

## СИНТЕЗ ТА ВИКОРИСТАННЯ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ З ЙМОВІРНІСНИМ КОДУВАННЯМ СТРУКТУРИ

**Леощенко С. Д.** – аспірант кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя Україна.

**Олійник А. О.** – канд. техн. наук, доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

**Субботін С. О.** – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

**Гофман Є. О.** – старший науковий співробітник науково-дослідної частини Національного університету «Запорізька політехніка», Запоріжжя Україна.

**Ільяшенко М. Б.** – канд. техн. наук, доцент кафедри комп'ютерних систем та мереж Національного університету «Запорізька політехніка», Запоріжжя Україна.

### АНОТАЦІЯ

**Актуальність.** Розглянуто задачу кодування інформації моделей на базі штучних нейронних мереж для подальшого пересилання та використання таких моделей. Об'єктом дослідження є процес кодування штучних нейронних мереж з використанням ймовірнісних структур даних.

**Мета роботи** полягає у розробці методу кодування нейронних мереж для зменшенні ресурсоемності процесу нейроevolюційного синтезу моделей.

**Метод.** Запропоновано метод кодування нейронних мереж на основі ймовірнісних структур даних. На початку метод використовує основні принципи підходу прямого кодування інформації про мережу та базуючись на секвенуванні кодує матрицю міжнейронних зв'язків у вигляді біополімерів. Потім використовуються ймовірнісні структури даних для більш компактного представлення початкової матриці. Для цього використовуються хеш-функції, початкова матриця проходить через процес хешування, що дозволяє значно знизити вимоги до ресурсів пам'яті. Метод дозволяє скоротити витрати пам'яті при пересилці штучних нейронних мереж, що значно розширює практичне використання таких моделей, запобігаючи різкому зменшенню точності їх роботи.

**Результати.** Розроблений метод реалізовано та досліджено при вирішенні задачі класифікації стану південнонімецьких кредиторів. Використання розробленого методу дозволило збільшити швидкість синтезу нейромоделі на 15–17,6%, в залежності від використовуваних обчислювальних ресурсів. Також метод дозволив скоротити долю пересилок інформації на 8%, що також свідчить про прискорення та більш раціональне використання ресурсів.

**Висновки.** Проведені експерименти підтвердили працездатність запропонованого математичного забезпечення і дозволяють рекомендувати його для використання на практиці при кодуванні моделей на основі штучних нейронних мереж для подальшого вирішення задач діагностування, прогнозування, оцінювання та розпізнавання образів. Перспективи подальших досліджень можуть полягати в попередній обробці даних для більш жорсткого контролю процесу кодування з метою мінімізації втрат якості роботи моделей на основі нейронних мереж.

**КЛЮЧОВІ СЛОВА:** нейроevolюція, кодування, ймовірнісні структури даних, нейронні мережі, генетичний алгоритм.

### АБРЕВІАТУРИ

NEAT – Neuroevolution of augmenting topologies;  
RAM – Random Access Memory;  
ГНМ – глибокі нейронні мережі;  
ДНК – дезоксирибонуклеїнова кислота;  
ЙСД – ймовірнісні структури даних;  
НЕС – нейроevolюційний синтез;  
ПМГА – паралельний модифікований генетичний алгоритм;  
РНК – рибонуклеїнова кислота;  
РНМ – рекурентні нейронні мережі;  
ШНМ – штучна нейронна мережа.

### НОМЕНКЛАТУРА

$\varepsilon_{estimation}$  – помилка оцінки;  
 $\delta$  – ймовірність для значення із Count-min sketch;  
 $cardinality_{hashfunction}$  – потужність хеш-функції;  
 $depth_{sketch}$  – глибина (висота) ескізу;

$G$  – популяція або генерація штучних нейронних мереж;  
 $n$  – кількість вхідних ознак, що характеризують екземпляри вибірки;  
 $N_i$  – множина нейронів на вході мережі;  
 $N_{i_l}$  – нейрон на вході мережі;  
 $N_o$  – множина нейронів на виході мережі;  
 $N_{o_p}$  – нейрон на виході мережі;  
 $N_h$  – множина нейронів прихованого шару мережі;  
 $N_{h_r}$  – нейрон прихованого шару мережі;  
 $NN$  – штучна нейронна мережа;  
 $NN_{param}$  – мета-параметри (інформація) про мережу;  
 $NN_{struct}$  – структура мережі;  
 $l$  – кількість нейронів на вході мережі;

$m$  – кількість залежних (категоріальних) ознак екземплярів вибірки;

$p$  – кількість нейронів на виході мережі;

$q$  – кількість міжнейронних зв'язків у мережі;

$r$  – кількість нейронів у прихованому шарі мережі;

*Sample* – вибірка даних;

$t$  – розмір популяції;

$w$  – множина міжнейронних зв'язків;

$w_q$  – міжнейронний зв'язок мережі;

$width_{sketch}$  – ширина ескізу;

$x_n$  – незалежна ознака екземпляру вибірки;

$X$  – множина незалежних змінних – ознак;

$y_m$  – значення залежної змінної (ознака) екземпляру вибірки;

$У_{очік}$  – реальне значення залежної змінної;

$У_{моделі}$  – значення залежної змінної отримане завдяки моделі;

$У_{моделі'}$  – значення залежної змінної отримане завдяки моделі після кодування та пересилки;

$Y$  – множина значень залежних змінних.

## ВСТУП

У наш час для синтезу моделей на основі нейронних мереж широкого використання набули еволюційні методи [1–9]. Такий підхід не вимагає наявності експертних знань про структуру моделей, дозволяє працювати з недиференційованими функціями активації, забезпечує змогу виходу з локальних екстремумів, а також можливості розв'язання завдань як параметричного, так і структурного синтезу моделей на основі нейронних мереж [10–14]. Важливим етапом нейроеволюційного синтезу діагностичних та розпізнавальних моделей є кодування інформації, тобто подання інформації про структуру моделі та її параметри для можливості застосування еволюційного пошуку. Проблема кодування інформації про нейронні мережі виникає при послідовному та паралельному нейроеволюційному синтезі, а також при збереженні та передачі ШНМ для подальшого їх опрацювання (рис. 1).

Усі існуючі методи кодування ШНМ можна розділити на прямі методи кодування та непрямі методи кодування [15, 16].

Обидві групи методів кодування мають свої переваги та недоліки. Пряме кодування не враховує тісний зв'язок між генним складом і індивідуальною продуктивністю, а тільки доводить, що його кодування конструкція може сприяти ефективній еволюції нейронної мережі. А непряме кодування вимагає розробки набору правил кодування і декодування для трансформації послідовностей генів і окремих фенотипів, тому необхідно більше розуміти генетичні та еволюційні механізми біології. Спосіб кодування генів організмів надзвичайно ефективний, і дуже коротка послідовність генів може керувати складними фенотипами індивідуумів. Багато сучасних досліджень непрямого кодування знаходяться на дослі-

дницькій стадії, але з безперервним розвитком досліджень механізмів біологічної еволюції дослідження нейроеволюційних методів, заснованих на обох напрямках кодування, все ще мають великий потенціал.

Крім того, ряд труднощів із вибором типу кодування пов'язані із сучасними топологіями ШНМ [15–21]. Так, класичні методи прямого кодування майже неможливо застосовувати для РНМ [21], де окрім прямих зв'язків між нейронами є ще і зворотні. При кодуванні ГНМ [10] виникають проблеми як при використанні прямих методів, так і при застосуванні непрямих методів кодування, оскільки в такому випадку необхідно не тільки закодувати приховані нейрони, а і зберегти структуру нейромоделі, забезпечуючи розподіл за шарами.

Саме тому актуальною є науково-прикладна задача розробки нових методів кодування нейромоделей, які дозволяють кодувати інформацію про різні топології ШНМ не витрачаючи при цьому великої кількості комп'ютерних ресурсів.

