НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«НЕЙРОННІ МЕРЕЖІ В ЗАСОБАХ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНО-ТЕХНІЧНИХ СИСТЕМАХ»

Виконав: здобувач 4 курсу
Рівень бакалавр
Галузь знань 12 «Інформаційні технології»
спеціальність 125 «Кібербезпека»
Заставнюк Мирослав Андрійович

Керівник: к.ф.-м.н., доцент
Черевко Євген Володимирович

Рецензент: к.м.-ф.н., доцент
Чепурна Олена Сергіївна

Одеса – 2024

АНОТАЦІЯ

Кваліфікаційна робота на тему «Нейронні мережі в засобах захисту інформації в інформаційно-технічних системах» на здобуття першого (бакалаврського) рівня вищої освіти за спеціальності 125 «Кібербезпека».

Містить 12 рисунків, 2 таблиці, 11 літературних джерел за переліком посилань. Робота виконана на сторінках загального тексту та сторінках основного тексту.

Метою даної роботи є ознайомлення та аналіз нейронних мереж для захисту інформації в інформаційно-технічних системах, а також створення власної нейронної мережі для задачі захисту інформації, а саме для фільтрації електронних листів.

У роботі розглядаються теоретичні відомості про нейронні мережі, їх будову та приклади використання, методи навчання і також практичне застосування на декількох прикладах.

Проаналізовано ефективність використання нейронних мереж для захисту інформації, зокрема роль нейронних мереж, в задачах захисту інформації у інформаційно-технічних системах.

Також будуть розглянуті стратегії оптимізації навчання та подальшого використання нейронних мереж, оцінка їх ефективності. У висновках будуть представлені основні результати дослідження та рекомендації щодо подальшого вдосконалення та розвитку нейронних мереж для захисту інформації у інформаційно-технічних системах.

НЕЙРОННІ МЕРЕЖІ, ІНФОРМАЦІЙНО-ТЕХНІЧНІ СИСТЕМИ, СИНАПС, ОДНОНЕЙРОННА МЕРЕЖА.

ABSTRACT

Qualification work on the topic «Neural networks in information security in information technology systems» for the first (bachelor's) level of higher education in the speciality 125 «Cybersecurity».

Contains 12 figures, 2 tables, 11 references according to the list of references. The work is performed on the pages of the general text and the pages of the main text.

The purpose of this work is to review and analyse neural networks for information security in information technology systems, as well as to create our own neural network for the task of information security, namely for filtering emails.

The paper discusses theoretical information about neural networks, their structure and examples of use, training methods, and practical application on several examples.

The paper analyses the effectiveness of using neural networks for information security, in particular the role of neural networks in information security tasks in information technology systems.

Strategies for optimising the training and further use of neural networks, as well as evaluating their effectiveness, will also be considered. The main results of the study and recommendations for further improvement and development of neural networks for information security in information and technical systems will be presented in the conclusions.

NEURAL NETWORKS, INFORMATION AND TECHNICAL SYSTEMS, SYNAPSE, SINGLE-NEURON NETWORK.

ЗМІСТ

ВСТУП

Актуальність роботи: У нинішній час цінність інформації зростає. Ріст кількості кіберзагроз та складність методів кібератак роблять питання із захисту інформації одним з найважливіших у сфері інформаційної безпеки.

Мета роботи: Метою роботи є дослідження можливостей застосування нейронних мереж для захисту інформації в інформаційно-технічних системах.

В сучасному світі інформаційно-технічні системи пронизали всі сфери життя, від особистих до корпоративних та державних структур. Їхня складність та важливість стрімко зростають, роблячи їх ласим шматочком для кіберзловмисників. Зростає рівень загроз для безпеки та конфіденційності інформації, яку вони обробляють та зберігають. Тому виникає гостра потреба в ефективних методах захисту, здатних реагувати на сучасні атаки в режимі реального часу.

Нейронні мережі, натхненні людським мозком, за останні роки здобули значне визнання та успішно застосовуються в багатьох галузях, включаючи інформаційну безпеку. Ці складні моделі здатні навчатися, розпізнавати патерни та проводити складний аналіз даних. Автоматичне навчання на основі отриманих даних це дійсно важливий та потужний інструмент. Однак, до 2006 року існували значні обмеження в навчанні нейронних мереж. Вони не могли конкурувати з традиційними методами програмування, за винятком деяких вузьконаправлених задач. Але ситуація кардинально змінилася у 2006 році з появою методів навчання “глибоких” нейронних мереж. Ці методи, які зараз відомі як “глибоке навчання”, постійно вдосконалюються, і на сьогоднішній день глибокі нейронні мережі демонструють приголомшливі результати в багатьох сферах, таких як комп'ютерний зір, розпізнавання мови та обробка природної мови. Їх активно використовують такі технологічні гіганти, як Google, Microsoft та Facebook.

1 ТЕОРЕТИЧНІ ОСНОВИ НЕЙРОННИХ МЕРЕЖ

1.1 Будова та використання нейронних мереж

З курсу біології нам відомо, що нейрон це клітина нервової системи, яка отримує сигнали від інших нейронів, обробляє їх та генерує власний сигнал, який потім передається до інших нейронів. Нейрони пов’язані між собою синапсами, які дозволяють їм передавати сигнали один одному.

Штучний нейрон, на відміну від біологічного не має фізичної структури. Це математична функція, яка приймає на вхід декілька сигналів, обробляє їх та генерує вихідний сигнал. Нейрони з’єднуються між собою та утворюють нейронні мережі, які можуть навчатися на даних, знаходячи закономірності та роблячи прогнози.

Нейронні мережі - це потужні інструменти, які можуть використовуватися для вирішення багатьох складних задач,наприклад:

1) Розпізнавання образів: Нейронна мережа розпізнає об’єкт на зображенні, наприклад обличчя або якийсь предмет.

2) Обробка мови: Зараз нейронні мережі можуть перекладати мови з відео/ аудіо файлів, генерувати тексти та мають можливість спілкування в режимі онлайн.

3) Прогнозування: Нейронні мережі на основі даних можуть прогнозувати майбутні події, такі як ціни на акції, результати виборів або погода.

4) Управління: Використання нейронних мереж для керування роботами та іншими системами де в подальшому керування людьми не буде потрібне.

5) Захист інформації: Нейронні мережі можуть використовуватись для виявлення та запобігання кібератак.

Їх можливості постійно розширюються, що робить їх одним з найважливіших напрямків досліджень у штучному інтелекті. Також важливо зазначити що наразі нейронні мережі не є досконалими, вони можуть бути складними для навчання та налаштування, можуть бути проблеми з інтерпретацією а також вони можуть бути вразливими до атак. Але не зважаючи

на ці обмеження, нейронні мережі вже вважаються одним із найважливіших інструментів у штучному інтелекті.

1.2 Види архітектур та алгоритмів навчання нейронних мереж:

1) Одношарові нейронні мережі

Одношарові мережі складаються з одного шару штучних нейронів, які безпосередньо отримують вхідні дані і виробляють вихідні. Їх можна використовувати для простих завдань класифікації, наприклад, для визначення, чи є зображенням кішка або собака.

Структура:

- Вхідний шар: складається з нейронів, які отримують вхідні дані.
- Вихідний шар: складається з нейронів, які виробляють вихід.
- Ваги: визначають силу впливу одного нейрона на інший.

Алгоритми навчання

Персептрон - це один з найпростіших алгоритмів навчання для одношарових нейронних мереж. Він використовує правила керованого навчання для оновлення ваг мережі.

Переваги:

- Простий і легкий в реалізації.
- Корисний для простих завдань класифікації.

Недоліки:

- Не може виконувати складне навчання або аналіз даних.
- Не може обробляти багатовимірні дані.

*2) Багатошарові нейронні мереж*і

Багатошарові нейронні мережі складаються з двох або більше шарів штучних нейронів, з'єднаних зваженими зв'язками. Їх можна використовувати для складних завдань, таких як розпізнавання мови, обробка зображень та прогнозування часових рядів.

Структура виглядає наступним чином

- Вхідний шар: складається з нейронів, які отримують вхідні дані. Прихований шар: складається з нейронів, які обробляють вхідні дані і видають проміжні результати.
- Вихідний шар: складається з нейронів, які виробляють вихідні дані. Ваги: визначають силу впливу одного нейрона на інший.

Алгоритм навчання:

Зворотне поширення: один з найпоширеніших алгоритмів навчання для багатошарових нейронних мереж. Для оновлення ваг мережі використовуються правила навчання під наглядом.

Переваги:

- Дозволяє проводити складне навчання та аналіз даних.
- Може працювати з багатовимірними даними.
- Корисний для широкого спектру завдань.

Недоліки:

Схильний до перенавчання, якщо не використовуються методи регуляризації.

3) Згорткові нейронні мережі

Згорткові нейронні мережі - це тип багатошарової нейронної мережі, спеціально розроблений для обробки зображень. Вона використовує згорткові та об'єднувальні шари для виявлення локальних особливостей зображення.

Структура

Згортковий шар: застосовує до зображення згортковий фільтр для виявлення локальних особливостей.

Агрегувальний шар: зменшує розмірність зображення, зберігаючи найважливішу інформацію.

Повноз’єднальний шар: використовується для класифікації та прогнозування на основі виявлених ознак.

Алгоритми навчання:

Зворотне поширення помилки: використовується для оновлення ваг мережі.

Переваги.

- Корисний для обробки зображень та відео.
- Може виявляти локальні особливості на зображеннях.
- Стійкий до шуму та спотворень.

Недоліки:

- Складніший і дорожчий в обчислювальному плані, ніж інші типи нейронних мереж.
- Може вимагати великих обсягів даних для навчання.

4) Рекурентні нейронні мережі

Рекурентні нейронні мережі - це тип нейронних мереж, які спеціально розроблені для обробки послідовностей даних, таких як текст або мова. Вони здатні враховувати контекст попередніх елементів послідовності при обробці поточного елементу.

