі границь яких визначаються параметри нечітких термів.

При обчисленні значень параметрів функцій належності нейро-нечіткої моделі будемо враховувати значущість n -го терму Δp_{mn} m -ї ознаки p_m для розпізнавання екземплярів вибірки $S = \langle P, T \rangle$. Значущість V_{mn} n -го терму m -ї ознаки визначимо як добуток величин

$\frac{Q|_{p_m \in \Delta p_{mn}}}{Q}$ та $(1 - \text{Entr}(\Delta p_{mn}))$, що характеризують, відповідно щільність розташування екземплярів множини S в інтервалі Δp_{mn} ознаки p_m і ступінь впливу терму Δp_{mn} на значення вихідного параметру T (1):

$$V_{mn} = \frac{Q|_{p_m \in \Delta p_{mn}}}{Q} (1 - \text{Entr}(\Delta p_{mn})). \quad (1)$$

Ентропія $\text{Entr}(\Delta p_{mn})$ інтервалу Δp_{mn} характеризує міру невизначеності вихідного параметра T за умови, що $p_m \in \Delta p_{mn}$, та визначається за формулою (2):

$$\text{Entr}(\Delta p_{mn}) = - \sum_{q=1}^{N_{Q, p_m \in \Delta p_{mn}}} \rho_q \log \rho_q. \quad (2)$$

Ймовірність ρ_q того, що вихідний параметр T прийме значення q в множині $S_{p_m \in \Delta p_{mn}}$ (за умови, що $p_m \in \Delta p_{mn}$) можна визначити за формулою (3):

$$\rho_q = \frac{Q|_{T=q; S_{p_m \in \Delta p_{mn}}}}{Q|_{p_m \in \Delta p_{mn}}}. \quad (3)$$

Найбільш ресурсомістким, як правило, є етап оцінювання поточної множини рішень $R^{(i)}$, на якому витрачається велика кількість комп'ютерних та часових ресурсів при обчисленні значень цільової функції G для кожного k -го ($k = 1, 2, \dots, N_\chi$) рішення $\chi_k^{(i)}$: $G_k^{(i)} = G(\chi_k^{(i)})$. Оскільки даний етап має високу обчислювальну складність, виконується повільно і не вимагає обміну даними між рішеннями $\chi_k^{(i)}$, його раціонально виконувати паралельно. Оцінювання рішень $\chi_k^{(0,j)}$ з множини $R^{(0)}$ виконується шляхом перетворення $\chi_k^{(0,j)} \rightarrow NFN_k^{(0,j)}$, в результаті якого зі структури $\chi_k^{(0,j)}$ формується нейро-нечітка ме-

режа $NFN_k^{(0,j)}$. В якості критерію оцінювання $G_k^{(0,j)}$, що характеризує рішення $\chi_k^{(0,j)}$, використовується помилка розпізнавання (якщо вихід T в множині S є дискретним) або середньоквадратична помилка (якщо вихід T приймає дійсні значення). Це вимагає обчислення значень вихідного параметра $T_q(NFN_k^{(0,j)})$ за синтезованою моделлю $NFN_k^{(0,j)}$ для кожного q -го спостереження множини $S = \langle P, T \rangle$. Отримана таким чином оцінка $G(NFN_k^{(0,j)})$ буде характеризувати пристосованість $G_k^{(0,j)}$ рішення $\chi_k^{(0,j)}$ в множині $R^{(0)}$: $G_k^{(0,j)} = G(\chi_k^{(0,j)}) = G(NFN_k^{(0,j)})$, що надалі дозволить відбирати найбільш пристосовані рішення для формування нових множин $R^{(i+1)}$.

Важливим етапом є створення нової множини рішень $R^{(i+1)}$. З метою більш детального дослідження областей локальних оптимумів доцільним є розбиття поточної множини рішень $R^{(i)}$ на підмножині $R^{(i,j)}$ з подальшим пошуком оптимуму в кожній з них (4):

$$R^{(i)} \rightarrow \{R^{(i,1)}, R^{(i,2)}, \dots, R^{(i,N_{pr})}\}. \quad (4)$$

Розбиття (4) пропонується виконувати з урахуванням апріорної інформації про розташування рішень $\chi_k^{(i)}$ в просторі елементів g_l ($l = 1, 2, \dots, N_g$). Такий підхід, на відміну від застосування острівної моделі еволюційного пошуку [7, 12], що припускає випадкове формування підпопуляцій $R^{(i,j)}$, дозволяє враховувати інформацію про просторове розташування рішень $\chi_k^{(i)}$ в множині $R^{(i)}$ і більш детально дослідити області можливих оптимумів. Кількість рішень $|R^{(i,j)}|$ в кожній підмножині $R^{(i,j)}$ визначається як відношення загальної кількості рішень N_χ в множині $R^{(i)}$ до кількості процесів N_{pr} (5):

$$|R^{(i,j)}| = N_\chi / N_{pr}. \quad (5)$$

Таким чином, задача розбиття (4) зводиться до необхідності формування N_{pr} підмножин $R^{(i,j)}$ (кожна з яких складається з $|R^{(i,j)}|$ рішень) в заданому просторі елементів g_l ($l = 1, 2, \dots, N_g$) (6):

$$R^{(i,j)} = \left\{ \chi_1^{(i,j)}, \chi_2^{(i,j)}, \dots, \chi_{|R^{(i,j)}|}^{(i,j)} \right\}, \quad j = 1, 2, \dots, N_{pr}. \quad (6)$$

Розбиття $R^{(i)} \rightarrow \{R^{(i,1)}, R^{(i,2)}, \dots, R^{(i,N_{pr})}\}$ пропонується виконувати за допомогою модифікації методу нечітких c -середніх (FCM, Fuzzy c -means) [13–15], оскільки він передбачає можливість віднесення одного екземпляру

(рішення $\chi_k^{(i)}$) до різних кластерів з різним ступенем належності, а також може бути використаний для побудови нейро-нечітких систем при визначенні нечітких множин [13, 16]. На відміну від базового методу нечітких c -середніх [15], в якому поточне розбиття на початковій ітерації отримують шляхом випадкової генерації елементів u_{kj} , що визначають належність k -го екземпляра до j -го кластера, в розробленому методі пропонується враховувати інформацію про розташування параметрів g_{lk} , що настроюються. Для визначення початкових значень величин u_{kj} обчислимо ранги $Rg(g_{lk}^{(i)})$ рішень $\chi_k^{(i)}$ за l -ю ознакою, що відповідають номерам інтервалів розбиття діапазону значень параметра g_{lk} .