**Об'єкт дослідження** – процес кодування ШНМ з використанням ЙСД.

Існуючі методи кодування ШНМ цілком або ж окремих мета-параметрів мережі характеризуються значною ресурсоемністю та є достатньо складними для імплементації еволюційних операторів. Крім того, при використанні існуючих [15, 16] часто виникає проблема із кодуванням інформації про РНМ або ГНМ.

**Предмет дослідження** – метод кодування ШНМ з використанням ЙСД.

На сьогоднішній день існує два основні підходи до кодування ШНМ. Проте більшість таких методів відрізняються такими недоліками: ресурсоемність та складність інтерпретування. Тому, у роботі запропоновано підхід, що базується на поданні інформації про ШНМ та кодуванні її з використанням ЙСД.

**Мета роботи** – розробка методу кодування нейронних мереж для зменшенні ресурсоемності процесу нейроеволюційного синтезу моделей.

## 1 ПОСТАНОВКА ПРОБЛЕМИ

Нехай задана вибірка вхідних даних  $Sample = \langle X, Y \rangle$ , де  $X = \{x_1, x_2, \dots, x_n\}$  – множина незалежних змінних – ознак,  $Y = \{y_1, y_2, \dots, y_m\}$  – множина значень залежних змінних,  $n$  та  $m$  – кількість вхідних ознак, що характеризують екземпляри вибірки.

Спираючись на це представимо проблему наступним чином: для синтезу ШНМ ( $NV$ ) необхідно визначити множину нейронів  $N = \{N_i, N_o, N_h\}$ , що складається із підмножин вхідних вихідних прихованих нейронів  $N_i = \{N_{i_1}, N_{i_2}, \dots, N_{i_l}\}, l = 1, 2, \dots, |N_i|$ ,  $N_o = \{N_{o_1}, N_{o_2}, \dots, N_{o_p}\}, p = 1, 2, \dots, |N_o|$ ,  $N_h = \{N_{h_1}, N_{h_2}, \dots, N_{h_r}\}, r = 1, 2, \dots, |N_h|$  та множини ваг зв'язків між нейронами  $w = \{w_q\}$ . Визначи-

вши значення елементів множин, можна вважати синтез ШНМ – завершеним.

Для цього будемо використовувати НЕС на базі ПМГА. В цьому випадку будемо оперувати популяціями ШНМ ( $G = \{NN_1, NN_2, NN_3, \dots, NN_t\}$ ). Тоді постає проблема, як представити структуру ( $NN_{struct}$ ) та інформацію про ШНМ ( $NN_{param}$ ), щоб переслати її з головного ядра на задіяні потоки  $NN = \{NN_{struct}, NN_{param}\} = \{N, w, NN_{param}\}$ .

## 2 ОГЛЯД ЛІТЕРАТУРИ

У класичній генетиці, зазвичай виділяється генотип і фенотип. Генотип – це генетичне представлення істоти, а фенотип – актуалізоване фізичне представлення істоти [21, 22]. Еволюційні алгоритми завжди сильно відображають біологію, нейроеволюція нічим не відрізняється в цьому плані [10–12].

Питання кодування походить від питання про те, як представляти ШНМ генетично під час роботи. Спосіб, яким кодують ШНМ, визначає шлях, за яким метод буде обробляти ключові еволюційні процеси: відбір, мутацію і кросовер (або рекомбінацію). Будь-яке кодування потрапляє в одну з двох категорій, пряме або непряме.

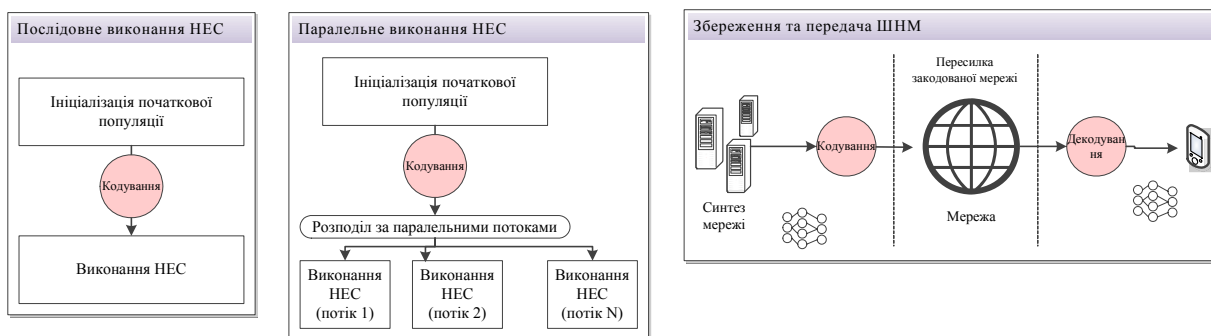


Рисунок 1 – Кейси використання кодування ШНМ

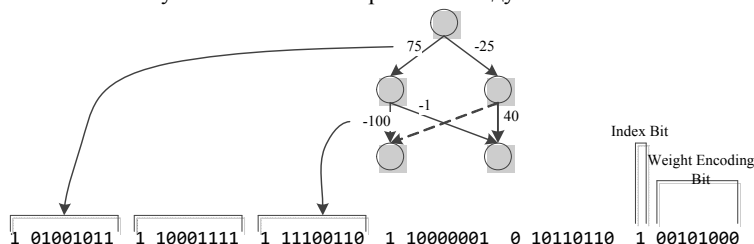


Рисунок 2 – Приклад прямого кодування ШНМ

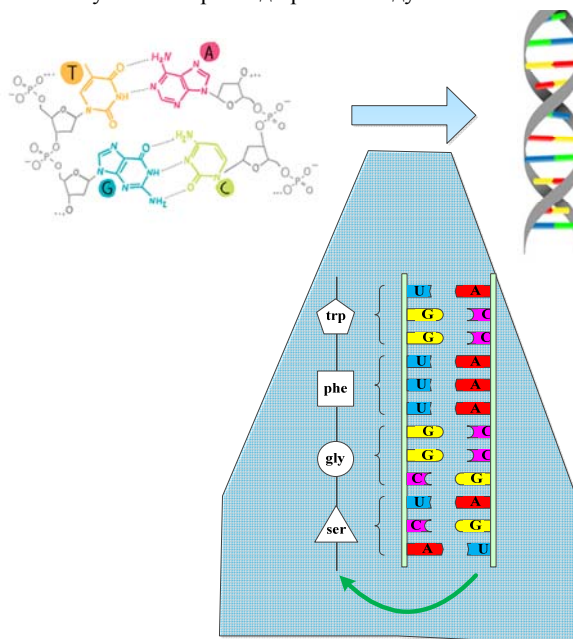


Рисунок 3 – Процес трансляції ДНК

Пряме кодування оперує хромосомами, що представляють деяке лінійне представлення ШНМ, в якому в явному вигляді вказані всі нейрони, ваги і зв'язку ШНМ [10, 12]. Таким чином, завжди можна побудувати взаємно-однозначну відповідність між структурними елементами ШНМ (нейронами, зв'язками, вагами та ін.), тобто фенотипом, і відповідними ділянками хромосоми, тобто генотипом.

Цей спосіб представлення нейронної мережі є найбільш простим і інтуїтивним, а також дозволяє застосовувати до отриманих хромосом вже наявний апарат генетичного пошуку (наприклад, оператори кросинговеру і мутації). З найбільш очевидних мінухів такої схеми кодування можна відзначити «розпухання» генотипу при збільшенні кількості нейронів і зв'язків ШНМ і, як наслідок, низьку ефективність за рахунок значного збільшення простору пошуку [10].

В прикладі The Index Bit використовується для вказівки того, чи існує з'єднання чи ні. Weight Encoding Bits кодують ваги в двійковій формі. Ряд дослідників запропонували стратегію кодування, яка кодує ваги в дійсних числах, а також визначені деякі мутаційні оператори, які підходять для кодування.

