

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет
Кафедра

Комп'ютерної інженерії та управління
Комп'ютерних інтелектуальних технологій та систем

КВАЛІФІКАЦІЙНА РОБОТА **Пояснювальна записка**

рівень вищої освіти

другий (магістерський)

Інтелектуальна система підтримки прийняття рішень
для збору працездатних гроху-адресів

Виконав:

студент 2 курсу, групи КІТм-20-1

Васюк Д.В.

Спеціальність 123 Комп'ютерна
інженерія Тип програми освітньо-
професійна Освітня програма

Комп'ютерні інтелектуальні технології

Керівник проф. Корабльов_М.М.

Допускається до захисту

(підпис)

Зав. кафедри

(підпис)

проф. Руденко О.Г.

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління
Кафедра _____ Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти _____ другий (магістерський)
Спеціальність (напрямок) _____ 123 – Комп'ютерна інженерія
(код і назва)
Освітня програма _____ Комп'ютерні інтелектуальні технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Васюку Дмитру Вячеславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інтелектуальна система підтримки прийняття рішень для збору працездатних гроху-адресів

затверджена наказом по університету від “ 08 ” листопада 2021 р. № 1666 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____ 10.12.2021

3. Вхідні дані до роботи _____

- 1) виявлення основних умов працездатності гроху-адрес;
- 2) побудова тестової моделі нейронної мережі для виявлення працездатних гроху-адресів;
- 3) середовище моделювання – Matlab;
- 4) мова програмування – PHP;
- 5) багатofункціональний фреймворк Symfony.

4. Перелік питань, що потрібно опрацювати в роботі _____

- 1) огляд предметної області;
- 2) аналіз предмету дослідження;
- 3) дослідження нейронних мереж;
- 4) дослідження інтелектуальних систем підтримки прийняття рішень;
- 5) дослідження роботи гроху-серверів;
- 6) розробка інтелектуальної системи підтримки прийняття рішень;
- 7) експериментальні дослідження;
- 8) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Плакати - 16 арк. ф. А4

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача та узгодження теми проєкту	08.11.2021	1
2	Огляд стану проблеми та постановка задачі	01.09-14.09	2
3	Аналіз літератури за напрямком магістерської роботи	14.09-21.09	3
4	Аналіз роху-серверів	21.09-28.09	4
5	Аналіз інтелектуальних систем підтримки прийняття рішень	21.09-28.09	5
6	Розробка тестової нейронної мережі	28.09-12.10	6
7	Експериментальні дослідження	12.10-02.11	7
8	Підготовка графічного матеріалу	23.11-07.12	8
9	Перевірка виконаного проєкту керівником	10.12.2021	9
10	Захист проєкту		10

Дата видачі завдання 08 листопада 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Корабльов М.М..

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 82 с., 25 рис., 7 табл., 15 джерел.

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, НЕЙРОННА МЕРЕЖА, PROXY-СЕРВЕР, ШТУЧНИЙ ІНТЕЛЕКТ, PHP, SYMFONY.

Предмет дослідження – методи прийняття рішень на основі використання нейронних мереж.

Метою кваліфікаційної роботи є розробка інтелектуальної системи підтримки прийняття рішень для збору працездатних проху-адресів.

У ході виконання кваліфікаційної роботи проводився аналіз методів підтримки прийняття рішень. Проаналізовано архітектури та типи нейронних мереж. Розроблений сервіс для пошуку працездатних проху-адресів з використанням нейронної мережі та перевірений на вирішення поставлених цілей. Представлені результати експериментальних досліджень, які підтвердили ефективність запропонованого підходу до збору працездатних проху-адресів.

ABSTRACT

Explanatory note of attestation work: 82 pages, 25 figures, 7 tables, 15 sources.

INTELLECTUAL DECISION SUPPORTING SYSTEMS, NEURAL NETWORK, PROXY-SERVER, ARTIFICIAL INTELLIGENCE, PHP, SYMFONY.

The subject of research – decision-making methods based on the use of neural networks.

The purpose of the certification work is to develop an intelligent decision support system for collecting working proxy addresses.

In the course of attestation work the analysis of methods of decision support was carried out. Architectures and types of neural networks are analyzed. Developed a service for finding working proxy-addresses using a neural network and tested for solving the set goals. The results of experimental studies have been presented, which confirmed the effectiveness of the proposed approach to collecting workable proxy addresses.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра Комп'ютерних інтелектуальних технологій та систем

АНОТАЦІЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

рівень вищої освіти другий (магістерський)

Інтелектуальна система підтримки прийняття рішень
для збору працездатних гроху-адресів

Виконав:

студент 2 курсу, групи КІТм-20-1

Васюк Д.В.

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна

Освітня програма Комп'ютерні
інтелектуальні технології

Керівник проф. Корабльов М.М.

2021 р.

Актуальність теми дослідження. Майже кожен ресурс в інтернеті має свою базу даних, котра з кожним днем стає все більше і більше, що в свою чергу уповільнює аналіз цих даних різними збирачами (парсерами). Потрібно більше ресурсів та часу для аналізу даних, через це компанії несуть величезні збитки. На порятунок прийшли проху-сервери. За допомогою проху-адрес, котрі виділяються проху-сервером, можна відправляти більше запитів і не хвилюватись про те, що збирач буде заблокований. Головним недоліком проху-серверів є те, що зазвичай вони не безкоштовні, іноді ціна оренди не великої кількості IP-адрес досягає декілька сотень доларів, але в інтернеті існують сервіси, котрі надають безкоштовні проху-адреси. Але навіть безкоштовні сервіси не є бездоганним вирішенням проблеми. Вони мають наступні недоліки:

- повільність при отриманні відповіді від ресурсу;
- короткий життєвий цикл (до 3 діб);
- використання декількома збирачами одного й того ж проху-адресу;
- не дійсні проху-адреси;
- не автоматична подача нових проху-адрес в систему збирача;
- довга ручна перевірка працездатності проху-адрес.

У магістерській роботі досліджено науково-прикладну проблему створення інтелектуальної систем підтримки прийняття рішень для виявлення працездатних проху-адрес, які повинні частково замінити платні сервіси по наданню проху-адрес.

Об'єктом дослідження є проху-сервери та проху-адреси.

Предметом дослідження є математичні моделі, методи та програмні комплекси, що орієнтовані на обробку інформації в процесі виявлення працездатних проху-адрес .

Дослідження базується на системному аналізі результатів сучасних теоретичних і прикладних розробок вітчизняних і зарубіжних учених в ІТ галузі. Для вирішення поставлених завдань використано: методи системного аналізу, методи побудови нейронних мереж, методи побудови

інтелектуальних систем підтримки прийняття рішень, методи об'єктно-орієнтованого програмування, методи побудови програмних застосунків з командним інтерфейсом.

Метою даної роботи є розробка проєкту інтелектуальної системи підтримки прийняття рішень здатної надати кожному збирачу працездатну гроху-адресу. Вимогами до системи є:

- збирати гроху-адреси з різних ресурсів;
- приймати рішення по працездатності зібраних гроху-адрес;
- зберігати інформацію по працездатності гроху-адрес;
- видавати збирачам працездатні гроху-адрес.

У першому розділі розглянуто аналіз предметної області і були поставлені задачі дослідження. Сучасні системи підтримки прийняття рішень – це системи максимально пристосовані до вирішення завдань повсякденної управлінської діяльності та є інструментом, призначеним для допомоги особам, які приймають рішення. Інтелектуальна система підтримки прийняття рішень це така система, яка за допомогою великих обсягів даних та їх аналізу може впливати на процес ухвалення рішень. Найбільш широко СППР використовується у сфері планування та прогнозування діяльності підприємств, завданням яких є забезпечення керівників належними засобами управління, що пробуджують інтерес, зацікавленість, цілеспрямованість менеджерів для досягнення кінцевого результату. Але суто економічним застосуванням СППР не обмежуються. Вони також використовуються у банківських справах, телекомунікаціях, страхуванні, медицині, правоохоронних органах та судовій справі, національній безпеці, митниці та поліції. В основі будь-якої інтелектуальної системи підтримки прийняття рішень лежить нейронна мережа. Відмінною особливістю мереж є те, що вони не мають певних правил для отримання кінцевого результату, вони не запрограмовані, але вони навчаються робити це на прикладах. Вони забезпечують діагностику, ідентифікацію та класифікацію складних об'єктів з чіткими, складними нелінійними технічними умовами. Зроблений огляд

сучасної літератури. Особлива увага приділяється інтелектуальним системам підтримки прийняття рішень. Був проведений огляд типів ІСППР. З цього огляду були зроблені висновки, що більш підходящим типом є ІСППР на основі машинного навчання. Також досліджувалися методи штучного інтелекту, які застосовуються в ІСППР, були визначені основні з них. Також були поставлені задачі котрі треба дослідити.

У другому розділі були досліджені загальні відомості про проху-сервери. Проху – це проміжний транзитний веб-сервер, який використовується як посередник між користувачем та кінцевим сервером. Проху-сервери забезпечують різні рівні функціональності, безпеки та конфіденційності залежно від вашого випадку використання, потреб або політики компанії. Їх можна налаштувати як веб-фільтри або брандмауери, захищаючи ваш комп'ютер від інтернет-загроз, таких як зловмисне програмне забезпечення. Були розглянуті основні види проху-серверів та проху-адресів, та виявлені основні переваги й недоліки.

У третьому розділі йдеться про створення ІСППР для визначення працездатності проху-адресів. Були досліджені типи моделей нейронних мереж. На основі цих досліджень була обрана найоптимальніша модель – одношаровий перцептрон з сигмоїдальною функцією активації на прихованому шарі та ступінчатою функцією активації на виході мережі. Було розглянуто методи відправки запитів на ресурс, а також фактори, котрі можуть впливати на відповідь з ресурсу. На основі цих досліджень, були виявлені основні умови завдяки яким можна відрізнити працездатний проху-адрес від не працездатного. У результаті чого була побудована нейромережева модель перцептрона на вхід якого приходять 8 умов працездатності з одним прихованим шаром та одним вихідним значенням.

Четвертий розділ присвячений експериментальним дослідженням. Моделювання нейронної мережі проводились в середовищі Matlab. Сам процес навчання відбувався наступним чином. Спочатку були створені випадкові ваги (це потрібно для подальшого їх корегування). Після створення

ваг, були створені навчальні набори, завдяки яким нейронна мережа буде вчитись та корегувати ваги. Із результатів першого навчання нейронної мережі були зроблені висновки, що не достатньо 30 навчальних наборів щоб виявити працездатні проху-адреси, в результаті чого було створено ще 30 додаткових навчальних наборів. Результати другого навчання були ліпшими за перші, але все одно були далекими від ідеалу, після чого було прийнято рішення додати ще 60 навчальних наборів. Результати третього навчання нейронної мережі показали, що нейронна мережа закінчила своє навчання і може виявляти працездатні проху-адреси. В сумі знадобилось 120 навчальних наборів.

На основі побудованої нейронної мережі було створено інтелектуальну систему підтримки прийняття рішень. Дана система представлена як консольний додаток. Сам процес виявлення працездатних проху-адресів відбувається в декілька етапів. Перший етап – це етап збору проху-адрес з різних ресурсів за допомогою збирачів. Другий етап – етап виявлення працездатних проху-адрес. Третій етап – збереження результатів. Дана система показала, що вона може виконувати поставлені завдання.

Ключові слова: проху-сервер, проху-адреса, нейронна мережа, інтелектуальна система підтримки прийняття рішень, процес виявлення працездатних проху-адрес.

Список опублікованих робіт за темою магістерської роботи:

Васюк Д.В. Система підтримки прийняття рішень на основі штучних нейронних мереж// XXV Міжнародний молодіжний форум «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ», 21-23 квітня 2021 р. м. Харків

Васюк Д.В. Огляд методів збору інформації // Матеріали Міжнародної науково-практичної конференції молодих учених, аспірантів та студентів «Інформаційні технології в сучасному світі: дослідження молодих вчених» 18 – 19 березня 2021 р. м. Харків

ЗМІСТ

ВСТУП	14
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	16
1.1 Основні відомості про системи підтримки прийняття рішень.....	16
1.2 Інтелектуальні системи підтримки прийняття рішень.....	19
1.2.1 ІСППР на основі ШІ.....	20
1.2.2 ІСППР на основі сховища даних	21
1.2.3 ІСППР на основі аргументації випадків	22
1.2.4 Характеристики ІСППР	22
1.3 Методи штучного інтелекту, які застосовуються в ІСППР	23
1.3.1 Штучні нейронні мережі для підтримки інтелектуальних рішень ...	24
1.3.2 Нечітка логіка для підтримки інтелектуальних рішень	26
1.3.3 Експертні системи для підтримки інтелектуальних рішень.....	28
1.3.4 Еволюційні обчислення для підтримки інтелектуальних рішень.....	29
1.3.5 Інтелектуальні агенти для підтримки інтелектуальних рішень	30
1.4 Постановка задачі дослідження.....	33
2 PROXY-СЕРВЕРИ	34
2.1 Загальні відомості	34
2.2 Види та типи проху-серверів	36
2.3 Переваги та недоліки проху-сервера	39
3 СТВОРЕННЯ ІСППР ДЛЯ ВИЗНАЧЕННЯ ПРАЦЕЗДАТНОСТІ PROXY-АДРЕСІВ	41
3.1 Типи моделей нейронних мереж	41
3.2. Методи визначення працездатності проху-адресу.....	49
3.3 Методи отримання відповіді від ресурсу	50
3.4 Пошук умов для визначення працездатності проху-адрес.....	52
3.5 Побудова нейромережевої моделі.....	53

3.6 Алгоритм навчання перцептрона	56
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	58
4.1 Опис середовища для проведення експериментальних досліджень	58
4.2 Тестування моделі нейронної мережі	58
4.3 Розробка додатку інтелектуальної системи підтримки прийняття рішень	64
ВИСНОВКИ.....	68
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	70
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	72
ДОДАТОК Б	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

- ЕС – Експертна система
- ІА – Інтелектуальний агент
- ІСППР – Інтелектуальна система підтримки прийняття рішень
- НМ – Нейронна мережа
- ОПР – Особа, яка приймає рішення
- СППР – Система підтримки прийняття рішень
- ШІ – Штучний інтелект
- FF – Feed-Forward Neural Network, нейронна мережа із прямим зв'язком
- HTTP – HyperText Transfer Protocol, протокол передачі даних
- HTTPS – HyperText Transfer Protocol Secure, розширення протоколу HTTP з підтримкою шифрування
- MAS – Multi-agent systems, багатоагентна система
- RNN – Recurrent Neural Networks, рецидивні нейронні мережі
- SOCKS – мережевий протокол сеансового рівня моделі OSI, який дозволяє пересилати пакети від клієнта до сервера через проху-сервер прозора

ВСТУП

Сьогодні, у складному динамічному середовищі, що характеризується постійною невизначеністю та мінливістю політичних, економічних та соціальних факторів, основою успішного функціонування суб'єктів господарювання є прийняття адекватних управлінських рішень. Сучасні системи підтримки прийняття рішень – це системи максимально пристосовані до вирішення завдань повсякденної управлінської діяльності та є інструментом, призначеним для допомоги особам, які приймають рішення.

