

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧОРНОМОРСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ПЕТРА  
МОГИЛИ

**Стовманенко Владислав Олегович**

УДК 004.8

**Проектування нейронних мереж із використанням генетичних  
алгоритмів**

Спеціальність 122 – Комп'ютерні науки

магістерської кваліфікаційної роботи на здобуття освітньої кваліфікації  
«Магістр комп'ютерних наук»

Миколаїв, 2021

Магістерська кваліфікаційна робота є рукопис.

Робота виконана в Чорноморському національному університеті імені Петра Могили Міністерства освіти і науки України на кафедрі інтелектуальних інформаційних систем.

Науковий керівник: канд. тех. наук, доцент кафедри ІІЗ  
Давиденко Євген Олександрович.

Рецензент: д-р. техн. наук, професор кафедри ІІЗ  
Коваленко Ігор Іванович.

Захист відбудеться 22 лютого 2021 р. о 9<sup>30</sup> год. на засіданні екзаменаційної комісії (ауд. 2-403) у Чорноморському національному університеті імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

З магістерською кваліфікаційною роботою можна ознайомитися в бібліотеці Чорноморського національного університету імені Петра Могили за адресою: 54003, м. Миколаїв, вул. 68-ми Десантників, 10.

Автореферат представлений «15» лютого 2021 р.

Секретар  
екзаменаційної комісії,  
к.пед.н., доцент

Н. М. Болюбаш

## ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Дана робота полягала у розробці засобу, який би дозволяв спростити проєктування нейронних мереж використовуючи генетичні алгоритми для побудови структури та встановлення початкових коефіцієнтів зв'язків між нейронами у нейронній мережі. Розглянуто різні структури даних для представлення структури нейронної мережі, проаналізовано, які переваги та недоліки для можливих варіантів.

Окрім того, для перевірки роботи засобу, треба було реалізувати нейромережевий рушій, який би дозволяв створювати нейронні мережі відповідно до створеної структури не попередньому кроці. Створений нейромережевий рушій дозволяє виконувати використовувати нейронні мережі для аналізу вхідних сигналів, а також проводити їх навчання методом зворотного поширення помилки. В роботі було проведено проєктування засобу, яке передбачало створення незалежних компонентів для роботи з генетичними алгоритмами та нейромережевими алгоритмами. Проаналізовано наявні технології для реалізації засобу, розглянуто можливості, які вони надають для виконання поставленого завдання, досліджено засоби, які дозволяють розроблювати застосунки із використанням певної технології.

Було отримано додатковий досвід написання великих звітних робіт, які пояснюють характер наукового дослідження. Опановано поняття “пояснювальна записка”, “унікальність тексту”, “автореферат” та інші пов'язані поняття.

**Актуальність роботи.** Нейромережеві технології проникають у все більшу кількість інформаційних систем й використовуються для покращення користувацького досвіду під час роботи із системою (адаптованість системи під функції, які користувач використовує більше, стилізація відповідно до смаків користувача, тощо), виконання евристичних завдань у конкретній предметній області (розпізнавання облич на фотографії, знаходження закономірностей у

великих обсягах даних, штучний інтелект у іграх, очищення шуму в звуковому сигналі, тощо).

Компанії, які планують використати нейромережі у наявних продуктах або ентузіасти, які їх досліджують завжди стикаються із схожими проблемами. Кожна мережа має бути належним чином спроектованою аби навчитися працювати із набором вхідних даних і вміти формувати певні висновки про них. Якщо система спроектована не правильно не зможе навчитися або буде працювати правильно тільки на певній категорії вхідних даних. При тому не існує єдиного алгоритму, який дозволяє правильно обрати структуру мережі. Існують тільки загальні твердження про ефективність тієї чи іншої архітектури (мережі прямого поширення, рекурентні системи, системи з пам'яттю, тощо). Конкретна будова (кількість нейронів, функції належності, зв'язки між нейронами), має бути встановлена експериментальним шляхом на основі даних, які будуть аналізуватися. Такий підхід вимагає великої кількості спроб та не гарантує знаходження оптимальною чи субоптимальної структури. Часто розробники знаходять структуру достатню для вирішення їх конкретної задачі. Пошук варіантів, які працюватимуть швидше або даватимуть кращі результати ведеться не завжди. В результаті кінцеві користувачі будуть очікувати довше, а системи, які виконуватимуть розрахунки використовуватимуть більше енергії та часу для отримання результатів. Для розробки системи, яка б працювала ефективно необхідно залучати велику кількість експертів, які б враховували особливості задачі під час проектування і встановлювали зв'язки між певними параметрами.