Непряме кодування використовує більш «біологічний» принцип: в генотипі кодується не сам фенотип, але правила його побудови (умовно кажучи, якась програма) [13]. При декодуванні генотипу ці правила застосовуються в певній послідовності (найчастіше, рекурсивно і, найчастіше, застосовність правил залежить від поточного контексту), в результаті чого і будується нейронна мережа.

При використанні непрямих методів кодування генетичне представлення (а, відповідно, і простір пошуку для генетичних алгоритмів) виходить більш компактним, а сам генотип дозволяє кодувати модульні структури, що дає в певних умовах переваги в адаптивності отриманих результатів [14]. Натомість же отримуємо практичну неможливість простежити, які зміни в генотипі призвели до заданих змін у фенотипі, а також безліч труднощів з підбором генетичних операторів, збіжністю і продуктивністю.

Історично, пряме кодування було досліджено раніше і глибше, проте ряд недоліків цього підходу змушують дослідників все більш пильно придивлятися до непрямих методів кодування. Однак за своєю суттю непрямі методи досить складні для аналізу. Наприклад, одна і та ж мутація правила, розташованого на початку «програми», надає колосальний ефект, а застосована до «кінцевих» правил – ефекту не має зовсім, а в результаті – генетичний пошук має сильну тенденцію до передчасної збіжності. Підбір операторів кросинговеру також є нетривіальним завданням, тому що використання стандартних бінарних операторів, як правило, призводить до частой появи нежиттєздатних рішень.

Існує також низка інших методик нейроеволюції. Розглянемо найбільш популярні з наведених коротких описів кожної з них [10, 14]:

– Боєрс і Куйпер (Boers and Kuiper) – використання контексто-залежних І-систем;

– Ділаєрт і Бір (Dellaert and Beer) – підхід, аналогічний Кангелоссі і Елману, але з використанням випадкових булевих нейромереж (random boolean networks);

– Харп, Самад і Гуха (Harp, Samad and Guha) – позонне пряме кодування структури;

– Груау (Gruau) – використання граматичного дерева для завдання інструкцій при діленні клітин (чимось схоже на Кангелоссі, Парісі і Нолфі);

– Вааріо (Vaario) – ріст клітин задається L-системами.

В результаті непряме кодування зазвичай більш компактне. З іншого боку, встановлення правил для непрямих кодування може призвести до сильного зміщення в просторі пошуку, тому набагато важче створити непряме кодування без суттєвих знань про те, як буде використовуватися кодування.

### 3 МАТЕРІАЛИ ТА МЕТОДИ

З теорії генетики [21–23] відомо, що секвенування біополімерів (білків і нуклеїнових кислот – ДНК і РНК) – це визначення їх амінокислотної або нуклеотидної послідовності [21–23]. В результаті секвенування отримують формальний опис первинної структури лінійної макромолекули у вигляді послідовності мономерів в текстовому вигляді [21, 22]. Розміри секвенуємих ділянок ДНК зазвичай не перевищують 100 пар нуклеотидів і 1000 пар нуклеотидів при секвенуванні по Сенгеру. В результаті секвенування ділянок ДНК, що перекриваються, отримують послідовності ділянок генів, цілих генів, тотальної мРНК або повних геномів організмів.

У цій роботі новий метод кодування інформації про ШНМ пропонується організувати базуючись на схожому принципі. Для кодування зв'язків у генотипі особини представимо інформацію про ваги міжнейронних зв'язків нейромоделі. При цьому кожен ген буде містити інформацію про індекси початкового та кінцевого нейрону зв'язку, а також його вагу. У випадку, коли метод працює з рекурентними нейронними мережами, додається додаткова комірка з вагою зворотного зв'язку, а його індекс визначається номером вихідного нейрону.

Введемо правила індексування нейронів:

1) оскільки кількість входів і виходів мережі є фіксованою величиною, то індекси відповідних нейронів є постійними і приймають значення в інтервалі  $[0; N_i - 1]$  для вхідних нейронів, та  $[N_i; N_i + N_o - 1]$  – для вихідних, де  $N_i$  і  $N_o$  – кількість входів і виходів мережі, відповідно. Видалення вхідних і вихідних нейронів є неможливим;

