

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

магістра

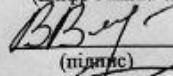
(освітній ступінь)

на тему: Методи та засоби опрацювання природної мови з  
використанням нейронних мереж з метою розпізнавання власних назв

Виконав: студент (ка) 6 курсу, групи СІм-61  
спеціальності 123

«Комп'ютерна інженерія»

(цифр і назва спеціальності)



(підпис)

Васьков В. О.

(прізвище та ініціали)

Керівник

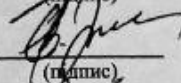


(підпис)

Лупенко С. А.

(прізвище та ініціали)

Нормоконтроль



(підпис)

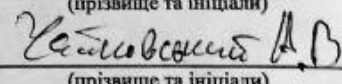
Тиш Є. В.

(прізвище та ініціали)

Рецензент



(підпис)



(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування повного навчального закладу)

Факультет Комп'ютерно-інформаційних систем і програмної інженерії  
Кафедра Комп'ютерних систем та мереж  
Освітній ступінь магістр  
Напрямок підготовки \_\_\_\_\_  
Спеціальність 123 „Комп'ютерна інженерія“

**ЗАТВЕРДЖУЮ**

Завідувач кафедри К.С. Сухівська Р.М.  
« 30 » \_\_\_\_\_ 2012 р.

**ЗАВДАННЯ  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Васькову Владиславу Александровичу

1. Тема роботи Методи та засоби оптимізації  
прискорення роботи з використанням нейронних мереж  
з метою розширення власних мереж

Керівник роботи Гуденко Сергій Анатолійович, доцент кафедри інформатичних наук

Затверджені наказом по університету від «27» 09 2019 року № 417-854

2. Термін подання студентом роботи 23 грудня 2019 р.  
3. Вихідні дані до роботи процес оптимізації прискорення  
роботи з метою розширення власних мереж

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)  
Вивчення предметної області оптимізації  
прискорення роботи з використанням нейронних мереж  
власних мереж, дослідження методів та засобів оптимізації  
прискорення роботи з використанням нейронних мереж  
власних мереж, розроблення алгоритмів для оптимізації  
прискорення роботи з метою розширення власних мереж

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)  
Світлолінійні малюнки алгоритмів наукової роботи  
розроблення алгоритмів оптимізації прискорення роботи  
власних мереж, дослідження методів оптимізації  
прискорення роботи з використанням нейронних мереж  
власних мереж, розроблення алгоритмів оптимізації  
прискорення роботи з метою розширення власних мереж

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Обґрунтув. екон. ефект. Екологія	Кирич Н.Б.	<i>[Підпис]</i>	<i>[Підпис]</i>
Профоне праця	Месенга О.М.	<i>[Підпис]</i>	<i>[Підпис]</i>
Безпека в НС	Осудьова Т.К.	21.11.19	24.11.19
	Струцьк В.С., ст. викл. кадр. ОУ	<i>[Підпис]</i>	<i>[Підпис]</i>

7. Дата видачі завдання 30.09.19

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Отримання завдання	30.09.19	виконано
2.	Аналіз завдання	1.10.19 - 6.10.19	виконано
3.	Дослідження предметної області опрацювання природної мови з використанням нейронних мереж	7.10.19 - 17.10.19	виконано
4.	Аналіз і дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж	18.10.19 - 1.11.19	виконано
5.	Реалізація програмного забезпечення для опрацювання природної мови з метою розпізнавання власних назв	2.11.19 - 28.11.19	виконано
6.	Вдосконалення економічної тривалості	29.11.19 - 1.12.19	виконано
7.	Вирозуміння проци та безпека в надзвичайних ситуаціях	2.12.19 - 8.12.19	виконано
8.	Висновки		виконано
9.	Варіювання розмовальної частини	8.12.19 - 16.12.19	виконано
10.	Варіювання графічної частини	17.12.19	виконано
11.	Фіналізування роботи	19.12.19	виконано
12.	Захист роботи	23.12.19	

Студент *[Підпис]* Васьков В.О.  
 (підпис) (прізвище та ініціали)

Керівник роботи *[Підпис]* Муренко С.А.  
 (підпис) (прізвище та ініціали)

## АНОТАЦІЯ

Методи та засоби опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв // Дипломна робота // Васьков Владислав Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІМ-61 // Тернопіль, 2019 // с. – 110, рис. – 28, табл. – 8, аркушів А1 – 10, додат. – 1, бібліогр. – 52.