Тому створення засобу для автоматичного або напівавтоматичного проектування (із можливостями для подальших модифікацій) дозволить створювати нейронні мережі швидше і забезпечить їх кращу якість у порівнянні із спроектованими в повністю ручному режимі.

**Мета бакалаврської дипломної роботи** створення прототипу засобу, який проектуватиме структуру нейронної мережі довільного типу відповідно до

вказаної кількості вхідних сигналів для аналізу, очікуваної вихідної кількості сигналів, вказаного стартового шаблону та заданої стратегії побудови нейронної мережі.

**Для досягнення поставленої мети необхідно вирішити такі задачі:**

- проаналізувати структури даних, які дозволять представляти структуру нейронних мереж у вигляді прийнятному для обробки генетичним алгоритмом;
- реалізувати генетичний алгоритм із підтримкою різних стратегії схрещування, відбору та можливістю використовувати довільну функцію пристосованості;
- реалізувати нейромережевий рушій, який дозволить будувати нейронні мережі відповідно до створеної структури, виконувати обчислення із наборами вхідних даних та проводити додаткове навчання мережі;
- спроектувати засіб таким чином аби частини не були залежними одна від одної й існувала можливість змінювати реалізацію кожного алгоритму незалежно від іншого;
- протестувати засіб на конкретній задачі для визначення впливу параметрів генетичного алгоритму на швидкість та ефективність роботи мережі.

**Об'єкт дослідження.** Об'єктом дослідження є алгоритми підбору оптимальної структури нейронних мереж.

**Предмет дослідження.** Предметом дослідження є генетичний алгоритм у ролі генератора оптимальної структури нейронної мережі.

Під час виконання роботи було написано декілька статей, які було роздруковано у різних наукових виданнях.

**Практичне значення отриманих результатів.** Реалізований засіб можна використовувати для проектування нейронних мереж на різних операційних системах та апаратних платформах для пришвидшення проектування нейронних мереж.

**Публікації та апробація.** Дану роботу було апробовано на двох конференціях та представлено статтю у фаховому виданні.

- Стовманенко В. О., Бойко Д. Д., Давиденко Є. О. Структури для даних представлення нейронної мережі. Могілянські читання – 2020 : досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти : XXIII Всеукр. наук.-метод. конф. : тези доповідей Комп'ютерні науки. Технічні науки, Миколаїв, 16-20 листоп. 2020 р. / ЧНУ ім. Петра Могили. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2020. – С. 89-92.
- Стовманенко В.О. Проектування нейронних мереж із використанням генетичних алгоритмів. Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів “Інтелектуальні інформаційні системи”: тези доповідей, Методи і засоби програмної інженерії, Миколаїв, 9 — 12 лютого 2021 року, ЧНУ ім. Петра Могили. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2021. – С. 136-138.

**Структура дипломної роботи.** Дипломна робота складається із вступу, трьох розділів, висновків, окремої спеціальної та методичної частин. Загальний обсяг роботи складає 100 сторінок, в ній 34 рисунки, 2 таблиці та 31 посилання на літературні джерела.

## ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** описано дипломний проект та базові характеристики наукової роботи (мета, об'єкт та предмет).

У **першому розділі (Актуальність проекту)** доводиться актуальність теми, яку обрано для дипломного проекту.