Інтелектуальна система підтримки прийняття рішень – система, яка за допомогою великих обсягів даних та їх аналізу може впливати на процес ухвалення рішень. В основі будь-якої інтелектуальної системи підтримки прийняття рішень лежить нейронна мережа.

Використання нейронних мереж відкриває новий рівень вивчення процесів у таких системах, як організм тварин або людей. Відмінною особливістю мереж є те, що вони не мають певних правил для отримання кінцевого результату, вони не запрограмовані, але вони навчаються робити це на прикладах.

Нейронні мережі можуть вирішувати математичні задачі з недоступною обробкою, класифікації образів, порівняння, здатністю до самонавчання та самоорганізації. Вони забезпечують діагностику, ідентифікацію та класифікацію складних об'єктів з чіткими, складними нелінійними технічними умовами.

Метою даної роботи є розробка інтелектуальної системи підтримки прийняття рішень для збору працездатних проху-адресів.

Досягнення поставленої мети передбачає реалізацію таких завдань:

- детальний аналіз інтелектуальних систем підтримки прийняття рішень, нейронних мереж, проху-серверів та проху-адресів;
- виявлення умов для класифікації працездатності проху-адресів;

- побудова тестової математичної та програмної моделі нейронної мережі для класифікації працездатності гроху-адресів;

- створення додатку на основі досліджень нейронної мережі по класифікації працездатності гроху-адресів для підтримки прийняття рішень.

Предметом дослідження є методи прийняття рішень на основі використання нейронних мереж.

У ході виконання кваліфікаційної роботи проводився аналіз методів підтримки прийняття рішень. Проаналізовано архітектури та типи нейронних мереж.

Як результат, розроблений сервіс для пошуку працездатних гроху-адресів з використанням нейронної мережі та перевірений на вирішення поставлених цілей. Представлені результати експериментальних досліджень, які підтвердили працездатність як нейронної мережі так і інтелектуальної системи підтримки прийняття рішень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Основні відомості про системи підтримки прийняття рішень

Система підтримки прийняття рішень (СППР) – спеціальне апаратне і програмне забезпечення, що дозволяє людям приймати виважені обґрунтовані рішення. Така програма функціонує як база даних з функціями їх накопичення, аналізу, формування зручних для роботи звітів. Вона дозволяє визначитися з вибором навіть у мінливій обстановці і при високому відсотку невизначеності.

СППР з'явилися внаслідок розвитку адміністративних інформативних систем і розроблені для підтримки процесів прийняття рішень. Вони відносяться до широкого спектру інтерактивних комп'ютерних систем, які допомагають особам, які приймають рішення, використовувати дані, моделі та знання для вирішення напівструктурованих, погано структурованих або неструктурованих проблем. Дані системи широко використовуються для організації ефективного управління бізнесом та полегшують роботу менеджерів зі збору та аналізу інформації, виявлення проблем та прийняття вірних рішень.

Найбільш широко СППР використовується у сфері планування та прогнозування діяльності підприємств, завданням яких є забезпечення керівників належними засобами управління, що пробуджують інтерес, зацікавленість, цілеспрямованість менеджерів для досягнення кінцевого результату. Але суто економічним застосуванням СППР не обмежуються. Вони також використовуються у банківських справах, телекомунікаціях, страхуванні, медицині, правоохоронних органах та судовій справі, національній безпеці, митниці та поліції [1].

СППР вирішує ряд важливих проблем, а саме:

- удосконалення рішень: СППР створюють умови для того, щоб за

допомогою комп'ютерних технологій з урахуванням часових і пізнавальних (когнітивних) обмежень та економічних лімітів менеджери розв'язували більше проблем та приймати кращі рішення.

- збільшення продуктивності праці творців рішень, тобто їх здатності створювати за менший період якісніші рішення.

- доповнення інструментальних засобів новими, більш ефективними можливостями у добуванні, формулюванні та створенні нових знань засобом аналізу та визначення проблем.

- полегшення виконання етапів прийняття рішень (збір інформації, проектування, відбір альтернатив).

- полегшення і упорядкування аналізу можливих шляхів розв'язування проблем.

- допомога творцям рішень у розв'язанні неструктурованих або напівструктурованих проблем.

- автоматизація управлінських процесів [2].

Більшість СППР призначені для вирішення конкретної проблеми або загального класу проблем, тому існує багато різновидів, спеціалізованих для різних типів користувачів і проблем. Не існує єдиної класифікації, тому наведу деякі основні характеристики за якими проводиться класифікація:

а) На рівні користувача:

- 1) активна – не лише надають інформацію, а й самі можуть пропонувати рішення та альтернативи;

- 2) пасивна – підтримують процес прийняття рішень, але самі ці рішення не формулюють і не пропонують, а лише надають інформацію та звіти: рішення ОПР приймає сама;

- 3) комбіновані – забезпечують ітеративний процес взаємодії між ОПР та СППР для досягнення консолідованого рішення: ОПР (або його консультант) може змінювати, доповнити чи уточнити пропозиції СППР та відправляти їх назад до системи для перевірки. Цей процес триває доти, доки не буде отримано оптимальне рішення.

б) На технічному рівні:

1) СППР рівня підприємства – працюють з великими сховищами даних і застосовуються багатьма користувачами;

2) персональна настільна СППР – є невеликими системами та підходять для управління з персонального комп'ютера одного користувача.

Всі різновиди СППР характеризуються однотипною структурою, зображеною на рис. 1.1.

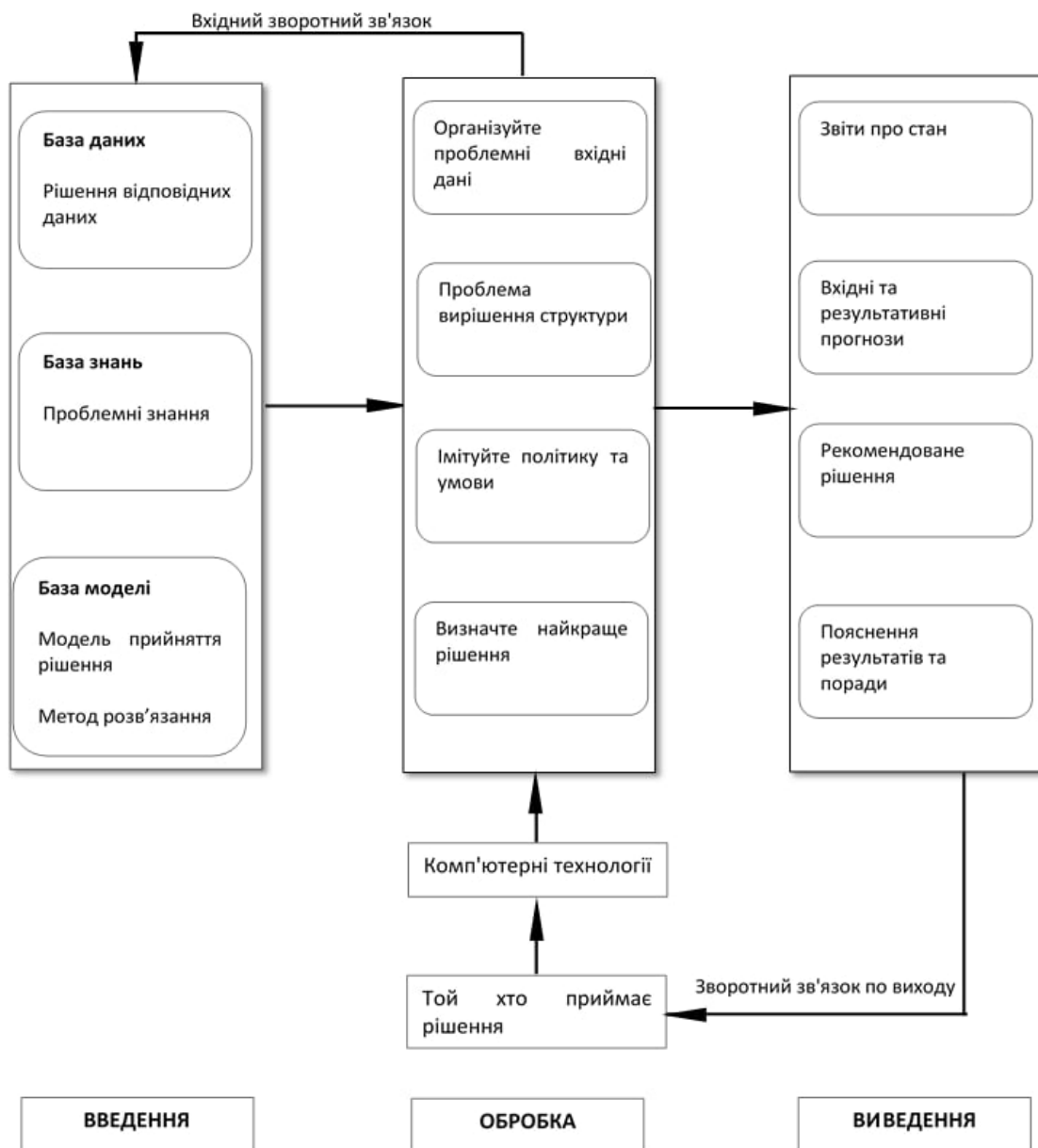


Рисунок 1.1 – Структура системи підтримки прийняття рішень

Вхідні дані включають базу даних, базу знань і базу моделей. База даних

містить дані, що мають відношення до проблеми рішення, тоді як база знань може включати, наприклад, рекомендації щодо вибору альтернатив. База моделей містить формальні моделі, алгоритми та методи отримання результатів. Обробка включає використання моделей рішень для моделювання або вивчення різних станів та пошуку найкращого рішення з урахуванням обмежень. Зворотній зв'язок з обробки може надати додаткові вхідні дані, які можна оновлювати в режимі реального часу для покращення вирішення проблеми. Вихідні дані можуть генерувати прогнози та пояснення для обґрунтування рекомендацій та порад. Результати надаються особі, яка приймає рішення і може взаємодіяти з системою, щоб надати додаткову інформацію або запросити результати.

Розрізняють 5 видів комп'ютерних СППР:

- Комунікативні: орієнтовані на одночасну роботу кількох фахівців, зайнятих одним спільним завданням.
- Інформаційні: зосереджені на збиранні та обробці даних, переважно аналізі часових рядів, функціонують, як СУБД у межах однієї компанії.
- Документальні: призначені для обробки та аналізу документів різного формату зі структурованими та не структурованими даними.
- Інтелектуальні: містять дані про рішення аналогічних завдань, норми та правила, на підставі яких вони приймалися, пропонують готові алгоритми, виходячи з накопиченого досвіду.
- Модельовані: підбирають моделі бізнес-процесів за заданими умовами – статистичні, фінансові, аналітичні.

У даній кваліфікаційній роботі розглянуто саме інтелектуальні системи підтримки прийняття рішень.

1.2 Інтелектуальні системи підтримки прийняття рішень

Інтелектуальні системи підтримки прийняття рішень (ІСППР) використовують методи штучного інтелекту для посилення та покращення

підтримки осіб, які приймають рішення. Вони включають в себе чітку процедуру прийняття рішення, засновану на наборі теоретичних принципів, які обґрунтовують «раціональність» цієї процедури.

Інтелект можна розглядати так, що це насамперед питання раціональних дій, щоб інтелектуальна система зробила найкращі можливі дії у даній ситуації. ІСППР має деякі можливості, що вказують на «розумну поведінку», такі як:

- навчання на досвіді;
- розуміння через двозначність або протиріччя;
- своєчасне та належне реагування на нову ситуацію;
- використання міркувань для розв'язування задач і раціональних висновків;
- справлятися зі складними ситуаціями;
- застосування знань для розуміння або зміни навколишнього середовища;
- визнання відносної важливості різних факторів у прийнятті рішення.

Відповідно до методу інтелектуального прийняття рішень, ІСППР можна розділити на 3 типи:

- 1) ІСППР на основі штучного інтелекту;
- 2) ІСППР на основі сховища даних;
- 3) ІСППР на основі аргументації випадку.

1.2.1 ІСППР на основі ШІ

Інтелектуальні системи підтримки прийняття рішень на основі штучного інтелекту включають наступні типи:

- 1) ІСППР на основі експертних систем. Експертна система складається з бази знань, механізму висновків і бази даних. Експертна система використовує декількісні логічні твердження для вираження знань і використовує автоматичне міркування для вирішення питань. ІСППР в основному

використовує метод кількісної оцінки для моделювання проблем і пропонує підтримку прийняття рішень з використанням результату розрахунку ціннісної моделі.

2) ІСППР на основі машинного навчання. Машинне навчання передбачає отримання знання про вирішення людських проблем за допомогою комп'ютера, що моделює навчання людини. Машинне навчання може автоматично здобувати знання. Таким чином, можна певною мірою усунути вузьке місце в отриманні знань в експертній системі.

3) ІСППР на основі агента. На даний момент агент є центром досліджень у сфері штучного інтелекту, включаючи дослідження інтелектуального агента, дослідження системи Multi-Agent та дослідження агентно-орієнтованих програм.

1.2.2 ІСППР на основі сховища даних

Завдяки узагальненню, збору та інтеграції інформації про джерела даних, сховище даних створює інтегрований, змінний і довговічний набір даних, присвячений темі, і пропонує інформацію, корисну для прийняття рішень. Онлайн аналітична обробка (OLAP) одночасно розвивається зі сховищем даних. Завдяки багатовимірному та комплексному запиту та синтетичному аналізу сховища даних у реальному часі можна дізнатися загальні характеристики та тенденції розвитку, приховані в даних. Архітектура ІСППР на основі аналізу даних показана на рис. 1.2.



Рисунок 1.2 –Архітектура ІСППР на основі аналізу даних

1.2.3 ІСППР на основі аргументації випадків

Міркування на основі випадку полягає у тому, щоб знайти метод вирішення поточних проблем із попереднього досвіду. Сукупність попередніх інцидентів може сформувати базу випадку, а саме модель вирішення проблем. Поточні проблеми – це цільові справи; проблеми або обставини в пам'яті є вихідними випадками. У той час як CBR розглядає проблеми, шукає вихідні випадки, які мають ті самі атрибути, що й цільові випадки з бази випадків; потім відрегулює, зіставляючи випадки. Міркування, засноване на прикладах, спрощує процес набуття знань, повторно використовує попередній процес розв'язування і відіграє хорошу функцію для задач, які можна вирішити за допомогою обчислень [3-4].