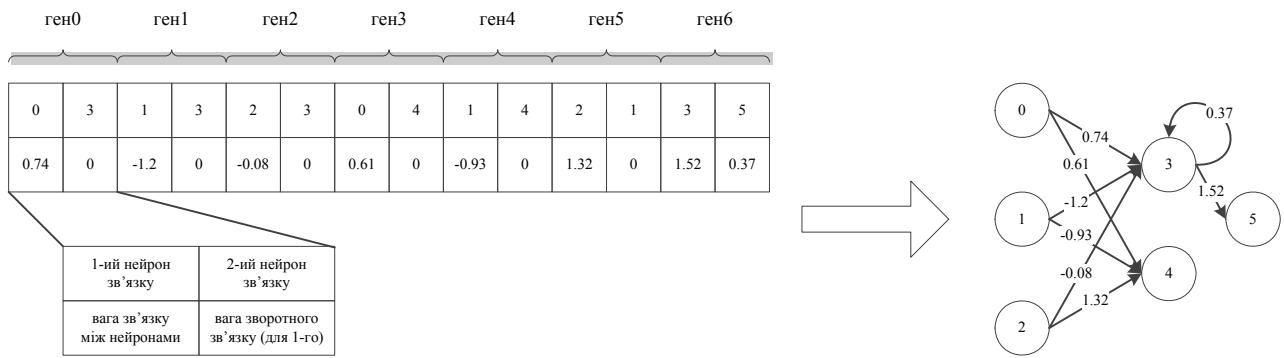


Рисунок 4 – Приклад кодування мережі

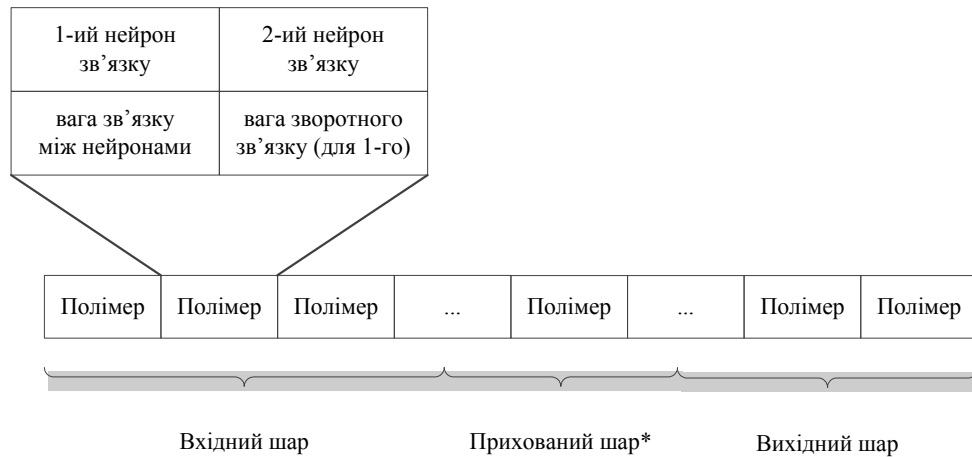


Рисунок 5 – Секвенування послідовностей за шарами ШНМ

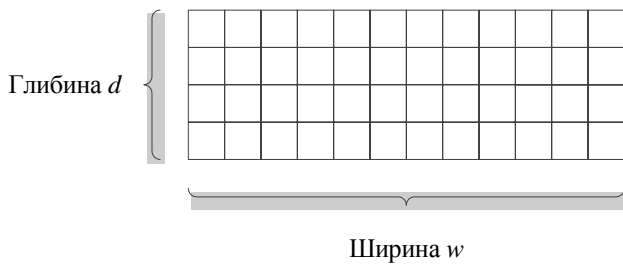


Рисунок 6 – Count-min sketch з глибиною  $d$  та шириною  $w$

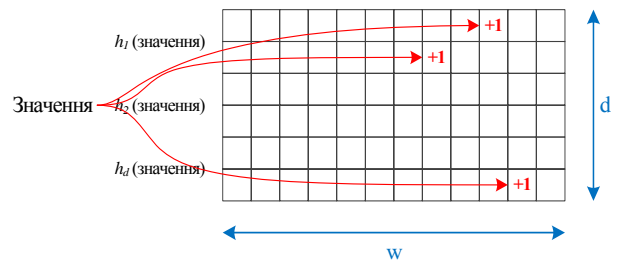


Рисунок 7 – Розподіл значень у структурі

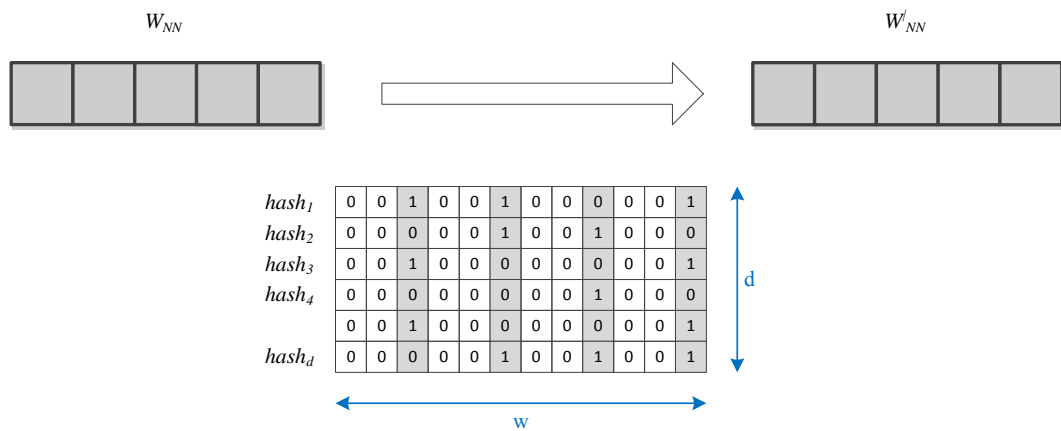


Рисунок 8 – Кодування та декодування інформації про ШНМ з використанням Count-min sketch

2) нові нейрони, що з'являються в результаті мутацій, отримують мінімальний можливий індекс. Наприклад, якщо особина представляє мережу з трьома входами, трьома виходами і не містить прихованих нейронів, то новому нейрону в цій мережі буде присвоєно індекс «5», наступному, що з'явився – «6» і т. д.;

3) індекси нейронів в мережі не можуть містити пропущені значення, тобто не може бути ШНМ з нейронами, що мають, наприклад, індекси  $N_0, N_1, N_2, N_5, N_6$ . Якщо такий випадок виникає, наприклад, після видалення з мережі нейрона з індексом 4, індекси нейронів, що залишилися коригуються таким чином:  $N_5 \rightarrow N_4$ ,  $N_6 \rightarrow N_5$ , при цьому змінюються дані у зв'язках, що належать до цих нейронів.

Тож отримаємо дворядковий список, де кожні чотири комірки (по дві з рядка) зберігають інформацію про нейрон (рис. 4).

Додатково слід відзначити, що друге правило забезпечує певне впорядкування шарів – секвенування послідовностей в списках (рис. 5).

На наступному етапі кодування використовуються ймовірнісні структури даних. Робота [24] пропонує використання модифікації фільтра Блума. Проте такий підхід не дозволяє в подальшому закодувати зворотні зв'язки, а використання такого підходу під час мутацій з видаленням певних нейронів вимагає запровадження додаткової матриці – другого фільтра Блума для підрахунку в зворотньому порядку [25]. Тож кількість обрахунків збільшується. З огляду на це більш доцільним є використання структури: Count–min sketch [26, 27]. Count–min sketch – це ймовірнісна структура даних, яка представляє собою таблицю частот подій в потоці даних. Вона використовує хеш-функції для зіставлення подій з частотами, але на відміну від хеш-таблиці використовує тільки сублінійний простір, за рахунок перерахунку деяких подій через колізій [25–27].

Структура Count–min sketch може представлятися у вигляді ескізу із глибиною  $d$  та шириною  $w$  (рис. 6).

Тоді представимо розподіл значень у масиві Count–min sketch, виходячи з принципу, зображеному на рисунку 7. Такий розподіл можна вважати ескізом.

Загалом матимемо представлення із розподілом за наступним правилом (1) та (2).

$$\varepsilon_{estimation} \leq 2 \cdot \max cardinalit_{y_{hashfunction}} / width_{sketch} \quad (1)$$

$$\delta = 1 - 0.5^{depth_{sketch}} \quad (2)$$

Варто відзначити, що ширина ескізу обмежує величину помилки (1), а висота (глибина) контролює ймовірність того, що оцінка проб'є цю межу (2).

Таким чином представимо масив даних про ШНМ у вигляді Count–min sketch та декодуємо його після цього (рис. 8).

Таким чином запропонований метод кодування розпочинається із початкового представлення матриці, що будується з так званих полімерів – комірок, що містять інформацію про вхідні-вихідні зв'язки між нейронами та ваги таких зв'язків, додатково для РНМ комірки будуть зберігати ваги зворотних зв'язків.

Далі визначаються хеш-функції (за замовченням рекомендується брати 4 хеш-функції) і створюється матриця для їх виведення, як показано на рис. 8.

Після цього для кожного елементу потоку даних про ШНМ обчислюються хеш-виходи і збільшується відповідний лічильник в матриці.

Таким чином, збільшуючи відповідні відліки в матриці, отримаємо оновлену матрицю.

У деяких випадках через хеш-колізію можлива ситуація, коли отримана частота елемента є дещо більшою, ніж очікувана. Точність кодування буде залежати від того, наскільки унікальними є хеш-функції, що повертають значення елемента. Крім того, чим більше хеш-функція, тим точнішою буде частота.

У цьому випадку ймовірнісна структура даних Count–min sketch дозволяє обчислити частоту потоків великих даних в суб-лінійному просторі, з часовою складністю, що оцінюється  $O(1)$ , тобто з постійною часовою складовою, що не залежить від параметрів нейромережевої моделі.

#### 4 ЕКСПЕРИМЕНТИ

Виконано експериментальне дослідження розробленого методу шляхом його порівняння з відомими аналогами. Слід зазначити, що найбільш схожим та часто використовуваним методом кодування інформації при нейроеволюційному синтезі є метод NEAT [10–12]. Тож в подальших експериментальних дослідженнях будемо спиратися на нього.

На сьогоднішній день також є паралельні інтерпретації методу NEAT. Тому під час експериментального дослідження будемо аналізувати швидкість та долю пересилок під час паралельного виконання синтезу авторського методу [28–31] з новим видом кодування та NEAT.

Наступним етапом експериментального дослідження буде використання нового методу для кодування попередньо синтезованої мережі. Після кодування така мережа буде пересилатися за допомогою мережі Інтернет на іншу робочу станцію де буде протестована точність отриманої мережі.

Для першого етапу тестування буде використовуватися апаратне забезпечення кафедри програмних засобів Національного університету «Запорізька політехніка»: процесор Xeon E5-2660 v4 (14 ядер), оперативна пам'ять 4x16 ГБ DDR4.

У якості вхідних даних буде використовуватися вибірка даних про Південнонімецький кредитний стан (South German Credit Data Set) [32, 33]. Оновлений набір даних [34–36] містять корекцію і деяку довідкову інформацію, засновану на поданні відкритих даних Ludwig-Maximilians-Universität München (2010) тих же

даних і декількох інших німецькомовних ресурсів. Характеристики вибірки даних наведено у табл. 1.