нейронні мережі можна використовувати у будь-яких галузях де є потреба аналізувати зображення, шукати на них певні об'єкти, локалізувати їх (не тільки визначати чи є той чи інший об'єкт на зображенні, але і вказувати конкретне місце цього предмета у зображенні) і, можливо, знаходити певну інформацію про знайдені предмети.

Окрім подібних програмних засобів, існують і програмно-апаратні рішення, які спроектовані таким чином аби операції, які необхідно виконувати під час роботи нейронної мережі виконувалися на апаратному рівні (тобто існує машинна команда, яка виконує вказану операцію і не потребує явної реалізації). Подібні рішення дозволяють отримати значно кращий рівень швидкодії порівняно із програмно реалізованими засобами (проте такі рішення не можна перепрограмувати і всі функції, які будуть використовуватися мають бути спроектованими заздалегідь).

Нейронні мережі можуть використовуватися у різноманітних галузях для автоматизації прийняття рішень та аналізу даних. І у кожному з розглянутих прикладів перед фактичним застосуванням експерти мають сформувати структуру нейронної мережі, яка здійснюватиме конкретну операцію. Тому засіб для автоматизованого або напів автоматизованого проектування нейронної мережі можна буде використовувати у будь-якій галузі, яка потребує використання нейромережових інструментів.

У **другому розділі (Теоретична частина)** описано підходи до побудовий нейронних мереж та принципів роботи покладених у основу їх роботи. Описано роботу штучних нейронів, поширення сигналів (пряме та зворотнє) та

популярні архітектури (як наприклад мережі прямого поширення та рекурентні мережі), а також генетичний алгоритм (загальний опис роботи та інтеграція із нейронними мережами) та відповідні принципи. В реалізації генетичного алгоритму враховано кількісні зміни показників та структурні зміни (утворення нових зв'язків між нейронами, руйнування наявних зв'язків та зміна їх кількості) Кількісні зміни виконуються за допомогою логічних операцій.

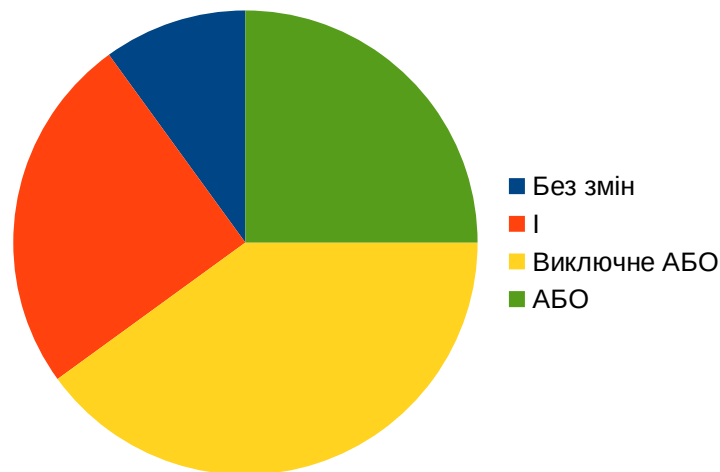


Рис. 1. Діаграма засобів модифікації, що використовуються для якісних змін Окрім того, з імовірністю 2% здійснюються мутації.

За структурні зміни відповідають:

- Заміна ділянки коду. Ймовірність 1%

Це класичний двоточковий кросовер, що змінює дві ділянки коду, випадкової довжини.

- Перестановка колонки та рядку. Ймовірність 1%

Зміна фіксованої ділянки, що відповідає зв'язкам конкретного нейрона.

- Зміна розміру генотипу. Ймовірність 0,5%

Додавання нового нейрона, що включається в мережу випадковим чином.

Новий нейрон не є вхідним та вихідним. Під час додавання формується зв'язок із двома випадковими нейронами.

**У третьому розділі (Практична частина)** проаналізовано технічні засоби, які можна використовувати для реалізації засобу, обґрунтовано вибір засобів, які було використано для використання завдання, розглянуто можливі



середовища розробки, які дозволяють реалізовувати застосунки із використанням обраних технічних засобів.