Для визначення початкових значень u_{kj} , що відповідають належності k -го екземпляра до j -го кластера, будемо використовувати формулу (7):

$$u_{kj} = \frac{N(Rg(g_{lk}^{(i)}) = j)}{N_{pr}}, \quad (7)$$

де $N(Rg(g_{lk}^{(i)}) = j)$ – кількість рангів $Rg(g_{lk}^{(i)})$, які дорівнюють j . Формула (7) забезпечую виконання умови $\sum_{k=1}^{N_\chi} u_{kj} = 1$.

Далі для кожного j -го кластера визначається його центр $C_j = \{C_{1j}, C_{2j}, \dots, C_{N_g j}\}$, де C_{lj} – l -та координата центра j -го кластера, що обчислюється за формулою (8):

$$C_{lj} = \frac{1}{\sum_{k=1}^{N_\chi} (u_{kj})^{mp}} \sum_{k=1}^{N_\chi} (u_{kj})^{mp} g_{lk}, \quad (8)$$

де mp – параметр методу, як правило, $mp = 2$.

Потім обчислюється значення функції $J(R^{(i)}, u^{(i)}, C^{(i)})$ (9), що характеризує якість нечіткого розбиття:

$$J(R^{(i)}, u^{(i)}, C^{(i)}) = \sum_{k=1}^{N_\chi} \sum_{j=1}^{N_{pr}} (u_{kj})^{mp} D^2(\chi_k^{(i)}, C_j). \quad (9)$$

Як правило, відстань $D(\chi_k^{(i)}, C_j)$ між рішенням $\chi_k^{(i)}$ і центром j -го кластера C_j обчислюється на основі Евклідової метрики. Після цього формується нове нечітке розбиття шляхом перерахунку значень u_{kj} , використовуючи формулу (10):

$$u_{kj} = \left(\frac{\sum_{t=1}^{N_{pr}} \left(\frac{D(\chi_k^{(i)}, C_j)}{D(\chi_k^{(i)}, C_t)} \right)^{\frac{2}{mp-1}}}{\sum_{t=1}^{N_{pr}} \left(\frac{D(\chi_k^{(i)}, C_t)}{D(\chi_k^{(i)}, C_t)} \right)^{\frac{2}{mp-1}}} \right)^{-1}. \quad (10)$$

При цьому, якщо $D(\chi_k^{(i)}, C_j) = 0$, то $\chi_k^{(i)}$ є центром j -го кластера, тому встановлюються такі значення параметрів u_{kj} : $u_{kj} = 1$, для всіх інших кластерів: $u_{kj} = 0, k' \neq k$. Потім, для отримання нечіткого розбиття u_{kj} ($k = 1, 2, \dots, N_\chi$, $j = 1, 2, \dots, N_{pr}$) використовуючи формули (8)–(9) розраховуються нові центри нечітких кластерів $C_j = \{C_{1j}, C_{2j}, \dots, C_{N_{\chi j}}\}$ і значення критерію $J(R^{(i)}, u^{(i)}, C^{(i)})$. Процес продовжується, доки кількість ітерацій не перевищить максимально допустимого або доки зміна значення критерію $J(R^{(i)}, u^{(i)}, C^{(i)})$ носить суттєвий характер (11):

$$|J_{old}(R^{(i)}, u^{(i)}, C^{(i)}) - J(R^{(i)}, u^{(i)}, C^{(i)})| \leq \epsilon_J, \quad (11)$$

де $J_{old}(R^{(i)}, u^{(i)}, C^{(i)})$ та $J(R^{(i)}, u^{(i)}, C^{(i)})$ – значення критерію якості розбиття J на попередній та поточній ітераціях, відповідно; ϵ_J – мінімально прийнятна зміна значення критерію J .

Після отримання прийнятних значень величин u_{kj} , що характеризують належність k -го екземпляра $\chi_k^{(i)}$ до j -го кластера виконується формування підмножин $R^{(i,j)}$ ($j = 1, 2, \dots, N_{pr}$). При цьому пропонується враховувати як значення належності u_{kj} до кожного з кластерів $R^{(i,j)}$, так і значення цільової функції $G_k^{(i)} = G(\chi_k^{(i)})$. Це дозволить підвищити ймовірність відбору при формуванні підмножин $R^{(i,j)}$ таких рішень $\chi_k^{(i)}$, які геометрично відповідають j -му кластеру (просторово розташовані близько до його центру) і володіють при цьому прийнятною пристосованістю $G_k^{(i)}$. Рішення $\chi_k^{(i)}$ будемо відносити в підмножини $R^{(i,j)}$ ($R^{(i,j)} = R^{(i,j)} \cup \chi_k^{(i)}$) при виконанні умови (12):

$$G_k^{(i)} + u_{kj} \leq \text{rand}[0; 2]. \quad (12)$$

Перегляд множини $R^{(i)}$ продовжується та повторюється доки не будуть послідовно сформовані всі підмножини $R^{(i,j)}$, кожна з яких складається з N_χ / N_{pr} рішень.

Після розбиття в головному процесі множини $R^{(i)}$ на підмножини $\{R^{(i,1)}, R^{(i,2)}, \dots, R^{(i,N_{pr})}\}$ виконується їх передача на інші процеси для паралельного стохастичного пошуку екстремумів.

Для формування нового набору $R^{(i,j)}$ на j -му процесі будемо відбирати рішення $\chi_k^{(i,j)} \in R^{(i,j)}$, виходячи зі значень їх цільової функції $G_k^{(i,j)} = G(\chi_k^{(i,j)})$. Рішення $\chi_k^{(i,j)} \in R^{(i,j)}$ пропонується вводити в набір рішень $R_{selected}^{(i,j)}$, допущених для формування нового покоління $R^{(i,j)}$ при виконанні умови (13):

$$G_{k,norm}^{(i,j)} \leq \text{rand}[0; 1]. \quad (13)$$

Таким чином, чим меншою є помилка $G_k^{(i,j)}$ синтезованої мережі $NFN_k^{(i,j)} = NFN(\chi_k^{(i,j)})$, тим вищою є ймовірність рішення $\chi_k^{(i,j)} \in R^{(i,j)}$ бути відібраним для формування нової множини $R^{(i,j)}$.

Нову множину рішень $R^{(i,j)}$ аналогічно еволюційному підходу до пошуку оптимальних рішень [11] пропонується формувати з найбільш пристосованих наборів $\chi_k^{(i,j)}$ (елітних), а також наборів, отриманих в результаті застосування еволюційних операторів схрещування та мутації. Для створення нової множини рішень пропонується використовувати оператори, розроблені в [17].