1.2.4 Характеристики ІСППР

Інтелектуальна система підтримки прийняття рішень має такі характеристики:

- Певна здатність до самостійного навчання. Особам, які приймають рішення, дозволяється модифікувати та розширювати знання в базі знань. Таким чином, здатність розв'язувати задачі може бути додатково покращена.

- ІСППР має механізм міркування. Він імітує процес мислення осіб, які приймають рішення, і застосовує відповідні знання, щоб направляти ОПР, вибирати правильні моделі прийняття рішень шляхом співбесіди «людина-машина» відповідно до вимог осіб, які приймають рішення.

- ІСППР має інтелектуальну функцію керування моделлю. Вона керує моделлю як різновидом структури знань і спрощує інтерфейс між різними підсистемами.

- ІСППР будує універсальну структуру системи підтримки прийняття рішень, щоб розширити поле обслуговування системи та зробити систему краще адаптованою до зміни середовища та зміни форми рішень [5].

Використання інструментів і моделей штучного інтелекту забезпечує прямий доступ до досвіду, а їхня гнучкість робить їх здатними підтримувати процеси навчання та прийняття рішень. Інтеграція з числовими та/або статистичними моделями в єдину систему забезпечує більш високу точність, надійність і корисність.

1.3 Методи штучного інтелекту, які застосовуються в ІСППР

Інтелектуальні системи підтримки прийняття рішень стають корисними для практичних і важливих застосувань і використовують різноманітні методи штучного інтелекту. Застосування варіюються від підтримки охорони здоров'я до бізнес-рішень, і все це дозволяє покращити прийняття рішень людьми.

У даному підрозділі розглянуто основи найбільш поширених методів штучного інтелекту та їх внесок у прийняття рішень.

1.3.1 Штучні нейронні мережі для підтримки інтелектуальних рішень

Штучні нейронні мережі – сукупність тісно взаємопов'язаних блоків обробки, які називаються нейронами і працюють разом для вирішення проблеми. Нейронні мережі були натхненні тим, як мозок обробляє інформацію і, як правило, складається з шарів нейронів, як показано на рис. 1.3. Перевагою НМ є їхня здатність представляти будь-яку обмежену безперервну функцію з будь-якою невеликою помилкою апроксимації.

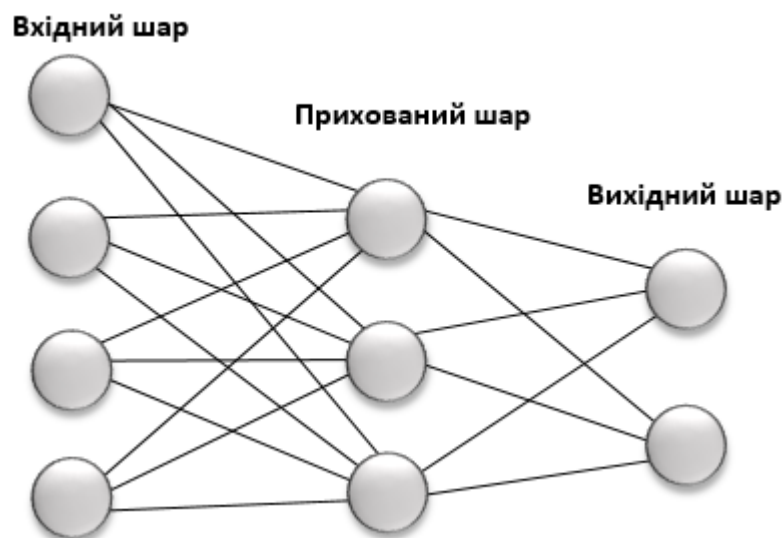


Рисунок 1.3 – Нейронна мережа з прихованим шаром

Основною одиницею нейронної мережі є нейрон або вузол. Кожен нейрон отримує вхідний сигнал x_i як стимул, з пов'язаною вагою w_i , що виражає відносну важливість x_i , від іншого нейрона або зовнішнього джерела. Вага може бути позитивним або негативним значенням, тобто збудливим або гальмуючим. Нейрон обчислює зважену суму всіх вхідних даних нейрона, тобто для j -го нейрона, де Y_j представляє рівень активації нейрона (формула 1.1).

$$Y_j = \sum_i x_i w_{ij} \quad (1.1.)$$

Інформація може бути модифікована та передана з нейрона за допомогою кількох різних типів функцій перетворення, які можуть бути нелінійними. Вагові коефіцієнти w_i мають вирішальне значення для навчання в НМ, оскільки вони коригуються у міру появи нових даних. Зазвичай НМ піддається ітераційному впливу навчального набору даних, який включає фактичні пов'язані вихідні дані, щоб розробити набір вагових коефіцієнтів, доки вихідні дані НМ не відповідають фактичним результатам до бажаного рівня точності. Потім НМ можна використовувати для прогнозування майбутніх станів за допомогою набору вхідних даних або «навчитися» переналаштувати вагові показники, коли йому надаються додаткові набори вхідних/ вихідних даних. Ця функція дозволяє нейронній мережі знаходити закономірності та узагальнювати на основі попередніх випадків, а також класифікувати на основі спостережень, так само, як люди використовують емпіричні спостереження, щоб запропонувати результати минулої історії чи спостережуваної поведінки.

Дві загальні топології НМ – це мережі з прямим зв'язком і рекурентним або зворотним зв'язком. Як впливає з назви, у прямому зв'язку нейронної мережі сигнали надходять від входів до виходів i , можливо, через кілька прихованих шарів. Наприклад, на рисунку 1.3 показано чотири входи, які протікають через прихований шар, що складається з трьох вузлів, до вихідного шару. НМ з прямим зв'язком найчастіше використовуються в проблемах прийняття рішень, оскільки потік інформації узгоджується з процесом прийняття рішень. Нейронна мережа зі зворотним зв'язком може мати сигнали, що рухаються в обох напрямках.

НМ відрізняється від послідовних, заснованих на логіці підходів, які передбачають певний тип взаємозв'язку між входами та виходами. Їх фундаментальна перевага полягає у тому, що в процесі налаштування ваг вони можуть природно відображати нелінійність. НМ пропонує підтримку вирішення важливих проблем, таких як виявлення шахрайства, які майже неможливо змоделювати за допомогою логічних підходів. З іншого боку, НМ

не підходять для таких операцій, як обробка даних.

Нейронні мережі «вивчають» основну функцію в даних, використовуючи одну з трьох стратегій з «даними навчання»: навчання без нагляду, під наглядом або навчання з підкріпленням. Неконтрольоване навчання відбувається, коли НМ отримують тільки вхідні дані і не мають відповідних вихідних даних. Мета полягає у тому, щоб визначити основну структуру даних. У керуваному навчанні НМ отримує вхідні дані та відповідні вихідні дані, щоб вона намагалася налаштувати ваги на різних входах, щоб вихідні дані, вироблені НМ, були в межах бажаної похибки апроксимації вихідних даних вибірки (цілей). Визначення ваги на входах дозволяє використовувати НМ для прогнозування з новим набором вхідних даних. Однак у багатьох практичних ситуаціях детальні n -кортежі введення-виведення недоступні, і може бути лише обмежена або навіть рідкісна кількість виходів після великої кількості вводів. Навчання з підкріпленням використовується для вирішення цієї ситуації та надання певного зворотного зв'язку НМ, щоб оцінити, чи правильно вибрані ваги. Під час навчання НМ проблемі ухвалення рішення слід уникати перенавчання (чи занадто точного зіставлення навчальних даних), щоб НМ можна було узагальнити і використовуватиме цілей прогнозування.

Нейронні мережі іноді називають «чорними ящиками», що означає, що інтерпретація моделі складна для особи, яка приймає рішення. Оскільки обчислення розподіляється між кількома або багатьма вузлами та прихованими шарами, правила прийняття рішень і відносини між змінними не можуть бути ідентифіковані з НМ [6].

1.3.2 Нечітка логіка для підтримки інтелектуальних рішень

Нечітка логіка розширює підтримку прийняття рішень, дозволяючи відображати вхідні дані або змінні в проблемі рішення так, як люди міркують про них. Особи, які приймають рішення, часто стикаються з проблемами, коли

вхідні дані є неточними або невизначеними. Наприклад, погода може бути сонячною, частково сонячною, переважно хмарною або похмурою. Для порівняння, булева логіка – це система символічної логіки, яка керує логічними функціями на комп'ютері і заснована на двійковій системі 0 (повністю хибна) і 1 (повністю істинна). Нечітка логіка дозволяє відобразити невизначеність, дозволяючи вхідним даним мати діапазон значень від 0 (повністю хибно) до 1 (повністю істинно).

Нечітка логіка може бути корисною в ІСППР, забезпечуючи:

- гнучкість врахувати несподіванки;
- варіанти інтуїції, такі як «ймовірно» або «дуже добре»;
- здатність уявляти сценарії «що, якщо»;
- низький ризик для неправильного вибору, оскільки існує діапазон значень на вибір;
- підходи до моделювання для задач з невизначеністю, які важко представити в математичних моделях.

Нечітка логіка гнучкіша у поданні, тому особа, яка приймає рішення, має широкий вибір і може вільно оцінювати значення вхідних даних. У нечіткій логіці немає внутрішньої структури, тому нелінійні зв'язки можна інкапсулювати природним чином без попереднього планування. У міру появи нової інформації значення можна легко уточнювати та змінювати, що дає особі, яка приймає рішення, природний спосіб упоратися з невизначеністю. Нечітка логіка забезпечує спосіб представлення поведінки, заснованої на правилах, наприклад, знання від експерта, щоб у потрібний момент можна було отримати досвід і надати особі, яка приймає рішення. Нечітка логіка також може бути поєднана з НМ, щоб інтерпретація змінних рішення була більш очевидною. Наприклад, вхідні змінні можуть бути описані трьома значеннями, такими як максимальне, мінімальне та найбільш ймовірне значення. Це описи природною мовою, які покращують здатність особи, яка приймає рішення, передавати знання предметної області моделі та інтерпретувати результат.

Нейронні мережі з нечіткою логікою є категорією моделей прийняття рішень, які можуть надати багат шарової мережі прямого зв'язку зі зрозумілим для людини значенням. Агрегативні нейрони виконують логіку AN-OR, тоді як референтні нейрони підтримують обробку на основі предикатів, наприклад, менше, більше або подібне, виражене мовою нечіткої логіки. Нечітка логіка НМ допомагає усунути деякі недоліки НМ щодо прозорості для покращення прийняття рішень.

1.3.3 Експертні системи для підтримки інтелектуальних рішень

Експертна система (ЕС) – це комп'ютерна система, яка намагається вирішити проблеми, які зазвичай вирішує людина-експерт. Часто термін ЕС використовується для опису системи, яка включає інтелект одного або кількох ідентифікованих експертів. Розробнику системи необхідно вивчити, як людина-експерт приймає рішення, а потім впровадити ці знання в комп'ютерну систему.

Компоненти ЕС показані на рисунку 1.4. Він показує експерта з предметної області, який надає знання для модуля отримання знань. Ці знання закодовані в базі знань, як правило, як частина процесу розробки.

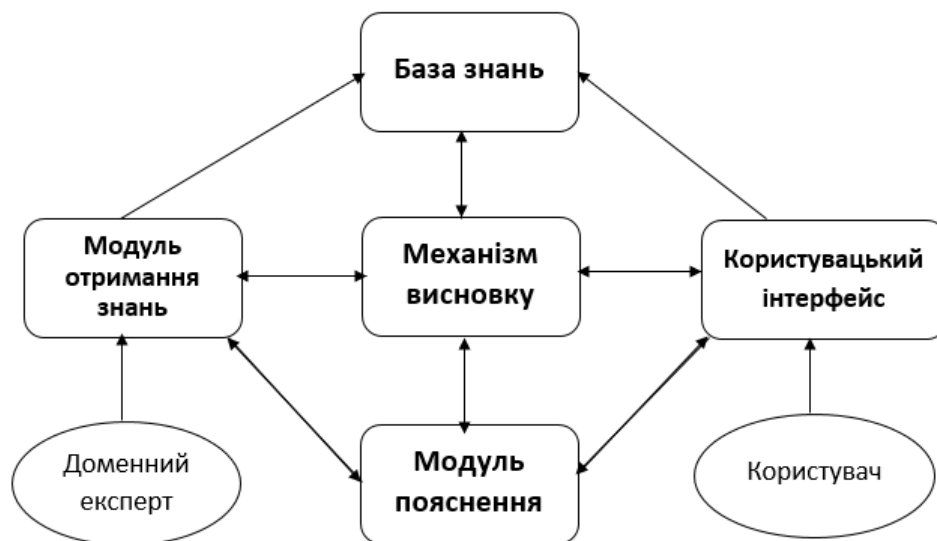


Рисунок 1.4 – Компоненти експертної системи

Користувач або особа, яка приймає рішення, входить до системи через інтерфейс. Потім користувач може отримати прямий доступ до бази знань для минулих випадків або до механізму виведення, щоб зробити висновок від минулих випадків до нового. Користувач може «деталізувати» для пояснення висновку з модуля пояснення. Таким чином, ЕС служить для одержання, збору та отримання знань від експерта в предметній області та передачі цього досвіду особі, яка приймає рішення [7].

1.3.4 Еволюційні обчислення для підтримки інтелектуальних рішень

Еволюційні обчислення черпають натхнення з природної еволюції, коли особини в популяції еволюціонують, щоб збільшити рівень свого виживання, підвищуючи рівень пристосованості до мети. Методи штучного інтелекту намагаються імітувати ці характеристики адаптації до навколишнього середовища, моделюючи появу, виживання та вдосконалення популяції індивідів. Генетичні алгоритми є одними з найбільш використовуваних для вирішення проблем. Після ініціалізації популяції наступні покоління взаємодіють, спілкуються та впливають один на одного, щоб краще адаптуватися до навколишнього середовища.

Узагальнена блок-схема для генетичних алгоритмів показана на рис. 1.5. Кінцева сукупність випадковим чином ініціалізується в момент $t = 0$, і встановлюється мета. Кожна людина оцінюється відповідно до мети і характеризується значенням придатності, яке представляє пристосованість індивіда до навколишнього середовища. Чим вище показник пристосовуваності, тим більша ймовірність того, що людина виживе і стане батьком для наступних поколінь. Батьками можуть стати і слабкі особи. Популяція продовжує розвиватися до тих пір, поки не буде виконано критерії зупинки або не буде досягнуто кількість поколінь. Згодом населення стає більш здоровим та зосередженим. Для уточнення популяції використовуються дві методики: розмноження і мутація. Розмноження вимагає, щоб окремі люди

обмінювалися частинами свого коду, щоб отримати потомство. Мутація викликає невеликі зміни в деяких компонентах коду індивіда.