Таблиця 1 – Загальна характеристика вибірки даних

Criterion	Characteristic
Data Set Characteristics:	Multivariate
Attribute Characteristics:	Integer, Real
Number of Instances:	1000
Number of Attributes	21

## 5 РЕЗУЛЬТАТИ

У таблиці 2 наведено результати порівняння виконання ПМГА із застосуванням компактного кодування та паралельної версії NEAT – Parallel NEAT.

Таблиця 2 – Результати експерименту

ПМГА					
Кількість ядер	1	2	4	8	16
Час синтезу, с	2361,870	1226,841	638,669	345,845	200,759
Доля пересилок	0,000	0,039	0,082	0,171	0,360
Час пересилок, с	0,000	47,691	52,136	59,288	72,273
Parallel NEAT					
Кількість ядер	1	2	4	8	16
Час синтезу, с	2716,510	1415,454	739,186	402,626	235,997
Доля пересилок	0,000	0,042	0,088	0,186	0,390
Час пересилок, с	0,000	59,608	65,370	74,773	92,039

Оскільки комп'ютерна система дозволяє змінювати кількість задіяних ядер – це бралось до уваги.

Зміни та різницю у часі пересилок при виконанні методів зображено у вигляді графіків, що представлені на рис. 9.

Експериментальні результати другого етапу наведено у табл. 3. Тут порівнюються результати синтезований ШНМ до та після пересилки з кодуванням. Засвідчимо, що швидкість передачі – не важлива.

Для повного відображення роботи із тестовою вибіркою використаємо нарізну діаграму, де один ряд значень це значення роботи моделі до пересилки, а дзеркальний ряд – значення після передачі та запуску мережі на іншій робочій станції. Нарізна діаграма зображена на рис. 10.

В такій діаграмі прогалини між дзеркальними стовбцями свідчать про певні неточності в роботі мережі, що була запущена після закодованої пересилки та декодування на новій робочій станції.

Таблиця 3 – Результати експерименту

№	Уочік	Умоделі	Умоделі'
1	63,40	62,16	62,16
2	60,60	59,36	59,36
3	63,84	62,56	62,56
4	64,25	63,02	63,02
5	58,48	57,28	57,28
6	61,79	60,53	60,53
7	63,72	62,46	62,46
8	61,97	60,77	60,77
9	58,25	57,02	57,02
...	...	...	...
100	59,63	59,03	59,03

## 6 ОБГОВОРЕННЯ

Експериментальна перевірка засвідчила, що ПМГА відпрацював швидше, хоча різниця у швидкості була не велика (табл. 2). Більш того, така різниця може бути пояснена саме меншою долею пересилок. Тобто під час роботи методу, менша кількість інформації пересилалася між потоками системи, а отже менше навантажувала пам'ять та систему загалом.

Проте, варто зазначити, що незважаючи ні на що, час пересилок зменшився, а отже інформація, що передавалася займала менше простору у пам'яті (рис. 9). Це додатково засвідчують графіки отримані на основі розподілу часу синтезу.

Тож перший етап експериментального дослідження довів результативність методу не в повній мірі.

Для другої частини експерименту буде використано попереднє апаратне забезпечення та додаткова робоча станція в наступній конфігурації: станція Acer Swift 3 під управлінням Intel Core i5-8250U із тактовою частотою 1,6–3,4 ГГц, RAM 16 ГБ (DDR4) та твердотільним накопичувачем на 128 ГБ. Ці дві системи будуть передавати закодовану ШНМ через мережу Інтернет. Засвідчимо, що швидкість передачі – не важлива.

Порівняємо помилку роботи ШНМ на тестовій частині вибірки. Як видно із табл. 3 у більшості випадків результати залишилися однаковими. Загалом же середня помилка збільшилася із 0,01811 до 0,1901. Додатково можемо у цьому пересвідчитися на нарізній діаграмі.

З нарізної діаграми (рис. 10) видно, що в деяких випадках модель отримана після декодування інформації після пересилки має певні неточності – це місця розривів стовбців на діаграмі. Проте таких випадків загалом достатньо мало. Таку ситуацію можливо отримати, коли після декодування були втрачені певні зв'язки.

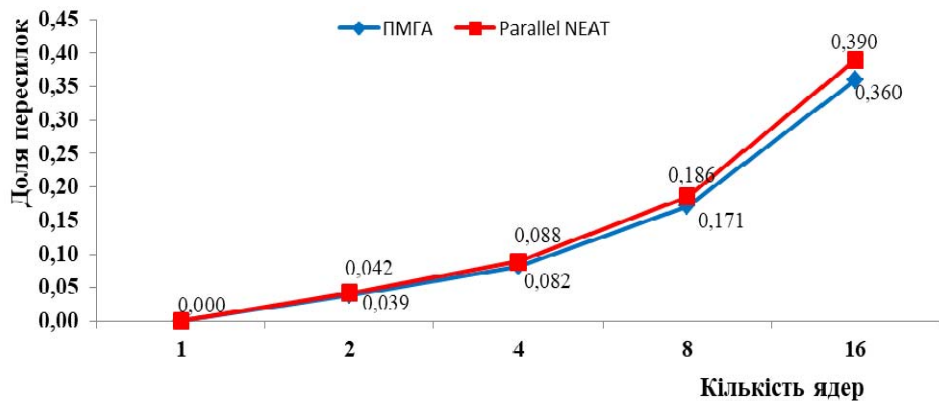


Рисунок 9 – Графіки розподілу пересилок під час паралельного синтезу ШНМ

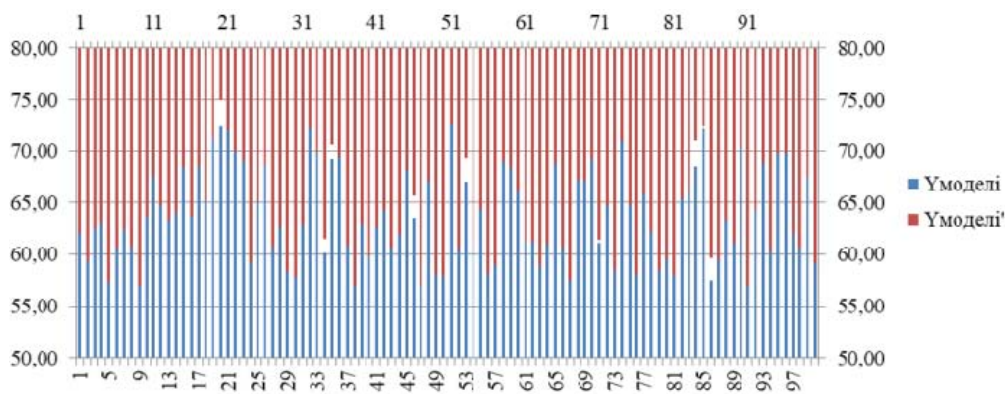


Рисунок 10 – Нарізна діаграма порівняння точності роботи моделей

Допустити такі незначні відхилення при втраті окремих нейронів в прихованому шарі – не можливо. Отже скоріше за все були втрачені саме зв'язки, які відповідали за певні взаємозв'язки у даних.

При таких результатах особливо цікаво було б відслідкувати зміни в роботі при дослідженні даних із певним періодом. До таких даних можливо віднести певну фінансову інформацію, коли наприклад, коливання курсу валют або вартості пакетів паперів має певний період, але все ж таки не повторюється повністю.

Аналізуючи загальні результати експериментів можна прийти до висновку, що запропонований метод компактного кодування не є досить ефективним для використання в паралельних системах, адже не самостійно не зменшує саме долю пересилок інформації, а лише частково пришвидшує цей процес. Більш того це можна пояснити виключно меншим об'ємом використаного ресурсу пам'яті.