Випадковий пошук з адаптацією в кожній j -й підмножині $R^{(i,j)}$ пропонується виконувати на j -му процесі паралельної комп'ютерної системи ($j = 1, 2, \dots, N_{pr}$) протягом N_{it} ітерацій. Після виконання N_{it} ітерацій випадкового пошуку в кожній підмножині $R^{(i,j)}$ виконується їх об'єднання $R^{(i)} = R^{(i,1)} \cup R^{(i,2)} \cup \dots \cup R^{(i,N_{pr})}$ в єдину популяцію з подальшим проведенням імовірнісної оптимізації над об'єднаною множиною. Це дозволяє виявити нові області, що містять локальні (можливо, і глобальний) оптимум. З метою скорочення часу імовірнісної оптимізації при роботі з об'єднаною множиною рішень пропонується обчислення значень цільової функції $G_k^{(i)}$ рішень $\chi_k^{(i)}$ виконувати на процесах в паралельній системі (розпаралелити).

Виходячи з зазначеного вище, представимо граф процесу параметричного синтезу нейро-нечітких моделей в ярусно-паралельній формі (рис. 2). Починаючи з ярусу перевірки критеріїв завершення процес виконується ітеративно.

На рис. 2 операція $RS(R)$ представляє собою випадковий пошук в множині рішень R . Остання операція на графі – $RS(R^{(i)})$ виконується паралельно. Одна ітерація процесу випадкового пошуку $RS(R^{(i)})$ в ярусно-паралельній формі представлена на рис. 3.

Пунктирні лінії на рис. 2 і 3 відокремлюють яруси графа один від одного. Операції на одному ярусі графа виконуються паралельно. Дуги графа демонструють залежності між операціями. Так, не можна приступити до виконання будь-якої операції доки не завершаться інші операції, дуги з яких орієнтовані в вершини розглянутих операцій.

Перехід між ярусами, при якому від однієї вершини графа дуги орієнтовані в кілька вершин наступного ярусу, пов'язаний з розсилкою (Scatter) рішень по процесах. Перехід іншого типу, при якому дуги орієнтовані від декількох вершин одного ярусу до однієї вершині наступного ярусу, виконується за допомогою збору (Gather) рішень в головному процесі. Як можна побачити з рис. 2 і 3, найбільш ресурсомісткі операції, а саме ініціалізація початкової множини рішень, оцінювання поточної множини рішень, випадковий пошук в підмножині рішень, виконуються одночасно на процесах паралельної системи.

Запропонований метод синтезу нейро-нечітких мереж на основі паралельних обчислень ґрунтується на стохастичному підході при пошуку значень параметрів, що настраюються, і полягає в розподілі найбільш ресурсомістких етапів по вузлах паралельної обчислювальної системи, що дозволяє скоротити час налаштування параметрів (значень вагових коефіцієнтів і параметрів функцій належності нейроелементів) синтезованих нейромоделей.

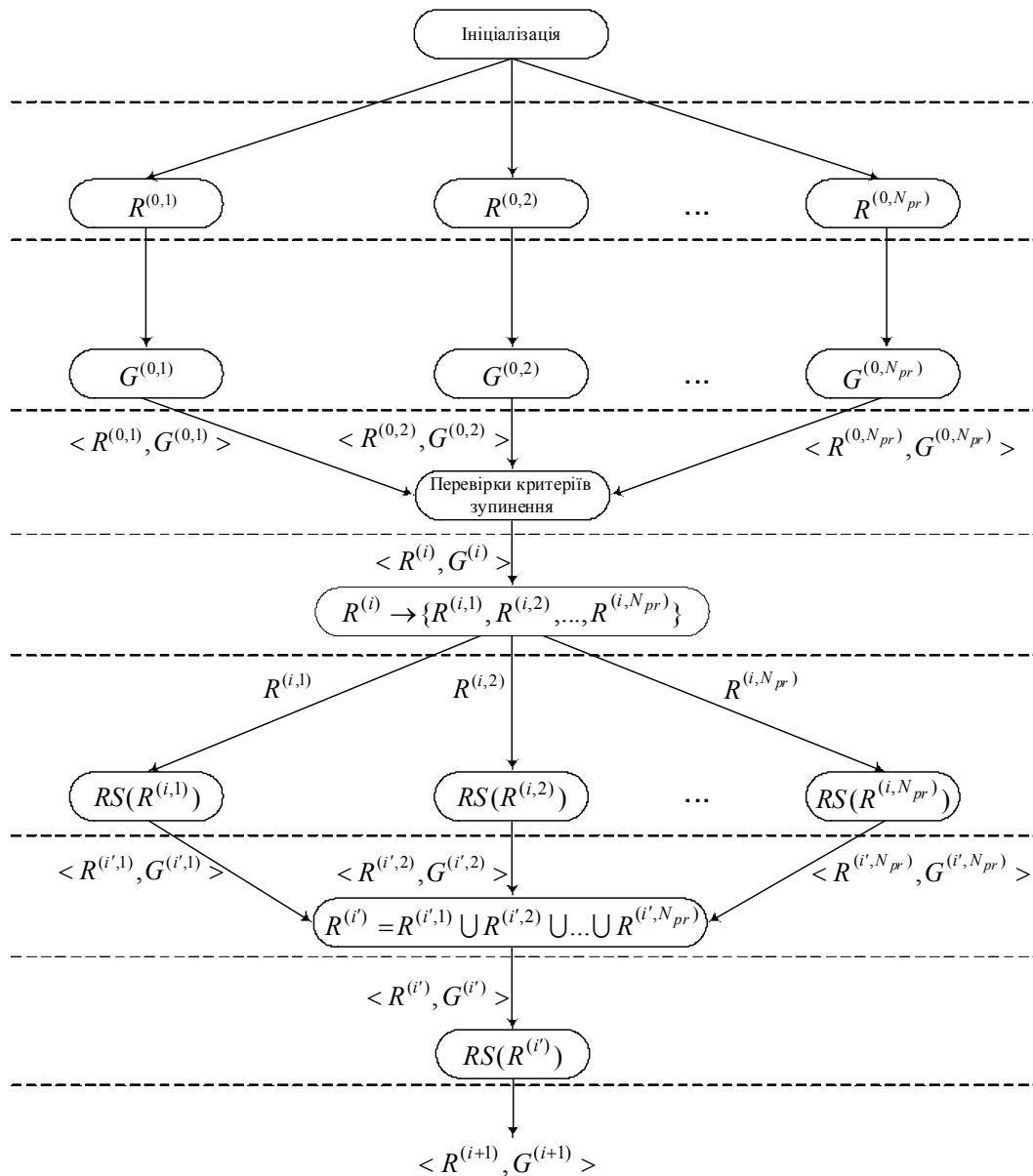


Рисунок 2 – Граф процесу параметричного синтезу нейро-нечітких мереж в ярусно-паралельній формі

4 ЕКСПЕРИМЕНТИ

Для виконання експериментального дослідження запропонованого методу синтезу нейро-нечітких мереж і вирішення задачі індивідуального прогнозування стану здоров'я хворого гіпертонічною хворобою був використаний кластер Інституту проблем моделювання в енергетику імені Г. Е. Пухова НАН України м. Київ. Було задіяно 16 вузлів наступної конфігурації: процесор – Intel Xeon 5405, оперативна пам'ять – 4×2 ГБ DDR-2 на кожен вузол, комунікаційне середовище InfiniBand 20Гб/с. На вузлах кластера встановлене middleware Torque і OMPI.