Для реалізації дипломної роботи було обрано платформу .Net та мову програмування C#. У порівнянні з Java та JVM існує ряд об'єктивних та суб'єктивних переваг:

- Наявність досвіду промислової розробки застосунків на платформі .Net.
- Застій у розвитку мови програмування Java (C# 3: лямбда-функції та анонімні типи, C# 4: динамічні типи, C# 5: асинхронні виклики, C# 6: Інтерполяція рядків та кортежі, C# 7: розпізнавання патернів, Nullable-класи (2007-2017 роки) в той час як Java 1.7.0 : динамічні типи, Java 1.8.0: лямбда-функції, анотації класів, вбудовування JS коду, Java 1.9.0: введення модульної системи, за приблизно той самий проміжок часу).
- Ліцензійні проблеми з правовласниками стандартів (Oracle).
- Наявність кращих засобів для створення графічного інтерфейсу користувача.

Створено функціональні вимоги до бібліотеки, спроектовано ієрархію класів та представлено схему роботи застосунку. Описано роботи з файлами, шаблонами багатопотокового програмування та колекціями на платформі .Net. Приведено результати роботи засобу із різними налаштуваннями генетичного алгоритму, проаналізовано вплив параметрів роботи генетичного алгоритму на ефективність роботи засобу.

Загальний алгоритм роботи застосунку представлено на рис 2.