Ключові слова: РОЗПІЗНАВАННЯ ВЛАСНИХ НАЗВ, ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ, НЕЙРОННА МЕРЕЖА, WORD2VEC, GLOVE, ТЕНЗОРНИЙ ПРОЦЕСОР.

Мета роботи полягає у обґрунтуванні та адаптації сучасних методів та засобів опрацювання природної мови в комп'ютерних системах для побудови програмної системи розпізнавання власних назв.

У дипломній роботі досліджено сучасні методи та засоби опрацювання природної мови з метою розпізнавання власних назв та встановлено їх переваги та недоліки.

Проаналізовано методи векторного представлення тексту, що використовуються для задач опрацювання природної мови з метою розпізнавання власних назв, що дало змогу обґрунтувати математичне забезпечення програмної системи для розпізнавання власних назв з використанням методів та сучасних засобів рекурентних нейронних мереж.

На основі попередньо обґрунтованого математичного забезпечення розроблено архітектуру програмної системи для розпізнавання власних назв.

Обґрунтовано бібліотеку TensorFlow.js та API Keras, як засобів розробки програмної системи для розпізнавання власних назв, що уможливило ефективну її реалізацію.

Здійснено оцінку якості роботи програмного забезпечення за такими параметрами, як: точність, повнота та F-міра.

## ANNOTATION

Methods and tools of natural language processing due to neural networks use aimed at proper names recognition // Master diploma thesis // Vaskov Vladyslav // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information System and Software Engineering, group CIM-61 // Ternopil, 2019 // p. – 110, fig. – 28, tab. – 8, sheets A1. – 10, addit. – 1, bibliography – 52.

Key words: PROPER NAME RECOGNITION, NATURAL LANGUAGE PROCESSING, NEURAL NETWORK, WORD2VEC, GLOVE, TENSOR PROCESSING UNIT.

The purpose of the work is to substantiate and adapt modern methods and means of natural language processing in computer systems to build a software system for proper name recognition.

In the thesis investigates modern methods and means of natural language processing for the purpose proper names recognition and their advantages and disadvantages.

The methods of vector representation of text used for natural language processing tasks for the purpose of recognition of proper names are analyzed, which made it possible to substantiate the mathematical software of the system for recognition of proper names using methods and modern means of recurrent neural networks.

Based on pre-grounded mathematical software, a software system architecture was developed for recognizing proper names.

The TensorFlow.js library and API Keras have been substantiated as a means of developing a software system for identifying their own names, which has made it possible to implement it effectively.

The quality of the software was evaluated using such parameters as precision, recall and F-measure.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП .....	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ .....	11
1.1. Огляд предметної області.....	11
1.2. Дослідження особливостей опрацювання природної мови.....	13
1.2.1. Токенізація.....	16
1.2.2. Стоп-слова .....	16
1.2.3. Стемінг та лематизація .....	17
1.2.4. Виділення частин мови.....	17
1.2.5. Розпізнавання іменованих сутностей.....	18
1.2.6. Модель «торба слів» (BOW).....	18
1.2.7. TF-IDF .....	20
1.2.8. N-Gram .....	21
1.3. Дослідження побудови нейронних мереж для задач опрацювання природної мови.....	23
1.3.1. Загальні відомості про нейронні мережі .....	23
1.3.2. Активаційна функція нейронів.....	27
1.3.3. Рекурентна нейронна мережа .....	27
1.3.4. Висновки до розділу .....	30
РОЗДІЛ 2 АНАЛІЗ І ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ.....	31
2.1. Аналіз методів опрацювання природної мови.....	31
2.1.1. Метод Word2Vec .....	31
2.1.2. Метод Glove.....	33
2.1.3. Переваги та недоліків методів Word2Vec та Glove .....	35
2.2. Дослідження математичного забезпечення методів роботи нейронних мереж для задач опрацювання природної мови.....	36
2.2.1. Дослідження математичного забезпечення методу Word2Vec .....	36
2.2.2. Дослідження математичного забезпечення методу Glove.....	42
2.3. Аналіз апаратних засобів опрацювання природної мови з використанням нейронних мереж.....	44
2.3.1. Тензорний процесор.....	44