Для проведення експериментів розроблено програмне забезпечення мовою C++ із застосуванням бібліотеки MPI [18–20]. Для оцінювання витраченого на виконання методу часу використана бібліотечна функція MPI_Wtime(). Експерименти були виконані по 3 рази на одному, двох, чотирьох, восьми і шістнадцяти ядрах задіяних процесорів.

Як аналог запропонованому методу була узятя острівна модель еволюційного пошуку (Island Genetic Algorithm, IGA) [11, 12].

5 РЕЗУЛЬТАТИ

Результати експериментів наведено на рис. 4–10 та у табл. 1–2. На рис. 4 приведені усереднені часові витрати на виконання запропонованого методу і методу IGA на кластері.

Для встановлення причини нелінійного росту прискорення було виміряно комунікаційні витрати, а саме: пересилання між ядрами процесорів і синхронізацію потоків. Отримані результати наведено в таблиці 1 і на рис. 6.

Графік ефективності кластерної системи, у якій виконувалася реалізація запропонованого методу і методу IGA, представлений на рис. 7.

Запропонований метод по своїй суті добре реалізується на SIMD-системах: кожне ядро виконує ті самі інструкції для кожної реалізації хромосом, тому були проведені експерименти на графічному процесорі (GPU) NVIDIA GTX 285+, що програмувався за технологією CUDA [21]. При цьому задавалася кількість задіяних по-

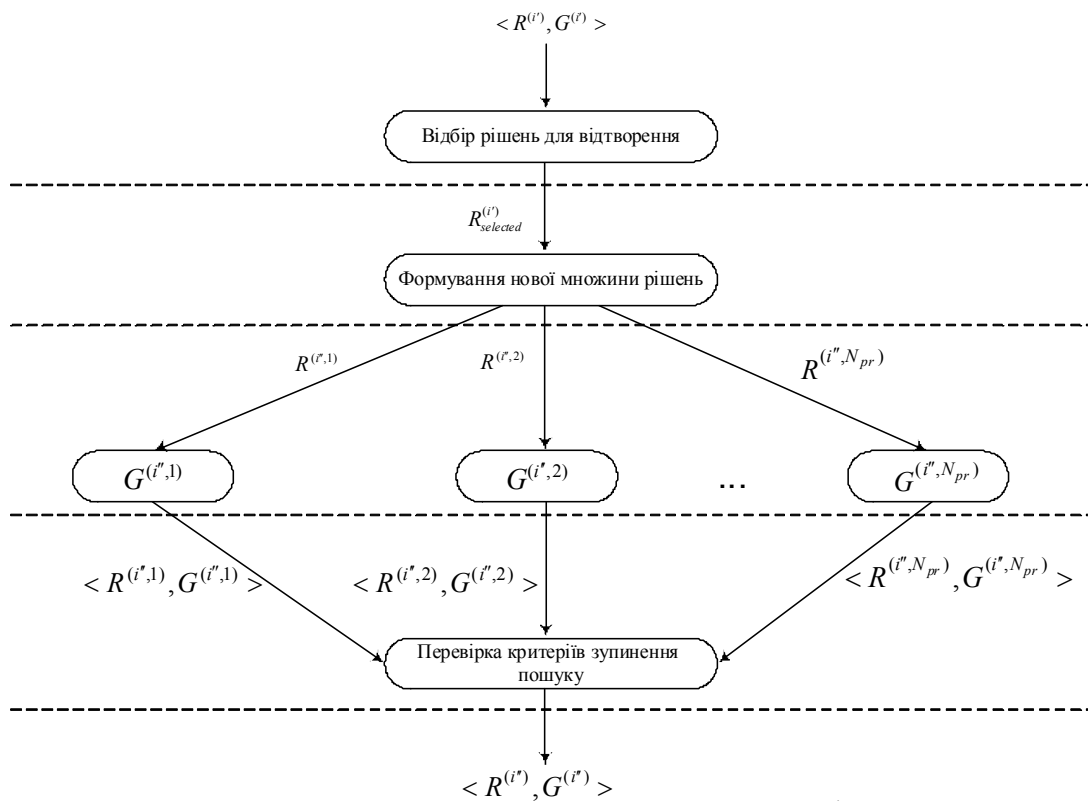


Рисунок 3 – Граф однієї ітерації випадкового пошуку $RS(R^{(i')})$

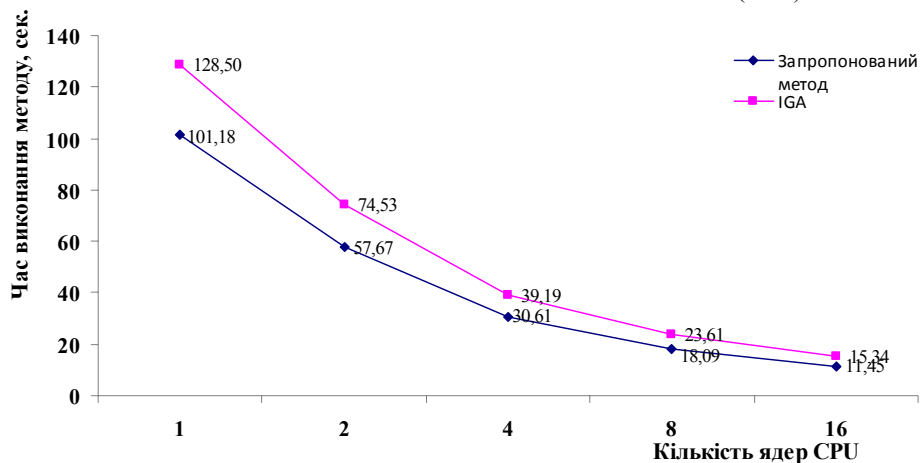


Рисунок 4 – Усереднений час виконання запропонованого методу і методу IGA на кластері