Проте значно кращі та важливіші результати було отримано саме під час використання методу для кодування інформації про ШНМ для пересилки з однієї робочої станції на іншу. Це значно розширює саме практичні межі використання ШНМ. Адже таким чином можна виконати високо ітеративний процес синтезу на обчислювальному сервері, а результуючу мережу переслати для використання на конкретну робочу станцію. Беручи до уваги, що такою робочою

станцією може бути, наприклад, мобільний пристрій такий підхід забезпечує високу швидкість передачі ШНМ та подальшого декодування та практичного імплементування такої моделі.

Подальшу роботу можна вести у напрямі саме жорсткого контролю процесу синтезу та початкового компактного кодування, щоби подальші втрати в точності були контрольованими та не критичними.

## ВИСНОВКИ

Вирішено актуальну науково-прикладну проблему компактного кодування нейромоделей в процесі нейроволюційного синтезу.

**Наукова новизна** полягає в тому що розроблено метод кодування нейронних мереж на основі ймовірнісних структур даних, який використовує принципи підходу прямого кодування. У запропонованому методі використовуються ЙСД для компактного зберігання інформації про ШНМ. Зберігання інформації про міжнейронні зв'язки підвищує логічну прозорість методу та робить можливим кодування інформації про РНМ та ГНМ. Використання ЙСД Count-min sketch дозволяє більш стисло та компактно кодувати матрицю початкових даних про ШНМ та прискорити процес передачі цих даних.

**Практична цінність** полягає в тому, що розв'язано практичні завдання кодування ШНМ, які в подальшому можуть використовуватися для діагнос-



тування, прогнозування, оцінювання та розпізнавання образів. Результати експериментів показали, що запропоновані методи кодування дозволяють більш компактно кодувати інформацію про ШНМ для її подальшої передачі на робочі станції для використання у якості моделі для діагностування, прогнозування, оцінювання та моделювання.

**Майбутні напрями дослідження та розробки** полягають у розробці більш жорстких процесів контролю синтезу та початкового кодування, щоби подальші втрати в точності були контрольованими та не критичними. Додатковим напрямом може стати попередня обробка вхідних даних для отримання більш явних зв'язків між даними та контролю відповідних міжнейронних зв'язків.

### ПОДЯКА

Робота була проведена за підтримки держбюджетного науково-дослідного проекту державного бюджету Національного університету «Запорізька політехніка» «Інтелектуальні методи та програмне забезпечення для діагностики та неруйнівного контролю якості військової та цивільної техніки» (номер державної реєстрації 0119U100360).

### ЛІТЕРАТУРА / LITERATURA

1. Kanaan M. T-Minus AI: Humanity's Countdown to Artificial Intelligence and the New Pursuit of Global Power / M. Kanaan. – Dallas : BenBella Books, 2019. – 270 p.
2. Taulli T. Artificial Intelligence Basics: A Non-Technical Introduction / T. Taulli. – New York : Apress, 2019. – 199 p.
3. Rothman D. Artificial Intelligence By Example: Acquire advanced AI, machine learning, and deep learning design skills / D. Rothman. – Birmingham : Packt Publishing, 2020. – 578 p.
4. Chang A.C. Intelligence-Based Medicine: Artificial Intelligence and Human Cognition in Clinical Medicine and Healthcare / A.C. Chang. – Cambridge : Academic Press, 2020. – 534 p.
5. Ponteves de P. AI Crash Course: A fun and hands-on introduction to machine learning, reinforcement learning, deep learning, and artificial intelligence with Python / P. de Ponteves. – Birmingham : Packt Publishing, 2019. – 360 p.
6. Artasanchez A. Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3.x / A. Artasanchez, P. Joshi. – Birmingham : Packt Publishing, 2020. – 618 p.
7. Additional training of neuro-fuzzy diagnostic models / [A. Oliinyk, S. Subbotin, S. Leoshchenko et al.] // Radio Electronics, Computer Science, Control. – 2018. – № 3. – P. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
8. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems / [S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko] // 2018 International Scientific-Practical Conference Problems of Informatics. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018 : proceedings. – Kharkiv : IEEE, 2018. – P. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
9. Method of Artificial Neural Network Synthesis for Using in Integrated CAD / [S. Leoshchenko, A. Oliinyk, S. Subbotin et al.] // 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, 26 February – 2 March 2019 : proceedings. – Lviv : IEEE, 2019. – P. 1–6. DOI: 10.1109/CADSM.2019.8779248
10. Iba H. Evolutionary Approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks / H. Iba. – New York : Springer, 2018. – 258 p.
11. Omelianenko I. Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms / I. Omelianenko. – Birmingham : Packt Publishing, 2019. – 368 p.
12. Bergel A. Agile Artificial Intelligence in Pharo: Implementing Neural Networks, Genetic Algorithms, and Neuroevolution / A. Bergel. – New York : Apress, 2020. – 407 p.
13. Blokdyk G. Neuroevolution of augmenting topologies: Second Edition / G. Blokdyk. – Ohio : 5STARCOoks, 2018. – 128 p.
14. Lockett A.J. General-Purpose Optimization Through Information Maximization (Natural Computing Series) / A. J. Lockett. – New York : Springer, 2020. – 579 p.
15. Rouhiainen L. Artificial Intelligence: 101 Things You Must Know Today About Our Future / L. Rouhiainen. – Scotts Valley : CreateSpace Independent Publishing Platform, 2018. – 300 p.
16. Koul A. Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow / A. Koul, S. Ganju, M. Kasm. – Newton : O'Reilly Media, 2019. – 620 p.
17. Singh A. Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS / A. Singh, R. Bhadani. – Birmingham : Packt Publishing, 2020. – 380 p.
18. Davies J. The Internet of Things: From Data to Insight / J. Davies, C. Fortuna. – Hoboken : Wiley, 2020. – 240 p.
19. Zheng N. Learning in Energy-Efficient Neuromorphic Computing: Algorithm and Architecture Co-Design / N. Zheng, P. Mazumder. – New York : Wiley-IEEE Press, 2019. – 296 p.
20. Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis (Springer-Briefs in Computer Science) / [F.M. Bianchi, E. Maiorino, M.C. Kampffmeyer, A. Rizzi, R. Jenssen]. – New York : Springer, 2017. – 81 p.
21. Ozkan L. RNA Sequencing: Principles and Data Analysis / L. Ozkan. – Traverse City : Independently published, 2020. – 118 p.
22. Robinson T.R. Genetics For Dummies / T.R. Robinson, L. Spock. – New York : For Dummies, 2020. – 400 p.
23. Tan T.W. Beginners Guide to Bioinformatics for High Throughput Sequencing / T.W. Tan, E. Lee. – Singapore : World Scientific Publishing Co Pte Ltd, 2018. – 300 p.
24. Weightless: Lossy Weight Encoding For Deep Neural Network Compression / [B. Reagen, U. Gupta, R. Adolf, M. Mitzenmacher, A. Rush et al.] // International Conference on Machine Learning (ICML 2018), Stockholm, 10–15 July : proceedings. – Stockholm, PMLR, 2018. – P. 1–10.
25. Gakhov A. Probabilistic Data Structures and Algorithms for Big Data Applications / A. Gakhov. – Madison : Books on Demand, 2019. – 220 p.

