МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ І.С. Скарга-Бандурова
«____»_____ 20__ р.

ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА

ПОЯСНЮВАЛЬНА ЗАПИСКА

НА ТЕМУ:

Інформаційна система класифікації повідомлень електронної пошти

Освітній рівень"бакалавр"
Спеціальність 123 "Комп'ютерна інженерії"

Науковий керівник роботи: _____ (підпис) Г.Ф.Кривуля (ініціали, прізвище)

Консультант з охорони праці: _____ (підпис) Я.О.Критська (ініціали, прізвище)

Здобувач вищої освіти: _____ (підпис) Н.А.Якимчук (ініціали, прізвище)

Група: КІ-16з

Сєвєродонецьк 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітній рівень Бакалавр
Спеціальність 123 "Комп'ютерна інженерія"
(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри _____
_____ С.О. Сафонова
«_____» _____20____р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА

Якимчук Наталії Андріївни
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система класифікації повідомлень електронної пошти

керівник проекту (роботи) Кривуля Геннадій Федорович, д.т.н., проф.
(прізвище, м.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «30» 04 2020 р. № 77/15.15
2. Строк подання студентом роботи 10.06.2020
3. Вихідні дані до роботи Матеріали переддипломної практики, теоретичні відомості про нейронні мережі, методи навчання нейронних мереж, опис технологій розробки, програмні засоби Java

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Актуальність нейронних мереж і технологій та методології щодо розробки застосунків, математична модель нейронної мережі та розпізнавання об'єкта методами машинного навчання, комп'ютерна модель реалізації методу класифікації повідомлень, охорона праці,висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 30.04.2020

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	01.05.2020-07.05.2020	
2	Аналіз завдання, огляд літератури	08.05.2020-10.05.2020	
3	Аналіз технічних засобів та розробка алгоритму	10.05.2020-13.05.2020	
4	Розробка частини проекту "Охорона праці"	13.05.2020-15.05.2020	
5	Реалізація системи	15.05.2020-01.06.2020	
6	Оформлення пояснювальної записки та презентації	2.06.2020-09.06.2020	

Здобувач вищої освіти _____ Н.А.Якимчук
(підпис) (прізвище та ініціали)

Науковий керівник _____ Г.Ф. Кривуляс
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка дипломної роботи містить: 80 с., 12 рис., 2 табл., 1 додаток, 28 джерел.

Робота присвячена вирішенню проблеми класифікації поштових повідомлень. Розроблено методи, які дозволяють проводити класифікацію повідомлень, які отримані з поштового сервісу. Повідомлення потрапляють до сервера на аналіз за допомогою мережевому протоколу прикладного рівня для доступу до електронної пошти – IMAP. Суть методів полягає в зчитуванні, обробці та розумінню мови, якою написане повідомлення. Розроблені методи дозволяють знаходити у повідомленнях заборонені слова та фрази, які будуть відправлені до «спаму». Знаходження заборонених слів та фраз бере на себе штучна нейронна мережа, яка використовує парадигму навчання: «з вчителем». Нейронна мережа має в своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги настроюються так, щоб мережа проводила відповіді якомога ближчі до відомих правильних відповідей.

Розроблені алгоритми реалізації запропонованих методів використано для створення спеціалізованого програмного засобу класифікації повідомлень.

Ключові слова: java, штучний інтелект, класифікація повідомлень.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ЗНМ (CNN) – згорткова нейронна мережа

НМ – нейронна мережа

BN – Batch-нормалізація

ВСТУП

На сьогоднішній день проблема деяких поштових сервісів значно помітна в плані сортування між «спамом» та «білим списком», бо як тільки не вигадають відправники, щоб обійти фільтр.

Реалізація мого фільтру опирається на використання штучної нейронної мережі, яка використовує парадигму навчання: «з вчителем». У якості вчителя використовується словник для навчання мережі, який має заборонені слова та фрази, які можна враховувати, як «спам».

Мій вибір пав на штучні нейронні мережі, тому що інтерес до них значно виріс за останні роки. Штучні нейронні мережі демонструють велику кількість властивостей нашого мозку. Вони навчаються, обчислюють попередні випадки на нові.

IMAP – служить для роботи з вхідними листами та забезпечує додаткові функції, наприклад, можливість пошуку зв ключовим словом без збереження пошти в локальній пам’яті. Поштовий клієнт, що використовує цей протокол, отримує доступ до сховища кореспонденції на сервер так, начебто вона розташована на комп’ютері одержувача. Електронними листами можна маніпулювати з комп’ютера користувача без постійного пересилання з сервера і назад файлів з повним змістом листів. Це дає мені змогу використати його для зчитування поштових повідомлень зі скриньки. Та подальшого аналізу вхідних повідомлень за допомогою штучної нейронної мережі.

У кінцевому результаті після аналізу повідомлення воно буде відправлено до «спаму» або залишено у переглянутому вигляді.

1 АКТУАЛЬНІСТЬ НЕЙРОННИХ МЕРЕЖ І ТЕХНОЛОГІЙ ТА МЕТОДОЛОГІЇ ЩОДО РОЗРОБКИ ЗАСТОСУНКІВ

1.1 Класифікація нейронних мереж

Нейронні мережі можна класифікувати за наступними показниками:

Таблиця 1.1 – Класифікація нейронних мереж

Характер навчання	Налаштування ваг	Тип вхідної інформації	Модель мережі
З вчителем Без вчителя З підкріпленням	Фіксоване Динамічне	Аналогова Двійкова	Прямого поширення Зворотного поширення (рекурентні) Самоорганізовані карти

Класифікація нейронних мереж за характером навчання:

– нейронні мережі, що використовують навчання з вчителем. Навчання з вчителем передбачає, що для кожного вхідного вектора існує цільовий вектор, що представляє необхідний вихід. Разом вони називаються навчальною парою. Зазвичай, мережа навчається на деякій множині таких навчальних пар. Пред'являється вихідний вектор, обчислюється вихід мережі і порівнюється з відповідним цільовим вектором. Далі ваги змінюються відповідно до алгоритму, що сприяє мінімізації похибки. Вектори навчальної множини пред'являються послідовно, обчислюються похибки і ваги підлаштовуються для кожного вектора, доки похибка по всій навчальній множині не досягне прийнятного рівня;

– нейронні мережі, що використовують навчання без вчителя. Навчання без вчителя є більш доречною моделлю навчання з точки зору

біологічних нейронних мереж. Таке навчання не потребує цільового вектору для виходів і, відповідно, не вимагає порівняння з відомими вірними відповідями. Навчальна множина складається лише з вхідних векторів. Навчальний алгоритм підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення достатньо близьких вхідних векторів надавало однакові виходи. Процес навчання виділяє статистичні властивості навчальної множини і групує подібні вектори у класи;

– нейронні мережі, що використовують навчання з підкріпленням. Навчання з підкріпленням – один із способів машинного навчання, в ході якого система навчається, взаємодіючи з певним середовищем. Відгуком середовища на прийняті рішення є сигнали підкріплення, тому таке навчання є окремим випадком навчання з вчителем, але вчителем є середовище або його модель. Деякі правила підкріплення базуються на неявних вчителів, через що їх можна віднести до навчання без вчителя.

Налаштування ваг:

– мережі з фіксованими зв'язками – вагові коефіцієнти нейронної мережі встановлюються певними значеннями одразу, виходячи з умов задачі;

– мережі з динамічними зв'язками – для них налаштування синаптичних ваг відбувається в процесі навчання.

Типи вхідної інформації:

– аналогова – вхідну інформацію представлено в формі дійсних чисел;

– двійкова – вся вхідна інформація в таких мережах представляється у вигляді нулів і одиниць.

Модель нейронної мережі:

– мережі прямого поширення – всі зв'язки і потік обробки скеровані від вхідних нейронів до вихідних. До таких мереж відносяться, наприклад: найпростіший перцептрон Розенблатта і багатошаровий перцептрон;

– мережі зворотного поширення (рекурентні) – сигнал з вихідних нейронів або нейронів прихованого прошарку частково передається назад на входи нейронів попереднього прошарку;

– саморганізовані карти або Мережі Кохонена – клас мереж, як правило, навчається без вчителя і успішно застосовується в задачах розпізнавання. Мережі такого класу здатні виявляти новизну у вхідних даних: якщо після навчання мережа зустрінеться з набором даних, несхожим з жодним відомим зразком, то вона не зможе класифікувати такий набір і тим самим виявить його новизну. Мережа Кохонена має лише два прошарки: вхідний і вихідний.

1.2 Типи нейронних мереж

Нейронні мережі прямого поширення (Feed Forward Neural Networks, FF або FFNN) і перцептрони (Perceptrons, P) є прямолінійними і передають інформацію від входу до виходу. Нейронні мережі, зазвичай, є багатошаровими (вхідний, приховані та вихідний прошарки). Нейрони одного прошарку не пов'язані між собою, а сусідні прошарки, зазвичай, повністю пов'язані.

Найпростіша нейронна мережа має два вхідних нейрони і один вихідний, і може використовуватися як модель логічного перемикання. FFNN, зазвичай, навчається за методом зворотного поширення похибки, в якому мережа отримує множини вхідних і вихідних даних. Похибка є різницею між отриманим значенням виходу та правильним.

Якщо у мережі є достатня кількість прихованих нейронів, вона теоретично здатна змоделювати взаємодію між вхідним і вихідними даними. Практично такі мережі використовуються рідко, але їх часто комбінують з іншими типами.

Архітектура FeedForward BackPropagation є популярною, ефективною та легкою моделлю навчання для складних, багатошарових мереж. Вона використовується у різних типах застосувань і породила великий клас

нейромереж з різними структурами та методами навчання. Мережа може моделювати функцію практично будь якої складності.

Типова мережа BackPropagation має вхідний прошарок, вихідний прошарок та принаймні один прихований прошарок. Теоретично, обмежень відносно кількості прихованих прошарків не існує, але практично застосовують один або два.

Нейрони організовані в пошарову структуру з прямою передачею сигналу. Кожний нейрон мережі продукує зважену суму своїх входів, пропускає цю величину через передатну функцію і видає вихідне значення.

Визначення числа проміжних прошарків і числа нейронів в них є важливим при моделюванні мережі. Більшість дослідників та інженерів, застосовуючи архітектуру до визначених проблем використовують загальні правила, зокрема:

– кількість входів та виходів мережі визначаються кількістю вхідних та вихідних параметрів досліджуваного об'єкту, явища, процесу, тощо. На відміну від зовнішніх прошарків, число нейронів прихованого прошарку $n_{прих}$ обирається емпіричним шляхом. В більшості випадків достатня кількість нейронів становить $n_{прих} \approx n_{вх} + n_{вих}$, де $n_{вх}$, $n_{вих}$ – кількість нейронів у вхідному і, відповідно, у вихідному прошарках;

– якщо складність у відношенні між отриманими та бажаними даними на виході збільшується, кількість нейронів прихованого прошарку повинна також збільшитись;

– якщо процес, що моделюється, може розділятись на багато етапів, потрібен додатковий прихований прошарок (прошарки). Якщо процес не розділяється на етапи, тоді додаткові прошарки можуть допустити перезапам'ятовування і, відповідно, невірне загальне рішення.

Після того, як визначено число прошарків і число нейронів в кожному з них, потрібно знайти значення для синаптичних ваг і порогів мережі, які спроможні мінімізувати похибку спродукованого результату. Саме для цього

існують алгоритми навчання, де відбувається підгонка моделі мережі до наявних навчальних даних.

Алгоритм діє ітеративне, його кроки називаються епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні приклади, вихідні значення мережі порівнюються з бажаними значеннями і обчислюється похибка. Значення похибки, а також градієнту поверхні станів використовують для корекції ваг, і дії повторюються. Процес навчання припиняється або коли пройдена визначена кількість епох, або коли похибка досягає визначеного рівня малості, або коли похибка перестає зменшуватись (користувач переважно сам вибирає потрібний критерій останову).

Нейронна мережа Хопфілда (Hopfield Network, HN) – це повнозв’язна нейронна мережа із симетричною матрицею зв’язків, що реалізує завдання асоціативної пам’яті. Деякий набір двійкових сигналів (зображень, звукових оцифровок, інших даних, що описують якийсь об’єкти або характеристики процесів), вважають зразковим. Мережа повинна вміти з 12агорткових сигналу, поданого на її вхід, виділити («пригадати» по частковій інформації) відповідний зразок або «дати висновок» про те, що вхідні дані не відповідають жодному із зразків.

Мережа Хопфілда використовує три прошарки: вхідний, прошарок Хопфілда та вихідний прошарок. Кожен прошарок має однакову кількість нейронів. Навчання мережі Хопфілда вимагає, щоб навчальний образ був представлений на вхідному та вихідному прошарках одночасно. Для правильного навчання мережі відповідні пари «вхід-вихід» мають відрізнятися між собою.

Під час отримання вхідних даних кожен вузол є входом, в процесі навчання він стає прихованим, а потім стає виходом.

Мережа навчається так: значення нейронів встановлюються відповідно до бажаного шаблону, після чого обчислюються ваги, які в подальшому не змінюються. Після того, як мережа навчилася на одному або кількох шаблонах, вона завжди буде зводитися до одного з них (але не завжди до

бажаного). Вона стабілізується в залежності від загальної «енергії» і «температури» мережі. В кожного нейрона є власний поріг активації, що залежить від температури, при проходженні якого нейрон приймає одне з двох значень (зазвичай -1 або 1, іноді 0 або 1). Така мережа часто називається мережею з асоціативної пам'яттю. Як людина, що бачить лише половину образу, може представити другу половину, так і ця мережа, маючи не повний образ, відновлює його до повного.

Якщо мережа Хопфілда використовується як пам'ять, що адресується за змістом вона має два головних обмеження:

– по-перше, число образів, що можуть бути збережені та точно відтворені є строго обмеженим. Якщо зберігається занадто багато параметрів, мережа може збігатись до нового неіснуючого образу, відмінному від всіх запрограмованих образів, або не збігатись взагалі. Межа ємності пам'яті для мережі приблизно 15% від числа нейронів у прошарку Хопфілда;

– другим обмеженням парадигми є те, що прошарок Хопфілда може стати нестабільним, якщо навчальні приклади є занадто подібними. Зразок образу вважається нестабільним, якщо він застосовується за нульовий час і мережа збігається до деякого іншого образу з навчальної множини. Ця проблема може бути вирішена вибором навчальних прикладів більш ортогональних між собою.

Спеціальний тип нейронної мережі, що дозволяє виробляти кластеризацію об'єктів. Мережа Кохонена складається з двох прошарків – вхідного і вихідного. Вихідний прошарок часто називається «прошарок Кохонена». При цьому кожен нейрон вхідного прошарку пов'язаний зі всіма нейронами вихідного, а всередині прошарків зв'язків немає. На нейрони вхідного шару подаються вектори ознак об'єктів, що кластеризуються.

Вхідні нейрони не беруть участі в процесі навчання і обробки даних, а просто розподіляють вхідний сигнал по нейронах вихідного шару. Число вхідних нейронів дорівнює розмірності вектора ознак (тобто числу ознак об'єкта).

Кількість вихідних нейронів мережі Кохонена дорівнює числу кластерів, яке повинно бути побудовано моделлю, і кожен нейрон асоційований з певним кластером. Виходи обробляються за принципом «переможець забирає все», тобто нейрон з максимальним значенням виходу видає одиницю, а всі інші – наближене до 0.

В результаті обробки пред'явленого мережі об'єкта, на виході одного з нейронів формується 1, а на виході інших – 0. Після чого об'єкт відноситься до кластеру, асоційованого з одиничним нейроном.

Навчання мережі Кохонена полягає в налаштуванні ваг зв'язків між нейронами, але виробляється методом конкурентного навчання.