Структура:

- Вхідний шар: отримує елементи послідовності даних.
- Прихований шар: зберігає стан мережі в залежності від попереднього елемента послідовності.
- Вихідний шар: генерує вихідні дані в залежності від поточних елементів послідовності та стану мережі.

Алгоритм навчання:

- Часове розширення: алгоритм створює копію мережі для кожного елемента послідовності та розширює рекурентну нейронну мережу з часом. Потім він застосовує поширення помилки для оновлення ваг мережі.
- Інші алгоритми навчання: Алгоритми навчання для рекурентної нейронної мережі включають контрольоване і неконтрольоване навчання.

Переваги:

- При обробці послідовностей даних можна враховувати контекст.
- Корисно для таких завдань, як машинний переклад, розпізнавання мови та аналіз тексту.

Недоліки:

- Може страждати від проблеми зникаючого градієнта, що ускладнює навчання мережі на довгих послідовностях.
- Складніше в навчанні, ніж інші типи нейронних мереж.

5) Додаткові типи нейронних мереж

Існує багато інших типів нейронних мереж, які використовуються для різних цілей, наприклад:

- Автокодери: навчаються стискати та відновлювати дані, що корисно для задач зменшення розмірності та виявлення аномалій.
- Генеративні моделі: навчаються генерувати нові дані, що може бути використано для створення реалістичних зображень, тексту або музики.
- Капсульні мережі: відносно новий тип нейронних мереж, які показують перспективні результати в розпізнаванні об'єктів та сегментації зображень.

Вибір типу нейронної мережі для конкретної задачі залежить від типу даних, яку потрібно обробити, та бажаного результату.

1.3 Застосування нейронних мереж в засобах захисту інформації

1) Виявлення вторгнень

Нейронні мережі можуть ефективно використовуватися для виявлення вторгнень в інформаційно-технічних системах. Їх можна навчити на великих наборах даних, що містять приклади нормальної та аномальної мережевої активності, щоб розпізнавати підозрілі дії, які можуть свідчити про кібератаку.

Переваги:

- Здатність навчатися на великих обсягах даних.
- Можливість виявляти складні та невідомі раніше загрози.
- Здатність адаптуватися до мінливих умов середовища.

Методи:

Аномальне виявлення поведінки (ABAS): нейронні мережі навчаються на базі нормальної поведінки користувачів та мережевого трафіку, щоб виявляти будь-які відхилення, які можуть свідчити про атаку.

Класифікація трафіку: нейронні мережі навчаються розрізняти легітимний та шкідливий трафік, ґрунтуючись на його характеристиках, таких як протоколи, адреси та порти.

Приклади:

- Система виявлення вторгнень на основі нейронних мереж, яка може виявляти атаки типу "людина в середині" (MitM).
- Система виявлення вторгнень на основі нейронних мереж, яка може виявляти атаки типу "відмова в обслуговуванні" (DoS).

2) Захист від шкідливого програмного забезпечення

Нейронні мережі також можуть використовуватися для захисту інформаційно-технічних систем від шкідливого програмного забезпечення. Їх можна навчити на великих наборах даних, що містять приклади шкідливого та легітимного програмного забезпечення, щоб розпізнавати та блокувати віруси, хробаки, трояни та інші загрози.

Переваги:

- Здатність навчатися на великих обсягах даних.
- Можливість виявляти нові та невідомі раніше типи шкідливого програмного забезпечення.
- Здатність адаптуватися до мінливих методів розробки шкідливого програмного забезпечення.

Методи:

- Статичний аналіз: нейронні мережі аналізують код програмного забезпечення, щоб визначити, чи є воно шкідливим чи легітимним.
- Динамічний аналіз: нейронні мережі спостерігають за поведінкою програмного забезпечення під час його виконання, щоб визначити, чи є воно шкідливим чи легітимним.

Приклади:

- Система захисту від шкідливого програмного забезпечення на основі нейронних мереж, яка може виявляти та блокувати нові та невідомі раніше типи вірусів.
- Система захисту від шкідливого програмного забезпечення на основі нейронних мереж, яка може виявляти та блокувати трояни, які намагаються вкрасти особисті дані або шпигувати за користувачами.

3) Фішинг та розсилка спаму

Нейронні мережі можуть використовуватися для боротьби з фішингом та розсилкою спаму. Їх можна навчити на великих наборах даних, що містять приклади легітимних та фішингових електронних листів, а також спаму та не спаму, щоб розпізнавати та блокувати шкідливі повідомлення.

Переваги:

- Здатність навчатися на великих обсягах даних.
- Можливість виявляти складні та невідомі раніше типи фішингових електронних листів та спаму.
- Здатність адаптуватися до мінливих методів фішингу та розсилки спаму.

Методи:

Аналіз тексту: нейронні мережі аналізують текст електронних листів, щоб визначити, чи є вони легітимними чи фішинговими, або спамом.

4) Захист даних

Нейронні мережі можуть використовуватися для захисту даних від несанкційованого доступу. Вони можуть застосовуватися для різних методів захисту даних, таких як:

- Шифрування: Нейронні мережі можуть генерувати складні ключі шифрування, які важко зламати. Вони також можуть використовуватися для оптимізації процесів шифрування та дешифрування.
- Стеганографія: Нейронні мережі можуть приховувати дані в інших, начебто нешкідливих файлах, таких як зображення або аудіофайли.

- Контроль доступу: Нейронні мережі можуть аналізувати дані користувачів та поведінку, щоб визначити, чи мають вони право доступу до певних ресурсів.

Переваги:

- Здатність генерувати складні ключі шифрування.
- Можливість приховувати дані більш ефективно, ніж традиційні методи стеганографії.
- Здатність адаптуватися до нових методів атак на дані.

Недоліки:

- Існує ризик того, що нейронні мережі, які використовуються для шифрування, можуть бути зламані в майбутньому, якщо розвинуться більш потужні комп'ютери.
- Стеганографічні методи на основі нейронних мереж можуть бути виявлені, якщо зловмисники дізнаються про ці методи.

Методи:

- Шифрування на основі нейронних мереж: Нейронні мережі генерують складні ключі шифрування, які стійкі до методів злому.
- Стеганографія з використанням нейронних мереж: Нейронні мережі приховують дані в зображеннях, аудіо- та відеофайлах, роблячи їх невидимими для нетренованого ока.

5) Автентифікація та авторизація

Нейронні мережі можуть використовуватися для більш надійної аутентифікації користувачів та авторизації доступу до ресурсів. Вони можуть аналізувати різні фактори, такі як відбитки пальців, розпізнавання обличчя, поведінку користувача під час введення пароля, щоб визначити, чи є користувач тим, за кого він себе видає.

Переваги:

- Здатність забезпечити багатофакторну аутентифікацію.
- Можливість виявляти атаки типу "brute force" та інші спроби несанкційованого доступу.

- Здатність адаптуватися до змін у поведінці користувачів.

Недоліки:

- Існує ризик того, що біометричні дані, такі як відбитки пальців або дані розпізнавання обличчя, можуть бути скомпрометовані.
- Системи аутентифікації на основі нейронних мереж можуть бути обмануті, якщо зловмисники навчаться підробляти поведінку користувачів.

Методи:

- Розпізнавання облич: нейронні мережі навчаються на великому наборі даних зображень обличчя людей, щоб виявляти та розпізнавати унікальні риси обличчя кожної людини.
- Відбитки пальців: Нейронні мережі навчаються на великому наборі даних зображень відбитків пальців, щоб виявляти та розпізнавати унікальні патерни ліній на пальцях кожної людини.

6) Аналіз даних:

Нейронні мережі здатні аналізувати великі обсяги даних про кібербезпеку. Вони можуть виявляти тенденції та закономірності, які допомагають у прогнозуванні та запобіганню майбутнім атакам.

Переваги:

- Здатність обробляти великі обсяги даних з різних джерел.
- Можливість виявляти складні взаємозв'язки між різними інцидентами кібербезпеки.
- Здатність прогнозувати майбутні атаки та вживати заходів для їх запобігання.

Недоліки:

- Для ефективного навчання нейронних мереж потрібні великі обсяги даних.
- Можливість помилкових спрацювань, коли система виявляє загрозу, якої насправді немає.

Методи:

- Виявлення аномалій: Нейронні мережі навчаються на базі нормальних даних про кібербезпеку, щоб виявляти будь-які відхилення, які можуть свідчити про атаку або інцидент.
- Прогнозування загроз: Нейронні мережі навчаються на історичних даних про кібербезпеку, щоб прогнозувати ймовірність майбутніх атак або інцидентів.

Нейронні мережі - це могутня технологія, яка має потенціал змінити світ багатьма способами. Важливо ретельно вивчити та обговорити етичні міркування, пов'язані з цією технологією, щоб забезпечити її відповідальне використання на користь людства.

1.4 Майбутнє нейронних мереж:

1) Поточні дослідження та розробки:

- Нейроморфні обчислення: Ця галузь досліджень прагне створити комп'ютерні системи, які імітують структуру та функції людського мозку. Це може призвести до більш енергоефективних та потужних нейронних мереж, які здатні до більш складних задач, таких як навчання та адаптація в режимі реального часу.
- Глибоке навчання з підкріпленням: Цей тип навчання дозволяє нейронним мережам навчатися на основі проб та помилок, взаємодіючи з динамічним середовищем. Це може призвести до більш автономних та інтелектуальних систем, які здатні до самостійного прийняття рішень та виконання складних завдань.
- Пояснений штучний інтелект: Ця область досліджень прагне зробити роботу нейронних мереж більш прозорою та зрозумілою. Це може допомогти людям краще зрозуміти, як нейронні мережі приймають рішення, а також підвищити довіру до цих систем.