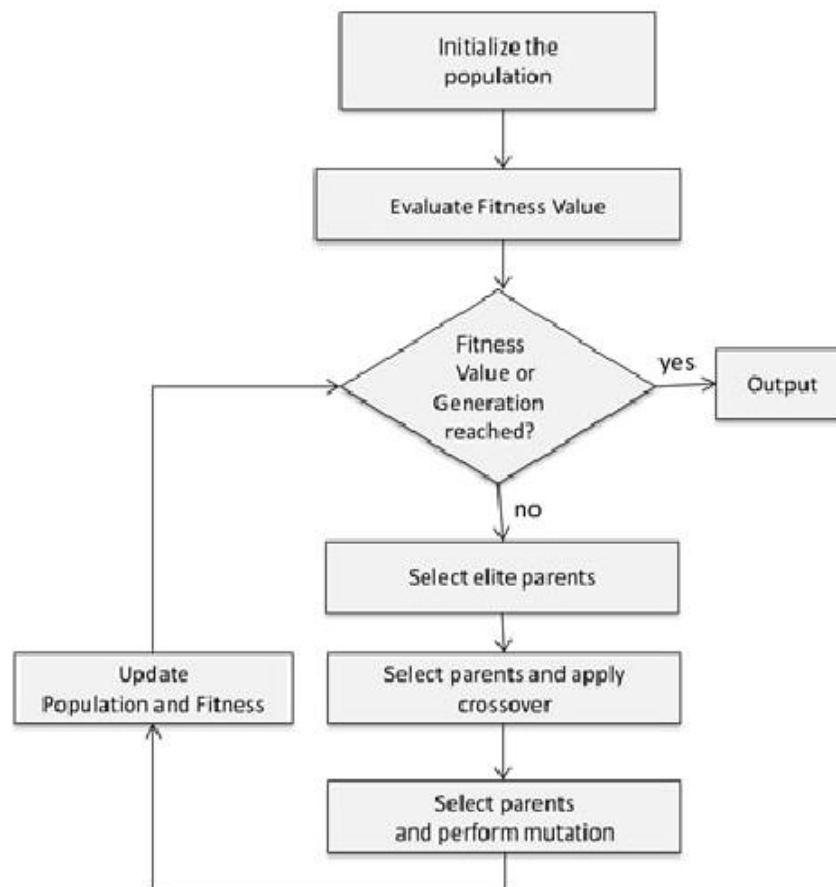


Рисунок 1.5 – Узагальнена блок-схема для генетичного алгоритму

1.3.5 Інтелектуальні агенти для підтримки інтелектуальних рішень

Із різних методів штучного інтелекту, інтелектуальні агенти (ІА) мають найширше застосування для вирішення проблем. За визначенням, агент – це об’єкт у системі, який «розташований у середовищі та може діяти автономно у цьому середовищі для досягнення своєї проектної мети». Ця внутрішня здатність автономної дії є людською характеристикою дій (або прийняття рішень) на основі контексту, а не наказової логіки, вбудованої в комп’ютерних програмах «якщо-то» (if-then).

У літературі розрізняють агента і розумного агента за допомогою концепцій слабких і сильних агентств. Слабка свобода дій включає такі

навички, як автономність, реактивність, адаптивність, проактивність і соціальні навички. Сильний агент додає до цих навичок більш просунуті риси, такі як комунікація, наполегливість, мобільність, раціональність та навчання. Автономія – це здатність приймати рішення незалежно від інших. Агенти, які демонструють реактивність і адаптивність, сприймають навколишнє середовище та реагують на його зміни. Проактивність означає, що за наявності конкретної інструкції ІА може вжити заходів для досягнення мети. Соціальні здібності дозволяють агентам взаємодіяти з іншими. Комунікація з іншими агентами представлена соціальною здатністю агента і призводить до таких властивостей, як співпраця та переговори. Постійність означає, що агент підтримує стан протягом тривалого періоду часу, а мобільність дозволяє агенту переміщуватися по всій системі для виконання завдань або пошуку нових знань. Раціональність – це здатність приймати рішення на основі навколишнього середовища, а навчання означає, що ІА може змінювати свої реакції на основі минулих взаємодій чи ситуацій. Для опису ІА використовуються такі людські терміни, як знання, намір та переконання, що вказує на можливу складну поведінку.

Потужним методом підтримки вирішення завдань у складних ситуаціях є створення багатоагентних систем (MAS), що складаються з груп агентів, які не мають повної інформації про середовище чи інших агентів. Кожен ІА може мати певні характеристики, а окремі агенти можуть об'єднуватися з іншими агентами, щоб діяти від імені особи, яка приймає рішення, або інших агентів з різними цілями та завданнями. Агентам може бути потрібна взаємодія, координація, переговори, навчання і навіть довіра іншим агентам для досягнення своїх цілей. Особи, які приймають рішення, можуть отримувати рекомендації від MAS або дозволяти агентам діяти від людини без спеціального дозволу. MAS є активною сферою досліджень у складних додатках і охоплюють питання, пов'язані з представленням проблеми прийняття рішення, розміром команди та реконфігурацією команд під час вирішення проблеми прийняття рішень.

Динамічні невизначені середовища особливо добре підходять для MAS. Команди агентів можуть сприймати навколишнє середовище, адаптуватися до змін, обмінюватися знаннями між членами команди та складати узгоджений план дій. Агенти можуть навчатися за допомогою контрольованого, неконтрольованого або навчання за винагороду. Подібно до НМ, під час контрольованого навчання система отримує правильні вихідні дані для управління системою. У неконтрольованому навчанні не надаються вихідні дані, тому системі потрібно визначити структуру вхідних даних. Під час навчання, заснованому на винагороді, MAS отримує позитивне чи негативне підкріплення, яке спрямовує систему.

Основними поняттями агентної технології є:

- Комунікація необхідна для передачі намірів ІА. Зв'язок може бути прямим або непрямим. Пряме спілкування може відбуватися за допомогою передачі повідомлень або спільних змінних. Непряма комунікація може відбуватися шляхом висновку, наприклад, шляхом спостереження агента за діями інших агентів.

- Необхідна координація для організації ІА, їх ресурсів та завдань. Координація використовується для вирішення конфліктів і підвищення продуктивності агента.

- Організація спільної роботи необхідна для кодифікації поведінки, такої як комунікація та координація ІА, щоб взаємодіяти та діяти як команда для досягнення спільної мети.

- Агенти, орієнтовані на людину, необхідні для повноцінної взаємодії між комп'ютерною системою та людиною, яка приймає рішення. Наприклад, підзавдання, які вимагають людського судження, можуть бути представлені людині, яка приймає рішення.

- Навчання необхідно для того, щоб зробити ІА більш схожим на людину та полегшити взаємодію між MAS та людьми, які приймають рішення.

1.4 Постановка задачі дослідження

Метою магістерської роботи є розробка інтелектуальної системи підтримки прийняття рішень для виявлення працездатних проху-адресів із застосуванням нейронних мереж.

Для досягнення мети дослідження необхідно вирішити наступні завдання:

- розглянути типи НМ;
- розглянути роботу проху-серверу;
- проаналізувати архітектури нейронних мереж;
- визначити умови працездатності проху-адрес;
- розробити нейронну мережу для виявлення працездатності проху-адрес;
- протестувати сформовану нейронну мережу за допомогою тестових даних;
- створити додаток на основі тестової НМ.

Розв'язання цих завдань дозволить створити ІСППР на основі нейронних мереж, яку можна використовувати для автоматичного пошуку та виявлення працездатних проху-адресів.

2 PROXY-СЕРВЕРИ

2.1 Загальні відомості

Прoxy (у перекладі з англійської як довірена особа) – це проміжний транзитний веб-сервер, який використовується як посередник між користувачем та кінцевим сервером (рис. 2.1).



Рисунок 2.1 – Роль проху у схемі передачі даних між користувачем та кінцевим ресурсом

Прoxy-сервери забезпечують різні рівні функціональності, безпеки та конфіденційності залежно від вашого випадку використання, потреб або політики компанії. Їх можна налаштувати як веб-фільтри або брандмауери, захищаючи ваш комп'ютер від інтернет-загроз, таких як зловмисне програмне забезпечення [8].

Послуги проху допомагають нам:

- обійти онлайн-блокування. Прoxy-сервери дозволяють отримати доступ до веб-сайту, до якого інакше не можна було б отримати доступ через блокування;

- отримати доступ до вмісту з географічними обмеженнями;
- отримати доступ до веб-сайтів, які обмежують кількість підключень.

Деякі файлообмінні служби навіть обмежують кількість завантажень з однієї IP-адреси;

- анонімно переглядати різні веб-сайти, не турбуючись про розголошення вашої IP-адреси;

- захиститися від шкідливого програмного забезпечення шляхом фільтрації вмісту.

Щоб підключитися до загальнодоступного проху-сервера, вам потрібно знати його IP-адресу. З його допомогою (і знаючи, який порт використовувати), ви можете налаштувати свій браузер або програму для маршрутизації своїх запитів через цей сервер.

Принцип роботи цього процесу залежить від браузера та від операційної системи. У Windows, наприклад, ви отримуєте спеціальну сторінку налаштувань проху-сервера, де ви можете ввести адресу, до якої потрібно підключитися (рис. 2.2).

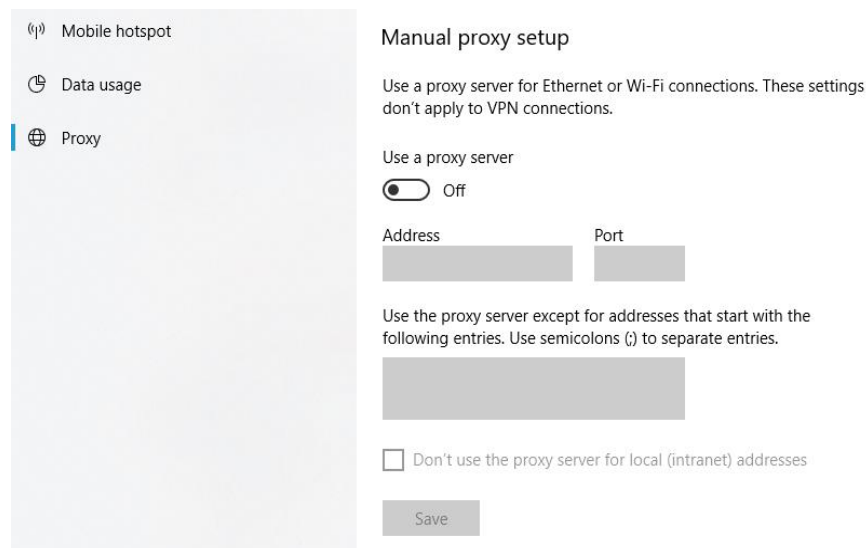


Рисунок 2.2 – Вікно сторінки налаштувань проху-сервера у Windows

Перш ніж почати пошук проху-серверів, ви повинні зрозуміти, що існує кілька доступних типів. Кожен виконує різні функції, тому важливо вибрати

правильний для ваших потреб.

Є апаратні та програмні версії. Апаратні з'єднання знаходяться між вашою мережею та Інтернетом, де вони отримують, надсилають та пересилають дані з Інтернету. Програмні проху-сервери зазвичай розміщуються у постачальника або знаходяться у хмарі. Ви завантажуєте та встановлюєте програму на свій комп'ютер, яка полегшує взаємодію з проху.

Часто програмний проху-сервер можна отримати за щомісячну плату. Іноді вони безкоштовні. Безкоштовні версії, як правило, пропонують користувачам менше адрес і можуть охоплювати лише кілька пристроїв, тоді як платні проху можуть задовольнити потреби бізнесу з багатьма пристроями.

2.2 Види та типи проху-серверів

Не всі проху-сервери працюють однаково. Важливо точно зрозуміти, які функції ви отримуєте від проху-сервера, і переконатися, що даний сервер відповідає вашим цілям використання. Проху можна розділити на кілька типів за різними критеріями:

а) За протоколом, що використовується:

1) FTP проху використовуються для завантаження даних на сервері FTP;

2) CGI проху (анонімайзери) допомагають відкрити будь-який веб-сайт у браузері. Жодних додаткових налаштувань не потрібно. Найчастіше такі проху виконані у вигляді веб-сайту, де можна ввести адресу сайту, яку необхідно відвідати;

3) SMTP, POP3 та IMAP проху використовуються для надсилання та отримання електронної пошти;

4) HTTP та HTTPS проху призначені для перегляду веб-сторінок;

5) SOCKS проху передає всі дані на кінцевий сервер як клієнт, тому вважається найанонімнішим протоколом. Протокол має три версії: SOCKS4, SOCKS4a і SOCKS5. Проху-сервери SOCKS5 мають підвищену безпеку та

пропонують три методи автентифікації: нульова автентифікація, автентифікація за допомогою пароля та автентифікація GSS-API.

HTTP, HTTPS та Socks проху використовуються найчастіше.

b) За рівнем анонімності:

1) Прозорий проху-сервер – не змінює запитувані або отримані дані, тому анонімність не передбачається, і веб-сайти можуть прочитати вашу IP-адресу. Прозорі проху зазвичай знаходяться в публічних точках доступу Wi-Fi, щоб надати доступ лише тим, хто має дозвіл. Багато серверів також використовують прозорі проксі для кешування вмісту. Їх легко налаштувати як на клієнтській, так і на серверній стороні. З іншого боку, прозорі проху-сервери більш сприйнятливі до певних загроз безпеки, таких як SYN-flood атаки відмови в обслуговуванні [9].

2) Анонімний проху – зосереджений на тому, щоб зробити Інтернет-активність невідстежуваною. Він працює шляхом заміни вашої IP-адреси своєю, приховуючи особу користувача та інформацію про комп'ютер, що допомагає запобігти крадіжці особистих даних і зберегти конфіденційність. Вони також можуть перешкодити веб-сайту надавати вам цільовий маркетинговий контент на основі вашого місцезнаходження. Веб-сайти можуть розпізнати, що проху-сервер обходить з'єднання, але не можуть визначити вашу початкову IP-адресу. Ці типи проху забезпечують мінімальну анонімність і можуть мати будь-який протокол [10].

3) Проху-сервер високої анонімності – це анонімний проху-сервер, який стирає вашу інформацію до того, як проху-сервер спробує підключитися до цільового сайту. Проху-сервери з високою анонімністю періодично змінюють IP-адресу, яку вони надають веб-серверу, що дуже ускладнює відстеження того, який трафік кому належить. Проху-сервери з високою анонімністю, такі як мережа TOR, є найбільш приватним і безпечним способом читання Інтернету. Сервер найкраще підходить для користувачів, для яких анонімність є абсолютною необхідністю, наприклад співробітників, які не хочуть, щоб їхня діяльність була відстежена до організації. З іншого

боку, деякі з них, особливо безкоштовні, є приманками, створеними для захоплення користувачів, щоб отримати доступ до їх особистої інформації або даних [11].