2.3.2. Структура тензорного процесора .....	46
2.3. Висновки до розділу .....	51
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З МЕТОЮ РОЗПІЗНАВАННЯ ВЛАСНИХ НАЗВ.....	52
3.1. TensorFlow .....	52
3.2. Keras.....	54
3.3. Опис навчання та функціонування розроблюваного програмного забезпечення .....	55
3.4. Оцінка якості розробленого програмного забезпечення .....	62
3.5. Висновки до розділу .....	64
РОЗДІЛ 4 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ.....	65
4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР .....	65
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи.....	67
4.3. Розрахунок витрат на електроенергію .....	71
4.4. Розрахунок витрат на матеріали.....	72
4.5. Розрахунок суми амортизаційних відрахувань.....	72
4.6. Обчислення накладних витрат.....	73
4.7. Складання кошторису витрат та визначення собівартості НДР .....	74
4.8. Розрахунок ціни НДР.....	74
4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень.....	75
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	77
5.1. Охорона праці.....	77
5.2. Аварія на радіаційно небезпечних підприємствах із джерелом іонізуючого випромінювання, їх несанкціоноване перевезення, зберігання, використання та виявлення в непристосованих місцях .....	81
5.3. Умови і фактори виникнення пожеж та суцільних пожеж. Заходи попередження пожеж та суцільних пожеж.....	86
РОЗДІЛ 6 ЕКОЛОГІЯ .....	93
6.1. Застосування екологічних знань у різних галузях соціально-політичного життя.....	93
6.2. Вимоги до моніторів (ВДТ) і ПЕОМ .....	96
ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	99
ДОДАТОК А ТЕЗИ КОНФЕРЕНЦІЙ.....	104

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ALU	Arithmetic logic unit
API	Application programming interface
ASIC	Application-specific integrated circuit
BOW	Bag of words
CBOW	Continuous bag of words
CISC	Complex instruction set computing
CNN	Convolutional neural network
CPU	Central processing unit
GPU	Graphics processing unit
MXU	Matrix multiply unit
NER	Named-entity recognition
NLP	Natural-language processing
RISC	Reduced instruction set computing
RNN	Recurrent neural network
SM	Streaming multiprocessor
TF	Tensor flow
TF-IDF	Term Frequency – Inverse Document Frequency
TPU	Tensor processing unit
UB	Unit buffer



## ВСТУП

**Актуальність теми.** Ми живемо у час, коли об'єми згенерованої людством інформації є більшими, чим коли небудь й кількість таких даних збільшується з кожним днем. Однак велику користь із цих даних можна отримати лише, якщо правильно здійснити опрацювання й аналіз цих даних.

На сьогодні щосекунди по всьому світу генеруються гігабайти нових даних різного виду: створюються нові знімки, відеозаписи, записується сотні відгуків під записами у соціальних мережах й багато іншого. Й більша частина цих даних у «сирому» вигляді є практично безкорисливою. Щоб отримати з цих даних певну користь, їх потрібно відфільтрувати і опрацювати. У часи, коли технології ще не були настільки розвинуті, все це здійснювалось вручну. На це витрачалися години, дні, неділі, а й інколи місяці. А, якщо врахувати, що раніше й самої інформації для опрацювання було у рази менше, то неважко зрозуміти, що сьогодні опрацьовувати такі об'єми інформації вручну просто неможливо. Тому було розроблено велика кількість методів і засобів, які дозволяють здійснювати це з допомогою комп'ютерної техніки.

**Мета і завдання дослідження.** Обґрунтування та адаптація сучасних методів та засобів опрацювання природної мови в комп'ютерних системах для побудови програмної системи розпізнавання власних назв.