Автокодувальник (Autoencoder, AE) – спеціальна архітектура штучних нейронних мереж, що дозволяє застосовувати навчання без вчителя при використанні методу зворотного поширення похибки. Основною ідеєю є автоматичне кодування (в сенсі стиснення, не шифрування) інформації. Подаються вхідні дані і задана похибка дорівнює різниці між входом і виходом. Архітектура автокодувальника є симетричною мережею прямого поширення, без зворотних зв'язків, містить вхідний, прихований і вихідний прошарки. Мережа за формою нагадує пісочний годинник, в ній вихідний прошарок повинен містити стільки ж нейронів, скільки у вхідному, а кількість нейронів у прихованому прошарку є меншою ніж в них.

Основним принципом навчання мережі є отримання на вихідному прошарку відгук, найближчий до вхідного. Щоб рішення не виявилося тривіальним, на проміжний прошарок автокодувальника накладають обмеження: проміжний шар повинен мати меншу розмірність, ніж вхідний і вихідний, або штучно обмежується кількість одночасно активних нейронів проміжного потоку.

Ці обмеження примушують мережу шукати узагальнення та кореляцію вхідних даних, виконувати їх стиснення. Таким чином, мережа автоматично навчається виділяти з вхідних даних загальні ознаки, які кодуються в значеннях ваг мережі. Так, при навчанні мережі на наборі різних вхідних

зображень, мережа може самостійно навчитися розпізнавати лінії та смуги під різними кутами.

Мережа типу «Deep Belief» (Deep Belief Networks, DBN) – це назва, яку отримав тип архітектури, в якій мережа складається з кількох з'єднаних 15агорткових15ьників. Такі мережі навчаються по блоках, причому кожному блоку потрібно лише вміти закодувати попередній. Для попереднього навчання глибоких (багатошарових) мереж застосовують каскадно автокодувальники. Прошарки мережі навчаються послідовно, починаючи з перших. До кожного нового прошарку на час навчання надається додатковий вихідний шар, що доповнює мережу до автокодувальника, після чого на вхід мережі подаються набір даних для навчання. Ваги ненавченого та додаткового прошарків налаштовуються за методом зворотного поширення похибки. Далі прошарок автокодувальника відключається і створюється новий, що відповідає наступному ненавченому прошарку мережі. На вхід мережі знов подається той же набір даних, навчені перші прошарки мережі залишаються без змін і працюють як вхідні для наступного прошарку автокодувальника. Навчання продовжується для всіх прошарків за винятком останніх. Останні прошарки мережі навчаються без використання авто кодувальника за методом зворотного поширення похибки та на відомих даних (навчання з учителем).

Така техніка називається «жадібним навчанням», яка полягає у виборі локальних оптимальних рішень, що не гарантують оптимальний кінцевий результат. Також мережу можна навчити (методом зворотного поширення помилки) відображати дані у вигляді ймовірнісної моделі. Якщо використовувати навчання без вчителя, стабілізовану модель можна використовувати для генерації нових даних.

Згорткові нейронні мережі (Convolutional Neural Networks, CNN) і глибинні згорткові нейронні мережі (Deep Convolutional Neural Networks, DCNN) сильно відрізняються від інших видів мереж. Зазвичай, вони використовуються для обробки зображень, рідше для аудіо. Типовим

способом застосування CNN є класифікація зображень: якщо на зображенні є кіт, мережа видасть «кіт», якщо є собака – «собака».

Такі мережі зазвичай використовують «сканер», який не обробляє всі дані за один раз. Наприклад, якщо є зображення 200 × 200 пікселів, він не буде відразу обробляти всі 40 тисяч пікселів. Замість цього мережа зчитує квадрат розміру 20 x 20 (зазвичай з лівого верхнього кута), потім зсунеться на 1 піксель і зчитує новий квадрат, і т.д. Ці вхідні дані далі передаються через згорткові прошарки, в яких не всі вузли з'єднані між собою. Ці шари мають властивість стискуватися з глибиною, причому часто використовуються ступені двійки: 32, 16, 8, 4, 2, 1. На практиці до кінця CNN прикріплюють FFNN для подальшої обробки даних. Такі мережі називаються глибинними (DCNN).

Розгорткові нейронні мережі (Deconvolutional Networks, DN), також звані зворотними графічними мережами, є зворотними до 16агорткових нейронних мереж. Наприклад, якщо передати мережі слово «кіт», то вона генерує картинки з котами, що подібні на реальні зображення котів. DNN також можна об'єднувати з FFNN. Варто зауважити, що в більшості випадків мережі передається не рядок, а який бінарний вектор: наприклад, <0, 1> - це кіт, <1, 0> - собака, а <1, 1> - і кіт, і собака.

Як і в звичайних нейронних мережах, в згорткових мережах нейрони мають зв’язок з усіма функціями активації з попереднього шару. Активації ж шарів класифікації можуть бути обчислені за допомогою множення матриць, що супроводжується зміщенням.

Відмінність між шаром класифікації і шаром згортки полягає у тому, що нейрони шару згортки з'єднані тільки з локальною областю на вході, і що нейрони цього шару можуть спільно використовувати параметри. Однак нейрони в обох шарах, незважаючи на свої особливості, підраховують скалярний добуток, тому їх функціональна форма ідентична. Більш того, можна виконати конвертацію між повнозв'язним і згортковим шарами.

Класифікатор повинен запам'ятовувати всі навчальні дані та зберігати їх для подальших порівнянь із даними з тестового запуску. Це дуже ресурсозатратно, оскільки набори даних можуть бути розміром у гігабайтах.

Лінійний класифікатор (рис. 1.1) дає оцінку класу як зважену суму всіх значень пікселів у трьох його кольорових каналах. Залежно від того, які значення встановлено для цих ваг, функція класифікатору має здатність позитивно або негативно оцінювати (залежно від знаку кожної ваги) певні кольори у певних положеннях зображення. Наприклад, людина стверджуватиме, що клас "корабель" може бути більш імовірним, якщо з боків на зображенні є багато синього кольору (що може відповідати воді).

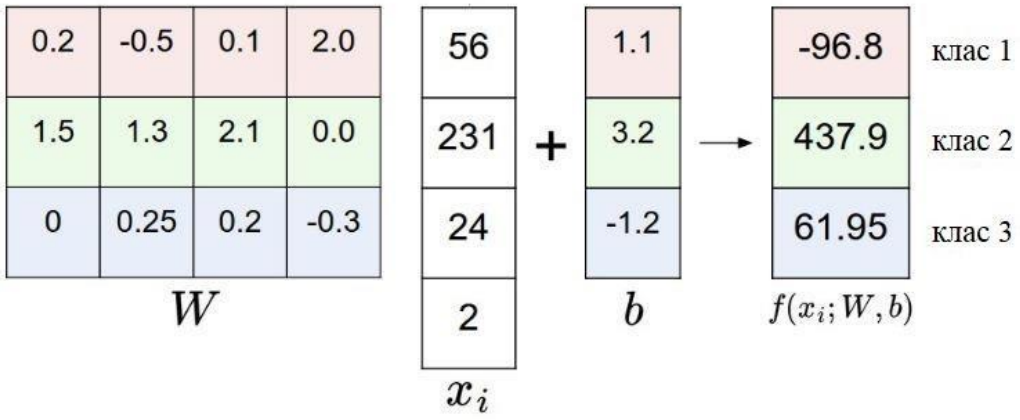


Рисунок 1.1 - Лінійний класифікатор

Ще один спосіб організації класифікатору – шаблонізація. Особливість способу стосується ваг W і полягає в тому, що кожен рядок W відповідає шаблону (або іноді також називається прототипом) для одного з класів. Оцінка кожного класу для зображення отримується шляхом порівняння кожного шаблону, один за одним, із зображенням за допомогою внутрішньої ознаки (або точки-ознаки), щоб знайти те, що "підходить" найкраще.

1.3 Огляд технологій

1.3.1 Огляд мови програмування Java

Java – сурово типізована об'єктно-орієнтована мова програмування, яка була розроблена компанією Sun Microsystems (яку в подальшому придбала компанія Oracle). Програми Java зазвичай транслюються в спеціальний байт-код, тому вони можуть працювати на будь-якій комп'ютерній архітектурі, за допомогою віртуальної Java-машини. Дата офіційного випуску – 23 травня 1995 року.

Редакції платформи Java. Існують три редакції платформи Java, що дозволяють розробникам застосунків, постачальникам послуг і виробникам апаратного забезпечення створювати рішення, що відповідають вимогам конкретних груп користувачів:

Java SE (Java Platform, Standard Edition). Використовуючи Java SE, ви можете створювати і розгортати Java – застосунки для настільних комп'ютерів і серверів, а також розробляти вбудоване програмне забезпечення і програми для систем реального часу;

Java EE (Java Platform, Enterprise Edition). Ця корпоративна версія платформи допомагає розробникам створювати і розгортати переносимі, надійні, масштабовані і безпечні серверні застосунки на Java;

Java ME (Java Platform, Micro Edition). Java ME надає середовище для виконання застосунків, створених для широкого кола мобільних і вбудованих систем, наприклад мобільних телефонів, кишенькових комп'ютерів, телевізійних приставок і принтерів.

Мова значно запозичила синтаксис із C та C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено

віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

1.3.2 Опис патернів проектування

Шаблони проектування – це перевірені і готові до використання рішення часто виникаючих в повсякденному програмуванні задач. Це не клас і не бібліотека, яку можна підключити до проекту, це щось більше. Шаблон проектування, що підходить під завдання, реалізується в кожному конкретному випадку. Крім того, він не залежить від мови програмування. Слід, однак, пам'ятати, що такий шаблон, будучи застосованим неправильно або до невідповідної задачі, може принести чимало проблем. Проте, правильно застосований шаблон допоможе вирішити задачу легко і просто.

Існує три типи патернів:

– породжуючі патерни (Creational), призначені для створення нових об’єктів в системі;

– структурні патерни (Structural), вирішують завдання компонування системи на основі класів і об’єктів;

– патерни поведінки (Behavioral), призначені для розподілу обов’язків між об’єктами в системі.

Породжуючі патерни:

– Абстрактна Фабрика (Abstract Factory) – надає простий інтерфейс для створення об’єктів, які належать до того чи іншого сімейства;

– Будівельник (Builder) – вимальовує стандартний процес створення складного об’єкта, відділяючи логіку будування об’єкта від його представлення;

– Фабричний Метод (Factory Method) – вирішує, яку реалізацію інстанціювати. Вирішують або нащадки Фабричного Методу, або він сам, приймаючи якийсь параметр;

– Прототип (Prototype) – дозволяє нам створювати копії об’єктів, що уже визначені на стадії дизайну (наприклад, список можливих типів зустрічей) або ж визначаються під час виконання програми («п’ятнична вечірка»), таким чином, відпадає необхідність заповнювати всі елементи об’єкту від А до Я. Уже створені або визначені екземпляри об’єкту називаються прототипічними екземплярами (prototypical instances);

– Одинак (Singleton) – забезпечує існування єдиного екземпляру класу та єдиного доступу до нього.

Структурні патерни:

– Адаптер (Adapter) – надає можливість користуватися об’єктом, який не є прийнятним у нашій системі і який не можна змінити. Ми адаптуємо його функціональність через інший, відомий нашій системі, інтерфейс;

– Міст (Bridge) – дозволяє розділити імплементацію від її абстракції, таким чином реалізація може бути змінена окремо від абстракції, оскільки вона не наслідується від неї напряму;

– Компонувальник (Composite) – дозволяє нам зберігати деревовидну структуру і працювати однаково із батьками та дітьми в дереві;

– Декоратор (Decorator) – використовується для надання деякої додаткової функціональності нашим об’єктам;

– Фасад (Facade) – надає єдину «точку доступу» до підсистеми, тим самим спрощуючи її використання та розуміння;

– Легковаговик (Flyweight) – забезпечує підтримку великої кількості об’єктів шляхом виокремлення спільної інформації для збереження в одному екземплярі;

– Проксі (Proxy) – підміняє реальний об’єкт та надсилає запити до нього тоді, коли це потрібно. Проксі також може ініціалізувати реальний об’єкт, якщо він до того не існував.

Патерни поведінки:

– Ланцюжок Відповідальностей (Chain of Responsibility) – забезпечує обробку об’єкта шляхом передачі його по ланцюжку доти, доки не буде здійснена обробка якоюсь із ланок;

– Команда (Command) – дозволяє інкапсулювати всю інформацію, необхідну для виконання певних операцій, які можуть бути виконані пізніше, використавши об’єкт команди;

– Інтерпретер (Interpreter) – дозволяє описати граматику певної мови, за допомогою чого можна записати речення на цій мові та інтерпретувати його значення;

– Ітератор (Iterator) – дозволяє доступатися почергово до елементів будь-якої колекції без вникання в суть її імплементації;

– Медіатор (Mediator) – централізує взаємодію між компонентами, таким чином послаблюючи їхню зв’язність;

– Хранитель (Memento) – використовується тоді, коли ви хочете скасовувати операції без відображення внутрішньої структури Хазяїна (Originator). Координація операцій здійснюється Опікуном (Caretaker), який надає можливість простого збереження миттєвих станів системи без уявлення про те, чим ці стани є;

– Спостерігач (Observer) – дозволяє автоматично реагувати багатьом об’єктам на зміну стану певного іншого об’єкта;

– Стан (State) – дозволяє винести логіку визначення стану об’єкту та його поведінку, характерну для цього стану, в інші класи;

– Стратегія (Strategy) – зберігає сім’ю алгоритмів і дозволяє змінювати їх незалежно та переключатися між ними;

– Шаблонний Метод (Template Method) – задає покроковий алгоритм, а елементи алгоритму можуть бути довизначені в похідних класах;

– Відвідувач (Visitor) – дозволяє відділити певний алгоритм від елементів, на яких алгоритм має бути виконаний, таким чином дозволяючи легко додати або ж змінити алгоритм без зміни елементів системи.

1.4 JavaMail. IMAP. POP3. SMTP

JavaMail – це Java API призначене для отримання і відправки електронної пошти з використанням протоколів SMTP, POP3 та IMAP. JavaMail є частиною платформи Java EE.

SMTP – це широко використовуваний мережевий протокол, призначений для передачі електронної пошти в мережах TCP / IP.

POP3 – стандартний інтернет-протокол прикладного рівня, який використовується клієнтами електронної пошти для отримання пошти з віддаленого сервера по TCP-з'єднання.

IMAP – мережевий протокол прикладного рівня для доступу до електронної пошти. Аналогічно до POP3, служить для роботи з вхідними листами, однак забезпечує додаткові функції, зокрема, можливість пошуку за ключовим словом без збереження пошти в локальній пам'яті. IMAP надає користувачеві великі можливості для роботи з поштовими скриньками, розташованими на центральному сервері. Поштовий клієнт, що використовує цей протокол, отримує доступ до сховища кореспонденції на сервер так, начебто ця кореспонденція розташована на комп'ютері одержувача. Електронними листами можна маніпулювати з комп'ютера користувача (клієнта) без постійного пересилання з сервера і назад файлів з повним змістом листів. Для відправки листів використовується протокол SMTP.

1.5 Методологія Scrum

Scrum – звід правил, на яких будується весь процес роботи. У 1986 році вперше про такий підхід розповіли Хіротака Такеуті і Ікудзіро Нонака. Вони опублікували його в Harvard Business Review (Гарвардському бізнес-огляді). Далі напрямок доповнили і впровадили в роботу Джеф Сазерленд і Кен Швабер.