с) За походження IP:

1) Proxu центру обробки даних – генеруються на серверах, призначених для обробки проху. Сервери спеціально налаштовані на підключення з високою пропускнуою здатністю та спеціалізоване програмне забезпечення для максимальної продуктивності. Proxu-сервер існує у фізичному центрі обробки даних, і запити користувача маршрутизуються через цей сервер. Proxu центрів обробки даних підходять для більшості бізнес-цілей, особливо для ботів.

2) Житловий проху-сервер – надає вам IP-адресу, яка належить певному фізичному пристрою. Потім усі запити передаються через цей пристрій. Вони також пов'язані з фізичними місцями проживання та використовують Інтернет-провайдерів для отримання IP-адрес. Житлові проху-сервери добре підходять для користувачів, яким потрібно підтвердити рекламу, яка розміщується на їхньому веб-сайті, щоб ви могли блокувати файли cookie, підозрілі чи небажані оголошення.

d) За способом використання:

1) Публічний проху – доступний кожному безкоштовно і не потребує жодного методу автентифікації під час налаштування. Загальнодоступні проху найкраще підходять для користувачів, для яких вартість є основною проблемою, а безпека та швидкість – ні. Хоча вони безкоштовні та легкодоступні, вони часто працюють повільно. Використовуючи публічний проху-сервер, ви також ризикуєте отримати доступ до вашої інформації іншими в Інтернеті.

2) Приватний проху – це проху, які захищені різними методами автентифікації, щоб запобігти будь-якому несанкціонованому використанню людьми без доступу. Закриті проху можна придбати у багатьох постачальників в Інтернеті. Закриті проху-сервери можуть бути двох типів (спільні та

приватні) залежно від кількості людей, між якими використовуються проксі-сервери.

е) На основі тривалості:

1) Необоротний або статичний IP-проху – призначаються лише конкретному користувачеві, і користувач може використовувати проху-сервери по черзі за допомогою ботів або плагінів.

2) Оборотний або змінний IP-проху – проху автоматично змінюються через певний час і замінюються новими. Це дозволяє робити кожен запит з новою IP-адресою і мати набагато кращу анонімність, швидкість і уникати обмежень у роботі ботів.

ф) На основі модифікації даних:

1) Прямий проху-сервер – приймає запит даних від клієнта, а потім пересилає запит на веб-сервер для обробки та знову пересилають відповідь від веб-сервера клієнту. Найкраще підходить для внутрішніх мереж, які потребують єдиної точки входу. Він забезпечує безпеку IP-адрес для тих, хто в мережі, і дозволяє здійснювати простий адміністративний контроль.

2) Зворотний проху – знаходиться на стороні веб-сервера для контролю трафіку до сервера, автентифікації або розшифровки запитів на з'єднання чи просто для кешування вмісту. Він працює шляхом перехоплення запитів від користувача на межі мережі веб-сервера. Потім він надсилає запити на вихідний сервер і отримує відповіді від нього. Зворотні проху також можуть розподіляти трафік на різні сервери, щоб збалансувати навантаження вхідних запитів [12].

2.3 Переваги та недоліки проху-сервера

Проху мають ряд переваг:

- Посилений захист: може діяти як брандмауер між вашими системами та Інтернетом. Без них зловмисники мають легкий доступ до вашої IP-адреси, яку вони можуть використовувати для проникнення на ваш комп'ютер або

мережу.

- Приватний перегляд, прослуховування та покупки.
- Доступ до вмісту, пов'язаного з обмеженням місцеположення.
- Можна використовувати для обмеження доступу до певних сайтів.

(Часто використовується в компаніях для блокування доступу до веб-сайтів, які зазвичай відволікають співробітників від важливих завдань. Деякі організації блокують сайти соціальних мереж, як-от Facebook та інші, щоб усунути спокуси, які витрачають час).

Недоліками є:

- Відстеження – дані кешу, які використовують проху, можуть запам'ятовувати всю особисту інформацію, включаючи паролі і є ймовірність, що співробітники, які працюють під проху, зловживають цією інформацією. Тому рекомендується завжди використовувати проху від законного постачальника послуг.

- Безпека – хоча проху забезпечують переваги анонімності, їм не вистачає шифрування. Більшість проху використовують сертифікати SSL для шифрування даних. Цього недостатньо, щоб запобігти сьогоденним атакам (Особливо від атак, відомих як видалення SSL).

- Несумісність – проху-сервер не завжди може бути сумісним з вашою локальною мережею.

- Вартість – налаштування та обслуговування проху -сервера можуть бути дорогими. Незважаючи на те, що великі організації можуть легко покрити ці витрати, це буде важко для малого бізнесу.

- Конфігурації проху попередньо запрограмовані для однієї конкретної мети. Тому має бути певне кодування, яке має бути виконано, щоб виконати свою вимогу. Але конфігурація проху може бути досить складною. Це повинно бути ідеально зроблено таким чином, щоб жодні порти не залишалися відкритими, щоб жодні хакери не могли шпигувати за вашою особистою інформацією [13].

3 СТВОРЕННЯ ІСППР ДЛЯ ВИЗНАЧЕННЯ ПРАЦЕЗДАТНОСТІ PROXY-АДРЕСІВ

3.1 Типи моделей нейронних мереж

В цілому НМ розділяються на два типи, ті котрі мають зворотній зв'язок або ні. Існують мережі прямого поширення сигналів (Feed-Forward Neural Network) FF без зворотних зв'язків та рекурентні (Recurrent Neural Network) RNN мережі або мережі зі зворотними зв'язками [14].

До основних типів нейронних мереж відносяться:

- багат шаровий або одношаровий перцептрон;
- згорткові нейронні мережі;
- рекурентні нейронні мережі.

Одношаровий перцептрон – це перша запропонована нейронна модель, де вміст локальної пам'яті нейрона складається з вектора ваг. Обчислення єдиного шару перцептрону виконується над розрахунком суми вхідного вектора, кожен з яких має значення, помножене на відповідний елемент вектора ваг. Значення, яке відображається на виході, буде введенням функції активації (рис. 3.1).

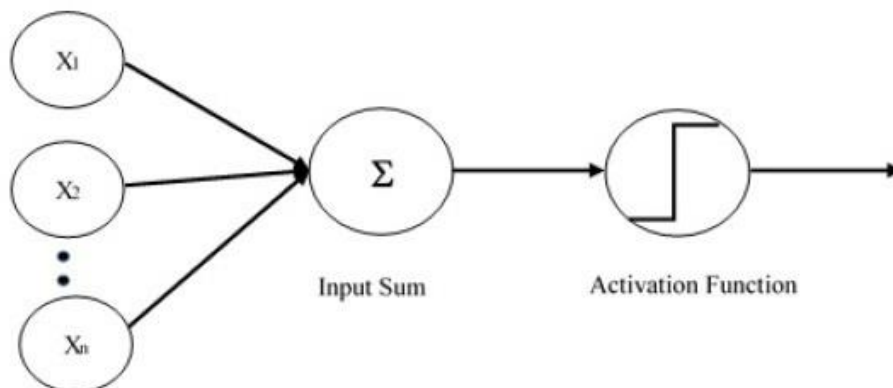


Рисунок 3.1 – Модель одношарового перцептрона

Багатошаровий перцептрон – це нейронна мережа, де картографування між входами та виходом є нелінійним.

Багатошаровий перцептрон має вхідні та вихідні шари і один або декілька прихованих шарів з багатьма нейронами, укладеними разом. Також в перцептроні міститься, нейрон, що повинен мати функцію активації, яка накладає поріг проходження (рис.3.2).

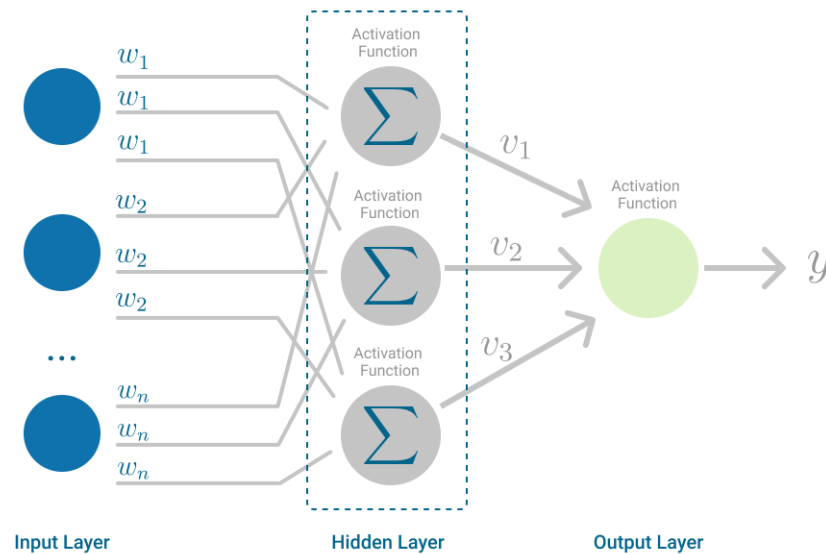


Рисунок 3.2 – Модель багатошарового перцептрону

Багатошаровий перцептрон підпадає під категорію алгоритмів живлення (FF), оскільки входи поєднуються з початковими вагами в зваженій сумі та піддаються функції активації. Але різниця полягає в тому, що кожна лінійна комбінація поширюється на наступний шар.

Кожен шар годує наступного з результатом їх обчислення, їх внутрішнього представлення даних. Це йде весь шлях через приховані шари до вихідного шару.

Якщо алгоритм обчислює лише зважені суми в кожному нейроні, розмножені результати до вихідного шару, і там зупинився, він не зможе вивчити ваги, які мінімізують функцію витрат. Якщо алгоритм обчислюється лише однією ітерацією, то не буде фактичного навчання.

Нейрони у багатошаровому перцептроні можуть використовувати будь-

яку довільну функцію активації, так само як і в одношаровому.

Функції активації:

- сигмоїд;
- порогова функція активації;
- лінійна функція активації;
- гіперболічний тангенс;
- ReLu;
- Leaky ReLu.

Сигмоїд – нелінійна за своєю природою, а комбінація таких функцій виробляє теж нелінійну функцію (рис. 3.3). Вона не бінарна, що робить активацію аналоговою. Для сигмоїду характерний гладким градієнт. Функція описується виразом:

$$f(S) = \frac{1}{1+e^{-x}}, \quad (3.1)$$

де x – параметр нахилу функції. Змінюючи цей параметр можна побудувати функції різної крутизни.

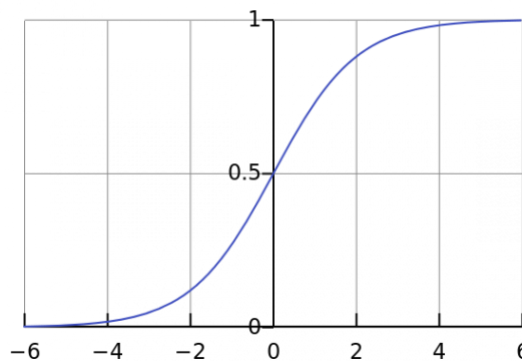


Рисунок 3.3 – Графік сигмоїдної функції активації

При наближенні до кінців сигмоїдальних значень, значення u , як правило, погано реагує на зміни v . Це означає, що градієнт u в таких областях займає невеликі значення. І це, у свою чергу, призводить до проблем з

градієнтом зникнення. У цьому випадку значення градієнта мало або зникає. Нейронна мережа відмовляється вчитись далі або робить його надзвичайно повільним.

Порогова функція активації (рис. 3.4) – описується як:

$$f(S) = \begin{cases} 1, & \text{якщо } S \geq 0 \\ 0, & \text{якщо } S < 0 \end{cases} \quad (3.2)$$

Якщо значення S більше деякого порогового значення, то вважається що нейрон активований. В іншому випадку нейрон неактивний.

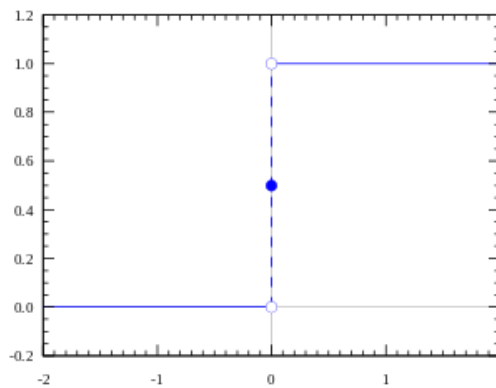


Рисунок 3.4 – Графік порогової функції активації

Лінійна функція активації (рис. 3.5) – це пряма лінія, пропорційна входу (тобто зваженої сумі на вході нейрона). Такий вибір активаційної функції дозволяє отримувати спектр значень, а не тільки бінарну відповідь. Можна з'єднати декілька нейронів разом і, якщо більше одного нейрона активовано, рішення приймається на основі застосування операції max (або softmax).

Дана функція активації описується як:

$$f(S) = \begin{cases} a, & \text{якщо } S > a \\ S, & \text{якщо } -a \leq S \leq a \\ -a, & \text{якщо } S < -a \end{cases} \quad (3.3)$$

Вихідний сигнал нейрону має такі значення:

- a , коли лінійна комбінація S більша за порогове значення a ;
- a , коли лінійна комбінація S менша за порогове значення $-a$;
- S в діапазоні значень $[-a; a]$.

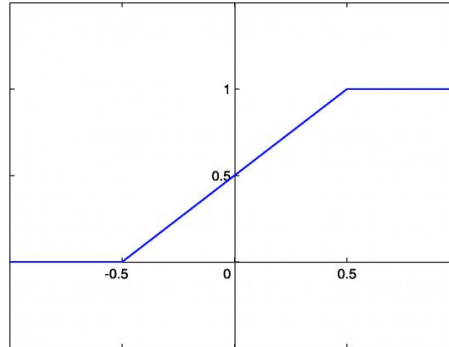


Рисунок 3.5 – Графік лінійної функції активації

Функція активації гіперболічного тангенсу (рис. 3.6). Описується наступним виразом:

$$f(S) = \tanh\left(\frac{S}{\alpha}\right) = \frac{e^S - e^{-S}}{e^S + e^{-S}} \quad (3.4)$$

Гіперболічний тангенс дуже схожий на сигмоїд. І дійсно, це скоригована сигмоїдна функція:

$$\tanh(x) = 2 * \text{sigmoid}(2x) - 1 \quad (3.5)$$

Тому така функція має ті ж характеристики, що й сигмоїд. Її природа нелінійна, вона добре підходить для комбінації шарів, а діапазон значень функції $(-1; 1)$. Тому немає сенсу турбуватися, що активаційна функція перевантажиться від великих значень. Однак варто зазначити, що градієнт функції гіперболічного тангенсу більше, ніж у функції сигмоїда (похідна крутіше).