Для досягнення поставленої цілі, необхідно розв'язати наступні задачі:

- дослідити сучасні методи та засоби опрацювання природної мови з метою розпізнавання власних назв та встановити їх переваги та недоліки;
- проаналізувати методи векторного представлення тексту, що використовуються для задач опрацювання природної мови з метою розпізнавання власних назв;
- обґрунтувати математичне забезпечення програмної системи для розпізнавання власних назв з використанням методів та сучасних засобів рекурентних нейронних мереж;

- проаналізувати сучасні засоби розробки програмних систем для розпізнавання власних назв з використанням нейронних мереж;
- реалізувати програмне забезпечення з метою розпізнавання власних назв;
- здійснити оцінку якості роботи розробленого програмного забезпечення.

**Об’єкт дослідження** – процес опрацювання природної мови з метою розпізнавання власних назв.

**Предметом дослідження** є методи, програмні і апаратні засоби для опрацювання природної мови з метою розпізнавання власних назв.

**Методи дослідження.** Для виконання задач дипломної роботи використано наступні методи: теоретико-емпіричний, системного аналізу, теорії проектування нейронних мереж, й також математичного та комп’ютерного моделювання.

**Наукова новизна одержаних результатів:**

- проведено компаративний аналіз сучасних методів та засобів опрацювання природної мов, що дало змогу обґрунтувати на основі кількісних та якісних показників найбільш ефективних із них для вирішення задачі розпізнавання власних назв;
- серед множини методів векторного представлення тексту для задач опрацювання природної мови з метою розпізнавання власних назв обґрунтовано використання та адаптацію методів рекурентних нейронних мереж, як найбільш ефективних, що становлять основу математичного забезпечення програмної системи для розпізнавання власних назв;
- з використанням методів інженерії якості програмного забезпечення, а саме із застосування критеріїв таких, як повнота, точність та F-міра здійснено валідацію програмної системи для розпізнавання власних назв.

**Практичне значення отриманих результатів.** Розроблено програмне забезпечення, яке дозволяє опрацьовувати природню мову з метою розпізнавання власних назв.

**Публікації.** Результати роботи апробовано на VIII Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» м. Тернопіль 27-28 листопада 2019 року та VII науково-технічна конференція «Інформаційні моделі, системи та технології» м. Тернопіль 11-12 грудня 2019 року.

**Структура роботи.** Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається із вступу, шести розділів, висновків, списку використаних джерел та додатку. Обсяг роботи: пояснювальна записка – 110 аркушів формату А4, графічна частина – 10 аркушів формату А1.

## РОЗДІЛ 1

# ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

### 1.1. Огляд предметної області

Виділення іменованих сутностей (named entity recognition) – одне з ключових завдань отримання інформації з неструктурованих документів. Його основна ціль – знайти у тексті власні назви, ідентифікатори об'єктів певного типу [1].

Однією з перших робіт в даній області прийнято вважати статтю Лізи Рау (1991). Вона запропонувала використовувати евристичні підходи і набір правил для виділення назв компаній в тексті. З тих пір було запропоновано величезну кількість рішень і стратегій по розпізнаванні власних назв. Задача була представлена на секціях різних конференцій: Message Understanding Conference (MUC), Conference on Natural Language Learning (CoNLL), International Conference on Language Resources and Evaluation<sup>2</sup> (LERC).

Також, хороша оглядова робота була виконана Девідом Надю і Сатоші Секіном [2]. Автори провели докладний огляд методів, що використовуються в області виділення і розпізнавання сутностей, за період з 1991 по 2006 роки.

Під терміном іменована сутність (named entity) ми розуміємо об'єкт певного типу, що має ім'я, або ідентифікатор. Які типи виділяє система, визначається в рамках конкретного завдання. Для новинного домену зазвичай це люди (PER), місця (LOC), організації (ORG) й т. д. В загальному випадку на вхід системи надходить текст, на виході система повідомляє інформацію про положення імен в тексті та інформацію про класи, які їм відповідають. Зауважимо, що набір класів ми фіксуємо заздалегідь.