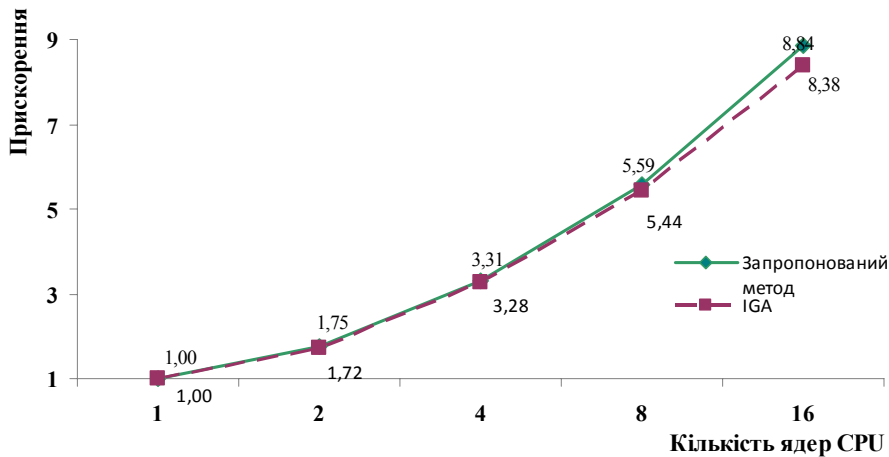


Рисунок 5 – Графік прискорення обчислювального процесу на кластері

Таблиця 1 – Комунікаційні витрати обчислювального процесу, що реалізує запропонований метод

Кількість задіяних ядер процесорів	Час пересилань та синхронізацій, с		Частка пересилань та синхронізацій	
	Запропонований метод	IGA	Запропонований метод	IGA
1	0	0	0	0
2	8,07	11,92	0,14	0,16
4	6,43	8,62	0,21	0,22
8	7,78	11,10	0,43	0,47
16	9,27	13,96	0,81	0,91

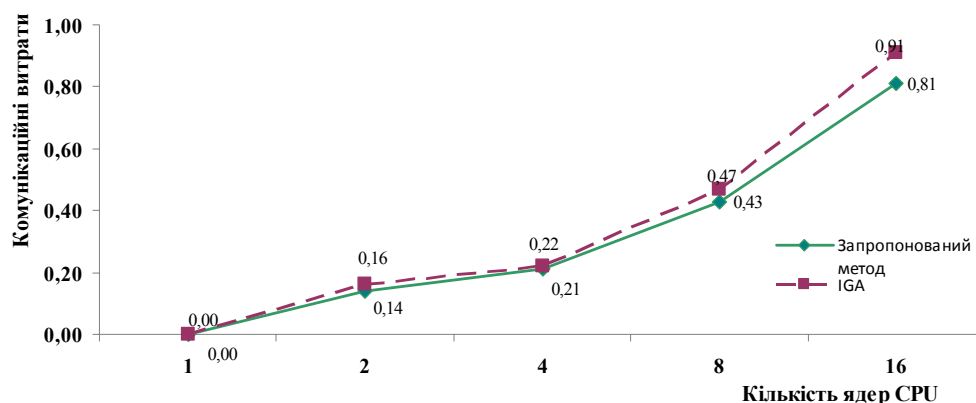


Рисунок 6 – Графік залежності комунікаційних витрат від числа задіяних ядер процесорів

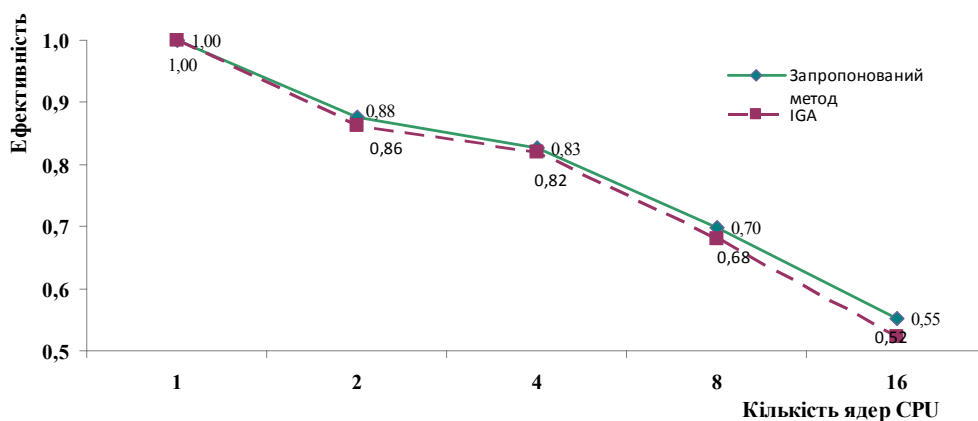


Рисунок 7 – Ефективність кластерної системи, що реалізує запропонований метод і метод IGA

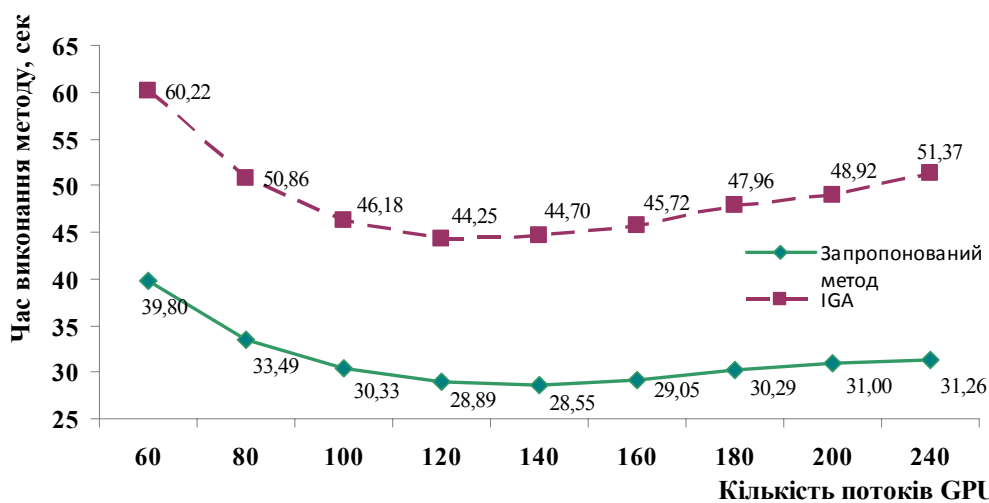


Рисунок 8 – Усереднений час виконання запропонованого методу і методу IGA на графічному процесорі

токів GPU і вимірювався час виконання методу за допомогою убудованих у пакет CUDA функцій. Експерименти для запропонованого методу і методу IGA були повторені по 2 рази на 60, 80, 100, 120, 140, 160, 180, 200 і 240 потоках. Усереднений час виконання методів відображено на рис. 8.