2) Потенційні наслідки для суспільства та економіки:

- Автоматизація: Нейронні мережі можуть призвести до автоматизації багатьох завдань, які зараз виконуються людьми. Це може призвести до втрати робочих місць у деяких сферах, але також може створити нові можливості для працевлаштування в інших.
- Персоналізація: Нейронні мережі можуть використовуватися для персоналізації продуктів, послуг та маркетингових повідомлень для кожного користувача. Це може призвести до більш кращого досвіду для користувачів, але також може викликати занепокоєння щодо конфіденційності та дискримінації.
- Збільшення можливостей: Нейронні мережі можуть використовуватися для вирішення складних проблем, які раніше вважалися нерозв'язними, таких як розробка нових ліків, прогнозування природних катаклізмів та створення штучного загального інтелекту.

3) Етичні міркування:

- Упередженість: Нейронні мережі можуть бути упередженими, якщо вони навчені на даних, які містять упередженість. Це може призвести до дискримінації певних груп людей.
- Прозорість: Нейронні мережі можуть бути складними і незрозумілими, що ускладнює розуміння того, як вони приймають рішення. Це може призвести до проблем з довірою та відповідальністю.
- Зловживання: Нейронні мережі можуть використовуватися в зловмисних цілях, таких як створення глибоких підробок або розробка автономної зброї.

Додаткові можливості нейронних мереж:

Нейронні мережі мають значний потенціал для вирішення проблем кібербезпеки, які виходять за рамки типових задач, таких як виявлення кібератак та захист даних. Ось декілька прикладів додаткових можливостей нейронних мереж в цій сфері:

- Розслідування кіберзлочинів:

Аналіз даних про кіберзлочини: нейронні мережі можуть аналізувати великі обсяги даних, пов'язаних з кіберзлочинами, включаючи веб-трафік,

електронні листи, записи журналів та файли з комп'ютерів, заражених шкідливим ПЗ. Це може допомогти слідчим виявити закономірності, зв'язки та сліди, які важко знайти власноруч.

- Захист від дезінформації:

Виявлення фейкових новин: нейронні мережі можуть аналізувати текст, зображення та відео, щоб виявити ознаки дезінформації, такі як неправдиві твердження, маніпулятивні зображення та вступні емоційні заголовки.

- Захист від ботів:

Виявлення бот-трафіку: нейронні мережі можуть аналізувати мережевий трафік, щоб виявити ознаки бот-активності, такі як незвичайні обсяги трафіку, підозрілі IP-адреси та автоматизовані запити.

- Захист пристроїв інтернету речей:

Виявлення вразливостей пристроїв інтернету речей: нейронні мережі можуть аналізувати дані з пристроїв інтернету речей, щоб виявити вразливості, які можуть бути використані хакерами.

2 ПРИКЛАДИ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ

2.1 Будова нейронних мереж:

Для більш глибокого розуміння потенціалу та найбільш ефективного використання нейронних мереж, важливо дослідити їх внутрішню будову та принципи роботи. Отримавши знання про те, як працює нейронна мережа можна буде краще оцінити її можливості та обмеження у задачах із захисту інформації. Це розуміння допоможе нам використовувати нейронні мережі більш ефективно, мати змогу розробляти інноваційні рішення в кібербезпеці та будувати більш стійкі та безпечні інформаційно-технічні системи. Тому давайте більш детально розглянемо будову нейронних мереж та приклади їх використання.

Нейронна мережа - це складна система, яка складається з трьох основних компонентів:

1) Штучні нейрони:

Штучний нейрон - це базовий обчислювальний елемент нейронної мережі, який імітує роботу біологічних нейронів.

Кожен штучний нейрон складається з трьох основних частин:

- Вхідні сигнали: Нейрон отримує сигнали від інших нейронів або від зовнішнього збудника.
- Активація: Ця функція обробляє вхідні сигнали та генерує вихідний сигнал.
- Вихідний сигнал: Нейрон передає вихідний сигнал іншим нейронам або для виконання задачі.

Різні типи нейронів використовують різні типи функцій активації, які визначають як нейрон реагуватиме на вхідні сигнали.

2) З’єднання:

Зв'язки між нейронами визначають, як сигнали передаються між ними.

Кожен зв'язок має вагу, яка визначає силу впливу одного нейрона на інший.

Ваги зв'язків можуть бути додатними, від'ємними або нульовими.

Позитивні ваги посилюють вплив нейрона, а негативні - послаблюють.

Ваги зв'язків налаштовуються під час навчання нейронної мережі.

3) Шари:

Нейрони в нейронній мережі організовані в шари.

Сигнали проходять між шарами, що дозволяє мережі виконувати складні обчислення.

Типи шарів:

- Вхідний шар: Отримує дані ззовні.
- Приховані шари: Обробляють і передають дані між собою.
- Вихідний шар: Генерує вихідні дані.

Кількість шарів і кількість нейронів у кожному шарі визначають архітектуру нейронної мережі.

2.2 Типи нейронних мереж:

Тепер давайте розглянемо типи нейронних мереж. Нейронні мережі бувають різних типів залежно від задач які вимагають їх використання. Ось деякі з найпоширеніших типів нейронних мереж:

1) Штучна нейронна мережа прямого поширення:

- Це найпростіший тип нейронних мереж, де сигнали проходять лише в одному напрямку, від вхідного шару до вихідного.
- Штучна нейронна мережа прямого поширення складаються з трьох типів шарів:
  - Вхідний шар: Отримує дані ззовні
  - Прихований шар: Обробляє та передає дані між собою
  - Вихідний шар: Генерує вхідні дані.

Цю нейронну мережу використовують для задач класифікації, таких як розпізнавання образів або прогнозування ймовірності події.

2) Згорткові нейронні мережі:

- На відміну від нейронних мереж прямого поширення, згорткові нейронні мережі мають зворотні зв’язки, які дозволяють сигналам проходити назад через мережу.
- Ці зворотні зв’язки використовуюся для налаштування ваг з’єднань під час навчання, що робить цю нейронну мережу більш потужною ніж нейронну мережу прямого направлення.
- Згорткові нейронні мережі використовують для задач з навчанням з учителем, де мережа навчається на наборі даних з правильними відповідями.
- Приклади для яких можуть використовуватися згорткові нейронні мережі:
  - Розпізнавання мови
  - Прогнозування часових рядів
  - Обробка зображень

3) Рекурентні нейронні мережі

- Ці нейронні мережі використовують для обробки послідовних даних, таких як текст або мова.
- Рекурентні нейронні мережі використовують рекурентні шари, які дозволяють їм врахувати контекст попередніх даних при обробці поточних даних.
- Цей тип нейронних мреж використовуються для задач з обробки природної мови, таких як:
  - Машинний переклад
  - Розпізнавання мови
  - Аналіз тексту

Обираючи яку нейронну мережу використовувати потрібно враховувати багато факторів, таких як:

Типи даних(зображення, послідовні дані, обробка природної мови тощо)

Розміри та складність наборів даних

Виконавчі ресурси(обчислювальні потужність, час навчання)

Доступність даних про навчання з мітками

Також важливо розуміти сильні та слабкі сторони різних типів нейронних мереж, щоб обрати ту, яка найкраще підходить до конкретно поставленої задачі.

2.3 Навчання нейронних мереж:

Для навчання нейронних мереж використовують машинне навчання, яке є необхідною частиною нейронних мереж. Вони навчаються на даних, для виконання різних задач, таких як розпізнавання образів, прогнозування та прийняття рішень. Процес навчання є ключовим для того щоб нейронні мережі стали корисними та потужними інструментами. Існує декілька способів, завдяки яким машинне навчання використовується в нейронних мережах:

- Навчання під наглядом: У цьому типі навчання нейронна мережа отримує набір даних, в якому кожен зразок має мітку або правильну відповідь. НМ навчається на цих даних, щоб встановити зв'язок між вхідними даними та вихідними мітками. Наприклад, нейронну мережу можна навчити розпізнавати об'єкти на зображеннях, надавши йому набір даних зображень, кожне з яких має мітку, що вказує на тип об'єкта, який на ньому зображено.
- Навчання без нагляду: У цьому типі навчання нейронна мережа не отримує міток для даних. Замість цього він повинен самостійно знайти структуру або приховані закономірності в даних. Наприклад, нейронну мережу можна навчити групувати дані на основі спільних ознак, без жодних міток, які б вказували, до якої групи належить кожен зразок.
- Навчання з підкріпленням: У цьому типі машинного навчання нейронна мережа вчиться виконувати дії в навколишньому середовищі, отримуючи зворотний зв'язок про те, наскільки правильними є його дії. НМ намагається максимізувати загальний кумулятивний зворотний зв'язок, навчаючись на своїх помилках з плином часу. Наприклад, нейронну мережу можна навчити грати в гру, надавши йому доступ до ігрового середовища і заохочуючи його за правильні рухи і караючи за неправильні.

- Глибоке навчання: Глибоке навчання відрізняється від інших методів навчання нейронних мереж, таких як навчання з учителем або навчання з підкріпленням, тим, що використовує нейронні мережі з кількома шарами. Ці шари дозволяють нейронним мережам вивчати складні нелінійні зв'язки між вхідними даними та вихідними мітками, що робить їх більш потужними та універсальними, ніж нейронні мережі з одним шаром або з невеликою кількістю шарів.

Давайте більш детально розглянемо спосіб глибокого навчання. Цей спосіб як і більшість має свої переваги та недоліки, ось деякі із них:

- Переваги глибокого навчання:
  - Висока точність: Глибока нейронна мережа може досягти високої точності в багатьох завданнях, які раніше вважалися складними для комп'ютерів.
  - Універсальність: Глибоке навчання можна застосовувати до широкого кола завдань, що робить його цінним інструментом для дослідників і розробників.
  - Постійне вдосконалення: Сфера глибокого навчання стрімко розвивається, і постійно розробляються нові алгоритми і методи, які роблять нейронні мережі ще більш потужними.
- Недоліки глибокого навчання:
  - Високі обчислювальні вимоги: Навчання глибоких нейронних мереж може бути складним з точки зору обчислень, що вимагає більш потужних комп'ютерів.
  - Потреба у великих наборах даних: Глибоке навчання, як правило, вимагає великих обсягів даних для навчання, що може бути складним для деяких завдань.
  - Складність інтерпретації: Буває важко зрозуміти, як системи глибокого навчання приймають рішення, що може обмежувати їхню прозорість і надійність.