Основні принципи Scrum

– за встановлені невеликі відрізки часу (спринти) замовник отримує готовий продукт з можливостями, які мають максимальний пріоритет;

– в процесі роботи команда збирається для обговорень. На них здійснюється детальна перевірка виконаних завдань, встановлюються чергові цілі, виконується коригування всього процесу;

– проект можна доопрацьовувати, попередньо визначивши нові цілі, які перетворюються в завдання, і встановлюють тривалість спринту;

– важливі терміни. Резерв проекту - перелік вимог до функціональності об'єкту розробки. Резерв спрінта - список функціональних можливостей з резерву проекту, необхідних замовнику. Пункти розставлені в залежності від важливості;

– в процесі розробки визначаються ролі: Product Owner, ScrumMaster і Scrum - команда. Перший відповідає за інтереси замовників. Другий проводить зустрічі для обговорення, відстежує дотримання встановлених принципів і параметрів роботи і вирішує питання і протиріччя, які виникають.

Група кваліфікованих фахівців виконує завдання проекту. Процес роботи за методологією Scrum наведено на рисунку 1.2.

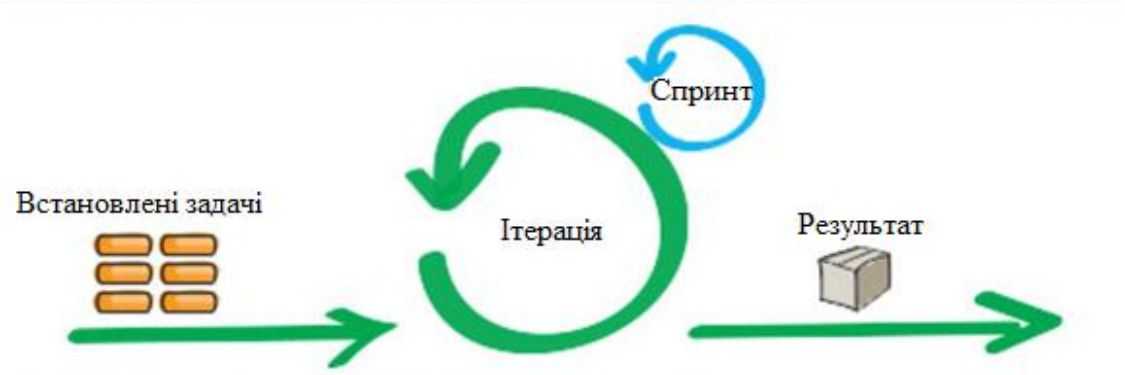


Рисунок 1.2 - Процес роботи за метологією Scrum

Такий підхід дозволяє максимально контролювати процес розробки, за відносно невеликий термін отримувати намічений результат і при необхідності його коригувати.

Сильні сторони Scrum: швидкий запуск проекту, чіткі терміни виконання завдань, складання планів і розгляд підсумків, мінімізація бюджету за рахунок розстановки пріоритетів, працездатність продукту на виході, постійний контроль над ходом проекту.

Слабкі сторони Scrum: зниження командного духу в разі недостатньо хорошої роботи когось із фахівців, вірогідність виконання зайвих операцій, швидкий і жорсткий графік, велика кількість часу, що приділяється для обговорення на шкоду реальній роботі.

1.6 Постановка задачи

Використання спам-фільтрів є актуальною темою на сьогоднішній день. Тому ставиться завдання розробки програмного застосунку класифікації повідомлень електронної пошти за допомогою нейронних мереж.

Для цього необхідно вирішити такі завдання:

– провести аналіз існуючих спам фільтрів, які використовують Google, Yahhoo, Bing;

– розробити сервер, який буде підключатися до поштового сервісу Gmail та брати звідти непрочитані листи;

– організувати підтримання сесії з поштовим сервісом упродовж роботи сервера;

– навчити нейронну мережу відрізняти текст у листах;

– розробити метод взаємодії нейронної мережі з листами для їх аналізу та детектування заборонених;

– розробити метод переміщення заборонених листів до папки «Спам».

2 МАТЕМАТИЧНА МОДЕЛЬ НЕЙРОННОЇ МЕРЕЖІ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТА МЕТОДАМИ МАШИННОГО НАВЧАННЯ

2.1 Штучна нейронна мережа

Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних між собою різним чином шарів штучних нейронів, які організовують загальну активну структуру і функціонально впливають на роботу один одного. У більшості архітектур штучних нейромереж активність нейрона визначається перетворенням зовнішнього сумарного впливу інших нейронів на даний нейрон.

З моменту свого зародження технології штучних нейронних мереж розвивалися досить відокремлено від класичних методів, нерідко докорінно змінюючи уявлення про предмет і проблематику теорії машинного навчання і розпізнавання об'єктів, залишаючи значний вплив на теоретичний, термінологічний і методологічний апарати цих дисциплін. Через деякий час після розвитку базових моделей штучних нейронних мереж, відбувся значний поділ науки про нейромережі на види топологій архітектури мереж і методи навчання мереж.

У більшості архітектур штучних нейронних мереж функції активації нейронів фіксовані, а ваги синапсів є параметрами мережі. Деякі входи нейронів є зовнішніми входами сукупної мережі, а деякі виходи нейронів - виходами сукупної мережі. Завдання нейромережі полягає в перетворенні вхідного вектора у вихідний вектор, що здійснюється вагою і топологією мережі.

2.1.1 Модель нейрону та його функції активації

Штучний нейрон характеризується своїм поточним станом. Тут можна провести аналогію з нервовими клітинами головного мозку, які можуть бути пошкоджені або неправильно правцювати. Він володіє групою синапсів – односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, присутня така частина, як аксон – вихідний зв'язок штучного нейрона, з якого сигнал (збудження або гальмування) надходить на синапси наступних нейронів. Загальний вигляд штучного нейрона наведено на рисунку 2.1. Штучний нейрон спочатку імітує властивості, що дані біологічному нейрону. Тут безліч вхідних сигналів, позначених x_1, x_2, … , x_N, надходить на штучний нейрон. Ці вхідні сигнали, в сукупності позначаються вектором X, приходять до синапсів біологічного нейрона. Кожен синапс характеризується величиною синапсичного зв'язку або його вагою w_i.

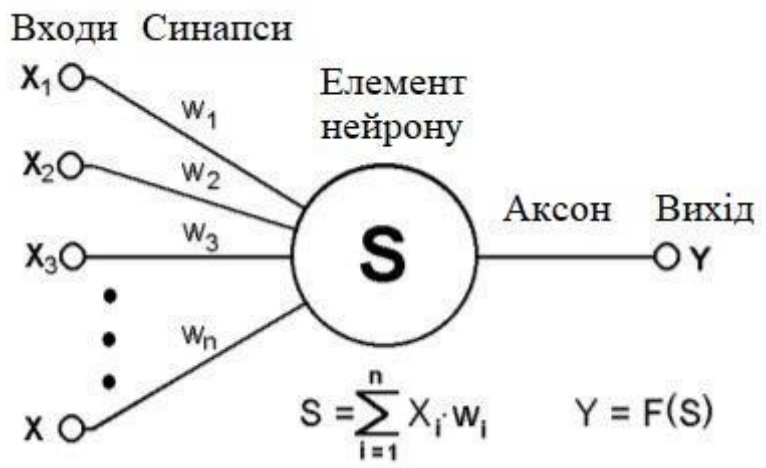


Рисунок 2.1 – Модель штучного нейрона

Стан нейрона визначається за формулою:

$$S = \sum_{i=1}^{n} x_i w_i \quad , \tag{2.1}$$

де n – число входів нейрона

x – значення і-го входу нейрона

w – вага і-го синапсу.

Потім визначається значення аксона нейрона за формулою:

$$Y = f(S), \tag{2.2}$$

де f – деяка функція, яка називається активаційною. Найбільш часто в якості активаційної функції використовується так званий сигмоїд, який має наступний вигляд:

$$f(x) = \frac{1}{1+e^{-ax}}. \tag{2.3}$$

Основна перевага цієї функції в тому, що вона диференційована на всій осі абсцис і має дуже просту похідну:

$$f'(x) = \alpha f(x)(1 - f(x)). \tag{2.4}$$

При зменшенні параметра а сигмоїд стає більш пологим, породжується в горизонтальну лінію на рівні 0,5 при а = 0. При збільшенні а сигмоїд все більше наближається до функції одиничного стрибка.

Для реалізації нелінійності при активації нейрона, його активність, крім різних видів суматорів і систем ваг на входах, визначається функцією одного аргументу – функцією активації. Нейрон в цілому реалізує скалярну функцію векторного аргументу, а вихідний сигнал нейрона визначається видом функції активації і може бути дійсним або цілим. Функція активації застосовується до зваженої суми постсинаптичних сигналів на вході нейрона. Таким чином, активність нейрона повністю визначається його параметрами – вагами і його функцією активації. Існує безліч передавальних функцій, що

застосовуються на практиці при розробці нейронних мереж, деякі з них служать для реалізації нелінійності системи. Вибір тієї чи іншої функції активації часто залежить від умов завдання і структури мережі. Деякі з розглянутих нижче функцій застосовуються тільки в застарілих системах або в навчанні, але вважаються класичними і згадуються щоразу при вивченні штучних нейронних мереж (рис. 2.2).

Порогова функція Хевісайда є найпростішою кусочно-лінійною передавальною функцією. Ця функція використовувалася в класичному персептроні, в даний час використовується в основному з метою навчання теорії нейронних мереж.

Лінійна функція активації. При використанні нескладної кусочно-лінійної функції сигнал на виході нейрона лінійно пов'язаний зі зваженою сумою сигналів на його вході. В даний момент лінійна функція на практиці також використовується дуже рідко.

Сигмоїдальна функція активації. Сигмоїд є монотонно зростаючою всюди диференційованою S-образною нелінійною функцією з насиченням. Забезпечує посилення слабких сигналів і запобігає насиченню сильних сигналів. Є однією з найбільш поширених передавальних функцій, часто використовується в нейронних мережах і сьогодні. Введення сигмоїдальних функцій було зумовлено недостатньою гнучкістю класифікаторів на основі порогових передавальних функцій і дозволило перейти від жорсткої однорозрядної логіки до більш гнучкої поведінки і адаптивної параметризації нейронних мереж. Прикладом сигмоидальной функції є логістична передавальна функція.

Функція гіперболічного тангенса відрізняється від логістичної кривої тим, що її область значень лежить в інтервалі (-1; 1), що в деяких випадках може спростити завдання навчання нейромерж.

ReLU (Rectified Linear Unit). У неглибоких нейромережах використовуються нелінійні функції активації. Часто зустрічаються різновиди сигмоїдальних і тангенціальних функцій є нелінійними, але на

практиці при навчанні глибоких нейромереж такі функції можуть привести до проблем із загасанням або збільшенням градієнтів. Функція ReLU є випрямленою лінійною функцією і на даний момент вважається набагато більш простим і ефективним з точки зору обчислювальної складності варіантом передавальної функції. Похідна цієї функції дорівнює або 0, або 1, від чого її застосування запобігає розростанню і загасанню градієнтів, і призводить до зменшення ваг, що позитивно позначається на обчислювальної здатності нейромережі. Передавальна функція ReLU (1.5) є одним з останніх успіхів в області методів налаштувань глибоких нейронних мереж.

$$f(x) = \max(0, x), \tag{2.5}$$

де x – вхід нейрона.

Сьогодні існує сімейство різних модифікацій ReLU, які вирішують проблеми надійності цієї функції при проходженні через нейрон великих градієнтів: Leaky ReLU, Parametric ReLU, Randomized ReLU.

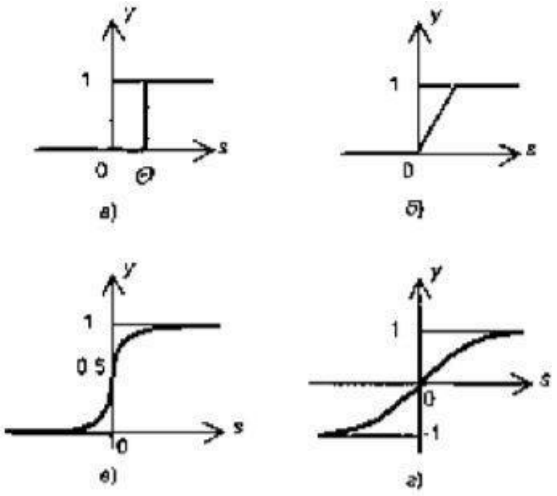


Рисунок 2.2 - Приклади функцій активації:

а) – функція одиничного стрибка; б) – лінійний поріг; в) – логістична функція; г) – гіперболічний тангенс

2.1.2 Модель нейронної мережі зворотного поширення

Нейронні мережі зворотного поширення – це потужний інструмент пошуку закономірностей, прогнозування, якісного аналізу. Таку назву – мережі зворотного поширення (back propagation) вони отримали через використовуваного алгоритму навчання, в якому помилка поширюється від вихідного шару до вхідного, тобто в напрямку, протилежному напрямку поширення сигналу при нормальному функціонуванні мережі.

Нейронна мережа зворотного поширення складається з декількох шарів нейронів, причому кожен нейрон шару і пов'язаний з кожним нейроном шару і + 1. У загальному випадку задача навчання НМ зводиться до знаходження якоїсь функціональної залежності Y = F (X) де X – вхідний, а Y – вихідний вектори. У загальному випадку така задача, при обмеженому наборі вхідних даних, має безліч рішень. Для обмеження простору пошуку при навчанні ставиться завдання мінімізації цільової функції помилки НМ, яка знаходиться за методом найменших квадратів:

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_i - d_i)^2, \tag{2.6}$$

де y – значення j-го виходу нейромережі,

d – цільове значення j-го виходу,

p – число нейронів у вихідному шарі.

Навчання нейромережі за допомогою методу градієнтного спуску, тобто на кожній ітерації зміна ваги здійснюється за формулою:

$$\Delta w_{ij} = -\eta * \frac{\partial E}{\partial w_{ij}}, \tag{2.7}$$

де η – параметр, що визначає швидкість навчання.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} * \frac{\partial y_i}{dS_j} * \frac{\partial S_j}{\partial w_{ij}}, \tag{2.8}$$

де y – значення виходу j-го нейрона,

S – зважена сума вхідних сигналів, що визначається за формулою (2.1).

При цьому множник:

$$\frac{\partial S_j}{\partial w_{ij}} = x_i, \tag{2.9}$$

де x – значення і-го входу нейрона.

Далі розглянемо визначення першого множника формули (1.8):

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{dy_k}{dS_k} * \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} * \frac{dy_k}{dS_k} * w_{jk}^{(n+1)}, \tag{2.10}$$

де k – число нейронів в шарі n + 1.

Ввівши допоміжну змінну:

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} * \frac{dy_j}{dS_j}. \tag{2.11}$$

Ми зможемо визначити рекурсивну формулу для визначення n-ного шару, якщо нам відомо наступного (n + 1) -го шару:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} * w_{jk}^{(n+1)} \right] * \frac{dy_j}{dS_j}. \tag{2.12}$$

Знаходження ж для останнього шару НС не представляє труднощі, так як нам відомий цільової вектор, тобто вектор тих значень, які повинна видавати НМ при даному наборі вхідних значень:

$$\delta_j^{(N)} = (y_i^{(N)} - d_i) * \frac{dy_j}{dS_j}. \quad (2.13)$$

І нарешті запишемо формулу (2.7) в розкритому вигляді:

$$\Delta w_{ij}^{(n)} = -\eta * \delta_j^{(n)} * x_i^n. \quad (2.14)$$

Розглянемо тепер повний алгоритм навчання нейромережі:

– подати на вхід НМ один із потрібних образів і визначити значення виходів нейронів нейромережі;

– розрахувати для вихідного шару НМ за формулою (2.13) і розрахувати зміни ваг вихідного шару N за формулою (2.14);

– розрахувати за формулами (2.12) і (2.14) відповідно і $\Delta w_{ij}^{(N)}$ для інших шарів НМ, n = N-1..1;

– скорегувати всі ваги НС за формулою (2.15)

– якщо помилка істотна, то перейти на крок 1.