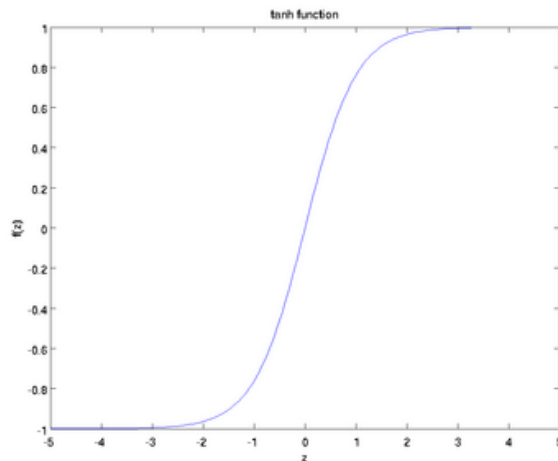


Рисунок 3.6 – Графік функції активації гіперболічного тангенсу

Функція активації ReLu (рис. 3.7). Описується наступним виразом:

$$f(x) = \begin{cases} 0, & \text{якщо } x < 0 \\ x, & \text{якщо } x \geq 0 \end{cases} \quad (3.6)$$

Стає зрозуміло, що ReLu повертає значення x , якщо x позитивно, і 0 в іншому випадку. ReLu нелінійна за своєю природою, а комбінація ReLu також нелінійна. Така функція є хорошим апроксиматором, так як будь-яка функція може бути апроксимована комбінацією ReLu. Область допустимих значень ReLu – $[0; \text{inf})$.

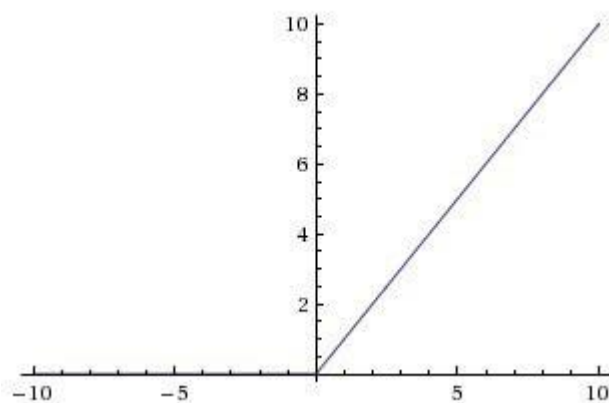


Рисунок 3.7 – Графік функції активації ReLu

Згортова нейронна мережа (Convolutional Neural Network – CNN) – це

глибокий алгоритм навчання, який може приймати вхідний образ, призначити важливість (вивчені ваги та упередження) до різних аспектів або об'єктів на зображенні та вміти диференціювати один з іншого. Попередня обробка в згортковій мережі набагато нижча в порівнянні з іншими алгоритмами класифікації.

Архітектура згорткової мережі аналогічна впливу нейронів у людському мозку. Індивідуальні нейрони реагують на подразники лише у обмеженій ділянці візуального поля, відомого як рецептивне поле. Колекція таких полів перекривається, щоб покрити всю візуальну область (рис. 3.8).

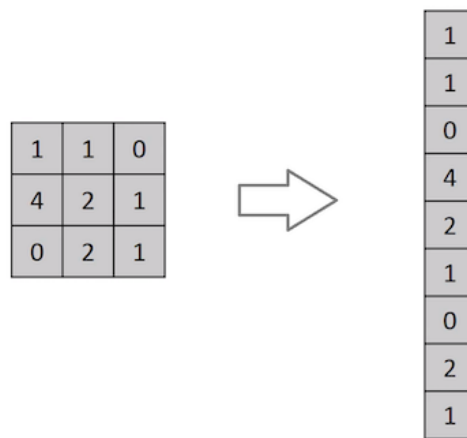


Рисунок 3.8 – Приклад конвертації даних у згортковій мережі

Згорткова мережа може успішно захопити просторові та тимчасові залежності у зображенні за допомогою застосування відповідних фільтрів. Архітектура виконує кращу придатну до набору зображень за рахунок зменшення кількості параметрів, пов'язаних та багаторазових.

Рекурентні нейронні мережі (Recurrent Neural Networks RNN) – нейронні мережі, призначені для роботи з текстом (рис. 3.9). Рецидивна мережа може допомогти дізнатися послідовну структуру тексту, де кожне слово залежить від попереднього слова, або слова в попередньому реченні.

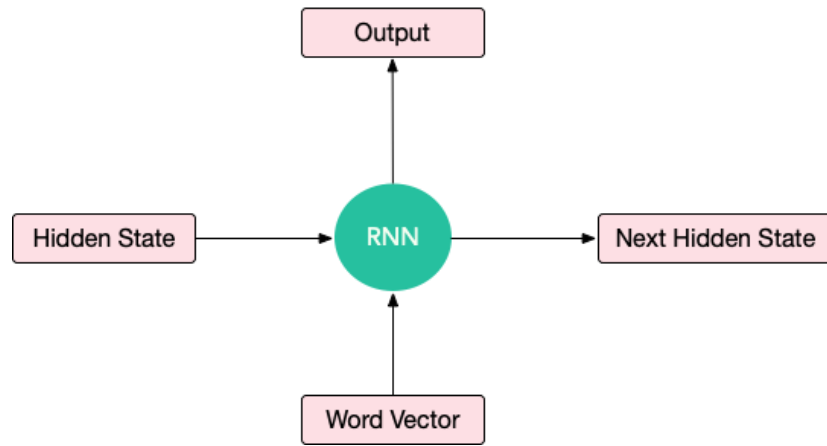


Рисунок 3.9 – Приклад рекурентної нейронної мережі

Кожну клітинку мережі можна представити як чорну коробку, яка бере значення прихованого стану (вектор), вектор слова, вихідний вектор та наступний прихований стан. У цій клітинці є деякі ваги, які необхідно налаштувати за допомогою зворотного пропусення втрат. Також однакова клітина застосовується до всіх слів, так що ваги поділяються на слова у реченні. Це явище називається обмін ваги.

На рисунку 3.10 наведено розширену версію клітинки рекурентної нейронної мережі. Кожна клітина мережі працює на кожному слові маркера і проходить прихований стан до наступної клітини.

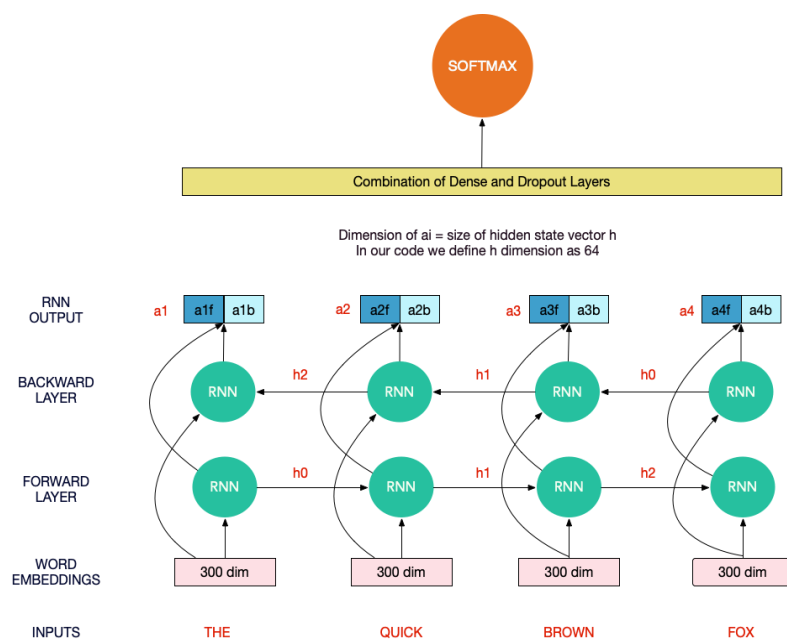


Рисунок 3.10 – Детальний приклад роботи рекурентної нейронної мережі

3.2. Методи визначення працездатності проху-адресу

Під працездатністю проху мається на увазі можливість відправляти запити на різні ресурси або сервіси та отримувати відповіді (request/response). Якщо запит був відправлений, а відповідь так і не прийшла, можна зробити висновок, що проху, котрий використовувався в запиті не працездатний. У випадку отримання відповіді можна зробити висновок, що проху працездатний. Чи все так однозначно? З логічної точки зору, якщо ми отримали результат, то проху працездатний, та ні, якщо результат був не отриманий (1 та 0). Але не у нашому випадку. На відправлення запиту, з використанням проху-адресу, та отримання відповіді від ресурсу можуть впливати наступні фактори:

1) connection timeout – проміжок часу, протягом якого слід встановити зв'язок між відправником запиту та ресурсом відповідачем;

2) request timeout – час очікування відповіді від ресурсу. Після встановлення з'єднання між відправником та ресурсом, відправник повинен періодично інформувати ресурс, про те що він ще там, посылаючи інформацію на цей інформаційний ресурс. Якщо відправник не може надсилати будь-яку інформацію на ресурс у вказаний час, ресурс просто знімає це підключення, оскільки він думає, що відправника більше немає, щоб не витратити марно ресурси;

3) завантаженість проху-адресу – кількість користувачів, що одночасно використовують одну й ту ж проху-адресу. Від цього фактору залежать двое попередніх;

4) завантаженість ресурсу – максимальна кількість, одночасно оброблюваних запитів;

5) швидкість проху-адресу – залежить від швидкості інтернету;

6) нестабільне з'єднання з інтернет мережею. Цей фактор може бути властивий всім трьом сторонам.

Беручи до уваги той факт, що ми будемо використовувати проху-адреси,

які знаходяться у вільному доступі, можна доповнити список факторів наступними:

- 1) життєвий період – зазвичай проху-адреси з вільного доступу мають життєвий період в 2-3 дні. Після третього дня, ймовірність працездатності проху-адреси майже нульова;
- 2) можлива наявність авторизації на проху-сервері;
- 3) підробка проху-адресу – деякі сервіси дають безкоштовно користуватись їх проху-адресами, але заздалегідь разом з працездатними адресами йдуть і підробки;
- 4) старі проху-адреси – ресурс з, якого беруться адреси, може рідко оновлювати свій список.

Відштовхуючись від цих факторів, можна впевнено сказати, що неможливо однозначно зробити висновок по працездатності проху-адресу, якщо брати до уваги всього лише факт наявності відповіді від ресурси та її відсутності.

3.3 Методи отримання відповіді від ресурсу

Як вже було визначено, проху-адрес може мати два стана або він працездатний або ж ні. Для підтвердження працездатності досить однієї, та певно найголовнішою умови – наявність відповіді від ресурсу. Тобто, якщо було отримано відповідь з ресурсу можна впевнено сказати, що проху-адреса є працездатною. Для підтвердження не працездатності проху-адресу не достатньо однієї умови, а саме відсутності відповіді від ресурсу. На працездатність проху-адресу можуть впливати різні, як зовнішні так і внутрішні фактори.

Один з варіантів це відправка декілька послідовних запитів (рис. 3.11), але навіть це не гарантує сто відсоткове отримання результату.



Рисунок 3.11 Послідовне відправлення запитів до ресурсу

Уявімо що ми відправляємо 3-5 тестових запитів з максимальним часом очікування відповіді 30 секунд, таким чином можна вирахувати мінімальний та максимальний час очікування відповіді. Мінімальним часом відповіді можна вважати проміжок часу від 0 до 5 секунд, максимальним часом можна вважати добуток максимального часу очікування відповіді та кількості тестових запитів:

$$t_{max} = t_{timeout} * R$$

Окрім послідовних запитів існують асинхронні, або як їх ще називають паралельні запити (рис. 3.12). Даний підхід може поліпшити ситуацію з отриманням відповіді, але так само як і у випадку з послідовними запитами не гарантує сто відсоткове її отримання.



Рисунок 3.12 – Паралельне відправлення запитів до ресурсу.

У асинхронного метода відправки запитів, з використання проху-адресу, є свої мінуси:

- 1) одночасне використання одного й того ж проху-адресу декількома клієнтами;
- 2) обмеженість в кількості запитів;
- 3) можливість блокування проху-адресу та занесення його у чорні списки.

У будь якому випадку асинхронний метод відправки запитів ліпше за послідовний, завдяки його швидкодії при тих самих, або й більших, кількостях запитів до ресурсу.

3.4 Пошук умов для визначення працездатності проху-адрес

На основі факторів, котрі можуть впливати на відсутність відповіді, можна виділити наступні умови для підтвердження або спростування працездатності проху-адресу:

- 1) наявність відповіді від ресурсу;
- 2) чи був проху-адрес доданий раніше, чи поточний адрес був доданий вперше;
- 3) наявність відповіді в попередній раз;
- 4) перевірка дати додання в базу даних. Якщо дата додання більше або менше трьох днів;
- 5) частота оновлювання проху-адресу. Як часто проху-адрес потрапляє в базу даних;
- 6) відсоткове відношення між кількістю успішно отриманих відповідей від ресурсу та кількістю невдалих, під час перевірки проху-адресу;
- 7) відсоткове відношення між кількістю успішно отриманих відповідей від ресурсу та кількістю невдалих, під час відправки запитів з клієнта;
- 8) перевірка на кількість невдало отриманих відповідей поспіль.

3.5 Побудова нейромережевої моделі

У підрозділі 3.4 були визначені умови для працездатності гроху-адресу. Дані умови будемо вважати за вхідні дані перцептрона, ці дані представлені як 0 та 1. Модель перцептрона представлена на рис. 3.13.