Графіки прискорення і комунікаційних витрат, що ілюструють роботу досліджуваних методів при використанні графічного процесора (GPU), наведено на рис. 9 і 10, відповідно.

У таблиці 2 наведено комунікаційні витрати обчислювального процесу, що реалізує запропонований метод на графічному процесорі.

6 ОБГОВОРЕННЯ

Як видно з рис. 4 і 5, запропонований метод у порівнянні з методом IGA дозволяє швидше виконати випадковий пошук, оскільки застосування модифікованих операторів створення нових рішень зменшує загальну кількість виконуваних на процесах операцій [17]. Прискорення обчислювального процесу, що реалізує запропонований метод і метод IGA, росте нелінійно зі збільшенням числа задіяних ядер процесорів.

Важливо відзначити, що обидва методи (запропонований метод і метод IGA) дозволили забезпечити побудову нейро-нечітких моделей із прийнятною точністю (при проведенні експериментів необхідно було синтезувати моделі, помилка прогнозування яких не перевищує 0,01). Помилка прогнозування на тестових даних склала 0,017 і 0,021 для запропонованого методу і методу IGA,

відповідно, що також є припустимим для розв'язуваної задачі індивідуального прогнозування стану здоров'я хворого гіпертонічною хворобою.

Частка комунікаційних витрат (рис. 6, табл. 1) в обчислювальному процесі, що реалізує запропонований метод і метод IGA, істотно росте зі збільшенням кількості ядер задіяних процесорів. При цьому на більше, ніж восьми ядрах, частка комунікаційних витрат значно перевищує частку цільових обчислень. Тому при реалізації запропонованого методу на кластері доцільно застосовувати чотири-шість ядер процесорів, щоб збалансувати відношення витрат до ефективності системи. Застосування методу IGA на кластері так само є виправданим, проте, при вирішенні поставленої задачі IGA затратив більше часу на пересилання в порівнянні з запропонованим методом.

З рис. 7 видно, що обидва методи продемонстрували прийнятну ефективність у кластерній системі. Для запропонованого методу (рис. 8) збільшення числа потоків понад 140 приводить до уповільнення обчислювального процесу. Це можна пояснити ростом комунікаційних витрат, а також простоем незадіяних потоків. Для методу IGA збільшення кількості потоків уже понад 120 сповільнює обчислювальний процес.

Дані, відображені у табл. 2 та на рис. 10, демонструють істотне зниження комунікаційних витрат при виконанні запропонованого методу і методу IGA на GPU у порівнянні з кластером, комунікаційні витрати якого подані в таблиці 1.

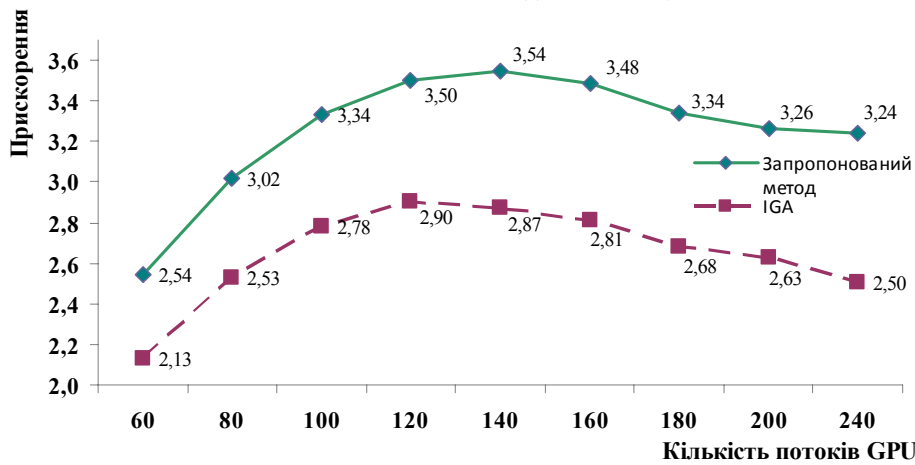


Рисунок 9 – Графік прискорення реалізації запропонованого методу і методу IGA на графічному процесорі

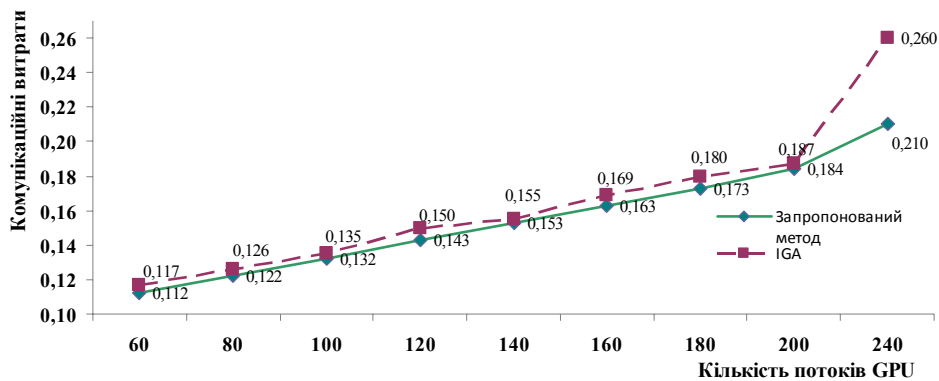


Рисунок 10 – Графік комунікаційних витрат на реалізацію запропонованого методу і методу IGA на GPU

Таблиця 2 – Комунікаційні витрати обчислювального процесу, що реалізує запропонований метод на GPU

Кількість задіяних потоків GPU	Час пересилань та синхронізацій, с		Частка пересилань та синхронізацій, с	
	Запропонований метод	IGA	Запропонований метод	IGA
60	4,46	7,05	0,112	0,117
80	4,09	6,41	0,122	0,126
100	4,00	6,23	0,132	0,135
120	4,13	6,64	0,143	0,150
140	4,37	6,93	0,153	0,155
160	4,74	7,73	0,163	0,169
180	5,24	8,63	0,173	0,180
200	5,70	9,15	0,184	0,187
240	6,56	13,36	0,210	0,260

Таким чином, застосування паралельних систем для реалізації запропонованого методу є доцільним, оскільки істотно знижується затрачуваний на виконання методу час (у 8,84 рази для 16-ти ядер кластеру та у 3,54 рази для 140 потоків GPU). При цьому SIMD-система на основі GPU є істотно дешевшою за кластер і може бути використана для одержання прийняттого прискорення реалізації запропонованого методу. У порівнянні з методом IGA запропонований метод продемонстрував кращу масштабованість як на кластері, так і на GPU. Крім того, запропонований метод дозволяє швидше одержати рішення й ефективніше задіє обчислювальні ресурси, забезпечуючи при цьому побудову моделей з прийнятною точністю прогнозування.