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t); \quad (2.15)$$

2.1.3 Підвищення ефективності навчання НМ зворотного поширення

Найпростіший метод градієнтного спуску, розглянутий вище, дуже неефективний у разі, коли похідні за різними ваг сильно відрізняються. Це відповідає ситуації, коли значення функції S для деяких нейронів близька по модулю до 1 чи коли модуль деяких ваг багато більше 1. У цьому випадку для плавного зменшення помилки треба вибирати дуже маленьку швидкість навчання, але при цьому навчання може зайняти недозволено багато часу. Найпростішим методом удосконалення градієнтного спуску є введення моменту m, коли вплив градієнта на зміну ваг змінюється з часом. Тоді формула (1.14) набуде вигляду:

$$\Delta w_{ij}^{(n)}(t) = -\eta * \partial_j^{(n)} * x_i^n + \mu \Delta w_{ij}^{(n)}(t-1). \qquad (1.16)$$

Додатковою перевагою від введення моменту є здатність алгоритму долати дрібні локальні мінімуми.

2.1.4 Одношарові штучні нейронні мережі

Можливість простого розпізнавання доступна одному нейрону, проте з'єднання тих же нейронів в мережі – заропука кращого результату. Одношаровою є мережа, що складається з сукупності нейронів, утворення яких формує шар, як зображено на рис. 2.3. Ліві вершини схеми зображеного нейрону служать для розподілу сигналів, що подаються на вхід. За цими вершинами не закріплено жодних обчислень, тому вони не вважаються шаром і позначені колами, щоб відрізняти їх від обчислювальних нейронів,

що позначаються квадратами. Існує сполучення окремою вагою кожен елементу з множини входів X з кожним штучним нейроном. За нейроном закріплена робота подачі зваженої суми входів в мережу. З метою спільності штучні та біологічні мережі мають відсутні з'єднання. Існують з'єднання між виходами і входами елементів в самому шарі. Ваги представлені елементами матриці *W*. Матриця має *n* рядків і m стовпців, де *n* – число входів, а *m* – число нейронів. Наприклад, w_{12} – це вага, що зв'язує перший вхід з другим нейроном. Таким чином, обчислення вихідного вектора *Y*, зводиться до матричного множення *Y = XW*.

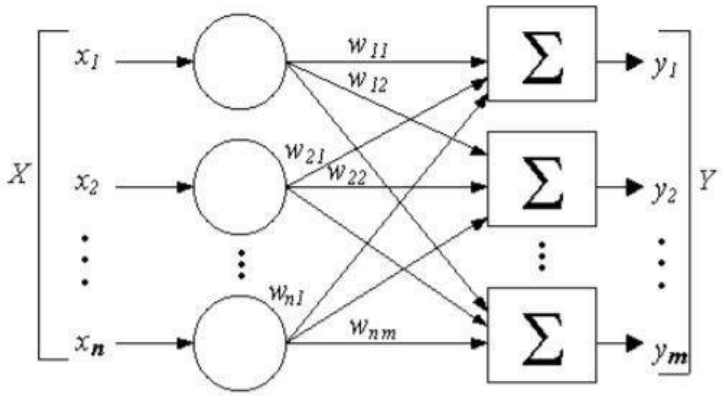


Рисунок 2.3 – Проста одношарова штучна нейронна мережа

2.1.5 Багатошарові штучні нейронні мережі

Відмінності обчислювальних процесів в нейронних мережах часто обумовлені способом взаємозв'язків нейронів. За сукупністю критеріїв на сьогоднішній день багатошарові архітектури можна розділити на статичні і динамічні. Кожен з класів архітектур нейронних мереж може включати безліч підкласів, реалізуючи різні підходи, нижче будуть наведені основні з них. До статичних архітектур відносять мережі прямого поширення, в яких реалізована однонаправлений зв'язок між шарами, відсутні динамічні елементи і зворотній зв'язок, а вихід навченої нейромережі однозначно визначається входом і не залежить від попередніх станів мережі.

Статичні штучні нейронні мережі прямого поширення:

- персептрон;
- нейронна мережа Кохонена;
- когнітрон та неокогнітрон;
- сучасна згорткова нейронна мережа.

На противагу статичним архітектурам, існують динамічні архітектури штучних нейромереж, що реалізують рекурентну структуру з використанням зворотніх зв'язків, завдяки чому стан мережі в кожний момент часу залежить від попереднього стану. Рекурентні нейромережі як правило базуються на багатошаровому персептроні.

2.2 Персептрон

Елементарний персептрон організовується на основі сенсорних даних на вході – S-елементів, асоціативних елементів - A-елементів, і реагуючих елементів на виході – R-елементів. Набір S-елементів, пов'язаний з A-елементом утворює асоціацію, і A-елемент активізується після того як досягнуто певне число сигналів від S-елементів. A-елемент передає зважений сигнал на R-елемент обрахунку суми, і в залежності від того, чи перевищує зважена сума деякий поріг, R-елемент видає результат роботи персептрону.

Багатошаровий персептрон організовується з додатковими прихованими шарами A-елементів, розташованими між S-елементами і R-елементами. Принципова складність завдань, що вирішуються багатошаровим персептроном, є найвищою для класу персептронів. Навчання елементарного і багатошарового персептрона полягає в зміні вагових коефіцієнт зв’язків A – R. Персептрон здатний працювати в режимі розпізнавання або узагальнення.

В одношаровому персептроні вхідні елементи безпосередньо пов'язані з вихідними за допомогою системи ваг, зв'язки S – A організовані за принципом однозначної відповідності. Одношаровий персептрон є окремим випадком класичного елементарного персептрону, найпростішою мережею прямого поширення - лінійним класифікатором, і має безліч принципових обмежень, таких як неможливість реалізації функції XOR.

2.3 Когнітрон та неокогнітрон

Когніторон був розроблений на основі будови біологічної зорової кори, має ієрархічну принципово багатошарову архітектуру. Нейрони між шарами когнітрону пов'язані тільки локально, і кожен шар реалізує різні рівні узагальнення: вхідні шари сприймають прості образи, такі як лінії, великі однорідні ділянки, їх орієнтацію і локалізацію в просторі вхідних даних, в той час як глибокі шари сприймають складніші абстрактні структури, незалежні від локалізації та інших простих ознак образу.

Когнітрон організовується з ієрархічно пов'язаних збудливих і гальмуючих шарів. Співвідношення збуджуючих і гальмуючих сигналів на вході нейрона визначає його стан збудження. Існують спрощені моделі когнітрону, що будуються з одновимірних шарів, але спочатку когнітрон конструювався як каскад двовимірних шарів. Пресинаптичний простір сигналів визначає виходи попереднього шару, постсинаптичний простір – входи наступного шару або площини. Нейрон когнітрону сприймає не весь постсинаптичний простір сигналів, а тільки його частину, реалізується принцип локальної зв'язності. Область пресинаптичного простору сигналів, що утворюють постсинаптичний простір сигналів, що впливають на стан даного нейрона, називається його локальним рецептивних полем. Рецептивні поля близьких один до одного постсинаптичних нейронів, звані зонами

конкуренції, перекриваються, тому активність даного пресинаптичного нейрона позначається на все більше поширення області постсинаптичних нейронів наступних шарів ієрархії. Розміри зон конкуренції обумовлюють кількість сприймання в просторі області ознак. Неокогнітрон є прямим розвитком ідеї, що лежать в основі когнітрону і точніше моделює структуру зорової кори головного мозку і є класифікатором, здатним до кращого розпізнавання об’єктів. Кожен шар неокогнітрона складається з площини простих S-нейронів і площини складних С-нейронів, що також організують локальну зв'язність.

Локальне рецептивне поле на площині S-нейронів наступного шару формується пресинаптичними сигналами площини С-нейронів попереднього шару. Локальні ознаки способу сприймаються S-нейронами, а спотворення локальних ознак компенсуються С-нейронами. В результаті цього процесу кожен шар після вхідного має своїм входом все більш узагальнену картину, утворену С-нейронами попередніх шарів. З кожним рівнем глибини первинні прості ознаки визначаються у більш складних з’єднаннях. Площину S-нейронів можна розглядати як один нейрон, ваги якого визначають ядро згортки, що застосовуються до попереднього шару у всіх можливих позиціях. Всі С-нейрони реагують на образ, відповідний ядру згортки, в їх рецептивному полі, тому він визначається інваріантно до його локалізації. Сучасні глибокі згорткові нейронні мережі засновані на ідеях, що лежать в основі неокогнітрона, і сьогодні застосовуються для вирішення широкого кола завдань: від промислових, корпоративних та дослідницьких до повсякденно побутових, включаючи завдання, які вирішуються мобільними пристроями.

2.4 Дещо з теорії розпізнавання об’єктів

Клас – множина об'єктів, що мають спільні властивості. Для об'єктів одного класу передбачається наявність «схожості». Для задачі розпізнавання може бути визначено довільну кількість класів, більше одного. Кількість класів позначається числом S. Кожен клас має свою ідентифікуючу мітку класу.

Класифікація – процес призначення міток класу об'єктам, відповідно до певного опису властивостей цих об'єктів. Класифікатор – це інструмент для присвоєння міток класам, який в якості вхідних даних отримує перелік ознак об'єкта. До одного з найпоширеніших способів класифікації можна віднести спосіб, що базується на описі об'єктів з використанням ознак, де кожен об'єкт характеризується набором числових або нечислових ознак. Проте існують типи даних для яких відкриті ознаки не дають високої точності класифікації, наприклад, колір точок зображень або цифровий звуковий сигнал. Загальна класифікація зображень собак і автомобілів є дуже простою для людини і водночас складною для машини. Причиною цього є можливість людини сприймати «приховані ознаки» недоступні для машини, такі як морда собаки або колеса автомобіля.

Верифікація – процес зіставлення досліджуваного об'єкта із однією моделлю об'єкта або описом класу.

3 КОМП'ЮТЕРНА МОДЕЛЬ РЕАЛІЗАЦІЇ МЕТОДУ КЛАСИФІКАЦІЇ ПОВІДОМЛЕНЬ

3.1 Основні принципи навчання нейронних мереж

Процес навчання штучний нейронних мереж розглядається як налаштування архітектури і ваг зв'язків між нейронами (параметрів) для ефективного виконання поставлених перед мережею завдань. Існує два великих класи навчання: клас детерменірованних методів і клас стохастичних методів.

До класу детерменірованих методи входять методи, в основі яких лежить ітеративна корекція параметрів мережі, в ході поточної ітерації яка грунтується на поточних параметрах. Основним детерменірованним методом і найпоширенішим методом навчання мереж сьогодні є метод зворотнього поширення помилки.

До класу стохастичних методів входять методи, що змінюють параметри мережі випадковим чином і зберігають тільки ті зміни параметрів, які призвели до поліпшення результатів роботи. Стохастичні алгоритми навчання реалізуються за допомогою порівняння помилок.

Навчання з учителем. Під час навчання мережі з учителем кожному прикладу з навчальної вибірки відповідає вектор, що характеризує однозначну правильну відповідь, що подається відразу на вихід мережі в обхід всієї її архітектури.

Після отримання власного результату мережі, алгоритм порівнює результуючий вектор з правильною відповіддю, на основі чого відбувається корекція подальших помилок. Правило корекції помилки, на якому заснований метод зворотного поширення помилки, є класичним прикладом навчання з учителем.

Навчання без вчителя у застосуванні до штучний нейромереж реалізується природним чином в процесі навчання, коли автоматичне налаштування параметрів мережею призводить до появи однакових результатів її функціонування при досить близьких вхідних значеннях, що на практиці можна порівняти зі зниженням розмірності даних в результаті ітераційного методу головних компонент. Правило Хеба, засноване на гіпотезі про посилення зв'язків між біологічними нейронами в разі їх одночасного збудження, і методи навчання при змаганні нейронів шляхом порівняння інтенсивності їх реакції є класичними прикладами методів навчання без учителя.

Метод зворотного поширення помилки (Backprop). Метод є класичним методом навчання з учителем, заснований на правилі корекції помилок і був розроблений як метод навчання багатошарового персептрона. Основна ідея полягає в поширенні сигналів помилки після її обчислення на виході мережі в напрямку, зворотньому прямому поширенні сигналів під час звичайного обчислювального процесу, від виходів мережі до її входів. При зворотньому проході синаптичні ваги налаштовуються з метою мінімізації помилки. Фактично, реалізується стохастичний градієнтний спуск, відбувається рух в багатовимірному просторі ваг до мінімуму помилки в сторону, протилежну градієнту. В процесі навчання циклічно знаходяться рішення на одно-критеріальні завдання оптимізації.

Для можливості реалізації методу передавальна функція нейронів повинна бути диференційована. Оскільки метод є модифікацією класичного методу градієнтного спуску, то може розглядатися як градієнтний спуск по поверхні помилки.

3.2 Алгоритм навчання нейронної мережі «з вчителем»

У своїй роботі я обрав нейронну мережу, яка використовує парадигму навчання: «з вчителем». Визначення методу навчання з учителем можна сформулювати так: задача машинного навчання у реалізації функції виведення на підставі відомих навчальних даних. Структура навчальних даних складається з набору прикладів для виконання навчання. В даному методі навчання кожен приклад представляє собою пару, що складається з вхідного об'єкта (зазвичай вектора) і бажаного вихідного значення (контрольний сигнал). Необхідно прорахувати значення функції на момент входу даних, потім порівняти значені із значенням очікуваного резузльту, вирахувати помилку і скорегувати параметри (вага нейронів) мережі на цю помилку.

Алгоритм навчання з учителем:

1) взяті дані для навчання мережі розділити на дві частини. Першою половиною буде навчальна вибірка, другою – тестова вибірка. (можливо розділити дані 70/30);

2) прорахувати значення функції для навчальної вибірки та знайти значення функції помилки;

3) провести корегування ваг синапсів у мережі;

4) повторювати пункти 2 і 3 до тих пір поки значення функції помилки не буде мінімальним. Повторювати ці дії можна як для кожного екземпляра навчальної вибірки так і для всієї вибірки в цілому. У першому випадку навчання буде повільніше, але з більшою точність. У другому випадку, показник точності занепадає, але навчання буде відбуватися швидше;

5) перевірити всі значення за тестовою вибіркою, щоб зрозуміти наскільки добре система змогла узагальнити дані (навчитися).

Якщо нейронна мережа навчається з використанням заздалегідь відомих правильних відповідей, то такий алгоритм навчання називається -

навчання з учителем. Необхідно відзначити, що при навчанні з учителем потрібна велика вибірка, щоб в достатній мірі сформувати робочу і гнучку нейронну мережу.

3.3 Показники нейронної мережі: точність, втрата та якість

Епоха – прямий і зворотний прохід по всім тренувальним прикладам.

Розмір серії (batch) – кількість тренувальних прикладів для однієї ітерації прямого і зворотного проходів. Кількість ітерацій – кількість проходів: кожен прохід використовує приклади (batch). Один прохід дорівнює прямому проходу та зворотньому проході. Тобто маючи 1000 прикладів, batch = 500, нам буде потрібно зробити дві ітерації, щоб завершити одну епоху. З математичної точки зору, навчання нейронних мереж - багатопараметрична задача нелінійної оптимізації.

Процес навчання мереж описують за допомогою графіку кривої, де горизонтальна вісь відображає кількість навчальних зразків, що отримує мережі, а вертикальна вісь відображає значення помилки, що отримано при класифікації вхідних даних.

На рисунку 3.1 зображено два криві: криву значень помилок на навчальному наборі (величина якої задана горизонтально) та криву значень помилок на тестовому або перевірочному наборі (яка має фіксований розмір). Чим більше даних використовується для навчання, тим більше помилок видасть архітектура мережі, що відповідатимуть навчальним даним. У той же час навчальні дані стають більш схожими на справжній розподіл даних, які слід зафіксувати за навчальними даними. У певний момент часу помилка на навчальному та тестовому наборах повинна бути приблизно однаковою.