Також глибоке навчання відрізняється від машинного навчання у деяких аспектах, а саме:

- Залежність від даних: Ключова відмінність глибинного навчання від традиційного машинного навчання - це його продуктивність при збільшенні обсягу даних. Алгоритми глибинного навчання не працюють так ефективно, коли обсяги даних невеликі, оскільки алгоритми глибинного навчання потребують великого обсягу даних, щоб досконало їх зрозуміти. І навпаки машинне навчання не буде працювати належним чином при великому обсязі даних.
- Залежності від оснащення: Алгоритм глибокого навчання вимагає багато матричних операцій. Процесор в основному використовується для ефективної оптимізації матричних операцій. Отже, графічний процесор - це апаратне забезпечення, необхідне для належної роботи глибинного навчання. Глибоке навчання більше покладається на потужні графічні процесори, ніж на традиційні алгоритми машинного навчання.
- Метод вирішення проблем: Застосовуючи традиційні алгоритми машинного навчання для вирішення проблем, машинне навчання зазвичай розбиває проблему на кілька під-проблем і вирішує під-проблеми, щоб отримати кінцевий результат. На противагу цьому, глибоке навчання виступає за пряме наскрізне вирішення проблеми.
- Час виконання: Загалом, навчання алгоритму глибокого навчання займає багато часу, оскільки в алгоритмі глибокого навчання є багато параметрів; отже, крок навчання займає більше часу. Найсучасніший алгоритм глибокого навчання, такий як ResNet, навчається рівно два тижні, тоді як навчання машинного навчання займає відносно небагато часу - від кількох секунд до кількох годин. Однак, час тестування є прямо протилежним. Алгоритми глибокого навчання потребують дуже мало часу для тестування. У порівнянні з деякими алгоритмами машинного навчання, час тестування збільшується зі збільшенням обсягу даних. Однак це стосується не всіх алгоритмів машинного навчання, оскільки деякі алгоритми машинного навчання мають короткий час навчання.

Також існує метод градієнтного спуску. Градієнтний спуск — це один із методів оптимізації, який дозволяє нейронній мережі навчатися. Зазвичай сигнали у нейронній мережі транспортується від вхідних рецепторів у перший і всі наступні приховані шари, доки не трансформуються у сигнал вихідного шару. Однак, якщо нейронна мережа не налаштована, ми отримуватимемо помилки. Для того щоб мінімізувати виникнення цих помилок, потрібно оптимізувати нейронну мережу.

Кінцевою метою оптимізації є знаходження таких параметрів, при яких функція помилки досягатиме мінімуму. Якраз одним із таких методів є метод градієнтного спуску. За його допомогою знаходиться напрямок, у якому функція помилки зменшується та оновлює параметри моделі відповідним чином. З математики відомо, що градієнт - це вектор, який спрямований у бік максимальної зміни функції. Але для нейронних мереж градієнт функції помилки допомагає визначати, як змінні одного чи іншого параметру впливають на значення функції помилки. Якщо уявити алгоритм градієнтного спуску то він буде виглядати наступним чином:

- Ініціалізація параметрів моделі випадковими значеннями
- Подання вхідних даних в модель та отримання передбачення
- Обчислення значення функції помилки, порівнюючи прогноз із фактичними значеннями
- Визначення градієнту функції помилки за кожним параметром моделі
- Оновлення значень параметрів функції за формулою:

нове значення параметра = старе значення параметра – швидкість навчання* градієнт.

- Повторення кроків з 2-го по 5-й для кожного етапу навчання або до досягнення критерію зупинки.

Найскладніше для цього методу це підібрати потрібний крок навчання, щоб градієнтний спуск не розходився і не забирав надто багато часу. Інколи градієнти, які передаються вагам мережі, можуть стати дуже малими або дуже

великими, що відповідно ускладнюватиме оновлення ваг та уповільнюватиме збіжність алгоритму.

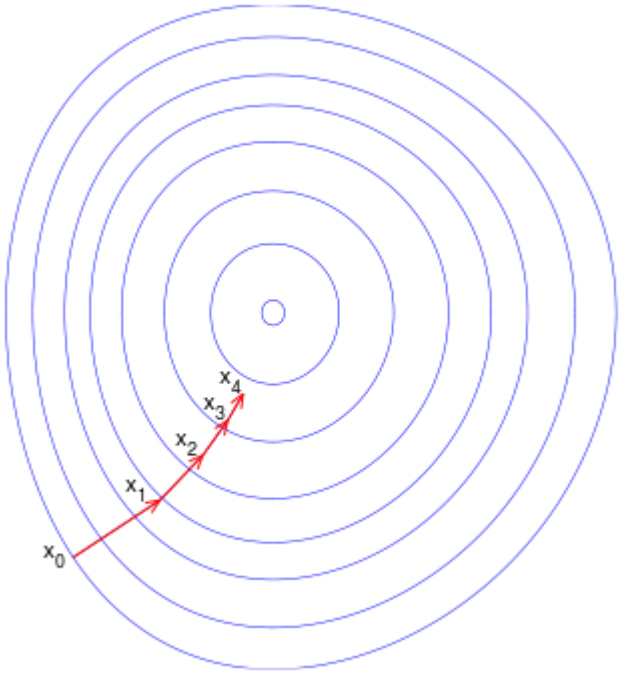

Рис.2.1 Приклад методу градієнтного спуску на ряду множин рівнів

2.4 Приклади використання нейронних мереж:

Давайте біль детально розглянемо деякі задачі для яких використовуються нейронні мережі.

В наш час однією із використовуваних нейронних мереж є нейронна мережа reCAPTCHA. reCAPTCHA це система, яка використовується для розрізняння живих користувачів від ботів. Вона заснована на нейронній мережі, яка навчається на величезному наборі даних зображень та тексту які надають люди.

Основними принципами роботи є:

1) Визначення CAPTCHA: Користувачу показують CAPTCHA, яка може містити:

Зображення: Розмиті, спотворені або з шумом.

Текст: Складний для читання або з помилками.

2) Введення інформації: Користувачу пропонують ввести текст із зображення, тексту, який він бачить, або обрати декілька зображень на яких є певний предмет(наприклад пожежний гідрант, світлофор тощо).

3) Надсилання відповіді: Відповідь користувача надсилається на сервери reCAPTCHA.

4) Аналіз за допомогою нейронної мережі: Нейронна мережа reCAPTCHA аналізує відповідь:

- Навчання: Мережа навчається на наборі даних який містить:
  - Приклади тексту та зображень, наданих людьми.
  - Приклади тексту та зображень, створених ботами.
- Визначення типу: Мережа визначає, чи є відповідь ймовірно людською.

5) Прийняття рішення:

- Людська відповідь: Якщо відповідь ймовірно людська, користувач вважається людиною і йому надається доступ до відповідного ресурсу.
- Відповідь бота: Якщо відповідь ймовірно не людська, для користувача:
  - Пропонується виконати додаткове завдання CAPTCHA.
  - Може бути відмовлено у доступі до ресурсу.

Переваги нейронної мережі reCAPTCHA:

- Стійкість до ботів: Нейронна мережа більш стійка до ботів, ніж традиційні CAPTCHA на основі правил.
- Простота використання: reCAPTCHA часто легше розгадати, ніж традиційні CAPTCHA, які можуть бути складними для читання і розуміння.
- Безперервне навчання: Нейронна мережа постійно навчається на новому наборі даних, що робить її більш стійкою до нових ботів.

Недоліки нейронної мережі reCAPTCHA

- Упередженість: нейронна мережа може бути упередженою щодо певних груп людей, якщо набір даних не є репрезентативним.

- Неточність: Нейронна мережа не є досконалою і іноді може помилково ідентифікувати людей як ботів.
- Незручність для людей з обмеженими можливостями: reCAPTCHA може бути складною для людей з вадами зору або слуху.

Детальний огляд компонентів:

- Нейронна мережа:
  - Тип: Багатошарова штучна нейронна мережа зі зворотним зв'язком.
  - Алгоритм навчання: Зазвичай використовується алгоритм зворотного поширення.
  - Функції активації: Зазвичай використовується ReLU або Sigmoid.
- Набір даних:
  - Містить зображення та текст, надані людьми та ботами.
  - Постійно оновлюється для включення нових даних.
  - Збалансований для представлення різних груп людей.
- CAPTCHA:

Можуть використовуватися різні типи CAPTCHA:

- Текстова CAPTCHA: Користувача просять ввести текст із зображення.
- Зображення CAPTCHA: Користувача просять вибрати зображення, що містять певний об'єкт.
- Аудіо CAPTCHA: Користувача просять ввести текст, який він чує.
- Оцінка ризику:
  - Нейронна мережа оцінює ризик того, що користувач є ботом, на основі його відповіді на CAPTCHA.
  - На оцінку ризику можуть впливати кілька факторів:
    - Якість відповіді користувача.
    - Історія поведінки користувача.
    - Інформація про пристрій та браузер користувача.

Використання нейронних мереж для розпізнавання рукописних цифр:

Зазвичай людина легко зможе розпізнати рукописні цифри, але це не такий легкий процес для нейронних мереж, оскільки людський мозок який пов’язаний із зоровою корою еволюціював протягом мільйонів років, і ця задача виконується на підсвідомому рівні. Але якщо це поручити комп’ютеру то це виявиться не такою ж і легкою задачею. Людське розуміння правопису цифр не так вже й легко виразити у вигляді алгоритмів. Майк Нільсен у своїй книзі “Neural Networks and Deep Learning” детально описав процес створення та навчання нейронної мережі для розпізнавання рукописних цифр. У цій книзі він визначив дві під-проблеми: перша - це винахід способу розбивання зображення на якому міститься певна кількість цифр на послідовність окремих зображень, а друга це класифікація окремих цифр. Перша під-проблема може вирішитися якщо ви маєте досить гарний спосіб класифікації окремих цифр.