26. Knebl H. Algorithms and Data Structures: Foundations and Probabilistic Methods for Design and Analysis / H. Knebl. – New York : Springer, 2020. – 360 p.
27. Introduction to Algorithms / [T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein]. – Cambridge : The MIT Press, 2009. – 1292 p.
28. Modification and parallelization of genetic algorithm for synthesis of artificial neural networks / [S. D. Leoshchenko, A. O. Oliinyk, S. A. Subbotin et al.] // Radio Electronics, Computer Science, Control. – 2019. – № 4. – P. 68–82. DOI: 10.15588/1607-3274-2018-3-12.
29. Synthesis of artificial neural networks using a modified genetic algorithm / [S. Leoshchenko, A. Oliinyk, S. Subbotin, N. Gorobii et al.] // 1st International Workshop on Informatics & Data-Driven Medicine (IDDM 2018), Lviv, 28–30 October, 2018 : proceedings. – Lviv : CEUR WS, 2018. – P. 1–13.
30. Implementation of Selective Pressure Mechanism to Optimize Memory Consumption in the Synthesis of Neuromodels for Medical Diagnostics / [S. Leoshchenko, A. Oliinyk, S. Subbotin et al.] // 2nd International Workshop on Informatics and Data-Driven Medicine (IDDM 2019), Lviv, 11–13 November, 2019 : proceedings. – Lviv : CEUR WS, 2019. – P. 109–120.
31. Leoshchenko S. Adaptive Mechanisms for Parallelization of the Genetic Method of Neural Network Synthesis / S. Leoshchenko, A. Oliinyk, S. Subbotin // 10th International Conference on Advanced Computer Information Technologies (ACIT 2020), Deggendorf, 16–18 November : proceedings. – Ternopil: IEEE, 2020. – P. 446–450. DOI: 10.1109/ACIT49673.2020.9208905.
32. South German Credit Data Set [Electronic resource]. – Access mode: <https://archive.ics.uci.edu/ml/datasets/South+German+Credit>
33. Grömping U. South German Credit Data: Correcting a Widely Used Data Set / U. Grömping // Reports in Mathematics, Physics and Chemistry, Department II. – Berlin : Beuth University of Applied Sciences Berlin, 2019. – 14 p.
34. Feature Selection Based on Parallel Stochastic Computing / [A. Oliinyk, S. Subbotin, V. Lovkin et al.] // 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT'2018), Lviv, 11–14 September 2018 : proceedings. – Lviv : IEEE, 2018. – P. 347–351. DOI: 10.1109/STC-CSIT.2018.8526729
35. Development of the indicator set of the features informativeness estimation for recognition and diagnostic model synthesis / [A. Oliinyk, S. Subbotin, V. Lovkin et al.] // 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET 2018), Slavsko, 20–24 February 2018 : proceedings. – Lviv : IEEE, 2018. – P. 903–908. DOI: 10.1109/TCSET.2018.8336342.
36. Methods of semantic proximity extraction between the lexical units in infocommunication systems / [S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko] // International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 10–13 October 2017 : proceedings. – Kharkiv : IEEE, 2017. – P. 7–12. DOI: 10.1109/INFOCOMMST.2017.8246137.

Стаття надійшла до редакції 14.04.2021.  
Після доробки 29.04.2021.

УДК 004.896

## СИНТЕЗ И ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ С ВЕРОЯТНОСТНЫМ КОДИРОВАНИЕМ СТРУКТУРЫ

**Леошенко С. Д.** – аспирант кафедры программных средств Национального университета «Запорожская политехника», Запорожье Украина.

**Олейник А. А.** – канд. техн. наук, доцент, доцент кафедры программных средств Национального университета «Запорожская Политехника», Запорожье Украина.

**Субботин С. А.** – д-р техн. наук, профессор, заведующий кафедрой программных средств Национального университета «Запорожская политехника», Запорожье Украина.

**Гофман Е. А.** – старший научный сотрудник научно-исследовательской части Национального университета «Запорожская политехника», Запорожье Украина.

**Ильяшенко М. Б.** – канд. техн. наук, доцент, доцент кафедры компьютерных систем и сетей Национального университета «Запорожская политехника», Запорожье Украина.

### АННОТАЦИЯ

**Актуальность.** Рассмотрена задача кодирования информации моделей на базе искусственных нейронных сетей для дальнейшей пересылки и использования таких моделей. Объектом исследования является процесс кодирования искусственных нейронных сетей с использованием вероятностных структур данных.

**Цель работы** заключается в разработке метода кодирования нейронных сетей для уменьшения ресурсоемкости процесса нейроэволюционного синтеза моделей.

**Метод.** Предложен метод кодирования нейронных сетей на основе вероятностных структур данных. В начале метод использует основные принципы подхода прямого кодирования информации о сети и, базируясь на секвенировании, кодирует матрицу межнейронных связей в виде биополимеров. Затем используются вероятностные структуры данных для более компактного представления исходной матрицы. Для этого используются хэш-функции, начальная матрица проходит через процесс хэширования, что позволяет значительно снизить требования к ресурсам памяти. Метод позволяет сократить затраты памяти при пересылке искусственных нейронных сетей, что значительно расширяет практическое использование таких моделей, предотвращая резкое уменьшение точности их работы.

**Результаты.** Разработанный метод реализован и исследован при решении задачи классификации состояния южногерманских кредиторов. Использование разработанного метода позволило увеличить скорость синтеза нейромоделей на 15–

17,6%, в зависимости от используемых вычислительных ресурсов. Также метод позволил сократить долю пересылок информации на 8%, что также свидетельствует об ускорении и более рациональном использовании ресурсов.

**Выводы.** Проведенные эксперименты подтвердили работоспособность предложенного математического обеспечения и позволяют рекомендовать его для использования на практике, при кодировании моделей на основе искусственных нейронных сетей, для дальнейшего решения задач диагностирования, прогнозирования, оценивания и распознавания образов. Перспективы дальнейших исследований могут состоять в предварительной обработке данных для более жесткого контроля процесса кодирования с целью минимизации потерь качества работы моделей, на основе нейронных сетей.

**КЛЮЧЕВЫЕ СЛОВА:** нейроэволюция, кодирование, вероятностные структуры данных, нейронные сети, генетический алгоритм.

UDC 004.896

#### SYNTHESIS AND USAGE OF NEURAL NETWORK MODELS WITH PROBABILISTIC STRUCTURE CODING

**Leoshchenko S. D.** – PhD student of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Oliinyk A. O.** – PhD., Associate Professor, Associate Professor of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Subbotin S. A.** – Dr. Sc., Professor, Head of the Department of Software Tools, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

**Gofman Ye. O.** – PhD, Senior Researcher of the Research Unit, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine

**Ilyashenko M. B.** – PhD, Associate Professor, Associate Professor of the Department of Computer Systems and networks, National University “Zaporizhzhia Polytechnic”, Zaporizhzhia, Ukraine.

#### ABSTRACT

**Context.** The problem of encoding information of models based on artificial neural networks for further transmission and use of such models is considered. The object of research is the process of coding artificial neural networks using probabilistic data structures.

**Objective** of this work is to develop a method for coding neural networks to reduce the resource intensity of the process of neuroevolutionary model synthesis.

**Method.** A method for encoding neural networks based on probabilistic data structures is proposed. At the beginning, the method uses the basic principles of the approach of direct encoding of network information and, based on sequencing, encodes a matrix of interneuronal connections in the form of biopolymers. Then, probabilistic data structures are used to represent the original matrix more compactly. For this purpose, hash functions are used, the initial matrix goes through the hashing process, which significantly reduces the requirements for memory resources. The method allows to reduce memory costs when sending artificial neural networks, which significantly expands the practical use of such models, preventing a sharp decrease in the accuracy of their operation.

**Results.** The developed method is implemented and investigated in solving the problem of classification of the state of South German creditors. The use of the developed method allowed increasing the rate of neuromodel synthesis by 15–17.6%, depending on the computing resources used. The method also reduced the share of information transfers by 8%, which also indicates faster and more efficient use of resources.

**Conclusions.** The conducted experiments confirmed the efficiency of the proposed mathematical software and allow us to recommend it for use in practice, when encoding models based on artificial neural networks, for further solving problems of diagnostics, forecasting, evaluation and pattern recognition. Prospects for further research may consist in pre-processing data for more strict control of the encoding process in order to minimize the loss of quality of models based on neural networks.