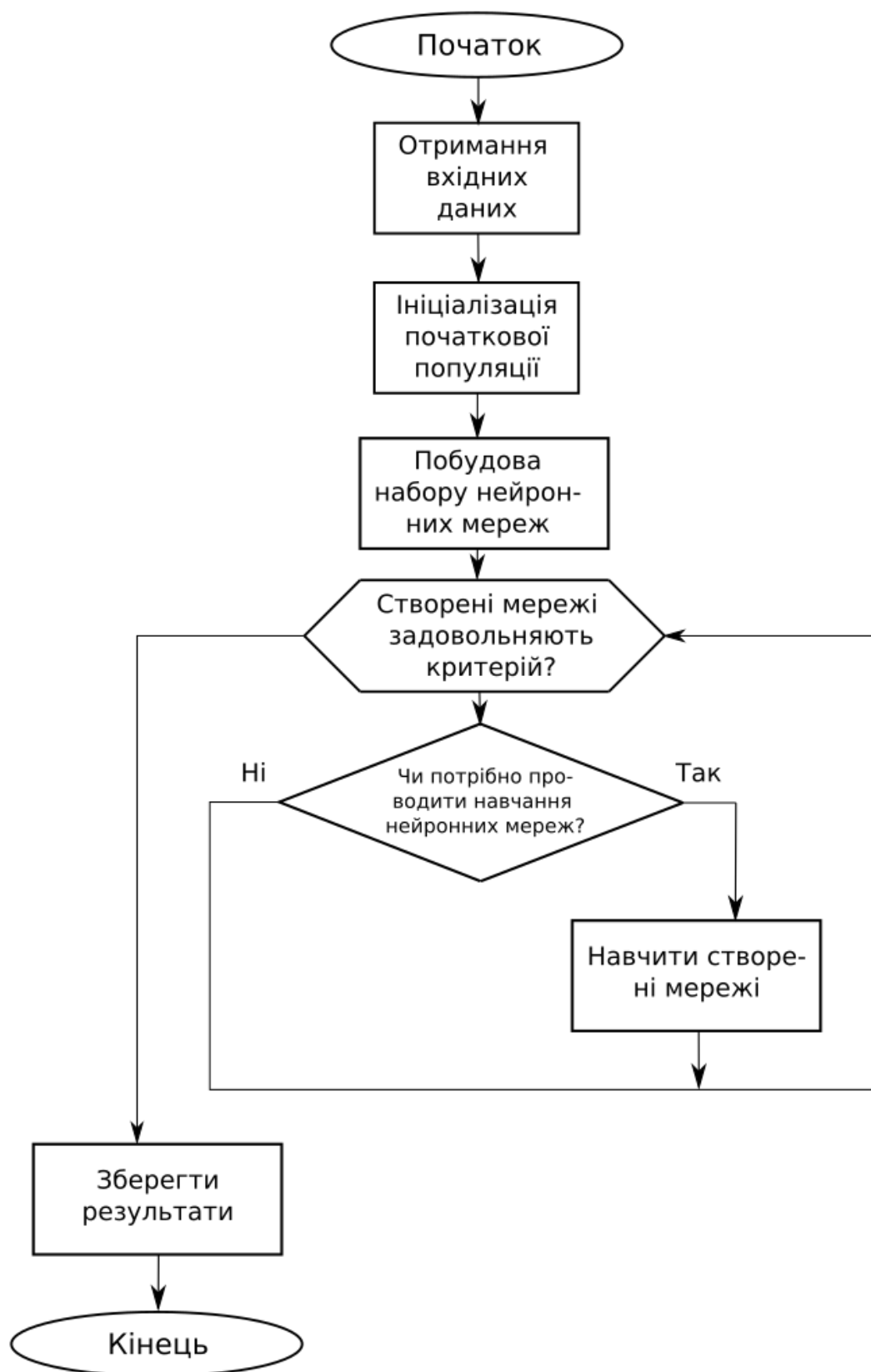


Рис. 2. Блок-схема роботи засобу для проектування нейронної мережі

Для роботи засобу необхідно надати змогу зберігати інформацію про зв'язки між нейронами. Окрім того треба зберігати вагу з'єднань та інформацію про функцію активації.

У оперативній пам'яті граф об'єктів представляється через наявність у одного об'єкта, посилання на інших об'єкт. Таким чином між ними встановлюється однонаправлений зв'язок. У найпростішому випадку єдиного об'єкту "граф" не існує. Є лише об'єкти в яких є посилання на інші об'єкти. Така структура має ряд недоліків [18]:

- неможливість оцінити зв'язки в графі без проходження її вершинами;
- додавання елемента вимагає мати посилання на об'єкт до якого треба додати новий елемент. У разі його відсутності вимагається прохід деревом допоки не буде знайдено потрібний елемент;
- видалення елемента вимагає проходження усіма об'єктами для того аби пересвідчитися в тому що жоден елемент з числа тих, що залишилися не має посилання на видалений елемент.

Для вирішення цих проблем вводиться єдиний об'єкт, який містить додаткову інформацію про структуру графу (при тому посилання між об'єктами один на одний можуть залишатися для зручності роботи з вершиною графою та її сусідами).

```
1 {
2   "matrix": [
3     [0,0,1],
4     [0,0,1],
5     [1,1,0]],
6   "vertices": [
7     {
8       "weights": [],
9       "func": "step"
10    },
11    {
12      "weights": [],
13      "func": "step"
14    },
15    {
16      "weights": [1,0.5],
17      "func": "step"
18    }
19  ]
20 }
```

Рис. 3. Приклад збереженого у JSON графу з матрицею суміжності

Одною із найбільш складних складових генетичних алгоритмів є представлення вхідних даних. Ці алгоритми працюють із генетичним представленням, яке у найпростішому вигляді представляється як одномірний масив байт. Здавалося б найпростішим варіантом було б зчитування представлення усіх об'єктів у оперативній пам'яті комп'ютера і передачі їх у генетичний алгоритм. Проте такий варіант може призводити до неочікваної повенінки пристрою, де виконувалося б виконання такої програми. До того ж не кожна мова може проводити подібні маніпуляції. Тому треба формувати певне користувацьке перетворення даних.

Загальний випадок перетворення графу з списком асоціативності неможливо перевести у двійковий вигляд. Якщо кожен елемент містить колекцію посилань на інші елементи або на відповідні вершини, то можна зіткнутися із тією ж самою проблемою, що і під час запису даних на диск (надмірне копіювання та складності з відновленням об'єкту із записаного стану). Конкретна реалізація перетворення має залежати від механізму розділення даних.