Для розпізнавання рукописів автор використовував ось таку тришарову нейронну мережу:

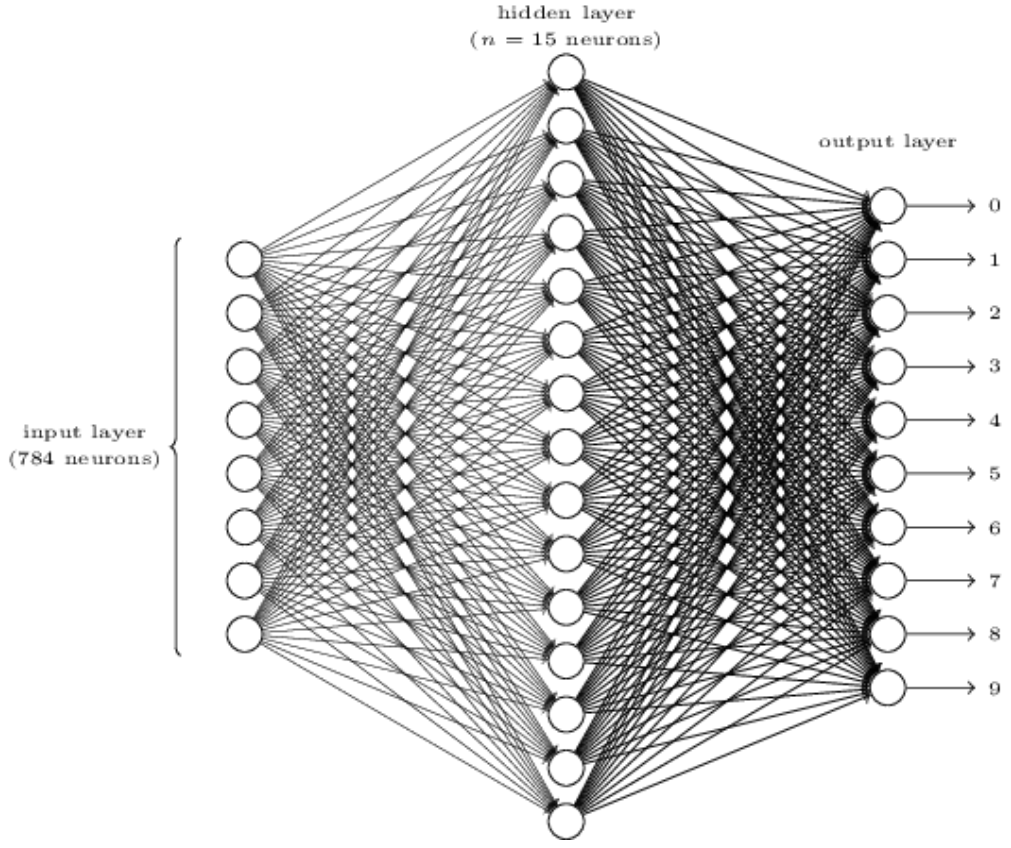

Рис.2.2 Схема тришарової нейронної мережі

Він зазначає, що вхідний шар мережі містить нейрони, які кодують значення вхідних пікселів. Навчальні дані для мережі будуть складатися з безлічі

зображень відсканованих рукописних цифр розміром 28 на 28 пікселів, і тому вхідний шар містить 784= 28×28 нейронів. Для простоти Майк Нільсен опустив більшість з 784 вхідних нейронів на схемі вище. Вхідні пікселі мають відтінки сірого, зі значенням 0.0 представляє білий колір, значення 1.0 - чорний, а проміжні значення представляють відтінки сірого, що поступово темнішають. Другий шар - прихований, і містить тільки n = 15 нейронів. Третій шар, тобто вихідний вміщає 10 нейронів(цифри від 0 до 9).

Також автор навчав свою нейронну мережу за методом який вже описувався в цій роботі, а саме метод градієнтного спуску. Для набору даних він використав набір даних MNIST, який має десятки тисяч відсканованих рукописних цифр разом із їх правильною класифікацією. Ось приклад цифр які обрав Майкл:

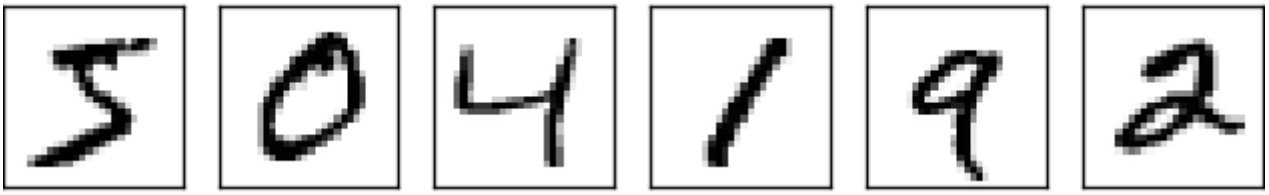

Рис.2.3 Обрані рукописні цифри для навчення нейронної мережі

Дані складалися із 2 частин: Перша частина - це відскановані зображення(60 тис.) почерку 250 людей(де половина це співробітники бюро перепису населення, а інша половина учні середньої школи), які використовувались як навчальні дані, а друга частина це тестові зображення(10 тис.) 28 на 28 у сірих відтінках, ці зображення використовуються для оцінки навичок розпізнавання рукописних цифр. Для позначення навчального введення використовується *x*. Для зручності *x* розглядається як 28*28 = 784 - комірний вектор, бажаний вихід це p= y(x), де y це 10-ти мірний вектор. Далі визначається формула витрат, яка виглядає наступним чином: $C(w,b) \equiv \frac{1}{2n}\sum_x \| p(x) - a \|^2$. Де *w* позначає сукупність усіх ваг мережі, *b* всі упередження, *p* загальна кількість входів, *a* - це вектор виходів з мережі при *x* - вхідні, а сума всіх навчальних вхідних даних *x*. В подальших формах автор замінив *w* і *b* на *v*. Припустивши мінімізацію функції *C(v)*. Це може бути будь яка дійсна функція

багатої кількості змінних $v = v1,v2,...$ Мінімізувавши $C(v)$ допомагає уявити C, як функцію лише двох змінних $v1,v2$ яе це зображено на зображені:

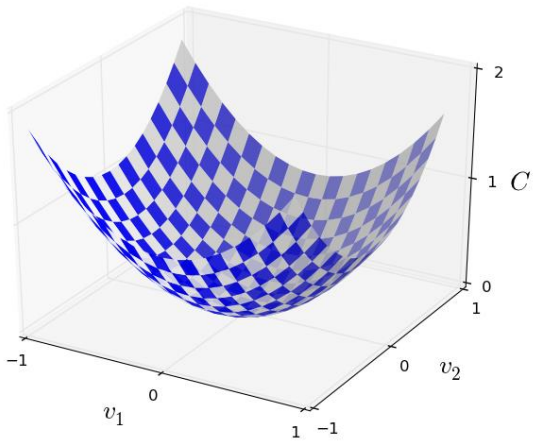

Рис.2.4 Схема мінімізації функції $C(v)$

Одним із способів розв'язку це використання обчислень для знаходження мінімуму. Але це не досить раціонально оскільки може з'явитись дуже багато змінних. Тому для цього автор використав наступну формулу: $\Delta C \approx \partial C \partial v1 \Delta v1 + \partial C \partial v2 \Delta v2$. Далі потрібно знайти спосіб для вибору $\Delta v1$ і $\Delta v2$, щоб ΔC ставав від'ємним. Для розуміння як робити такий вибір допоможе визначити Δv як вектор змін у v, $\Delta v \equiv (\Delta v1, \Delta v2)^T$, де T це операція транспортування, яка перетворює вектори-рядки на вектори-рядки. Також позначивши градієнт C як вектор часткових похідних $(\partial C/\partial v1, \partial C/\partial v2)^T$. В результаті отримали формулу $\nabla C \equiv (\partial C \partial v1, \partial C \partial v2)^T$. Далі знаходяться вагові коефіцієнти за допомогою градієнтного спуску w_k і упередження b_l які мінімізують вартість у рівнянні. Записавши правило оновлення градієнтного спуску в термінах компонентів отрмали: $w_k \to w'_k = w_k - \eta * (\partial C/\partial w_k)$ та $b_l \to b'_l = b_l - \eta * (\partial C/\partial b_l)$. Користуючись цим правилом можна сподіватись знайти мінімум функції вартості. Ця функція витрат має вигляд $C = 1n \sum x C_x$, тобто це середнє за витратами $C_x \equiv // y(x) - a // 22$ для окремих навчальних прикладів. На практиці, щоб обчислити градієнт ∇C потрібно обчислити градієнти ∇C_x окремо для кожного навчального входу x а потім зробити їх середніми, $\nabla C = 1n \sum x \nabla C_x$.

Також для прискорення навчання можна використати ідею стохастичного градієнтного спуску щоб пришвидшити навчання пропонується оцінювати градієнт функції втрат ∇C шляхом обчислення ∇C_x для невеликої випадкової вибірки навчальних даних. Отримані значення ∇C_x потім усереднюються, щоб отримати наближення справжнього градієнта ∇C. Цей метод дозволяє значно пришвидшити процес навчання.