Крива процесу навчання для встановлення є індикатором правильності роботи мережі, якщо велика кількість навчальних даних без будь-яких змін,

що в свою чергу покращує ефективність мережі. Побудувавши криву навчального набору, можна оцінити, чи здатна архітектурна модель мережі відповідати даним для потрібної класифікації помилки. Значення помилки при тестовому наборі ніколи не повинне бути значно нижчим, ніж значення помилки навчального набору. Якщо помилка на навчальному наборі занадто висока, то збільшення даних не допоможе вирішити проблему. Натомість стане зрозуміло, що сама архітектура моделі або алгоритм її навчання потребують коригування. У випадку, коли на графіку крива навчального набору значно перевищує криву тестового (перевірочного) набору даних необхідно або зменшити роздільну здатність зображень, або ввести додаткові навчальні зразки даних, або збільшити регуляризацію для вирішення даної проблеми.

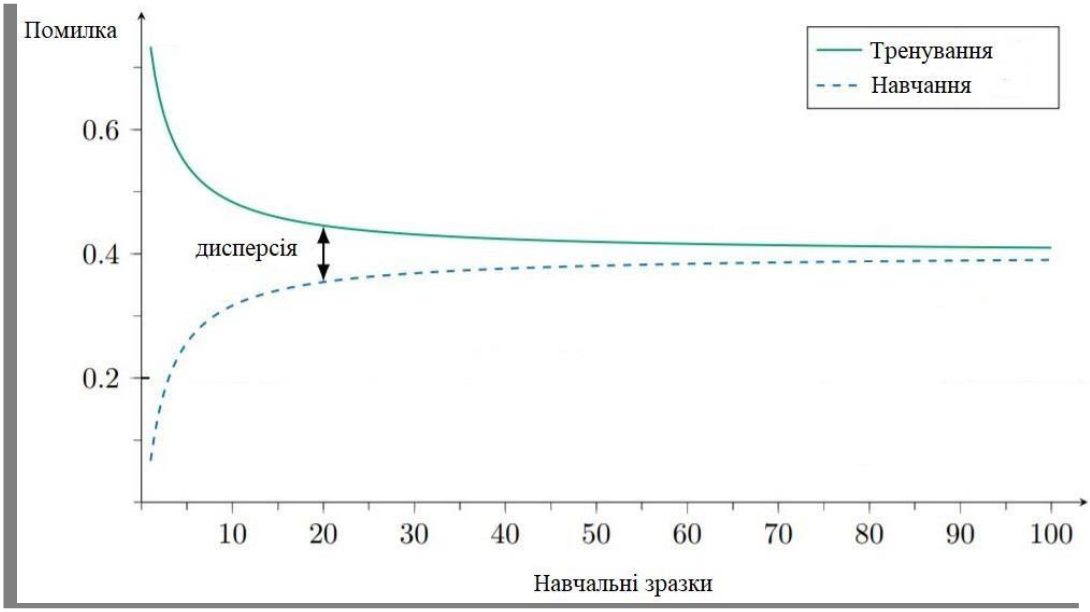


Рисунок 3.1 - Процес навчання мережі

Показники мережі слугують для обчислення та відображення гіперпараметрів (наприклад, навчальної епохи). Показники можна подати у вигляді графіків гіперпараметрів на горизонтальній осі та значення якостіна вертикальній осі. Точність розпізнавання, представлена як помилка = (1 точність) або втрата є типовими показниками якості. У випадку, коли

кількість навчальних епох розглядається як головний гіперпараметр, показники діють як індикатор довгого часу тренування та продуктивності мережі. Побудувавши графік значень помилки на наборі тренувань, а також значень помилки на наборі перевірки, можна також оцінити, чи є загрозливим перенавчання

Розглянемо детальніше рис.3.2 де зображено типову ситуацію перенавчання. Кількість епох навчання виступає як гіперпараметр мережі, а оцінка якості розпізнавання виражено через значення помилки. Чим довше буде навчатися мережа, тим краще вона звикне до тренувального набору даних. У якийсь момент мережа добре розпізнаватиме лише для навчальні дані і втратить здатність до узагальнення. На цьому етапі криві якості навчальної та перевірочної виборок на графіку розходяться. Поки класифікатор продовжує вдосконалювати показник якості на тренувальному наборі, він навпаки погіршує якість розпізнавання при застосуванні тестового набору. При ситуації, коли показник якості не поліпшується при достатній кількості епох варто змінити значення ініціалізованих ваг нейронів на початку мережі, або застосувати нормалізацію моделі.

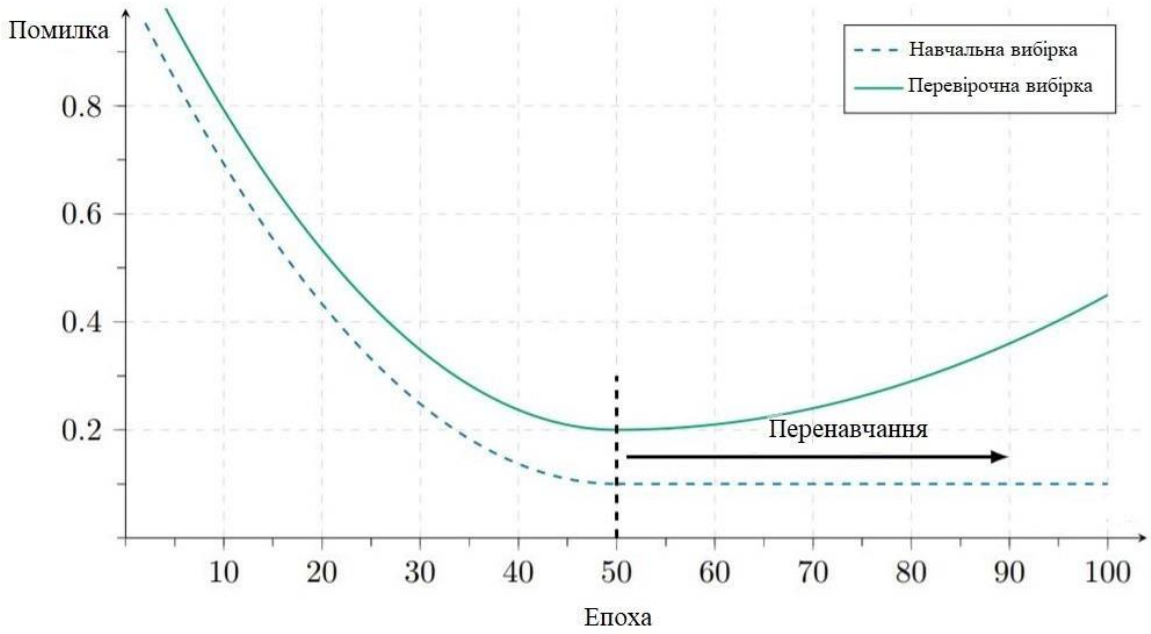


Рисунок 3.2 - Проблема перенавчання мережі

3.3.1 Функція втрат

Застосовані у лінійних класифікаторах функції визначення оцінки приналежності об’єкту до певного класу не надають певного контролю за вхідними даними, адже вони параметризовані значенням набором ваги нейронів W. Тому варто встановити контроль над цими вагами нейронів, і потрібно зробити це їх таким чином, щоб прогнозовані оцінки класів відповідали значенню з навчальних виборок.

Практичне застосування класифікаторів показало, що не завжди оцінка класу, що дійсно відповідає правильній ідентифікації об’єкта є вірною, тому рішенням стане обчислення не позитивного результату розпізнавання, а навпаки – негативного. Для цього і використовується функція втрат або цільова функція (функція вартості). Інтуїтивно, значення функції втрат буде високим, якщо роботу з класифікації навчальних даних завершилася помилками, і значення буде низьким, якщо все працюватиме вірно.

Функція втрат (англ. loss function), також називається функцією помилки або функцією вартості – це функція, яка вираховує дійсне значення для складної події, як-от передбачення приналежності до класу вхідного вектора. Вона використовується для визначення цільової функції. У пролематиці класифікації функція втрат подається як перехресна ентропія. С точки зору теорії інформації, навчанням є мінімізація значень перехресної ентропії стосовно реальних класів і гіпотезами моделі.

Значенні втрати даних є позитивним, коли класифікація є неправильною, але значення втрат є числом більшим для більш складних моделей мережі. Якщо дві моделі мереж розпізнають однаковий набір даних, то кращою визнається модель архітектура якої є простішою.

Приводом для зображення саме функції втрат методом кривої на графіку замість інших показників якості є те, що тоді графік міститиме більше інформації про якість мережі. Головним недоліком функції втрати є

те, що вона не має верхньої межі, як точність, і її важко інтерпретувати. Функція втрати лише показує відносний прогрес навчання, тоді як точність показує абсолютний прогрес.

Існують критерії покращення використання функції втрат, які можуть допомогти у вдосконаленні мережі. Якщо втрати протягом кількох епох не зменшуються, то темп навчання може бути занизьким. Процес оптимізації також може бути зафіксований у місцевому мінімумі. Неправильне значення функції втрат у вигляді числа, що не існує, може бути пов'язана із надто високими темпами навчання. Іншою причиною може бути поділ на нуль або логарифм нуля. В обох випадках додавання невеликої константи, як от 10−7, вирішує проблему. Якщо крива функції втрат в залежності від кількості епох напочатку досягла межі, погані значення ініціалізації ваги може бути цьому причиною.

3.3.2 Огляд бібліотек машинного навчання

На сьогоднішній день у реалізації нейронних мереж важливу роль реалізації посталених архітектур на налаштувань відіграють, як правило, спеціалізовані пакети, що надають вже повністю реалізовані базові структури та алгоритми для зручної розробки. Найбільш широко використовуються такі популярні бібліотеки, як Theano, Tensorflow та Torch та ін. Дані пакети реалізують програмний інтерфейс для реалізації низькорівневих розрахунків, тому прийняття рішень, в першу чергу, повинно базуватися на суперечностях про продуктивність та зручність розробки. Варто відзначити, що Torch реалізує інтерфейс для мови Lua, не дуже популярної серед програмістів, що істотно знижує зручність і швидкість розробки. Згідно дослідженням Theano and Torch показують приблизно однакову продуктивність на одну й тому ж наборі даних.

Окрім описаної вище незручності використання бібліотеки Torch, існує ще одна, також пов'язана зі зручністю розробки. Інші два згаданих вище інструменти здатні працювати з бібліотекою більш високого рівня Keras , що надає загальні, для Tensorflow і Theano, заздалегідь визначені алгоритми та структури. За результатами порівняння та аналізу документації бібліотек було прийнято рішення використовувати для програмної реалізації згорткової нейронної мережі зв'язок бібліотек Tensorflow і Keras.

Бібліотека Keras – програмний продукт, написаний мовою Python, що надає високорівневе API для роботи зі штучними нейронними мережами. Даний продукт створювався з метою запобіганню труднощів в розробці нейронних мереж, пов'язаних із синтаксичними особливостями конкретних пакетів. В січні 2016 року даний інструмент було придбано компанією Google і адаптовано як основне високорівневе надлаштування над бібліотекою Tensorflow.

Бібліотека Theano була написана в першу чергу як розширення мови Python, яка дозволяє ефективно розрахувати математичні вирази, що містять багатомірні масиви. В бібліотеці реалізується базовий набір інструментів для побудови нейронних мереж. Процес створення моделі та визначення її параметрів вимагає написання об'ємного коду, що включає реалізацію класу моделі, самостійного визначення її параметрів, реалізації методів, що визначають функцію помилок, правило обрахування градієнтів, способи зміни ваги нейронів.

Бібліотека Caffe реалізована на мові програмування C++. Топологія нейромереж, вихідні дані та спосіб навчання задаються за допомогою конфігураційних протоколів (застосовується технологія і протокол передачі даних) у прототиповому форматі. Побудова структури мережі виконується з простотою та зручністю. Бібліотека Caffe підтримується досить великою спільнотою розробників та користувачів і на сьогоднішній день є самою розповсюдженою бібліотекою глибинних навчальних програм широкого кола застосування.

Через те, що в останні роки методи машинного навчання отримали велику популярність і застосовуються в безлічі різних областей, існує великий вибір бібліотек для реалізації моделей машинного навчання. В таблиці 3.1 представлені найбільш відомі та поширені бібліотеки.

Таблиця 3.1 – Порівняння бібліотек глибинного навчання

Назва бібліотеки	*Theano*	*Caffe*	*Torch*	*TensorFlow*
Мова програмування	Python	Python, C++	Lua	Python, Java
Спеціалізація на методах машинного навчання	Різноманітні види регресій, нейронні мережі	Нейронні мережі	-	Різноманітні види регресій, нейронні мережі
Можливість глибинного навчаня	+	+	+	+
Швидкість роботи	Середня	Висока	Низька	Висока
Якість позпаралелювання між декількома пристроями	-	-	-	+
Допоміжі функції (автоматичне обрахування градієнта та ін.)	-	+ (зарахунок допоміжних бібліотек Python)	+ (зарахунок допоміжних бібліотек Python)	+

3.3.3 TensorFlow

TensorFlow – це бібліотека програмного забезпечення з відкритим програмним кодом для задач машинного навчання, розробленого компанією Google. Вона дозволяє створювати та навчати нейронні мережі різної архітектури, що знаходять своє застосування у виявленні та розпізнаванні образів, пошуку взаємозв'язків. TensorFlow також включає в себе TensorBoard, який представляє собою засоби візуалізації в браузері для оцінки ефективності навчання та мережевих параметрів моделі.

TensorFlow досягає своєї продуктивності завдяки паралельній обробці задач між центральними та графічними процесорами GPU (у тому числі декількох одночасно) і навіть горизонтальне масштабування за допомогою gRPC. Tensorflow підтримує мови програмування Python і C++. Кожне обчислення в TensorFlow представляється у вигляді графу потоку даних, графу обчислень. Граф обчислень є моделлю, яка описує як будуть виконуватися обчислення. Важливо зауважити, що складання графа обчислень і виконання операцій в заданій структурі – два різних процеси. Граф складається з плейсхолдерів (tf.Placeholder), змінних (tf.Variable) і операцій. У ньому виробляється обчислення тензорів – багатовимірних масивів, які можуть бути числом або вектором.

Графи виконуються в сесіях (tf.Session). Існують два типи сесій: звичайні та інтерактивні (tf.InteractiveSession); інтерактивна сесія підходить для виконання в консолі. Сесія зберігає стан змінних (Variables) і черг (queues). Явне створення сесій і графів гарантує належне звільнення ресурсів пам'яті. У графі кожна вершина має 0 або більше входів і 0 або більше виходів, і являє собою реалізацію операції. Тензорами є ребра графа, а саме масиви довільного розміру (тип масиву вказується під час побудови графа). Особливі вершини, що управляють залежності (control dependencies), також можуть бути в графі: вони вказують, що вихідний вузол для контрольної залежності повинен закінчити виконання до того, як вузол одержувача контрольної залежності почне виконуватися.

Кожна операція має назву і являє собою абстрактне обчислення (наприклад, операція суми). У операції можуть бути атрибути. Ядро – специфічна реалізація операції, яка може бути виконана на певному типі пристрою (центральний або графічний процесор).

Змінна – особливий вид операції, який повертає лічильник на постійно мінливий тензор: така змінна не зникає після одиничного використання графа. Лічильники на подібні тензори передаються чисельним операціями, які потім змінюють вказаний тензор. У завданнях машинного навчання,

параметри моделі зазвичай зберігають тензори в змінних, які оновлюються на кожному кроці навчання.