В такій матриці можна зберігати не тільки інформацію про наявність зв'язку, а й, наприклад, інформацію про вагу вхідного з'єднання між нейронами (тоді індекс рядку — це індекс конкретного нейрона, а індекс стовпця — індекс іншого нейрона, вихід якого передається на вхід першого). Це створює деякі складнощі для процесу кросоверу. Для їх виконання часто покладаються на логічні побітові операції для конкретного значення. .Net підтримує подібні операції для чисел із плаваючим положенням крапки, але, за замовчуванням, не дозволяє виконувати їх із користувацькими структурами даних. У першому випадку наявна реалізація спричиняє стрибкоподібні зміни значення пов'язані з особливостями збереження чисел із плаваючою комою в пам'яті комп'ютера. Тому для них треба реалізувати не стандартні логічні операції, які б дозволяли працювати з такими числами так само як з цілими числами (мається на увазі результат). Одним із таких варіантів є використання чисел із фіксованим положенням коми. На жаль такого типу немає в стандартній бібліотеці .Net.

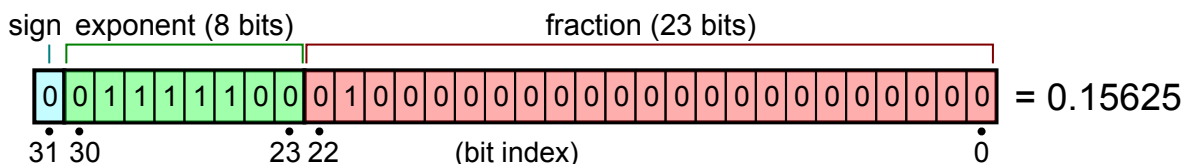


Рис. 4. Представлення чисел з плаваючою комою в пам'яті комп'ютера

У разі не стандартних наборів даних треба реалізувати користувацькі правила виконання логічних операцій і перевантажити відповідні оператори (платформа .Net підтримує перевантаження арифметичних та логічних операцій).

**У спеціальній частині (Охорона праці у надвичайних ситуаціях)** розглядалися умови праці у офіі компанії “Global Logic” у місті Миколаєві, та заходи, які здійснюються під час виникнення надзвичайної ситуації (пожежі), а також те, як відбувається підготовка та тренування робітників для забезпечення належного рівня виконання норм поведінки під час надзвичайних ситуацій.

**У методичній частині (реалізація засобу для роботи з графікою на .Net)** було сформовано практичну роботу, яка дозволяла б сформувавши засіб для роботи із графічними даними, який можна було б інтегрувати в засіб для автоматизованого проектування нейронних мереж. За роки розвитку з'явилося багато сторонніх бібліотек, які дозволяють працювати з графікою. Вони покладаються на роботу графічних бібліотек, які дають змогу працювати з графічними примітивами (Skia, Cairo, тощо). Це потужні інструменти, які можна використовувати на різних апаратних платформах та операційних системах. На їх основі формуються засоби для роботи з графікою, які надають інтерфейс для роботи з об'єктами враховуючи доменну область максимально інкапсулюючи роботу з апаратним забезпеченням (VTK, GTK). Проте малювання конкретних елементів усе одно покладається на користувача. Звісно, існує велика кількість уже готових рішень для візуалізації даних (Oxyplot для .NET, gnuplot для POSIX-сумісного оточення, JFreeChart для Java, тощо). Проте для кращого розуміння і максимального рівня налаштовуваності можна

створити власне рішення, яке вирішуватиме конкретно поставлені задачі і дозволить отримати практичний досвід роботи з графікою на більш “низькому” рівні.