Також давайте розглянемо приклад як працював би процес блокування банківської картки при неправильному вводі пін-коду у вигляді нейронної мережі. У нас буде два входи, один на правильне введення пін-коду інший на неправильне. Після вводу правильно пін-коду нас буде одразу переправляти в меню банкомату. А після неправильного введення від нас будуть вимагати ввести його ще раз, і якщо ще раз неправильно то вимагатиме ще раз, а якщо на третій раз неправильно ввести то картка заблокується. Розглянемо просту схему для повторення цієї дії :

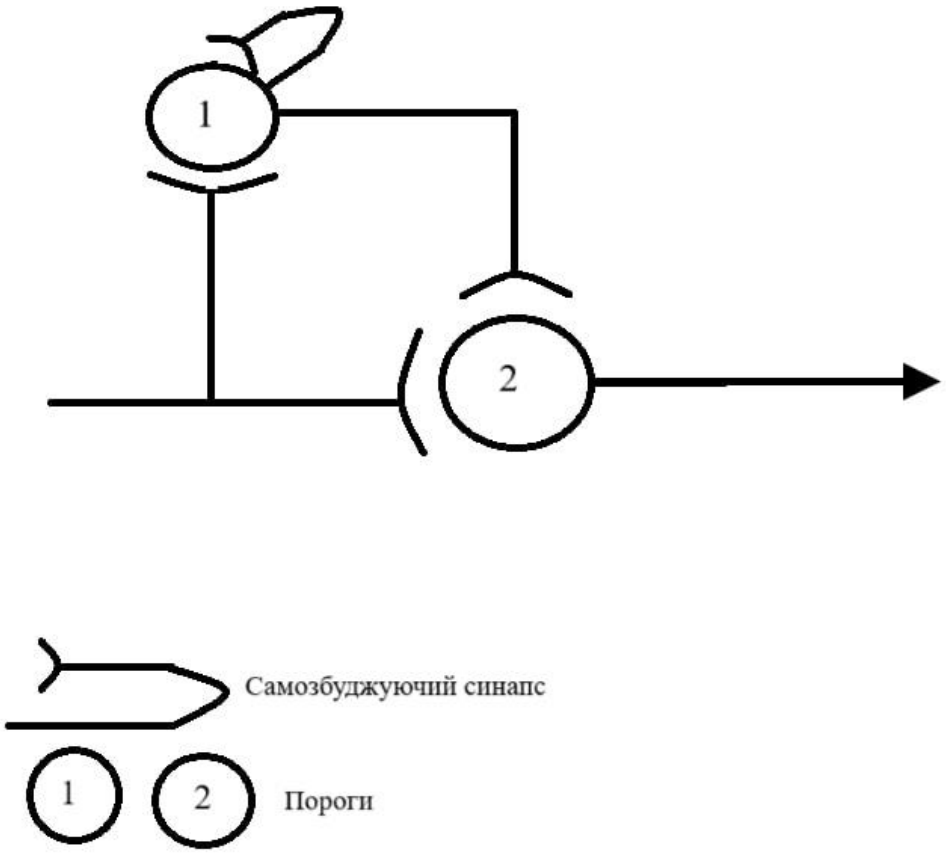

Рис.2.5 Схема із зворотнім колотералем

Під порогами ми вважаємо мінімальний сумарний сигнал в наслідок якого нейрон буде збудженим. Вважаємо за замовчуванням що вихідний сигнал нейрона дорівнюватиме 1. В цій схемі сигнал спочатку йде до нейрону зі значенням 1 він сам себе збуджує і далі посилає сигнал до нейрону зі значенням 2, але оскільки ми отримали лише 1 сигнал з нейрону зі значенням 2 інформація далі не поступатиме. Тільки після того як ми отримаємо новий сигнал який одразу піде до нейрону з порогом 2 інформація з цього нейрону піде далі. Отож, давайте схематично виразимо весь алгоритм роботи банкомату при неправильному вводі пін-коду:

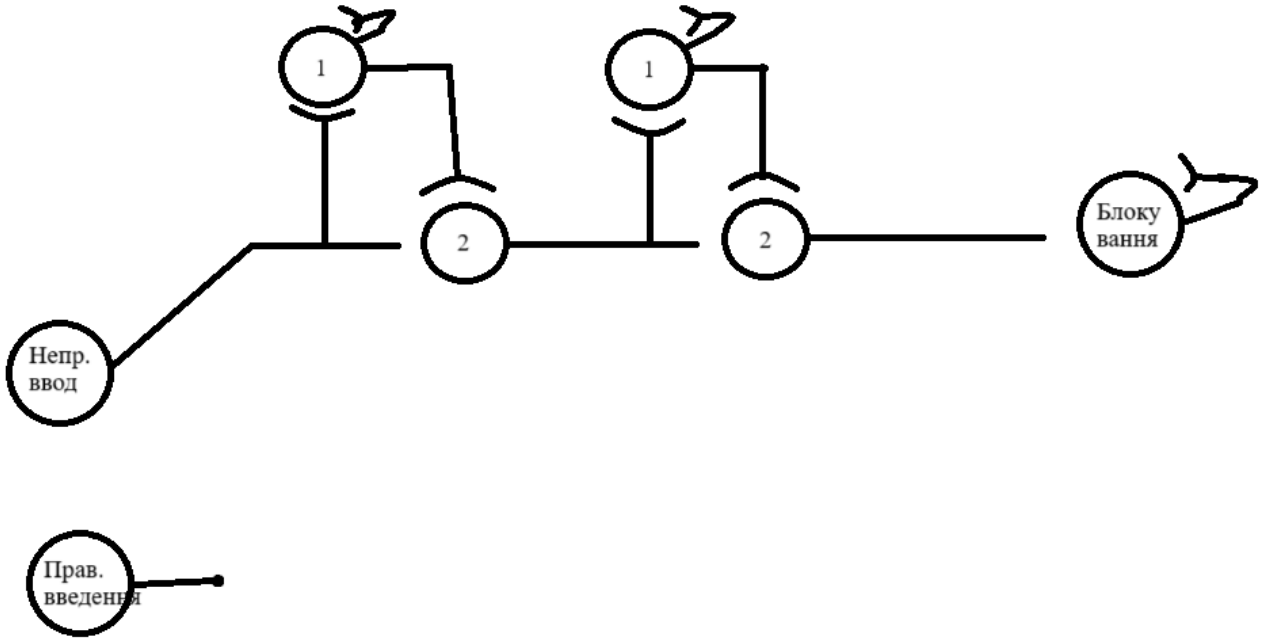

Рис. 2.6 Схема алгоритму роботи нейронної мережі при неправильному вводі пін-коду

Згідно схемі спочатку йде один сигнал після введення неправильного пін-коду до нейрону із порогом 1, який потім сам себе збуджує, після другої невдалої спроби введення коду сигнал йде до нейрону із порогом 2 і знову переходить до нейрону із порогом 1, який знову ж таки сам себе збуджує. Вже після третього

невдалого вводу пін-коду сигнал йде до нейрону із порогом 2 і одразу до нейрону із блокуванням картки. Водночас із блокуванням картки блокуються всі вихідні сигнали від нейрону правильного введення пін-коду.

А тепер давайте поглянемо як буде виглядати весь алгоритм роботи цієї нейронної мережі:

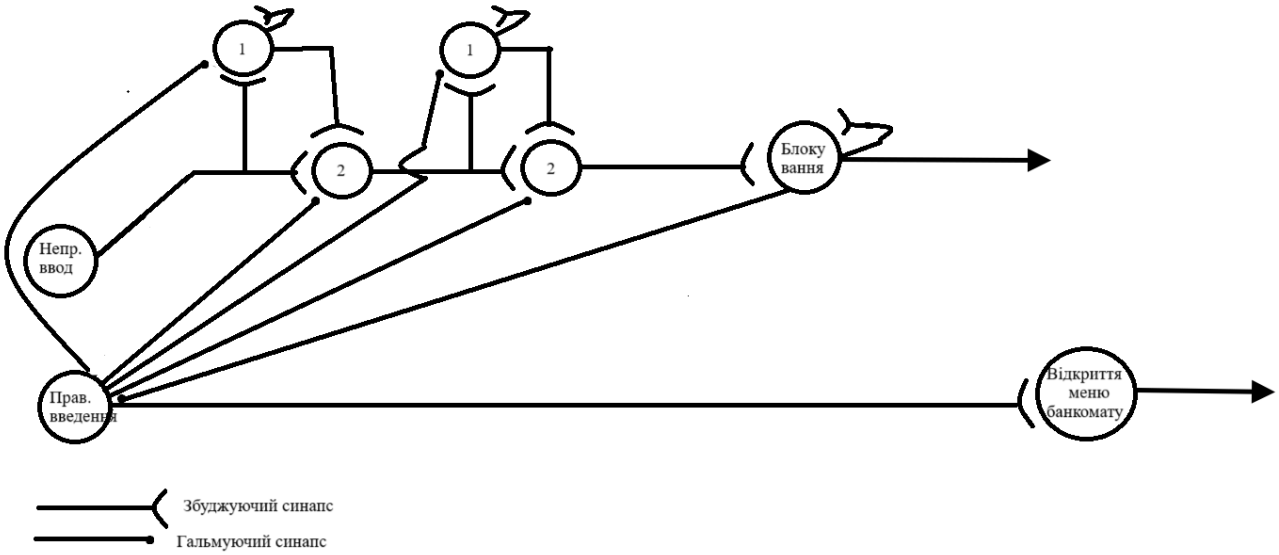

Рис.2.7 Схема повного алгоритму роботи банкомату у вигляді нейронної мережі

Виходячи зі схеми можна розказати цей повний алгоритм роботи. Якщо в нас пін-код введено правильно сигнал йде до нейрону який буде відкривати меню банкомату. Якщо в нас неправильно введено пін-код, то йде сигнал до нейрону із порогом 1 і він сам себе збуджує, але якщо з другої спроби пін-код ввели правильно, то сигнал із нейрону правильного введення буде блокувати вхідний сигнал до нейрону із порогом 2 і сигнал вже не йде до другого нейрону із порогом 1, така ж сама ситуація і для наступних неправильних введень

3 РЕАЛІЗАЦІЯ ОДНОНЕЙРОННОЇ МЕРЕЖІ ДЛЯ ЗАДАЧІ З ФІЛЬТРУВАННЯ ЕЛЕКТРОННИХ ЛИСТІВ.