3.4 Stanford NLP

Розглядаючи різні бібліотеки машинного навчання, я почала писати свою роботу з використанням TensorFlow, але в мові Java ця бібліотека досить сира і не стабільна. Я зіткнулася з багатьма труднощами, використовуючи її та в деякий момент вирішив спробувати використати іншу, досить нову бібліотеку, яка була написана під мову Java – Stanford NLP.

Обробка природної мови (NLP – Natural language processing) – область, що знаходиться на перетині computer science, штучного інтелекту та лінгвістики. Мета полягає в обробці і "розумінні" природної мови для перекладу тексту, пошуку слів і відповіді на питання.

З розвитком голосових інтерфейсів і чат-ботів, NLP стала однією з найважливіших технологій штучного інтелекту. Але повне розуміння і відтворення сенсу мови – надзвичайно складне завдання.

3.5 Тестування розробленої моделі

У ході тестування роботи серверу було виправлено багато помилок. При першому запуску сервера всі повідомлення вважалися, як спам та переміщувалися до папки «MYSPAM». Кожен наступний запуск породжував деякі помилки, виключення, та поступово було налагоджено роботу серверу.

Програмний застосунок передбачає знаходження та обробку повідомлень електронної пошти Gmail за допомогою нейронної мережі. Так як програмний застосунок реалізований, як сервер, ілюстрація буде у вигляді трасування та відображення повідомлень у привичному нам інтерфейсі поштової скриньки Gmail.

Отже, перед тим, як запустити наш сервер нам потрібно занести логін та пароль до файлу з назвою credentials.txt в одну лінію через пробіл та створити папку з назвою «MYSPAM» у скриньці Gmail.

Потім редагувати файл з назвою emailspamfilter_ner.txt, в нього можна вносити слова або фрази, які ви вважаєте за «спам» та допоміжне слово SPAMITEM.

Важливо ставити табуляцію між вашим словом та ключовим словом.

Приклад внесення:

акція SPAMITEM

распродажа SPAMITEM

…

Після того, як всі данні внесені ми можемо запустити наш сервер командою ./start.sh – Linux або відкрити виконуючий файл Start.bat – Windows.

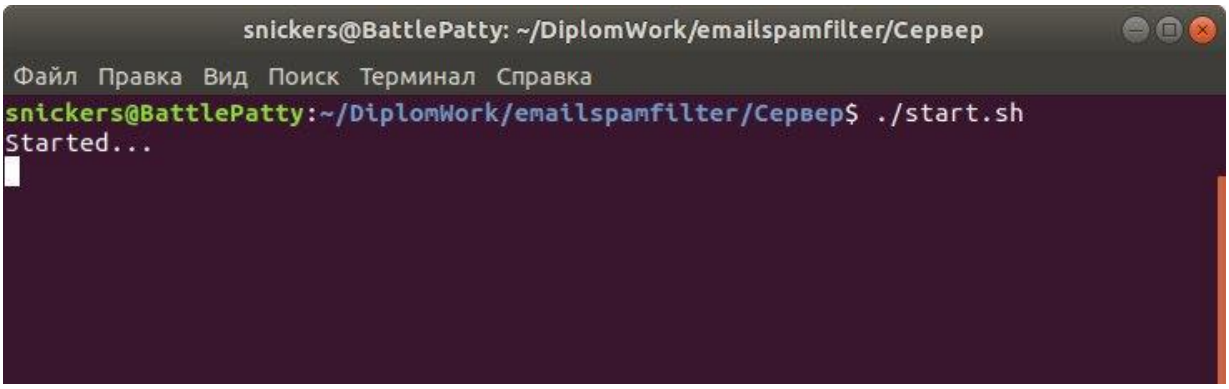


Рисунок 3.3 – Запуск серверу

Як можна побачити на рис. 3.3 за допомогою трасування, сервер запущено. Але поки не прийде повідомлення нейронна мережа не активна.

На рис. 3.4 можна побачити, що є нове повідомлення.

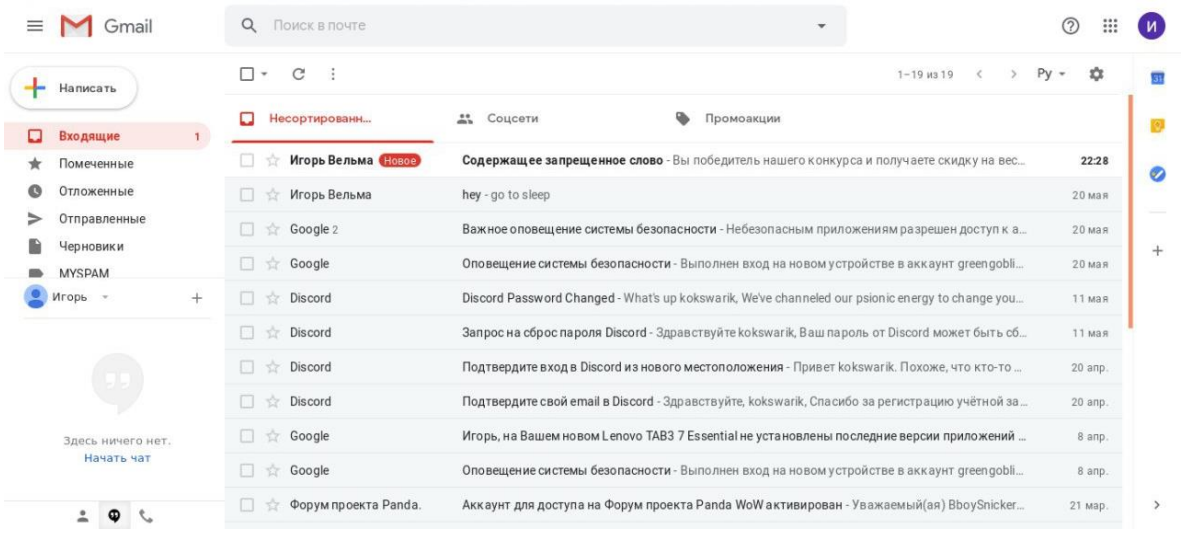


Рисунок 3.4 – Зображення скриньки

Після того, як надійшло повідомлення, наша нейронна мережа починає навчатися, на це відводиться певний час в залежності від того, скільки інформації занесено до файлу з назвою emailspamfilter_ner.txt.

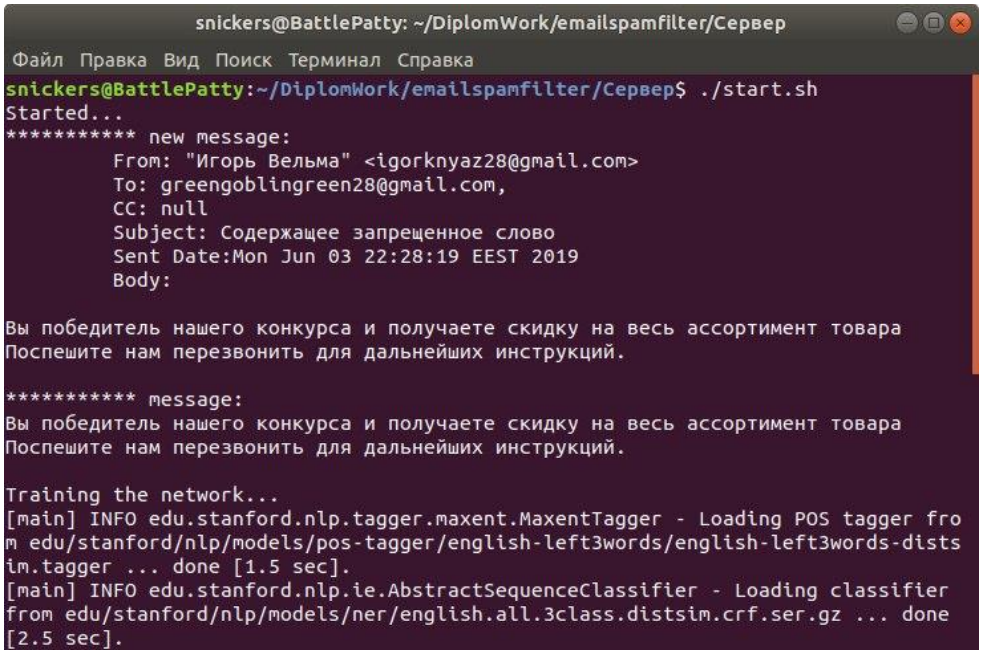


Рисунок 3.5 – Логування та процес навчання мережі

На рис. 3.5 видно, що нам надійшло нове повідомлення та його дані. Якщо глянути нижче, можна побачити у логах напис Training the network – це означає, що мережа навчається. Після того, як мережа навчиться вона проаналізує текст повідомлення та в логах в залежності від того повідомлення є спам чи ні, покаже:

marked as spam… – якщо повідомлення є спам.

not a spam… – якщо повідомлення не спам.

На рис. 3.6 у кінці трасування видно, що повідомлення було помічено, як спам.

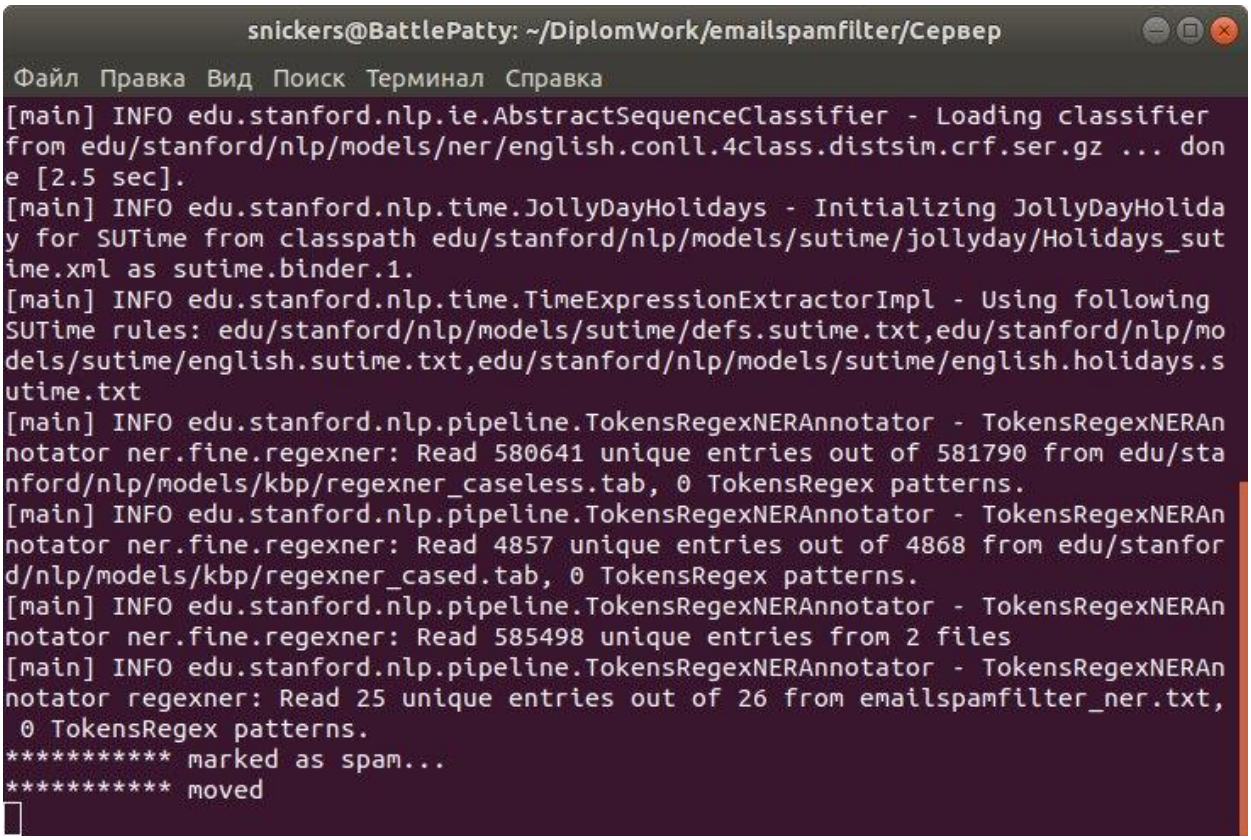


Рисунок 3.6 – Логування (відображення того, що лист помічений, як спам)

Після чого було надіслано ще декілька тестових повідомлень, які успішно не пройшли перевірку на спам та відправилися до папки «MYSPAM». На рисунку 3.7 можна побачити листи.

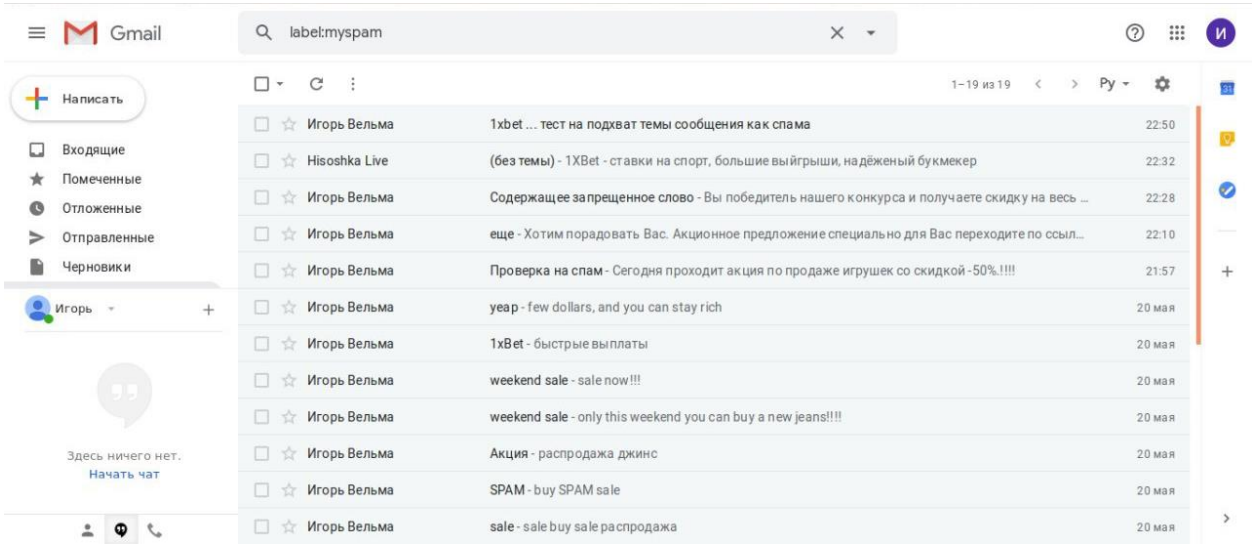


Рисунок 3.7 – Папка «MYSPAM»

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів , що впливають на персонал

Персональні ЕОМ типу IBM PC AT має наступні характеристики:

- споживана потужність 350 Вт;
- робоча напруга 220 В;
- напруга джерел живлення +12 В, -12 В, 5 В;
- робоча частота 50 Гц.

Виходячи з приведених характеристик, очевидно, що для користувача існує небезпека поразки електричним струмом у разі недбалого поводження з комп'ютером і порушення правил експлуатації (невиконання огляду відкритих частин ПЕВМ, що знаходяться під напругою або знятих для ремонту вузлів і т. д.).

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;

У відповідності з ДСН 3.3.6.042-99 [21] до легкої фізичної роботи відносяться всі види діяльності, вироблювані сидячи і не вимагаючи фізичної напруги. Робота користувача розробленого пакету програм відноситься до категорії Іа.

Згідно з ДСТУ Б А.3.2-13:2011 [22] приміщення для ПЕОМ по ступеню небезпеки поразки людини електричним струмом відноситься до приміщень без підвищеної небезпеки (немає струмопровідної полови, вогкості, підвищеної температури, можливості одночасного дотику до корпусів устаткування з “землею” і до струмонесучих частин).