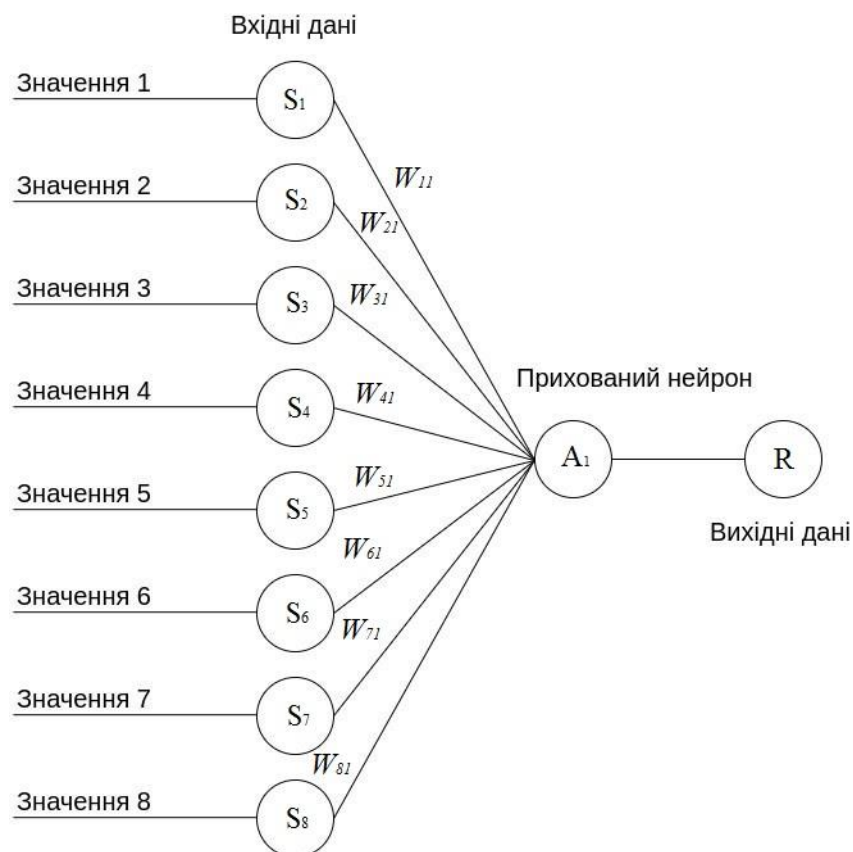


Рисунок 3.13 – Нейромережева модель для ІСППР

Значення, котрі поступають на вхід перцептрона – це конвертовані умови працездатності гроху-адресу. Кожне значення дорівнює 0 або 1.

Вхідні дані S_1 – Наявність відповіді від ресурсу. Якщо під час тестування гроху-адресу, відповідь з ресурсу була отримана значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_2 – Чи є дана гроху-адреса новою, чи ні. Якщо гроху-адреса нова, то значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_3 – Наявність відповіді під час перевірки в попередній раз.

Якщо відповідь була отримана в попередній раз значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_4 – Чи пройшло з початку додання проху-адресу, до бази даних 3 дні, чи ні. Якщо проміжок часу між датою додання та поточною більше або дорівнює три доби значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_5 – Наявність проху-адресу на різних джерелах. Якщо кількість джерел у сумі складає 35 і більше відсотків значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_6 – Кількість разів, коли було отримано відповідь від ресурсу, під час тестової перевірки проху-адресу. Якщо кількість разів у сумі складає 30 і більше відсотків значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_7 – Кількість разів, коли було отримано відповідь від ресурсу, під час відправки запитів клієнтом. Якщо кількість разів у сумі складає 7 і більше відсотків значенню буде присвоєно 1, якщо ні – 0.

Вхідні дані S_8 – Наявність п'яти поспіль не отриманих відповідей від ресурсу. Якщо умова буде виконана значенню буде присвоєно 1, якщо ні – 0.

Щоб перевірити правильність кінцевого результату нейронної мережі слід задати незмінний набір тестових вхідних значень.

В таблиці 3.1 зображені тестові набори значень. Ці набори дуже схожі на ті, що будуть в реальній системі.

Таблиця 3.1 – Набір тестових значень з очікуваним результатом

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_1	0	0	0	0	1	1	0	0	0
N_2	0	0	0	0	0	1	1	0	1
N_3	0	0	0	1	0	1	1	0	0
N_4	0	0	0	1	1	1	1	0	1
N_5	0	0	1	1	0	0	0	0	1
N_6	0	1	0	1	0	0	0	0	0

Продовження таблиці 3.1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_7	1	0	0	0	0	0	0	0	1
N_8	0	0	0	0	0	1	1	1	0
N_9	0	0	1	0	1	1	1	1	0
N_{10}	1	0	0	1	0	0	0	0	1

Для навчання нейронної мережі слід задати набір значень, по яким ця мережа буде проводити навчання. Навчальні набори представлені в табл. 3.2.

Таблиця 3.2 – Вхідні дані, для навчання нейронної мережі

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_1	0	0	0	0	0	0	0	0	1
N_2	0	0	0	0	0	0	0	1	0
N_3	0	0	0	0	0	0	1	0	0
N_4	0	0	0	0	0	0	1	1	0
N_5	0	0	0	0	0	1	0	0	0
N_6	0	0	0	0	0	1	0	1	0
N_7	0	0	0	0	0	1	1	0	0
N_8	0	0	0	0	0	1	1	1	0
N_9	0	0	0	0	1	0	0	0	0
N_{10}	0	0	0	0	1	0	0	1	0
N_{11}	0	0	0	0	1	0	1	1	0
N_{12}	0	0	0	0	1	1	0	1	0
N_{13}	0	0	0	0	1	1	1	0	1
N_{14}	0	0	0	0	1	1	1	1	0
N_{15}	0	0	0	1	0	0	0	0	0
N_{16}	0	0	0	1	0	0	0	1	0
N_{17}	0	0	0	1	0	0	1	0	0

Продовження таблиці 3.2

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_{18}	0	0	0	1	0	0	1	1	0
N_{19}	0	0	0	1	0	1	0	0	0
N_{20}	0	0	0	1	0	1	0	1	0
N_{21}	0	0	0	1	0	1	1	0	1
N_{22}	0	0	0	1	0	1	1	1	0
N_{23}	0	0	0	1	1	0	0	0	0
N_{24}	0	0	0	1	1	0	0	1	0
N_{25}	0	0	0	1	1	0	1	0	0
N_{26}	0	0	0	1	1	0	1	1	0
N_{27}	0	0	0	1	1	1	0	0	0
N_{28}	0	0	0	1	1	1	0	1	0
N_{29}	0	0	0	1	1	1	1	0	1
N_{30}	0	0	0	1	1	1	1	1	0

3.6 Алгоритм навчання перцептрона

Алгоритм навчання перцептрона має наступний вигляд.

Кожним вагам, з кожного входу (W_{ij}) та порогу (W_0) присвоюються випадкові значення. Наступним кроком є пред'явлення навчального екземпляру з навчальної множини:

$$x^p = \{x_{1p}, \dots, x_{Lp}\}, \quad (3.7)$$

де $p = 1, 2, \dots, t$ – навчальна множина, t – кількість екземплярів у навчальній виборці.

Далі обчислюється реальне значення на виході перцептрона:

$$y = \psi(\varphi(w, x)), \quad (3.8)$$

де $w = \{w_0, w_1, w_2, \dots, w_L\}$ – набір ваг, що утворюють пам'ять нейрона. Після обчислення реальних значень корегуються ваги перцептрона:

$$w_i = w_i + \alpha(y^s - y^s) * x_{ip} \quad (3.9)$$

$$i = 0, 1, \dots, L, x_0 = 1 \quad (3.10)$$

де α – позитивний коригувальний приріст (крок навчання).

Якщо досягнуто збіжність, то процедура корекції ваг закінчується, інакше алгоритм повертається до пред'явлення навчального екземпляру з навчальної множини.

Відповідно до даного методу спочатку здійснюється ініціалізація параметрів перцептрона випадковими значеннями. Потім по черзі пред'являються образи з відомою класифікацією, вибрані з навчальної множини, і коригуються ваги відповідно до формул 4.3 та 4.4.

Величина корекції визначається позитивним коригувальним приростом, конкретне значення якого вибирається досить великим, щоб швидше проводилася корекція ваг, і в той же час досить малим, щоб не допустити надмірного зростання значень ваг. Процедура навчання триває до тих пір, поки не буде досягнута збіжність, тобто поки не будуть отримані ваги, що забезпечують правильну класифікацію для всіх образів з навчальної множини.

Одношаровий перцептрон використовує як дискримінантну функцію зважену суму та як активаційну – порогову або сигмоїдну, рідше – лінійну функцію. У залежності від типу функції активації розрізняють дискретний перцептрон, що використовує порогову функцію активації, і дійсний перцептрон, що використовує дійсні функції активації, наприклад, сигмоїдну функцію. У нашому випадку будемо використовувати дійсну функцію активації – сигмоїдну функцію [15, с.44-54].

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1 Опис середовища для проведення експериментальних досліджень

Існує величезна кількість інструментів для проведення математичних досліджень. Це можуть бути як сервіси або додатки так і мови програмування. Однією з таких мов є Matlab.

Matlab – моделювальне середовище, котре дозволяє в декілька кроків змоделювати майже будь-яку систему.

Matlab ідеально підходить для побудови тестової моделі нейронної мережі.

4.2 Тестування моделі нейронної мережі

Подаємо на вхід набір тестових даних ($N_1 - N_{10}$), які були визначені в розділі 4.4. Ваги для кожного входу були обрані випадковими: $W_{11} = 0.6294$, $W_{21} = 0.8116$, $W_{31} = -0.7460$, $W_{41} = 0.8268$, $W_{51} = 0.2647$, $W_{61} = -0.8049$, $W_{71} = -0.4430$, $W_{81} = 0.0938$. Після задання ваг починаємо симуляцію. Результати симуляції НМ представлені в таблиці 4.1.

$S_1 - S_8$ вхідні дані.

\sum_w – сума ваг кожного вхідного значення.

F – результат функції активації.

R – вихідне значення.

Таблиця 4.1 – Результат навчання нейронної мережі з випадковими вагами

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	\sum_w	F	R
N_1	0	0	0	0	1	1	0	0	-8.9293	1.3244e-04	0

Продовження таблиці 4.1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_2	0	0	0	0	0	1	1	0	4.6324	0.9904	1
N_3	0	0	0	1	0	1	1	0	0.0561	0.5140	1
N_4	0	0	0	1	1	1	1	0	-4.4086	0.0120	0
N_5	0	0	1	1	0	0	0	0	4.7822	0.9917	1
N_6	0	1	0	1	0	0	0	0	0.5608	0.6366	1
N_7	1	0	0	0	0	0	0	0	5.1370	0.9942	1
N_8	0	0	0	0	0	1	1	1	-0.5044	0.3765	0
N_9	0	0	1	0	1	1	1	1	4.3895	0.9877	1
N_{10}	1	0	0	1	0	0	0	0	0.5607	0.6366	1

Із результатів таблиці 4.1 можна зробити висновки, що результат не збігається з поставленим. Щоб результат став більш точним додаємо ще 30 комбінацій навчальних значень. Навчальні набори представлені в таблиці 4.2.

Таблиця 4.2 – Додатковий набір даних для навчання нейронної мережі

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_1	0	0	0	1	1	1	1	0	1
N_2	0	0	0	1	1	1	1	1	0
N_3	0	0	1	0	0	0	0	0	1
N_4	0	0	1	0	0	0	0	1	0
N_5	0	0	1	0	0	0	1	0	1
N_6	0	0	1	0	0	0	1	1	0
N_7	0	0	1	0	0	1	0	0	1
N_8	0	0	1	0	0	1	0	1	0
N_9	0	0	1	0	0	1	1	0	1
N_{10}	0	0	1	0	0	1	1	1	0
N_{11}	0	0	1	0	1	0	0	0	1

Продовження таблиці 4.2

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_{12}	0	0	1	0	1	0	0	1	0
N_{13}	0	0	1	0	1	0	1	0	1
N_{14}	0	0	1	0	1	0	1	1	0
N_{15}	0	0	1	0	1	1	0	0	1
N_{16}	0	0	1	0	1	1	0	1	0
N_{17}	0	0	1	0	1	1	1	0	1
N_{18}	0	0	1	0	1	1	1	1	1
N_{19}	0	0	1	1	0	0	0	0	1
N_{20}	0	0	1	1	0	0	0	1	0
N_{21}	0	0	1	1	0	0	1	0	1
N_{22}	0	0	1	1	0	0	1	1	0
N_{23}	0	0	1	1	0	1	0	0	1
N_{24}	0	0	1	1	0	1	0	1	0
N_{25}	0	0	1	1	0	1	1	0	1
N_{26}	0	0	1	1	0	1	1	1	0
N_{27}	0	0	1	1	1	0	0	0	1
N_{28}	0	0	1	1	1	0	0	1	0
N_{29}	0	0	1	1	1	0	1	0	1
N_{30}	0	0	1	1	1	0	1	1	0

За допомогою алгоритму зворотного поширення були отримані нові, більш коректні ваги: $W_{11} = 0.6294$, $W_{21} = 0.8116$, $W_{31} = -0.7460$, $W_{41} = -4.8160$, $W_{51} = -4.8074$, $W_{61} = -4.8077$, $W_{71} = -4.8076$, $W_{81} = -4.8187$. Корегуємо ваги та починаємо симуляцію. Результати НМ представлені в таблиці 4.3.

Таблиця 4.3 – Результат навчання нейронної мережі

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_1	0	0	0	0	1	1	0	0	-6.4568	0.0016	0
N_2	0	0	0	0	0	1	1	0	-6.4568	0.0016	0
N_3	0	0	0	1	0	1	1	0	-11.2547	1.2946e-05	0
N_4	0	0	0	1	1	1	1	0	-14.4831	5.1292e-07	0
N_5	0	0	1	1	0	0	0	0	8.3244	0.9998	1
N_6	0	1	0	1	0	0	0	0	10.5064	1.0000	1
N_7	1	0	0	0	0	0	0	0	15.1222	1.0000	1
N_8	0	0	0	0	0	1	1	1	-23.1957	8.4379e-11	0
N_9	0	0	1	0	1	1	1	1	-13.3018	1.6716e-06	0
N_{10}	1	0	0	1	0	0	0	0	10.3242	1.0000	1

Із результатів таблиці 4.3 можна зробити висновки, що НМ не вертає потрібні значені і потребує більшої кількості навчальних наборів. Щоб швидше навчити НМ, додаємо шістдесят навчальних наборів, тим самим пришвидшуємо навчання в 2 рази. Додаткові навчальні набори представлені в таблиці 4.4.