ВИСНОВКИ

У роботі вирішено актуальну задачу автоматизації індивідуального прогнозування стану здоров'я хворого гіпертонічною хворобою.

Наукова новизна роботи полягає в тому, що запропоновано метод синтезу нейро-нечітких моделей на основі паралельних обчислень, який ґрунтується на стохастичному підході при пошуку значень параметрів, що настраюються, і полягає в розподілі найбільш ресурсномістких етапів по вузлах паралельної обчислювальної системи. Це дозволяє скоротити час настроювання параметрів (значень вагових коефіцієнтів і параметрів функцій належності нейроелементів) синтезованих нейромоделей.

Практична цінність отриманих результатів полягає в тому, що розроблено програмне забезпечення, яке реалізує запропонований паралельний метод і дозволяє виконувати побудову нейро-нечітких мереж на основі заданих наборів спостережень для синтезу моделей індивідуального прогнозування стану здоров'я хворого гіпертонічною хворобою. Результати проведених експериментів свідчать про прийнятні значення показників ефективності запропонованого паралельного методу синтезу нейро-нечітких мереж.

Перспективи подальших досліджень полягають у застосуванні запропонованого підходу до синтезу нейро-нечітких мереж з навчальних вибірок даних для вирішення практичних задач неруйнівного контролю якості.

ПОДЯКИ

Роботу виконано в рамках держбюджетної науково-дослідної теми Запорізького національного технічного університету «Інтелектуальні інформаційні технології автоматизації проектування, моделювання, керування та діагностування виробничих процесів і систем» (номер державної реєстрації 0112U005350) за підтримки міжнародного проекту «Centers of Excellence for young REsearchers» (CERES) програми «Tempus» Європейської Комісії (реєстраційний номер 544137-TEMPUS-1-2013-1-SK-TEMPUS-JPHES).

Навчальну вибірку надано професором Запорізького національного технічного університету проф. С. О. Субботіним.

СПИСОК ЛІТЕРАТУРИ

1. Beevers D. G. ABC of hypertension / D. G. Beevers, G. Y. Lip, E. O'Brien. – London : BMJ Books, 2007. – 88 p.
2. Camm A. J. The ESC Textbook of Cardiovascular Medicine / A. J. Camm, T. F. Luscher, P. Serruys. – Hoboken : Wiley-Blackwell, 2006. – 1136 p. DOI: 10.1093/med/9780199566990.001.0001.
3. Suzuki K. Artificial Neural Networks: Architectures and Applications / K. Suzuki. – New York : InTech, 2013. – 264 p. DOI: 10.5772/3409.
4. Hanrahan G. Artificial Neural Networks in Biological and Environmental Analysis / G. Hanrahan. – Boca Raton, Florida : CRC Press, 2011. – 214 p. DOI: 10.1201/b10515.
5. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посібник / С. О. Субботін. – Запоріжжя : ЗНТУ, 2008. – 341 с.
6. Baragona R. Evolutionary Statistical Procedures: An Evolutionary Computation Approach to Statistical Procedures Designs and Applications / R. Baragona, F. Battaglia, I. Poli. – Berlin : Springer, 2011. – 296 p.
7. Spall J. C. Introduction to Stochastic Search and Optimization / J. C. Spall. – Hoboken : Wiley, 2003. – 618 p. DOI: 10.1002/0471722138.
8. Дневник погоды [Електронний ресурс]. – Москва : ООО «НПЦ «Мэп Мейкер», 2014. – Режим доступа: <http://www.gismeteo.ru/diary/5093>.
9. Weber M. Observations of Solar Activity (Mg II Index) by GOME, SCIAMACHY, and GOME-2 [Electronic resource]. – Bremen: University of Bremen, 2014. – Access mode: <http://www.iup.uni-bremen.de/gome/gomemgii.html>
10. Smith S. Genetic and Evolutionary Computation: Medical Applications / S. Smith, S. Cagnoni. – Chichester : John Wiley & Sons, 2011. – 250 p. DOI: 10.1002/9780470973134.
11. Yu X. Introduction to Evolutionary Algorithms (Decision Engineering) / X. Yu, M. Gen. – London: Springer, 2010. – 418 p. DOI: 10.1007/978-1-84996-129-5.
12. Cantu-Paz E. Efficient and Accurate Parallel Genetic Algorithms / E. Cantu-Paz. – Massachusetts : Kluwer Academic Publishers, 2001. – 162 p. DOI: 10.1007/978-1-4615-4369-5.
13. Encyclopedia of artificial intelligence / Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol. 1–3. – 1677 p.
14. Encyclopedia of machine learning / [eds. C. Sammut, G. I. Webb]. – New York : Springer, 2011. – 1031 p.
15. Mumford C. L. Computational Intelligence / C. L. Mumford, L. C. Jain. – Berlin : Springer-Verlag, 2009. – 732 p. DOI: 10.1007/978-3-642-01799-5.
16. Bishop C.M. Pattern recognition and machine learning / C. M. Bishop. – New York: Springer, 2006. – 738 p.
17. Oliinyk A. O. Using Parallel Random Search to Train Fuzzy Neural Networks / A. O. Oliinyk, S. Yu. Skrupsky, S. A. Subbotin // Automatic Control and Computer Sciences. – 2014. – Vol. 48, Issue 6. – P. 313–323. DOI: 10.3103/S0146411614060078.

18. Gebali F. Algorithms and Parallel Computing / F. Gebali. – New Jersey: John Wiley & Sons, 2011. – 364 p. DOI: 10.1002/9780470932025.
19. Roosta S.H. Parallel Processing and Parallel Algorithms: Theory and Computation / S. H. Roosta. – New York : Springer-Verlag, 2000. – 566 p.
20. Quinn M. J. Parallel Programming in C with MPI and OpenMP / M. J. Quinn. – New York: McGraw-Hill, 2004. – 529 p.
21. NVIDIA CUDA Compute Unified Device Architecture 5.5 / Santa Clara: NVIDIA Corporation, 2014. – 117 p.

Стаття надійшла до редакції 20.03.2015.