У відповідності з ДСанПіН 3.3.2-007-98 [23] при обслуговуванні ПЕВМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

– підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;

– підвищений рівень статичної електрики;

– підвищений рівень електромагнітних випромінювань;

– підвищена або знижена температура повітря робочої зони;

– підвищена або знижена рухливість повітря;

– підвищена або знижена вогкість повітря;

– відсутність або недолік природного світла;

– підвищена пульсація світлового потоку;

– недостатня освітленість робочого місця;

– підвищений рівень шуму на робочому місці;

– розумове перенапруження;

– емоційні навантаження;

– монотонність праці.

В промислових приміщеннях повинні бути забезпечені санітарно-гігієнічними етичні вимоги до повітря робочої зони згідно з ДСН 3.3.6.042-99 [21].

4.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка усугубляє тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм надає на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Ступінь ураження людини електричним струмом залежить від наступних факторів:

– значення сили струму;

– електричного опору тіла людини і тривалості протікання через нього струму;

– роду і частоти струму;

– індивідуальних властивостей людини і навколишнього середовища.

Даним проектом передбачаються наступні технічні способи і засоби, застережливі поразки людини електричним струмом:

– заземлення електроустановок;

– занулення;

– захисне відключення;

– електричне розділення сітей;

– використовування малої напруги;

– ізоляція струмоведучих частин;

– огорожа електроустановок.

Проведемо розрахунок заземлюючого пристрою.

Початкові дані для розрахунку заземлюючого пристрою:

– напруга установки, що заземляється, - 220В;

– режим нейтралу мережі - з ізольованою нейтралою;

– питомий опір ґрунту – 100 Ом·м(суглинок);

– гранично допустимий опір заземлюючого пристрою - 4 Ом;

– характеристика кліматичної зони (III):

а) середня багаторічна низька температура, ^{o}C - від –14 до -10;

б) тривалість замерзання вод, дні - 150;

в) коефіцієнт сезонності для вертикального електроду завдовжки 3м -1,5.

Визначимо розрахунковий опір ґрунту (Ом·м) по формулі (4.1).

$$\rho_{расч} = \psi \cdot \rho = 1{,}5 \cdot 100 = 150 Ом \cdot м \tag{4.1}$$

де ρ - питомий опір ґрунту;

ψ_i – кліматичний коефіцієнт, що враховує стан ґрунту під час вимірювань (таблиця 4 [22]).

Розрахуємо опір розтіканню одиночного трубчастого заземлювача по формулі (4.2).

$$R_{з.1} = (\frac{\rho_{расч}}{2 \cdot \pi \cdot l}) \cdot \ln(4 \cdot \frac{l}{d}) \tag{4.2}$$

де l – довжина заземлювача (l=5м);

d – діаметр труби і стрижня (d=0,05м);

$$R_{з.1} = (\frac{\rho_{расч}}{2 \cdot \pi \cdot l}) \cdot \ln(4 \cdot \frac{l}{d}) = (\frac{150}{2 \cdot 3{,}14 \cdot 5}) \cdot \ln(4 \cdot \frac{5}{0{,}05}) = 28{,}6 \text{ Ом}$$

Розрахуємо кількість паралельно сполучених одиночних заземлювачей по формулі (4.3).

$$n = \frac{R_{з.1}}{R_{доп} \cdot \eta} = \frac{28{,}6}{4 \cdot 0.47} = 15{,}2 \tag{4.3}$$

де Rдоп=4. – самий допустимий опір заземлюючого пристрою;

η - коефіцієнт використовування ґрунтового заземлення (для шістки заземлювачей η=0,47).

Округлятимемо отримане значення у більшу сторону n=[15,2]=16.

Розрахуємо довжину горизонтальної сполучної смуги по формулі (4.4).

$$L = a \cdot (n-1) = 3 \cdot (16-1) = 45м \tag{4.4}$$

де a – відстань між вертикальними заземлювачами (a=3м);

n – кількість вертикальних заземлювачей (n=16).

Розрахуємо опір сполучної смуги по формулі (5.5).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln(\frac{L^2}{d \cdot h}) \tag{5.5}$$

де d – еквівалентний діаметр смуги шириною l=5 (d=0,05м);

h – глибина заставляння смуги (h=0,8м).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln(\frac{L^2}{d \cdot h}) = \frac{150}{2 \cdot 3{,}14 \cdot 5} \cdot \ln(\frac{45^2}{0{,}05 \cdot 0{,}8}) = 51{,}7 \text{ Ом}$$

Розрахуємо результуючий опір заземлюючого електроду з урахуванням сполучної смуги по формулі (4.6).

$$R_{гр} = \frac{R_{з.1} \cdot R_n}{R_{з.1} \cdot \eta_n + R_n \cdot n \cdot \eta_з} \le R_{доп} \tag{4.6}$$

де η_n – коефіцієнт використовування сполучної смуги (для 6-і заземлювачей $\eta_{n=0,27}$).

$$R_{гр} = \frac{R_{з.1} \cdot R_п}{R_{з.1} \cdot \eta_п + R_п \cdot n \cdot \eta_з} = \frac{26{,}6 \cdot 51{,}7}{26{,}6 \cdot 0{,}27 + 51{,}7 \cdot 16 \cdot 0{,}47} = 3{,}47 \text{ Ом}$$

3,47<4 $\Rightarrow$ умова забезпечення електробезпеки персоналу виконується.

Таким чином, остаточна кількість заземлювачей 15 шт.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Підвищення працездатності людини і збереження його здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень – це поєднання температури, вогкості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і пітовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

В приміщенні для виконання робіт операторського типу, пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (див. таблицю 4.1).

Таблиця 4.1 - Оптимальні параметри мікроклімату в робочій зоні виробничого приміщення для категорії робіт 1

Період року	Температура, оС	Відносна вогкість %	Швидкість руху повітря, м/с
Холодний	22.24	40.60	0,1
Теплий	23.25	40.60	0,1

Оскільки в приміщенні немає джерел виділення шкідливих речовин, можна використовувати природну вентиляцію. Площа приміщення складає 32 м2 . Для забезпечення прийнятних параметрів мікроклімату в приміщенні з такою площею можна використовувати 1 кондиціонер типу БК-2000.

Спектр випромінювання монітора комп'ютера включає рентгенівську, ультрафіолетову, інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння нехтує мала, оскільки цей вид випромінювання поглинається речовиною екрану.

Для зниження дії електромагнітного випромінювання пропонується захист часом і відстанню. Захист часом передбачає обмеження часу перебування людини в зоні дії полів. Тривалість роботи на ПЕОМ повинна складати не більше 3.5–4.5 години.

Також необхідно забезпечити раціональне освітлення в робочому приміщенні. В проекті, що розробляється, передбачається використовувати суміщене освітлення. В світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи володіють високою світловою віддачею до 75 Лам/Вт і більш, тривалим терміном служби до 10000 годин, спектральним складом випромінюваного світла, близьким до сонячного.

Зорова робота оператора ПЕВМ відповідно до ДБН В.2.5-28-2018 [25] відноситься до розряду Va з світловим потоком Фл=3120 кожна. Нормована освітленість на робочому місці (Ен) при загальному освітленні складає 200 лк.

Проведемо розрахунок кількості світильників в робочому приміщенні завдовжки a=6 м, шириною b=3 м, заввишки c=4 м. Формула розрахунку

штучного освітлення при горизонтальній робочій поверхні методом світлового потоку (4.7):

$$Фл = \frac{Ен \cdot S \cdot Z \cdot K}{N \cdot U \cdot M} \quad (4.7)$$

де *Фл* – світловий потік, Лм;

Ен – нормована освітленість;

S – площа підлоги, кв.м;

Z=1.1–1.3 - поправочний коефіцієнт світильника (для стандартних світильників);

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників;

N – число світильників;

U=0.55–0.6 – коефіцієнт використовування, залежний від типу світильника, показника індексу приміщення і др.;

M – число ламп в світильнику.

З формули (4.7) виразимо N і визначимо кількість світильників для даного приміщення:

$$N = \frac{Ен \cdot S \cdot Z \cdot K}{Фл \cdot U \cdot M}$$

$$N = \frac{200 \cdot 18 \cdot 1{,}2 \cdot 1{,}5}{3120 \cdot 0{,}6 \cdot 2} = 1{,}7 \quad (4.7)$$

Виходячи з цього, рекомендується використовувати 2 світильники. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. (4.1).

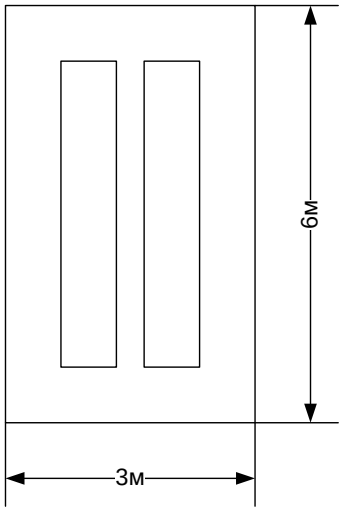


Рисунок 4.1 – Схема розташування світильників

Зниження шуму можна добитися раціонально розпланувавши приміщення, установкою устаткування на спеціальні амортизуючи прокладки. Згідно вимогам ДСН 3.3.6.037-99 [26] рівні звуку не повинні перевищувати 50 дБ.

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби передбачаються використовувати спокійні колірні поєднання і покриття, що не дають відблисків. Від електромагнітного випромінювання, витікаючого від ПЕОМ, використовуються захисні екрани.

Для забезпечення чистоти повітря і відповідних мікрокліматичних умов пропонується застосувати приточування-витяжну вентиляцію. Для зменшення дії шкідливих речовин і загазованості для роботи з розплавленими матеріалами робоче місце забезпечується примусовою витяжною вентиляцією. Цей метод забезпечує притоку потрібної кількості свіжого повітря (30 мм^3 /ч на одного працюючого).

Кількість повітря, яка необхідна подавати в приміщення для забезпечення необхідних параметрів повітряного середовища, визначається на підставі кількість тепла, вологи і шкідливих речовин, що поступають в

приміщення, а також враховуючи видалення повітря місцевими відсмоктуваннями від устаткування, загальнообмінною вентиляцією.

4.4 Рекомендації по пожежній профілактиці

Пожежі представляють небезпеку для життя людини і зв'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що спричиняє за собою порушення ходу технологічного процесу.

Горючими матеріалами в приміщенні, де розташовані ПЕОМ, є:

– поліамід - матеріал корпусу мікросхеми. Горюча речовина. Температура самозапалення 420 ^{o}C, енергія запалення 2мДж;

– полівінілхлорид - ізоляційний матеріал. Горюча речовина. Температура самозаймання 480 єС, енергія запалення 50мДж;

– склостоліт ДЦ - матеріал друкарської платні. Складногорючий матеріал;

– пластикат кабельний No.489 - матеріал ізоляції кабелю. Складногорючий матеріал. Температура самозаймання 1500 єС;

– плита деревостружкова - будівельний і обробний матеріал, матеріал з якого виготовлені меблі. Складнозапалений матеріал. Показник горючості 1.8;

– папір – довідкова і робоча документація, література. Горючий матеріал. Показник горючості більше 2.1.

Відповідно до ДСТУ Б В.1.1-36:2016 [27] приміщення відноситься до категорії В (пожежовибухонебезпечної).

Джерелами запалення можуть бути:

– іскри при замиканні і розмиканні ланцюгів;

– іскри і дуги коротких замикань;

– перегріви від тривалого перевантаження і наявності перехідного опору;

– розряди статичної електрики.

Для того, щоб зупинити реакцію горіння, порушують умови її виникнення і підтримки. Звичайно для гасіння використовуються порушення двох основних умов сталого стану – пониження температури і режим руху газів. Пониження температури може бути досягнутий шляхом введення речовин, які поглинають багато тепла в результаті випаровування і дисоціації (наприклад, вода, порошки).

При повному тому, що згоряє органічних сполук утворюються С, SO, Н Про, N, а при тому, що згоряє неорганічних з'єднань – оксиди. Залежно від температури плавлення і тривалості реакції можуть знаходиться або у вигляді розплавів (Al O, Ti O), або підійматися в повітря у вигляді диму (P O, Na Про, MgO).

Склад продуктів неповного згоряє горючих речовин складений і різноманітний. Це можуть бути горючі речовини:

– Н, З, СН;

– атомарний водень і кисень;

– різні радикали – ВІН, СН .

Продуктами неповного згоряє можуть бути також оксиди азоту, спирти, альдегіди, кетони і високотоксичні з'єднання, наприклад, синильна кислота.

Для захисту персоналу від дій небезпечних і шкідливих чинників пожежі проектом передбачено застосування промислового фільтруючого протигаза з коробкою марки В (жовтий).

До системи запобігання пожежі відносяться: запобігання утворення горючого середовища і освіти в горючому середовищі джерел запалення, забезпечення пожежебеспеки устаткування.

Щоб запобігти пожежі в обчислювальних центрах, проектом пропонується виконання наступних вимог:

– електроживлення ЕОМ має автоматичне блокування відключення електроенергії на випадок перегріву системи, що може бути результатом зупинки системи охолоджування і кондиціонування;

– система вентиляції обчислювальних центрів обладнується блокуючими пристроями, що забезпечують її відключення на випадок пожежі. Система обладнується вогнеперегороджуючими клапанами;

– застосування устаткування, що задовольняє вимогам електростатичної іскробезпеки [22];

– після закінчення роботи, перед закриттям приміщення, всі електроустановки і персональні комп'ютери відключаються від сіті електроживлення;

– в приміщеннях обчислювальних центрів забороняється:

1) влаштовувати електророзетки на основах, що згоряють;
2) використовувати синтетичні доріжки і килими;
3) користуватися побутовими електронагрівальними приладами;
4) захаращувати евакуаційні виходи і проходи;
5) влаштовувати на вікнах глухі грати;
6) залишати без нагляду включену в електромережу апаратуру, що використовується для вимірювань і нагляду.

Для протипожежного захисту проектом пропонується обладнати приміщення площею 18 м2, яке відноситься до категорії В, автоматичною протипожежною сигналізацією із застосуванням датчиків сповіщення РІД-1 (оповіщувач димовий іонізаційний) в кількості 1 штуки і застосовується в первинних засобах пожежегасінні. Площа контрольована оповіщувачем 150 м2.

Крім того, необхідно проводити навчання робочого персоналу правилам пожежної безпеки.

Розрахуємо вірогідність виникнення пожежі у виробничому приміщенні у разі запалювання транзистора:

$$Q = л \cdot T \cdot Ркз/отк \cdot Qвоспл \cdot Рзащ \qquad (4.8)$$

де л – інтенсивність відмов пожежеопасних EPI;

T – час роботи пожежеопасного EPI за оцінюваний інтервал часу;

Ркз/отк - умовна вірогідність виходу EPI в стан короткого замикання при його відмові;

Qвоспл - вірогідність запалювання EPI, що знаходиться в стані короткого замикання;

Рзащ – вірогідність відмови захисту пожежеопасного EPI. Якщо захист відсутній, Рзащ приймається рівній 1.

Вірогідність виникнення пожежі у разі запалювання транзистора:

$$Q = 1 \cdot 10^{-6} \cdot 1 \cdot 10^{-4} \cdot 0.1 \cdot 1 \cdot 10^{-4} = 1 \cdot 10^{-15}$$

Розрахована вірогідність виникнення пожежі значно менше допустимої, яка складає $1 \cdot 10^{-6}$.

В даному розділі були проаналізовані небезпечні і шкідливі виробничі чинники, що роблять вплив на персонал, розроблені заходи щодо техніки безпеки, заходу, забезпечуючи виробничу санітарію і гігієну праці, а також заходи щодо пожежної профілактики.

ВИСНОВКИ

У рамках дипломної роботи був розроблений і реалізований «спам» фільтр за допомогою використання нейронної мережі. У ході роботи було зроблено багато помилок та неоднозначності, але з часом були внесені виправлення.