Таблиця 4.4 Додаткові навчальні набори

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_1	0	0	1	1	1	1	0	0	1
N_2	0	0	1	1	1	1	0	1	0
N_3	0	0	1	1	1	1	1	0	1
N_4	0	0	1	1	1	1	1	1	0
N_5	0	1	0	0	0	0	0	0	0
N_6	0	1	0	0	0	0	0	1	0
N_7	0	1	0	0	0	0	1	0	0
N_8	0	1	0	0	0	0	1	1	0

Продовження таблиці 4.4

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_9	0	1	0	0	0	1	0	0	0
N_{10}	0	1	0	0	0	1	0	1	0
N_{11}	0	1	0	0	0	1	1	0	1
N_{12}	0	1	0	0	0	1	1	1	0
N_{13}	0	1	0	0	1	0	0	0	0
N_{14}	0	1	0	0	1	0	0	1	0
N_{15}	0	1	0	0	1	0	1	0	1
N_{16}	0	1	0	0	1	0	1	1	0
N_{17}	0	1	0	0	1	0	1	1	0
N_{18}	0	1	0	0	1	1	0	0	1
N_{19}	0	1	0	0	1	1	0	1	0
N_{20}	0	1	0	0	1	1	1	0	1
N_{21}	0	1	0	0	1	1	1	1	0
N_{22}	0	1	0	1	0	0	0	0	0
N_{23}	0	1	0	1	0	0	0	1	0
N_{24}	0	1	0	1	0	0	1	0	0
N_{25}	0	1	0	1	0	0	1	1	0
N_{26}	0	1	0	1	0	1	0	0	0
N_{27}	0	1	0	1	0	1	0	1	0
N_{28}	0	1	0	1	0	1	1	0	1
N_{29}	0	1	0	1	0	1	1	1	0
N_{30}	0	1	0	1	1	0	0	0	0
N_{31}	0	1	0	1	1	0	0	1	0
N_{32}	0	1	0	1	1	0	1	0	1
N_{33}	0	1	0	1	1	0	1	1	0
N_{34}	0	1	0	1	1	1	0	0	1
N_{35}	0	1	0	1	1	1	0	1	0

Продовження таблиці 4.4

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_{36}	0	1	0	1	1	1	1	0	1
N_{37}	0	1	0	1	1	1	1	1	0
N_{38}	0	1	1	0	0	0	0	0	1
N_{39}	0	1	1	0	0	0	0	1	0
N_{40}	0	1	1	0	0	0	1	0	1
N_{41}	0	1	1	0	0	0	1	1	0
N_{42}	0	1	1	0	0	1	0	0	1
N_{43}	0	1	1	0	0	1	0	1	0
N_{44}	0	1	1	0	0	1	1	0	1
N_{45}	0	1	1	0	0	1	1	1	0
N_{46}	0	1	1	0	1	0	0	0	1
N_{47}	0	1	1	0	1	0	0	1	0
N_{48}	0	1	1	0	1	0	1	0	1
N_{49}	0	1	1	0	1	0	1	1	0
N_{50}	0	1	1	0	1	1	0	0	1
N_{51}	0	1	1	0	1	1	0	1	0
N_{52}	0	1	1	0	1	1	1	0	1
N_{53}	0	1	1	0	1	1	1	1	0
N_{54}	0	1	1	1	0	0	0	0	1
N_{55}	0	1	1	1	0	0	0	1	0
N_{56}	0	1	1	1	0	0	1	0	1
N_{57}	0	1	1	1	0	0	1	1	0
N_{58}	0	1	1	1	0	1	0	0	1
N_{59}	0	1	1	1	0	1	0	1	0
N_{60}	0	1	1	1	0	1	1	0	1

Після симуляції були отримані нові ваги: $W_{11} = 15.1222$, $W_{21} = 15.3044$,

$W_{31} = 13.1223$, $W_{41} = -4.7979$, $W_{51} = -3.2284$, $W_{61} = -3.2284$, $W_{71} = -3.2284$, $W_{81} = -16.7389$. Корегуємо ваги та починаємо нову симуляцію. Результати НМ представлені в таблиці 4.5.

Таблиця 4.5 – Результат навчання нейронної мережі

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_1	0	0	0	0	1	1	0	0	-1.1027	0.2492	0
N_2	0	0	0	0	0	1	1	0	1.1191	0.7538	1
N_3	0	0	0	1	0	1	1	0	-0.7898	0.3122	0
N_4	0	0	0	1	1	1	1	0	0.6025	0.6462	1
N_5	0	0	1	1	0	0	0	0	5.9900	0.9975	1
N_6	0	1	0	1	0	0	0	0	-1.4840	0.1848	0
N_7	1	0	0	0	0	0	0	0	6.6357	0.9987	1
N_8	0	0	0	0	0	1	1	1	-2.6600	0.0654	0
N_9	0	0	1	0	1	1	1	1	-10.2696	3.4672e-05	0
N_{10}	1	0	0	1	0	0	0	0	3.8676	0.9795	1

Результати, котрі були отримані під час останнього навчання, збігаються з тестовими результатами таблиці 4.1.

Нові скореговані ваги: $W_{11} = 6.9357$, $W_{21} = 0.2841$, $W_{31} = 3.7581$, $W_{41} = -1.1681$, $W_{51} = 0.1115$, $W_{61} = 0.3712$, $W_{71} = 0.3878$, $W_{81} = -9.3886$.

Таким чином можна зробити висновок, що НМ успішно завершила навчання.

4.3 Розробка додатку інтелектуальної системи підтримки прийняття рішень

На основі результатів досліджень та експериментів можна приступити до створення додатку, котрий буде шукати в інтернеті ресурси з проху-адресами та перевірятиме їх на працездатність.

В основі додатку буде використовуватись мова програмування PHP та фреймворк Symfony. Фреймворк Symfony ідеально підходить для створення додатків, котрі можна запускати з командної консолі, а розширення curl, мови PHP, дуже легке в використанні для відправки повідомлень на сервер та отриманні відповіді з нього. Знайдені проху-адреси зберігатимуться в базі даних. СУБД котре буде використовуватися в додатку – MySQL. Це пов'язано з тим, що фреймворк Symfony має вбудований функціонал для швидкого підключення і безпосередньої роботи з реляційними базами даних, того MySQL є найоптимальнішим варіантом.

Після обрання всіх необхідних інструментів можна приступити до створення додатку.

У розділі 3.4 були визначені основні умов для визначення працездатності проху-адрес, на основі цих даних були створені класи конвертори, котрі оброблюють дані та вертають булеві значення – 0 або 1 (рис. 4.1).

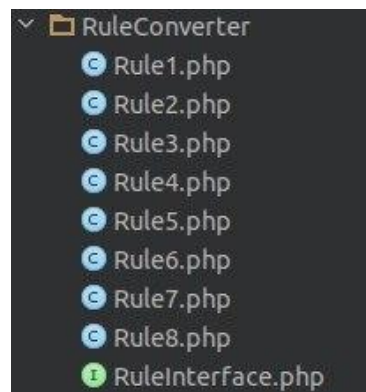


Рисунок 4.1 – Класи конвертори вхідних даних

У розділі 3.6 було розглянуто алгоритм побудови перцептрона, а в розділі 4.2 були проведені тестування та навчання даного алгоритму та системи в цілому. На основі цих дослідженні було побудовано сервіс «ProxyClassifier», котрий приймає масив класів, конверторів умов (рис. 4.2). Дані класи конвертують поточний стан проху-адреси в зрозумілі для нього значення – 0 або 1. Після конвертації відбувається процес аналізу проху-адреси

на її працездатність, у разі успішного підтвердження буде повернуто 1 або 0 у іншому випадку.

```

$classifier = new ProxyClassifier();

foreach($foundProxies as $proxy){
    $io->note(sprintf( format: "Proxy %s has status: %s",
        ...values: $proxy->getIp() . ':' . $proxy->getPort(),
        $classifier->getResult([
            new Rule1($proxy),
            new Rule2($proxy),
            new Rule3($proxy),
            new Rule4($proxy),
            new Rule5($proxy),
            new Rule6($proxy),
            new Rule7($proxy),
            new Rule8($proxy),
        ])
    ));
}

```

Рисунок 4.2 – Автоматичний процес класифікації знайдених проху-адресів

Даний додаток було реалізовано як команду, котру можна запускати з терміналу в реальному часі або за допомогою cron daemon, у внутрішніх процесах сервера. Результат виконання команди наведено на рис. 4.3.

```

! [NOTE] Proxy 145.76.72.10:4153 has status: 1
! [NOTE] Proxy 38.14.35.156:80 has status: 0
! [NOTE] Proxy 17.129.14.86:1256 has status: 1
! [NOTE] Proxy 56.56.12.145:8080 has status: 1
! [NOTE] Proxy 132.12.2.154:8080 has status: 0
! [NOTE] Proxy 202.166.206.59:5678 has status: 0
! [NOTE] Proxy 190.98.189.228:4145 has status: 1
! [NOTE] Proxy 202.131.233.187:5678 has status: 0
! [NOTE] Proxy 109.238.229.23:4145 has status: 1
! [NOTE] Proxy 183.88.219.206:34676 has status: 0

```

Рисунок 4.3 – Результат виконання команди

Результати виконання команди заносяться в базу даних, вони зображені на рис 4.4.

80426	3	34.201.103.100	49205	2021-10-18 22:20:06	2021-10-18 22:20:06
80427	3	89.250.149.114	60981	2021-10-18 22:20:06	2021-10-18 22:20:06
80428	3	37.1.43.116	53281	2021-10-18 22:20:06	2021-10-18 22:20:06
80429	3	154.72.199.202	41201	2021-10-18 22:20:06	2021-10-18 22:20:06
80430	3	202.57.37.197	59404	2021-10-18 22:20:06	2021-10-18 22:20:06
80431	3	52.250.1.188	80	2021-10-18 22:20:06	2021-10-18 22:20:06
80432	3	185.56.209.114	52342	2021-10-18 22:20:06	2021-10-18 22:20:06
80433	3	47.240.102.163	3128	2021-10-18 22:20:06	2021-10-18 22:20:06
80434	3	45.121.216.218	55443	2021-10-18 22:20:06	2021-10-18 22:20:06
80435	3	95.25.232.104	55443	2021-10-18 22:20:06	2021-10-18 22:20:06
80436	3	89.189.128.183	8080	2021-10-18 22:20:06	2021-10-18 22:20:06
80437	3	193.41.88.58	53281	2021-10-18 22:20:06	2021-10-18 22:20:06
80438	3	195.158.3.198	3128	2021-10-18 22:20:06	2021-10-18 22:20:06
80439	3	202.141.233.166	48995	2021-10-18 22:20:06	2021-10-18 22:20:06
80440	3	89.40.48.186	8080	2021-10-18 22:20:06	2021-10-18 22:20:06
80441	3	213.81.218.225	8080	2021-10-18 22:20:06	2021-10-18 22:20:06

Рисунок 4.4 – Збережені результати виконання команди

Розроблений додаток може бути модифікований або інтегрований, як сервіс в інші додатки.

ВИСНОВКИ

Метою магістерської кваліфікаційної роботи є розробка інтелектуальної системи підтримки прийняття рішень для виявлення працездатних гроху-адресів.

У роботі проводився аналіз методів класифікації нейронних мереж. Проаналізовано архітектури, типи нейронних мереж та їх функції активації. На основі аналізу було обрано архітектуру нейронної мережі. Даною архітектурою став багатошаровий перцептрон з сигмоїдальною функцією активації. Окрім цього було проаналізовано роботу гроху-адресів, гроху-серверів, умови при яких можна вважати гроху-адресу працездатною та при яких ні. Також було сформовано набори навчальних та тестових даних, що відповідають стану працездатності гроху-адрес, котрі знаходяться у вільному доступі. Виконано покрокове навчання НМ у програмному забезпеченні Matlab. Створений додаток з ІСППР для пошуку та класифікації працездатних гроху-адресів.

Створену НМ було перевірено на визначення стану працездатності гроху-адресів за допомогою тестових вхідних наборів даних. Отримані результати показують здатність НМ виявляти працездатні та не працездатні гроху-адреси.

Ефективність НМ оцінюється за наступними параметрами:

- ваги, котрі були відкориговані під час навчання мережі (за допомогою функції зворотного розповсюдження);
- ваги, котрі були отримані після сумарного підрахунку;
- результат функції активації;
- вихідні дані.

Визначено, що для навчання НМ досить сто двадцяти навчальних наборів та п'ятдесят тисяч епох з коефіцієнтом навчання 0.8.

Аналіз показав, що навчання мережі накладає обмеження на правильну

класифікацію в порядку помилок елементів у вхідному наборі. Якщо збільшується кількість навчальних даних, тоді ефективність розпізнавання зростає. Визначається ступінь змін помилок, що залежить від мінімальної кількості об'єктів у навчальному наборі.

Було визначено яким чином навчальний набір впливає на якість визначення працездатності гроху-адрес. Ефективність підтверджується точністю визначення стану працездатності об'єкту. Дослідження показали, що при додаванні великої кількості наборів тестових даних зростає точність визначення стану об'єкту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ситник В. Ф. Системи підтримки прийняття рішень: навч. посіб. Київ: КНЕУ, 2004. 614 с.
2. Братушка С. М., Новак С. М., Хайлук С. О. Системи підтримки прийняття рішень: навч. посіб. для самост. вивч. дисципліни: для студ. вищ. навч. закл. Суми: УАБС НБУ, 2010. 265 с.
3. Gloria Phillips-Wren. Intelligent Decision Support Systems. Multicriteria Decision Aid and Artificial Intelligence. 2013. p.25-44. URL: https://www.researchgate.net/publication/277703502_Intelligent_Decision_Support_Systems (дата звернення: 18.10.2021).
4. Gupta JND, Forgionne GA, Mora MT. Intelligent Decision-making Support Systems. London: Springer, 2006. 504 с.
5. Нестеренко О. В., Савенко О. І., Фаловський О. О. Інтелектуальні системи підтримки прийняття рішень: навч. посібн./ за ред. П.І. Бідюка – Київ: Національна академія управління, 2016. – 188 с.
6. He Changlin, Li Yufen. A Survey of Intelligent Decision Support System. Advances in Engineering Research, volume 122: 7th International Conference on Applied Science, Engineering and Technology (ICASET 2017), Qingdao, China, 29–30 July 2017. 2017. P. 201–206.
7. Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд/ пер. з англ. К.А. Птицын.. Москва: Вильямс, 2016. 1408 с.
8. Анонімний мережевий проксі сервер URL: <https://comuedu.ru/uk/windows/anonymous-network-proxy-server-proxy-security-or-should-i-trust-public-proxy-servers.html> (дата звернення 18.10.2021).
9. Прозрачный прокси сервер URL: <https://proxys.io/ru/blog/proksi-info/prozrachnyu-proksi-squid-chto-eto-gde-primenyaetsya> (дата звернення 18.10.2021).
10. Что такое анонимный прокси? URL:

<http://forumaboutproxy.com/obshie-voprosi/chto-takoe-anonimnii-proksi-tipi-proksi-prozrachnii-anonimnii-ili-elitnii> (дата звернення: 18.10.2021).

11. Proxy Server URL: [https:// www.fortinet.com/resources/cyberglossary/proxy-server](https://www.fortinet.com/resources/cyberglossary/proxy-server) (дата звернення: 20.10.2021).

12. Adrian R. Compare Different Types of Proxies. Best Proxy Reviews. URL: <https://www.bestproxyreviews.com/different-types-of-proxies/#types-of-proxies-based-on-ip-origin> (дата звернення: 02.11.2021).

13. Различные плюсы и минусы прокси-сервера URL: <https://itsecforu.ru/2020/08/13> (дата звернення: 02.11.2021).

14 7 архитектур нейронных сетей для решения задач NLP. URL: <https://neurohive.io/ru/osnovy-data-science/7-arhitektur-nejronnyh-setej-nlp> (дата звернення: 04.11.2021).

15. Субботін С. О. Нейронні мережі: теорія та практика: навч. посіб. – Житомир: Вид. О. О. Євенок, 2020. – 184 с.