У цьому розділі розглянемо приклад з використанням однонейронної мережі для фільтрації електронних листів(спаму) за допомогою JupyterLab. Для навчання цієї нейронної мережі ми будемо використовувати метод найменших квадратів, що вона мала змогу розрізняти спам та легітимні електронні листи. Для початку пояснімо, що таке однонейронна мережа та метод найменших квадратів. Однонейронна мережа - це найпростіша мережа, яка складається з одного нейрону з одним або декількома вхідними даними та одним вихідним значенням. Її можна буде використовувати для вирішення задач класифікації, якої є фільтрація електронних листів на предмет спаму. Важливість фільтрації спаму:

- Захист від небажаної пошти: Спам може захаращувати поштові скриньки користувачів, зменшуючи їх продуктивність та заважаючи їм знаходити важливі повідомлення.
- Захист від шкідливого контенту: Спам часто містить шкідливі посилання, які можуть призвести до зараження комп'ютера вірусами або фішингових атак.
- Збереження конфіденційності: Спам може містити особисту інформацію або фінансові дані, які можуть бути використані для шахрайства або крадіжки ідентичності.
- Економія ресурсів: Фільтрація спаму може допомогти зменшити навантаження на сервери електронної пошти та знизити витрати на Інтернет-трафік.

Методи фільтрації спаму:

- Фільтри на основі правил: Ці фільтри використовують правила для визначення того, чи є повідомлення спамом. Правила можуть бути засновані на

текстовому вмісті, заголовках, адресах, вкладеннях або інших характеристиках повідомлення.

◆ Статистичні фільтри: Ці фільтри використовують статистичні методи для аналізу тексту та інших характеристик повідомлення, щоб визначити, чи є воно спамом.

◆ Фільтри на основі машинного навчання: Ці фільтри використовують машинне навчання для автоматичного вивчення того, які характеристики повідомлень вказують на спам.

Структура однонейронної мережі складається з:

- Вхідний шар: Складається з одного нейрона який приймає вхідні дані. Вхідні дані можуть бути представлені у вигляді чисел, векторів або матриць.
- Нейрон: Це центральний елемент однонейронної мережі, який обробляє вхідні дані та генерує вихідні дані, нейрон складається з вагів, тобто, коефіцієнтів, які пов’язують вхідні дані з нейроном та активаційною функцією, яка перетворює суму зважених вхідних даних в вихідне значення нейрона.
- Вихідний шар: Складається з одного нейрона, який генерує вихідне значення однонейронної мережі.

Функціонування однонейронної мережі:

- Вхідні дані: Вхідні дані подаються на вхідні нейрони.
- Обчислення зважених сум: Кожен вхідний нейрон множить вхідне значення на відповідну вагу. Суми зважених вхідних даних з усіх нейронів обчислюються та підсумовуються.
- Активація: Сума зважених вхідних даних подається на активаційну функцію нейрона. Активаційна функція перетворює цю суму в вихідне значення нейрона.
- Вихідне значення: Вихідне значення нейрона є вихідним значенням однонейронної мережі.

Перевагами для однонейронних мереж є простота та легкість розуміння, швидкість навчання та ефективність для задач з малим обсягом даних.

Недоліками є обмежена здатність моделювання складних нелінійних зв’язків, чутливість до шуму та необхідність ретельного підбору гіперпараметрів.

Метод найменших квадратів - це алгоритм машинного навчання який використовується для знаходження оптимальних значень параметрів моделі. Метод найменших квадратів прагне знайти такі значення параметрів p, які мінімізують функцію помилки E(p): $E(p) = \sum(y_i - f(x_i, p))^2$. У випадку однонейронної мережі метод найменших квадратів використовується для визначення ваги з’єднання між вхідними даними та нейроном.

Для прикладу ми взяли ситуацію де в нас 8 ситуацій і також є 4 критерії на яких базуватиметься рішення щодо відправлення того чи іншого електронного листа до спаму, ці критерії ми виразимо як *x1, x2, x3, x4*, а рішення відправляти у спам чи ні на основі отриманих даних з цих критеріїв як *y*. Розпишемо це все у вигляді таблиці.

Таблиця 3.1

Чи знайомий відправник **x1**	Вкладення із підозрілим файлом **x2**	Чи на явні посилання **x3**	Чи академічна адреса **x4**	Чи переносити до спаму **y**
1	1	1	0	0
0	1	1	1	0
0	0	0	1	0
0	1	1	0	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
1	1	1	1	0

Для легшого розуміння ми позначили відповіді так через 1, а відповідь ні як 0.

У подальшому ми використовуватимемо ці дані у форматі матриці, для навчання нашої однонейронної мережі.

Для побудови нейрона використаємо формулу неоднорідного адаптивного суматора:

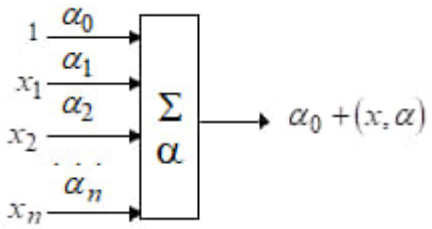

Рис.3.1 Адаптивний неоднорідний суматор

Отже для того щоб зробити матрицю додамо до нашої вже існуючої таблиці значення *x0*.

Таблиця 3.2

x0	**x1**	**x2**	**x3**	**x4**	**y**
1	1	1	1	0	0
1	0	1	1	1	0
1	0	0	0	1	0
1	0	1	1	0	1
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	1	1	1	1	0

Для матриці(X) використовуватимемо лише *x*:

1	1	1	1	0
1	0	1	1	1
1	0	0	0	1
1	0	1	1	0
1	0	0	1	0
1	0	0	1	1

1	0	1	0	0
1	1	1	1	1

Тепер нам потрібно цю матрицю транспонувати(X^t), після цього вона матиме такий вигляд:

1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1
1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	1
0	1	1	0	0	1	0	1

Далі отримаємо матрицю коефіцієнтів нормальних рівнянь(X^t*X) у результаті множення початкової матриці на транспоновану матриці, і вона матиме такий вигляд:

8	2	5	6	4
2	2	2	2	1
5	2	5	4	2
6	2	4	6	3
4	1	2	3	4

Також нам потрібно отримати стовпчик вільних членів(X^t*Y) шляхом множення транспонованої матриці на *y*, в результаті буде:

3
0
2
2
0

Реалізація цих кроків у застосунку JupyterLab:

```
[1]: import numpy as np

[2]: A = np.array([[1,1,1,1,0], [1,0,1,1,1], [1,0,0,0,1], [1,0,1,1,0], [1,0,0,1,0], [1,0,0,1,1], [1,0,1,0,0], [1,1,1,1,1]])

[3]: AT = A.T

[4]: print(AT)

     [[1 1 1 1 1 1 1 1]
      [1 0 0 0 0 0 0 1]
      [1 1 0 1 0 0 1 1]
      [1 1 0 1 1 1 0 1]
      [0 1 1 0 0 1 0 1]]

[5]: XtX = AT @ A

[6]: print(XtX)

     [[8 2 5 6 4]
      [2 2 2 2 1]
      [5 2 5 4 2]
      [6 2 4 6 3]
      [4 1 2 3 4]]

[7]: B = np.array ([0,0,0,1,1,0,1,0])

[8]: XtY = AT @ B

[9]: print(XtY)

     [3 0 2 2 0]
```

Рис.3.2 Операції із матрицями у JupyterLab

Далі нам потрібно отримати коефіцієнти вхідних даних для *x0,…,x4* помноживши обернену матрицю на стовпчик вільних членів і отримаємо:

$$X0 = 0{,}818181818$$

$$X1 = -0{,}545454545$$

$$X2 = 0{,}090909091$$

$$X3 = 0$$

$$X4 = -0{,}727272727$$

Виконання цього процесу в JupyterLab:

```
[10]: def inverse_matrix_method(A1, b):
          try:
              inv_A1 = np.linalg.inv(A1)
          except np.linalg.LinAlgError:
              print("Матриця не має оберненої!")
              return None
          x = inv_A1 @ b
          return x

[11]: A1 = XtX

[12]: b = XtY

[13]: x = inverse_matrix_method(A1, b)

[14]: if x is not None:
          print(f"Відповідь: {x}")

      Відповідь: [ 0.81818182 -0.54545455  0.09090909  0.         -0.72727273]
```

Рис. 3.3 Результат виконання множення оберненої матриці на стовпчик вільних чисел у JupyterLab

Також можна схематично відобразити як отримані результати взаємодіють із нейроном, а саме:

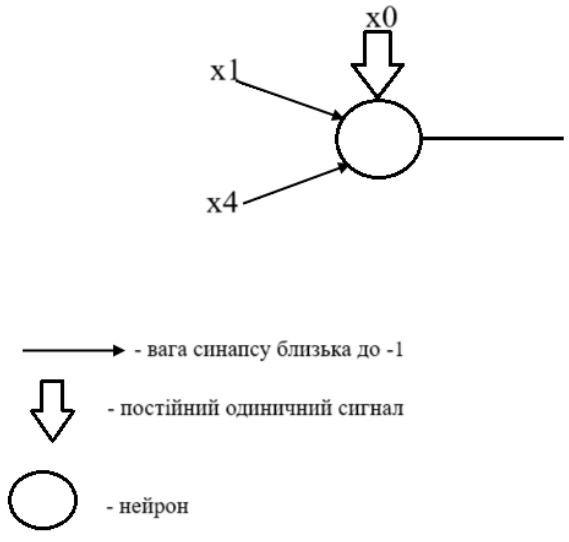

Рис.3.4 Схема взаємодії вхідних даних та нейрону

Після цього знайдемо теоретичні значення(yt), після підставлення добутку коефіцієнтів вхідних даних на матрицю X і отримаємо такий результат:

0,363636364

0,181818182

0,090909091

0,909090909

0,818181818

0,090909091

0,909090909

-0,363636364

В результаті всіх цих операцій ми отримали приблизні значення для нашої нейронної мережі, тобто якщо число ближче то 1 то ми цей лист відправляємо у спам а якщо ближче до 0 то з листом все добре і відправлення у спам для нього не потребуються.