Олейник А. А.

Канд. техн. наук, доцент, доцент кафедры программных средств, Запорожский национальный технический университет, Запорожье, Украина

СИНТЕЗ НЕЙРО-НЕЧЕТКИХ МОДЕЛЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ СОСТОЯНИЯ ЗДОРОВЬЯ БОЛЬНОГО НА ОСНОВЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Решена задача разработки математического обеспечения для автоматизации процесса индивидуального прогнозирования состояния здоровья больного гипертонической болезнью. Объектом исследования является процесс синтеза моделей для медицинского диагностирования. Предмет исследования составляют методы синтеза нейро-нечетких моделей для медицинского диагностирования. Цель работы: повысить эффективность процесса синтеза нейро-нечетких сетей для построения диагностических моделей на основе обучающих выборок. Предложен стохастический метод синтеза нейро-нечетких моделей на основе параллельных вычислений, основанный на стохастическом подходе при поиске значений настраиваемых параметров, и заключающийся в распределении наиболее ресурсоемких этапов по узлам параллельной вычислительной системы. Предложенный метод позволяет сократить время настройки параметров (значений весовых коэффициентов и параметров функций принадлежности нейроэлементов) синтезированных нейромоделей. Разработано программное обеспечение, реализующее предложенный метод. Проведены эксперименты по исследованию свойств предложенного метода. Результаты экспериментов позволяют рекомендовать предложенный метод для использования на практике.

Ключевые слова: множество решений, нейронная сеть, признак, параллельное программирование, прогнозирование, синтез модели.

Oliinyk A.

PhD., Associate Professor, Associate Professor of Department of Software Tools, Zaporizhzhya National Technical University, Zaporizhzhya, Ukraine

SYNTHESIS OF NEURO-FUZZY MODEL FOR PATIENT HEALTH PREDICTING BASED ON PARALLEL COMPUTING

The problem of mathematical support development is solved to automate the process of individual health prediction of hypertensive patient. The object of study is the process of model synthesis for medical diagnosis. The subject of study are methods of neuro-fuzzy model synthesis for medical diagnosis. The purpose of the work is to improve the efficiency of the process of neuro-fuzzy network synthesis for constructing diagnostic models based on training samples. The stochastic method for the synthesis of neuro-fuzzy models based on parallel computing is proposed. It uses the stochastic approach for finding the values of adjustable parameters, and consists in the distribution of the most demanding stages on the nodes in parallel computing system. The proposed method can reduce the time of parameters calculation (the weighting coefficients and parameters of membership functions of neurons) of synthesized neuromodels. The software implementing proposed method is developed. The experiments to study the properties of the proposed method are conducted. The experimental results allow to recommend the proposed method for use in practice.

Keywords: solution set, neural network, feature, parallel programming, prediction, model synthesis.

REFERENCES

- Beevers D. G., Lip G. Y., O'Brien E. ABC of hypertension. London, BMJ Books, 2007, 88 p.
- Camm A. J., Luscher T. F., Serruys P. The ESC Textbook of Cardiovascular Medicine. AHoboken, Wiley-Blackwell, 2006, 1136 p. DOI: 10.1093/med/9780199566990.001.0001.
- Suzuki K. Artificial Neural Networks: Architectures and Applications. New York, InTech, 2013, 264 p. DOI: 10.5772/3409.
- Hanrahan G. Artificial Neural Networks in Biological and Environmental Analysis. Boca Raton, Florida, CRC Press, 2011, 214 p. DOI: 10.1201/b10515.
- Subbotin S. O. Podannja j obrobka znan' u sistemah shtuchnogo intelektu ta pidtrimki prijnjattja rishen' : navch. posibnik. Zaporizhzhja, ZNTU, 2008, 341 p.
- Baragona R., Battaglia F., Poli I. Evolutionary Statistical Procedures: An Evolutionary Computation Approach to Statistical Procedures Designs and Applications. Berlin, Springer, 2011, 296 p.
- Spall J. C. Introduction to Stochastic Search and Optimization. Hoboken, Wiley, 2003, 618 p. DOI: 10.1002/0471722138.
- Dnevnik pogody [Elektronnyj resurs]. Moscow, OOO «NPC «Мер Меjker», 2014, Rezhim dostupa: <http://www.gismeteo.ru/diary/5093>.
- Weber M. Observations of Solar Activity (Mg II Index) by GOME, SCIAMACHY, and GOME-2 [Electronic resource]. Bremen, University of Bremen, 2014, Access mode: <http://www.iup.uni-bremen.de/gome/gomengii.html>
- Smith S., Cagnoni S. Genetic and Evolutionary Computation: Medical Applications. Chichester, John Wiley & Sons, 2011, 250 p. DOI: 10.1002/9780470973134.
- Yu X., Gen M. Introduction to Evolutionary Algorithms (Decision Engineering). London, Springer, 2010, 418 p. DOI: 10.1007/978-1-84996-129-5.
- Cantu-Paz E. Efficient and Accurate Parallel Genetic Algorithms. Massachusetts, Kluwer Academic Publishers, 2001, 162 p. DOI: 10.1007/978-1-4615-4369-5.
- Eds.: J. R. Dopico, J. D. de la Calle, A. P. Sierra. Encyclopedia of artificial intelligence. New York, Information Science Reference, 2009, Vol. 1–3, 1677 p.
- Eds. C. Sammut, G. I. Webb Encyclopedia of machine learning. New York, Springer, 2011, 1031 p.
- Mumford C. L., Jain L. C. Computational Intelligence. Berlin, Springer-Verlag, 2009, 732 p. DOI: 10.1007/978-3-642-01799-5.
- Bishop C. M. Pattern recognition and machine learning. New York, Springer, 2006, 738 p.
- Oliinyk A. O., Skrupsky S. Yu., Subbotin S. A. Using Parallel Random Search to Train Fuzzy Neural Networks, *Automatic Control and Computer Sciences*, 2014, Vol. 48, Issue 6, pp. 313–323. DOI: 10.3103/S0146411614060078.
- Gebali F. Algorithms and Parallel Computing. New Jersey, John Wiley & Sons, 2011, 364 p. DOI: 10.1002/9780470932025.
- Roosta S. H. Parallel Processing and Parallel Algorithms: Theory and Computation. New York, Springer-Verlag, 2000, 566 p.
- Quinn M. J. Parallel Programming in C with MPI and OpenMP. New York, McGraw-Hill, 2004, 529 p.
- NVIDIA CUDA Compute Unified Device Architecture 5.5 / Santa Clara. NVIDIA Corporation, 2014, 117 p.