**KEYWORDS:** neuroevolution, coding, probabilistic data structures, neural networks, genetic algorithm.

#### REFERENCES

1. Kanaan M. T-Minus AI: Humanity's Countdown to Artificial Intelligence and the New Pursuit of Global Power. Dallas, BenBella Books, 2019, 270 p.
2. Taulli T. Artificial Intelligence Basics: A Non-Technical Introduction. New York, Apress, 2019, 199 p.
3. Rothman D. Artificial Intelligence By Example: Acquire advanced AI, machine learning, and deep learning design skills. Birmingham, Packt Publishing, 2020, 578 p.
4. Chang A.C. Intelligence-Based Medicine: Artificial Intelligence and Human Cognition in Clinical Medicine and Healthcare. Cambridge, Academic Press, 2020, 534 p.
5. Ponteves de P. AI Crash Course: A fun and hands-on introduction to machine learning, reinforcement learning, deep learning, and artificial intelligence with Python. Birmingham, Packt Publishing, 2019, 360 p.
6. Artasanchez A., Joshi P. Artificial Intelligence with Python: Your complete guide to building intelligent apps using Python 3.x. Birmingham, Packt Publishing, 2020, 618 p.
7. Oliinyk A., Subbotin S., Leoshchenko S., Ilyashenko M., Myronova N., Mastinovskiy Y. Additional training of neuro-fuzzy diagnostic models, *Radio Electronics, Computer Science, Control*, 2018, № 3, pp. 113–119. DOI: 10.15588/1607-3274-2018-3-12.
8. Leoshchenko S., Oliinyk A., Subbotin S., Zaiko T. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems, *2018 International Scientific-Practical Conference Problems of Informatics. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018, proceedings*. Kharkiv, IEEE, 2018, pp. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
9. Leoshchenko S., Oliinyk A., Subbotin S., Shylo S., Shkarupko V. Method of Artificial Neural Network Synthesis for

- Using in Integrated CAD, *15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, 26 February – 2 March 2019, proceedings*. Lviv, IEEE 2019, pp. 1–6. DOI: 10.1109/CADSM.2019.8779248
10. Iba H. *Evolutionary Approach to Machine Learning and Deep Neural Networks: Neuro-Evolution and Gene Regulatory Networks*, New York, Springer, 2018, 258 p.
  11. Omelianenko I. *Hands-On Neuroevolution with Python: Build high-performing artificial neural network architectures using neuroevolution-based algorithms*, Birmingham, Packt Publishing, 2019, 368 p.
  12. Bergel A. *Agile Artificial Intelligence in Pharo: Implementing Neural Networks, Genetic Algorithms, and Neuroevolution*, New York, Apress, 2020, 407 p.
  13. Blokdyk G. *Neuroevolution of augmenting topologies: Second Edition*, Ohio, 5STARCOoks, 2018, 128 p.
  14. Lockett A.J. *General-Purpose Optimization Through Information Maximization (Natural Computing Series)*. New York, Springer, 2020, 579 p.
  15. Rouhiainen L. *Artificial Intelligence: 101 Things You Must Know Today About Our Future*, Scotts Valley, CreateSpace Independent Publishing Platform, 2018, 300 p.
  16. Koul A., Ganju S., Kasam M. *Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow*, Newton, O'Reilly Media, 2019, 620 p.
  17. Singh A., Bhadani R. *Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS*, Birmingham, Packt Publishing, 2020, 380 p.
  18. Davies J., Fortuna C. *The Internet of Things: From Data to Insight*, Hoboken, Wiley, 2020, 240 p.
  19. Zheng N., Mazumder P. *Learning in Energy-Efficient Neuromorphic Computing: Algorithm and Architecture Co-Design*, New York, Wiley-IEEE Press, 2019, 296 p.
  20. Bianchi F.M., Maiorino E., Kampffmeyer M.C., et al. *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis (SpringerBriefs in Computer Science)*, New York, Springer, 2017, 81 p.
  21. Ozkan L. *RNA Sequencing: Principles and Data Analysis*, Traverse City, Independently published, 2020, 118 p.
  22. Robinson T.R., Spock L. *Genetics For Dummies*, New York, For Dummies, 2020, 400 p.
  23. Tan T.W., Lee E. *Beginners Guide to Bioinformatics for High Throughput Sequencing*, Singapore, World Scientific Publishing Co Pte Ltd, 2018, 300 p.
  24. Reagen B., Gupta U., Adolf R., Mitzenmacher M., et al. *Weightless: Lossy Weight Encoding For Deep Neural Network Compression, International Conference on Machine Learning (ICML 2018), Stockholm, 10–15 July, proceedings*. Stockholm, PMLR, 2018, pp. 1–10.
  25. Gakhov A. *Probabilistic Data Structures and Algorithms for Big Data Applications*, Madison : Books on Demand, 2019, 220 p.
  26. Knebl H. *Algorithms and Data Structures: Foundations and Probabilistic Methods for Design and Analysis*, New York, Springer, 2020, 360 p.
  27. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. *Introduction to Algorithms*, Cambridge, The MIT Press, 2009, 1292 p.
  28. Leoshchenko S. D., Oliinyk A. O., Subbotin S. A., Lytvyn V. A., Shkaruplyo V.V. *Modification and parallelization of genetic algorithm for synthesis of artificial neural networks, Radio Electronics, Computer Science, Control, 2019, № 4, pp. 68–82. DOI: 10.15588/1607-3274-2018-3-12.*
  29. Leoshchenko S., Oliinyk A., Subbotin S., Gorobii N., Zaiko T. *Synthesis of artificial neural networks using a modified genetic algorithm, 1st International Workshop on Informatics & Data-Driven Medicine (IDDM 2018), Lviv, 28–30 October, 2018 : proceedings*. Lviv, CEUR WS, 2018, pp. 1–13.
  30. Leoshchenko S., Oliinyk A., Subbotin S., Gorobii N., Zaiko T. *Implementation of Selective Pressure Mechanism to Optimize Memory Consumption in the Synthesis of Neuro-models for Medical Diagnostics, 2nd International Workshop on Informatics and Data-Driven Medicine (IDDM 2019), Lviv, 11–13 November, 2019. proceedings*. Lviv, CEUR WS, 2019, pp. 109–120.
  31. Leoshchenko S., Oliinyk A., Subbotin S. *Adaptive Mechanisms for Parallelization of the Genetic Method of Neural Network Synthesis, 10th International Conference on Advanced Computer Information Technologies (ACIT 2020), Deggendorf, 16–18 November, proceedings*. Ternopil, IEEE, 2020, pp. 446–450, DOI: 10.1109/ACIT49673.2020.9208905.
  32. South German Credit Data Set [Electronic resource]. Access mode: <https://archive.ics.uci.edu/ml/datasets/South+German+Credit>
  33. Grömping U. *South German Credit Data: Correcting a Widely Used Data Set, Reports in Mathematics, Physics and Chemistry*, Department II, Berlin, Beuth University of Applied Sciences Berlin, 2019, 14 p.
  34. Oliinyk A., Subbotin S., Lovkin V., Leoshchenko S., Zaiko T. *Feature Selection Based on Parallel Stochastic Computing, 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT'2018), Lviv, 11–14 September 2018, proceedings*. Lviv, IEEE, 2018, P. 347–351. DOI: 10.1109/STC-CSIT.2018.8526729
  35. Oliinyk A., Subbotin S., Lovkin V., Leoshchenko S., Zaiko T. *Development of the indicator set of the features informativeness estimation for recognition and diagnostic model synthesis, 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET 2018) Slavsko, 20–24 February 2018, proceedings*. Lviv, IEEE, 2018, P. 903–908. DOI: 10.1109/TCSET.2018.8336342.
  36. Leoshchenko S., Oliinyk A., Subbotin S., Zaiko T. *Methods of semantic proximity extraction between the lexical units in infocommunication systems, 2017 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T) Kharkiv, 10–13 October 2017, proceedings*. Kharkiv, IEEE, 2017, pp. 7–12. DOI: 10.1109/INFOCOMMST.2017.8246137.