Виконання процесу знаходження теоретичних даних у JupyterLab:

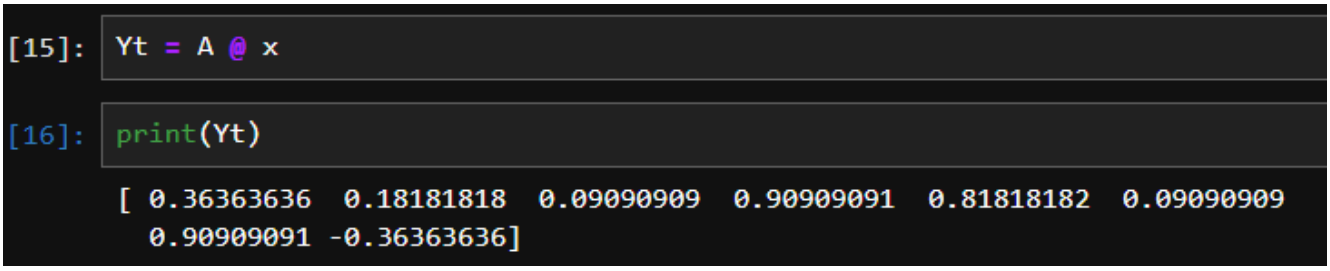

Рис. 3.5 Знаходження yt в JupyterLab

Отже на основі проведених операцій можна визначити сильні та слабкі сторони однонейронних мереж в задачі фільтрації електронних листів. Сильними сторонами є:

Простота та легкість реалізації: Однонейронні мережі – це прості нейронні мережі, які легко реалізувати та навчати. Це робить їх хорошим вибором для задач, де потрібна швидка та проста модель.

Ефективність для задач з невеликими наборами даних: Однонейронні можуть бути ефективними для задач з невеликими наборами даних, де складніші моделі можуть бути перенавчання.

Інтерпретованість: Однонейронні мережі мають просту структуру, що робить їх більш інтерпретованими, ніж складніші нейронні мережі. Це може бути корисно для розуміння того, як модель приймає рішення.

Слабкі сторони використання однонейронних мереж:

Обмежена здатність моделювати складні нелінійні зв'язки: однонейронні мережі не здатні моделювати складні нелінійні зв'язки між даними, що може обмежувати їх ефективність в деяких задачах.

Чутливість до шуму: однонейронні мережі можуть бути чутливими до шуму в даних, що може призвести до неточних результатів.

Необхідність ретельного підбору гіперпараметрів: Ефективність однонейронної сильно залежить від вибору гіперпараметрів, таких як швидкість навчання та вагова ініціалізація. Це може бути складним завданням і потребувати знань про машинне навчання.

Давайте схематично виразимо альтернативний алгоритм роботи нейронної мережі для фільтрації електронних листів:

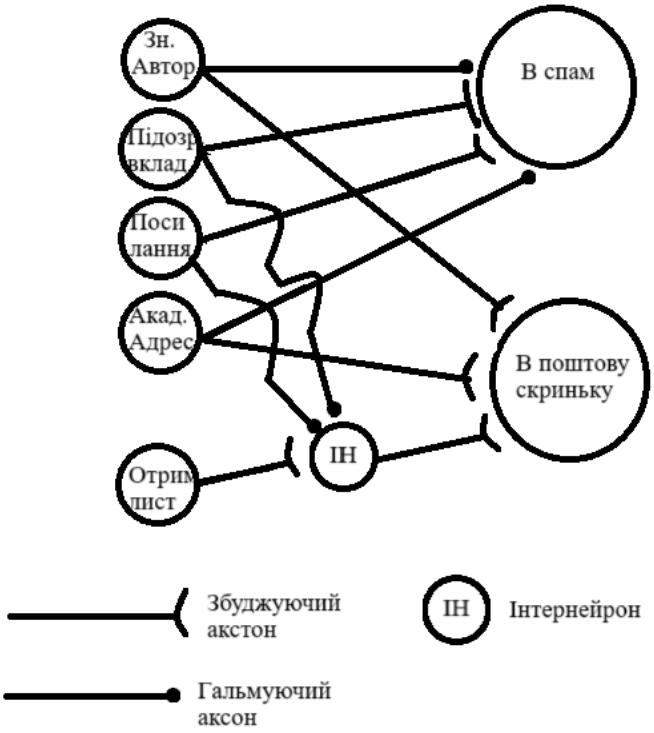

Рис. 3.6 Схема роботи алгоритму однонейронної мережі для фільтрування електронних листів

Виходячи зі схеми ми отримаємо, що отримавши лист інформація йде до інтернейрона та активує його, потім зважаючи на наші критерії сигнал або блокує передачу інтернейрону або навпаки передає інформацію до нейрону який відправляє листа до поштової скриньки. Також якщо ми знаємо автора листа та/або у відправника академічна адреса то в такому випадку ці нейрони будуть збуджувати нейрон для відправки цього листа до скриньки, і якщо нам не відомий відправник та він не має академічної адреси, але до листа прикріплене підозріле вкладення та/або посилання в такому випадку активується нейрон який відправляє цей лист до спаму.

ВИСНОВОК

В результаті виконання бакалаврської роботи було проведено комплексне дослідження застосування нейронних мереж для захисту інформації в інформаційно-технічних системах в процесі якого було наведено декілька прикладів використання нейронних мереж в засобах захисту інформації, описані різновиди архітектур нейронних мереж та наведено конкретні приклади їх використання в інформаційно-технічних системах. Також описано структуру та функціонування однонейронних мереж, включаючи активаційні функції, пояснено метод найменших квадратів який був використаний для навчання нейронної мережі, його математичну основу та застосування для однонейронних мереж. Розглянуто задачу з фільтрації спаму, її важливість та методи фільтрації.

Нейронні мережі мають значний потенціал для покращення захисту інформації в інформаційно-технічних системах. Їх здатність до навчання на великих обсягах інформації та їх можливість до виявлення складних закономірностей робить їх ідеальними інструментами для виявлення та запобігання кіберзагрозам. Для покращення та подальшого розвитку нейронних мереж в інформаційно-технічних системах розглянемо декілька рекомендацій:

1. Покращення стійкості до атак: Розробити методи навчання нейронних мереж, які роблять їх стійкими до атак спрямованих на дискредитацію інформаційно-технічної системи.

2. Підвищення продуктивності та масштабованості: Розробка нових алгоритмів навчання нейронних мереж, які будуть більш ефективними з точки зору обчислювальних ресурсів та зможуть обробляти більш великі обсяги даних в режимі реального часу.

3. Інтеграція нейронних мереж з іншими системами кібербезпеки: Розробити методи інтеграції нейронних мереж з іншими системами кібербезпеки, такими як система виявлення вторгнень та система запобігання вторгнень, для покращення загальної ефективності кіберзахисту, наприклад, нейронні мережі можна використовувати для аналізу трафіку мережі та журналів

систем виявлення вторгнень та запобігання вторгненням для виявлення аномалій та підозрілої активності, також можна використовувати нейронні мережі для навчання на даних системи виявлення вторгнень та системи запобігання вторгненням, для того щоб покращити їх здатність до виявлення нових та невідомих загроз.

На практиці продемонстровано процес навчання однонейронної мережі за допомогою методу найменших квадратів на тестовому наборі ситуацій з певними критеріями, оцінено ефективність нейронної мережі, проведено аналіз та визначені сильні та слабкі сторони використання однонейронних мереж для задач з фільтрації електронних листів.

Результати роботи демонструють перспективність використання нейронних мереж, зокрема розглянутих на прикладі однонейронних мереж, в засобах захисту інформації. Використаний метод найменших квадратів для навчання нейронної мережі показав свою ефективність в задачах фільтрації спаму.

ПЕРЕЛІК ПОСИЛАНЬ

1. Штучна нейронна мережа URL: https://uk.wikipedia.org/wiki/%D0%A8%D1%82%D1%83%D1%87%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0#%D0%A8%D1%82%D1%83%D1%87%D0%BD%D1%96_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%B8
2. Згорткова нейронна мережа URL: https://uk.wikipedia.org/wiki/%D0%97%D0%B3%D0%BE%D1%80%D1%82%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0
3. Рекурентна нейронна мережа URL: https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D0%B5%D0%BD%D1%82%D0%BD%D0%B0_%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0
4. Нейронна мережа прямого поширення URL: https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0_%D0%BF%D1%80%D1%8F%D0%BC%D0%BE%D0%B3%D0%BE_%D0%BF%D0%BE%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8F
5. Методи навчання нейронних мереж URL: https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D0%BE_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F_%D0%A8%D0%9D%D0%9C
6. Machine Learning and Deep Learning Methods for Cybersecurity URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8359287
7. Градієнтний спуск: алгоритм та приклад на Python URL: https://robotdreams.cc/uk/blog/331-gradiyentniy-spusk-algoritm-ta-priklad-na-python

8. reCAPTCHA URL: https://uk.wikipedia.org/wiki/ReCAPTCHA
9. Neural Networks and Deep Learning, Michael Nielsen, 2019. URL: http://neuralnetworksanddeeplearning.com/index.html
10. Метод найменших квадратів URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BD%D0%B0%D0%B9%D0%BC%D0%B5%D0%BD%D1%88%D0%B8%D1%85_%D0%BA%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D1%96%D0%B2
11. Simon S. Haykin Neural Networks: A Comprehensive Foundation 2Nd Ed, Prentice-Hall Of India Pvt. Limited, 1999

ДОДАТОК

```
import numpy as np
A = np.array([[1,1,1,1,0], [1,0,1,1,1], [1,0,0,0,1], [1,0,1,1,0],
[1,0,0,1,0], [1,0,0,1,1], [1,0,1,0,0], [1,1,1,1,1]])
AT = A.T
print(AT)
XtX = AT @ A
print(XtX)
B = np.array ([0,0,0,1,1,0,1,0])
XtY = AT @ B
print(XtY)
def inverse_matrix_method(A1, b):
    try:
        inv_A1 = np.linalg.inv(A1)
    except np.linalg.LinAlgError:
        print("Матриця не має оберненої!")
        return None
    x = inv_A1 @ b
return x
A1 = XtX
b = XtY
x = inverse_matrix_method(A1, b)
if x is not None:
print(f"Відповідь: {x}")
Yt = A @ x
print(Yt)
```