Вивчення нейронних мереж це щось нове для мене. Мені подобаєється, що нейронні мережі використовуються у багатоьх напрямах та развиваються з кожним днем все більше і більше.

У моїй роботі є як плюси так і мінуси, на мою думку. До плюсів слід віднести: всі повідомлення, які мають заборонене слово чи фразу попадають до спаму; достатньо швидка робота. До мінусів: чим більший буде словник, тим довше буде проходити процес навчання нейронної мережі; можливі похибки при аналізі тексту (лист може мати заборонене слово, але буде вважатися, як нормальне повідомлення); постійне оновлення словника, якщо буде потрібність.

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливи та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп’ютері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) Згорткова нейронна мережа [Електронний ресурс]. Режим доступу: http://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа.

2) Іван Гудфелов, Йоша Бенгіо, Арон Коурвілле. «Машинне навчання» MIT Press, 2016. Режим доступу: http://www.deeplearningbook.org.

3) Ліла В.Б. Алгоритм та программна реализація адаптивного метода навчання штучних нейронних мереж// Инженерний вестн. Дона, 2012.

4) Девід Е. Голберг «Дзен та мистецтво генетичного алгоритму». 3rd International Conference on Genetic Algorithms, pp. 80-85.

5) Шаблони проектування [Електронний ресурс]. Режим доступу: http://travelscode.com/shabloni-proektuvannya-pochatok/

6) Герберт Шилдт - Java, повний посібник, 9-те видання [Текст]: переклад з англ. - М. : ООО «І.Д. Вільямс», 2012.

7) Вікіпедія - Java [Електронний ресурс] режим доступу: https://uk.wikipedia.org/wiki/Java

8) Роберт Каллан - "Основные концепции нейронных сетей". “Вильямс", 2003. - 288с.

9) Зорін Ю. М. Конспект лекцій з предмету «Комп`ютерні системи штучного інтелекту» [Електронний ресурс] / Ю. М. Зорін – Режим доступу до ресурсу: http://m.scs.kpi.ua/course/view.php?id=107

10) Бодянський Є. В., Кулішова Н. Є., Руденко О. Г. Рекурентна прогнозуюча штучна нейронна мережа: архітектура та алгоритми навчання // Адаптивні системи автоматичного управління. – Дніпропетровськ: Системні технології, 1999. – Вип. 2 (22). – С. 129-137.

11) Моделі нейронних мереж. – Режим доступу: http://techn.sstu.ru/kafedri/подразделения/1/MetMat/Terin/neiro/neiro.htm

12) Моделі нейронних мереж. – Режим доступу: https://studme.com.ua/1246122010028/neural/models.htm

13) Галушкин А.И. Теория нейронных сетей. / А.И. Галушкин - М.: ИПРЖР, с2010. – 348 с.

14) Тархов Д.А. Нейронные сети. Модели и алгоритмы [Текст]: Радиотехника, 2010. – 82 с.

15) Лисе А.А., Степанов М.В. Нейронные сети и нейрокомпьютеры. / А.А. Лисе, М.В. Степанов [Текст]: ГЭТУ. - СПб., 2009. 64 с.

16) Комашинский В.И. Смирнов Д.А. Внедрение в нейро-информационные технологии [Текст]: СПб., 1999.

17) Горбань А.Н. Нейронные сети на персональном компьютере / А.Н. Горбань, Д.А. Россиев. – Новосибирск: Наука, 2006. – 230 с.

18) Борисов Ю.И. Нейросетевые методы обработки информации и средства их программно-аппаратной поддержки / Борисов Ю.И, Кашкаров В.М., Сорокин С.А. // Открытые системы. – 1997.– № 4. – С. 38 – 40.

19) Іван Гудфелов, Йоша Бенгіо, Арон Коурвілле. «Машинне навчання» MIT Press, 2016. Режим доступу: http://www.deeplearningbook.org.

20) Ліла В.Б. Алгоритм та программна реализація адаптивного метода навчання штучних нейронних мереж// Инженерний вестн. Дона, 2012.

21) ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих. Режим доступу: https://dnaop.com/html/34094/doc-ДСН_3.3.6.042-99

22) ДСТУ Б А.3.2-13:2011Система стандартів безпеки праці. Будівництво. Електро безпечність. Загальні вимоги. Режим доступу- : http://online.budstandart.com/ru/catalog/doc-page?id_doc=27973

23) ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин Режим Доступу- : https://zakon.rada.gov.ua/rada/show/v0007282-98

24) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров`я працівників під час роботи з екранними пристроями Міністерство доходів і зборів України Наказ від 05.09.2013 р. № 443 "Про затвердження Примірної

інструкції з охорони праці під час експлуатації електронно-обчислювальних машин") Режим доступу- : https://zakon.rada.gov.ua/laws/show/%D0%BD%D0%BF%D0%B0%D0%BE%D0%BF_0.00-7.15-18

25) ДБН В.2.5-28-2018. Природне і штучнеосвітлення. Режим доступу - : https://okna.ua/img_all/oknaua/dbn-V-2-5-28-2018-ed.pdf

26) ДСН 3.3.6.037-99 Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку. Режим Доступу-: https://zakon.rada.gov.ua/rada/show/va037282-99

27) ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: www. URL: https://dbn.co.ua/load/normativy/dstu/dstu_b_v_1_1_36/5-1-0-1759

Додаток А

Комп’ютерна презентація

Інформаційна система класифікації повідомлень електронної пошти

Виконав: ст.гр. КІ-16з Якимчук Н.А.

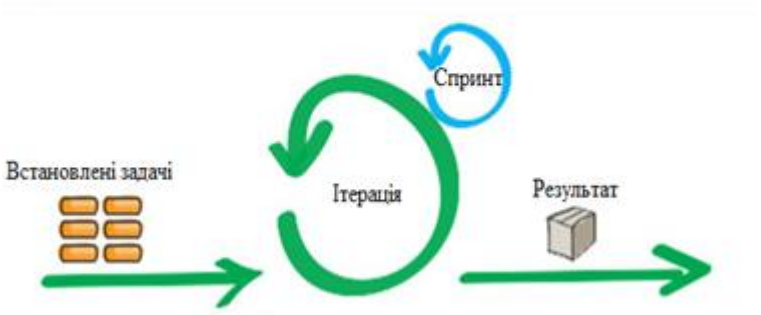
Керівник: проф. Кривуля Г.Ф.

Актуальність

Класифікація нейронних мереж

Характер навчання	Налаштування ваг	Тип вхідної інформації	Модель мережі
З вчителем	Фіксоване	Аналогова	Прямого поширення
Без вчителя	Динамічне	Двійкова	Зворотного поширення (рекурентні)
З підкріпленням			Самоорганізовані карти

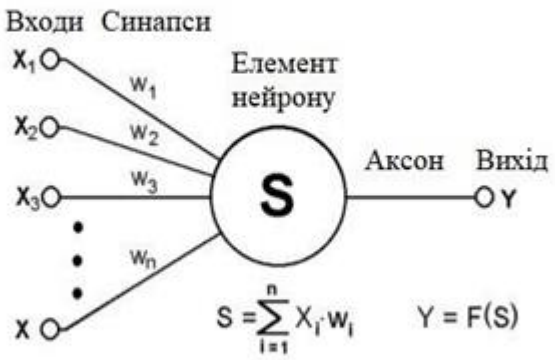
Процес роботи за метологією Scrum



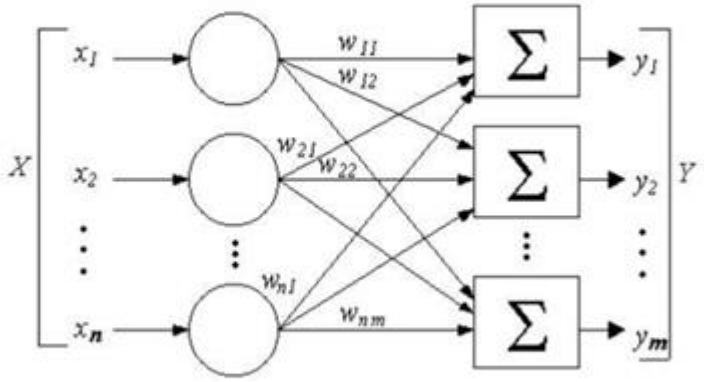
Постановка задачи

- ✓ провести аналіз існуючих спам фільтрів, які використовують Google, Yahhoo, Bing;
- ✓ розробити сервер, який буде підключатися до поштового сервісу Gmail та брати звідти непрочитані листи;
- ✓ організувати підтримання сесії з поштовим сервісом упродовж роботи сервера;
- ✓ навчити нейронну мережу відрізняти текст у листах;
- ✓ розробити метод взаємодії нейронної мережі з листами для їх аналізу та детектування заборонених;
- ✓ розробити метод переміщення заборонених листів до папки «Спам».

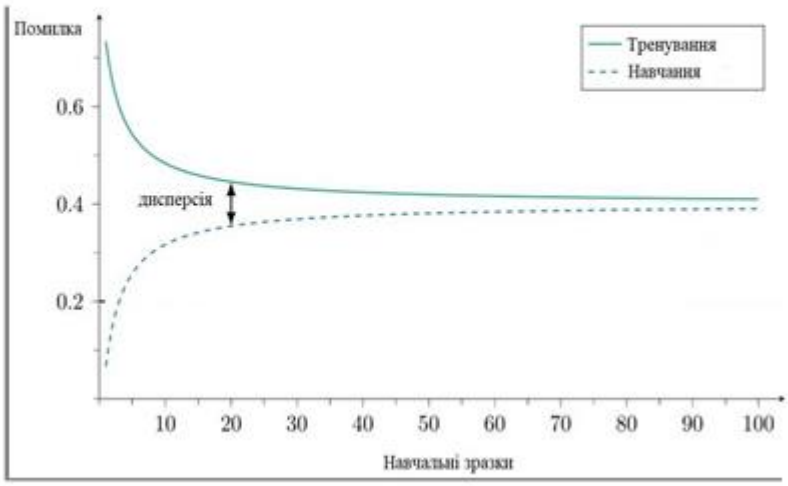
Модель штучного нейрона



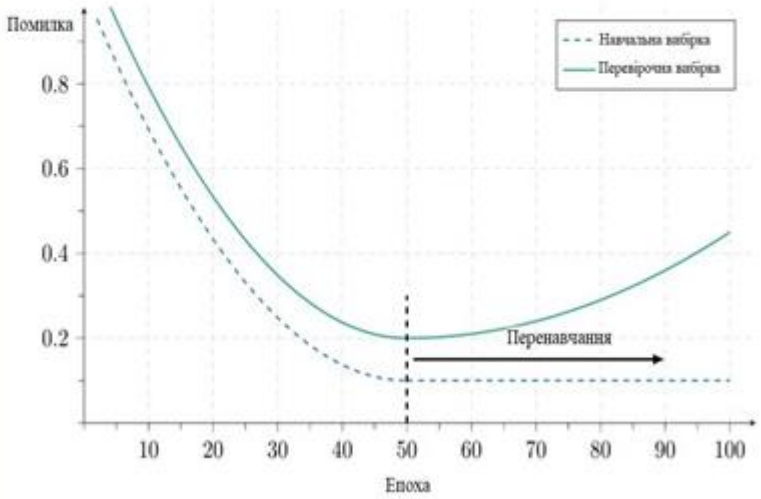
Проста одношарова штучна нейронна мережа



Процес навчання мережі



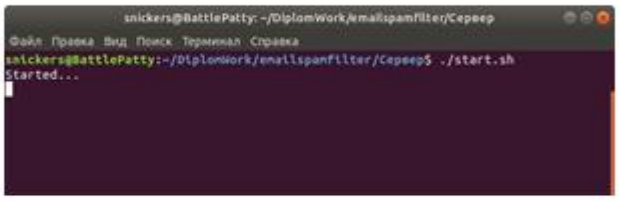
Проблема перенавчання мережі



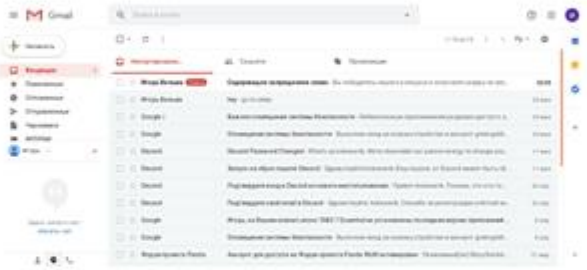
Stanford NLP


```
package org.emailspamfilter.textclassifier;

import java.util.List;
import java.util.Optional;

import edu.stanford.nlp.simple.Sentence;

public class EmailTextClassifier {

    private static EmailTextClassifier emailTextClassifierSingleton = null;

    private EmailTextClassifier() {
        super();
    }

    public static EmailTextClassifier getInstance() {
        if(emailTextClassifierSingleton == null) {
            emailTextClassifierSingleton = new EmailTextClassifier();
        }
        return emailTextClassifierSingleton;
    }

    public Optional<Integer> mailLanguageClassifier(String[] args) {

        String text = args[2];

        Sentence singleSentence = new Sentence(text);
        singleSentence.regexner("/home/gabriel/Projects/emailspamfilter/emailspamfilter_ner.txt", true);

        //
        List<String> emailSubject = singleSentence.nerTags("SPAM");

        List<String> emailBody = singleSentence.nerTags("SPAM");

        List<String> emailTextAttachments = singleSentence.nerTags("SPAM");

        List<String[]> spamEmails =
                emailSubject.stream()
                        .flatMap(i -> emailBody.stream()
                                               .filter(j -> text.indexOf(j) - text.indexOf(i) > 0)
                                               .flatMap(j -> emailTextAttachments.stream()
                                                                                 .map(k -> new String[]{i, j, k})
                                                        )
                                )
                        .collect(Collectors.toList());
        return Optional.of(spamEmails);
        */

        List<String> emailBody = singleSentence.nerTags("SPAMITEM");

        if(!emailBody.isEmpty())
            return Optional.of(new Integer(args[0]));
        else
            return Optional.empty();

    }
```

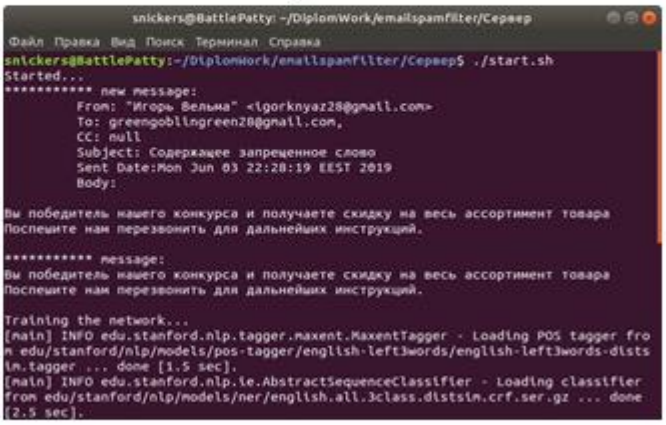
Тестування розробленої моделі. Запуск серверу



Тестування розробленої моделі. Зображення скриньки



Тестування розробленої моделі. Логування та процес навчання мережі



Тестування розробленої моделі. Логування (відображення того, що лист помічений, як спам)

Висновки

У рамках дипломної роботи був розроблений і реалізований «спам» фільтр за допомогою використання нейронної мережі. У ході роботи було зроблено багато помилок та неоднозначності, але з часом були внесені виправлення.

Плюси технології: всі повідомлення, які мають заборонене слово чи фразу попадають до спаму; достатньо швидка робота.

Мінуси технології: чим більший буде словник, тим довше буде проходити процес навчання нейронної мережі; можливі похибки при аналізі тексту (лист може мати заборонене слово, але буде вважатися, як нормальне повідомлення); постійне оновлення словника, якщо буде потрібність.