## ВИСНОВКИ

Під час виконання дипломної роботи було спроектовано та розроблено засіб для автоматичного проектування нейронних мереж із використанням генетичних алгоритмів. Його протестовано на задачі побудови структури нейронної мережі для знаходження значення логічної функції. Проаналізовано вплив різних (стратегія формування популяції, стратегія відбору, розмір популяції, тощо) параметрів роботи генетичного алгоритму на ефективність та швидкість роботи засобу. Реалізовано логічні операції для роботи із числами з плаваючою комою, які враховують особливості збереження таких чисел у пам’яті комп’ютерів з архітектурою x86. Закріплено навички роботи із діаграмами функціонального проектування IDEF. Використано шаблони проектування такі як (“фабрика”, “стратегія”, “хранитель” та ін) для реалізації збірки. Описано засоби для роботи із файловою системою платформи .Net, класи, особливості виконання побітових логічних операцій з різними типами даних, а також підходи до паралельного програмування і засоби синхронізації потоків. Закріплено досвід роботи із системами контролю версії (git) та розподіленими методологіями розробки. Код засобу ліцензовано відповідно до Загальної громадської ліценсії третьої версії (GPL v3).

## АНОТАЦІЯ

**Стовманенко Владислав Олегович. Проектування нейронних мереж із використанням генетичних алгоритмів** — на правах рукопису.

Магістерська кваліфікаційної робота на здобуття освітньої кваліфікації «Магістр комп'ютерних наук» – Чорноморський національний університет імені Петра Могили, Миколаїв, 2021.

В даній магістерській дипломній роботі було розглянуто підходи до проектування нейронних мереж. Проаналізовано сферу використання нейронних мереж. Розглянуто можливі варіанти представлення структури нейронної мережі у пам'яті комп'ютера та обрано варіант, який дозволить обробку генетичними алгоритмами. Сформовано набір функціональних вимог до засобу автоматичного проектування нейронних мереж, обґрунтовано вибір технологій для реалізації засобу. Спроектовано алгоритми для роботи із не стандартними наборами даних (логічні операції для чисел із змінним положенням коми та операції для складних типів даних). Реалізовано генетичний алгоритм із декількома стратегіями схрещування та відбору та рушій для побудови, виконання і тренування нейронних мереж відповідно до даної структури. Сформовано тестову нейронну мережу і проаналізовано вплив параметрів роботи генетичного алгоритму на швидкість та ефективність роботи засобу.

Дипломна робота складається із 99 сторінок, 34 рисунків, 2 таблиці, 31 посилань на літературні джерела, спеціальної частини, що складається з 12 сторінок та методичної частини, що складається з 10 сторінок.

**Об'єкт дослідження** — алгоритми підбору оптимальної структури нейронних мереж.

**Предмет дослідження** — генетичний алгоритм у ролі генератора оптимальної структури нейронної мережі.

**Ключові слова:** *нейронні мережі, генетичні алгоритми, структури даних, C#, dotnet, граф, автоматичне проектування, функція пристосованості, бінарне представлення даних.*

# ABSTRACT

**Stovmanenko Vladyslav Olegovich. Neural network designing with genetic algorithm** — manuscript copyright.

This article contains overview of available approaches for neural network designing. Analyzed area of it's application. Reviewed different options for neural network structure storing in computer's memory and chosen the one that allows to process neural network structure with genetic algorithm. A set of functional requirements created for tool that allows automatic or semi automatic neural network structure design and an appropriate set of technologies was chosen to implement this tool. Designed algorithms to apply basic logical operation to non standard (in terms of in memory representation) data types like floating point numbers and complex data types. Implemented genetic algorithm with multiple strategies for crossing, selection and mutation. Implemented neural engine to create neural networks according to given structure representation, process input data with created network and train that network if needed. Tool that was created was tested and it's output is analyzed. Reviewed influence of the genetic algorithm parameter on performance and efficiency of tool.

Article consists of 99 pages, 34 images, 2 table(s), 31 references, special part, that has 12 pages and methodological path that has 10 pages.

**The object of research** — algorithms for designing optimal structure of neural network.

**The subject of research** — genetic algorithm as neural network structure generator.

**Keywords:** *neural networks, genetic algorithms, data structures, C#, dotnet, graph, automatic designing, fitness function, binary data representation.*