Існує безліч стратегій і підходів до вирішення поставленого завдання. Особливо популярні методи машинного навчання з учителем. Така система

спочатку тренується на множині розмічених прикладів, натреновану модель тепер можна використовувати на довільних даних. Так як тренувальна колекція, зазвичай, створюється вручну і має обмежений обсяг, в деякій мірі справедливо, що успіх систем розпізнавання сутностей залежить від наявності додаткових ресурсів: словників сутностей, додаткових тренувальних наборів, колекцій нерозмічених текстів і т. д. Також, користь тренувальної колекції з роками може зменшуватися, наприклад, в новинному тренувальному набору 2000 року навряд чи можна зустріти згадку компанії, що була створена, скажімо у 2019 році, а отже постає питання, чи зможе система коректно розпізнати сутність.

Виділення сутностей – один з ключових етапів попереднього опрацювання тексту для побудови більш складнішого програмного забезпечення, що працює з аналізом інформації.

Предметна область і жанр текстів, також мають сильний вплив на систему виділення сутностей в цілому. Проектування методів, стійких до зміни домену оброблюваних текстів, залишається важким завданням [3]. Особливо це впливає на системи, заснованих на правилах. Зміна предметної області, як правило, спричиняє погіршення від 10% до 40% в точності і повноті систем. Однак, Дауме Хал III у своїй роботі запропонував метод, який підвищує стійкість статистичного аннотатора до зміни домену, і продемонстрував його працездатність зокрема на завданні виділення сутностей.

Для різних предметних областей властиві різні класи іменованих сутностей. Часто в якості сутностей розглядають імена людей, назви організацій і місць [4]. Однак, цим набір класів не обмежується; тип різне (MISC) був введений на CoNLL і включив в себе широкий набір сутностей: найменування національностей, назви подій, тощо. Додаткове дроблення вищевказаних класів привело до створення великих ієрархій іменованих сутностей.

Виділення сутностей може бути корисно і в інформаційному пошуку. Відомо, що слово «Ягуар» може трактуватися у різних контекстах як назва

компанії, музичної групи, марки машин або просто назва живої істоти. Якщо відповідна система інформаційного пошуку знає, про яке із значень слова «Ягуар» йде мова в тому чи іншому документі, то в залежності від запиту видача сторінок може бути обмежена тільки певним типом сутностей. Попереднє виділення сутностей з величезної колекції документів дозволяє поліпшити їх організацію і дати перше уявлення про зміст документа. Все перераховане робить завдання про виділення іменованих сутностей досить корисним і цікавим не тільки з точки зору дослідників, але і з точки зору компаній, що займаються інтелектуальним аналізом даних, інформаційним пошуком і т. д.

## 1.2. Дослідження особливостей опрацювання природної мови

Природна мова – це мова, яку люди використовують для спілкування один з одним. Це спілкування може бути словесним або текстовим. Наприклад, розмови віч-на-віч, твіти, блоги, електронні листи, веб-сайти, SMS-повідомлення. Однак, на відміну від людей, комп'ютери не можуть так легко зрозуміти природну мову. Для перекладу природної мови у формат, зрозумілий комп'ютерам, потрібні застосовувати досить складні прийоми та методи [5].

Опрацювання природної мови (NLP) – галузь штучного інтелекту, яка допомагає комп'ютерам зрозуміти, інтерпретувати та маніпулювати людською мовою.

NLP допомагає розробникам організувати та структурувати знання для виконання таких завдань, як переклад, узагальнення, розпізнавання власних назв об'єктів, розпізнавання мовлення, сегментація тем тощо.

П'ять основних компонентів опрацювання природної мови (рис.1.1):

- морфологічний та лексичний аналіз;
- синтаксичний аналіз;
- семантичний аналіз;

- інтеграція дискурсу;
- прагматичний аналіз.

Морфологічний та лексичний аналіз – це створення словника, що містить всі слова та вирази. Він включає поділ тексту на абзаци, слова та речення. Окремі слова аналізуються за їх компонентами, а неслівні лексеми, такі як розділові знаки, відокремлюються від слів.

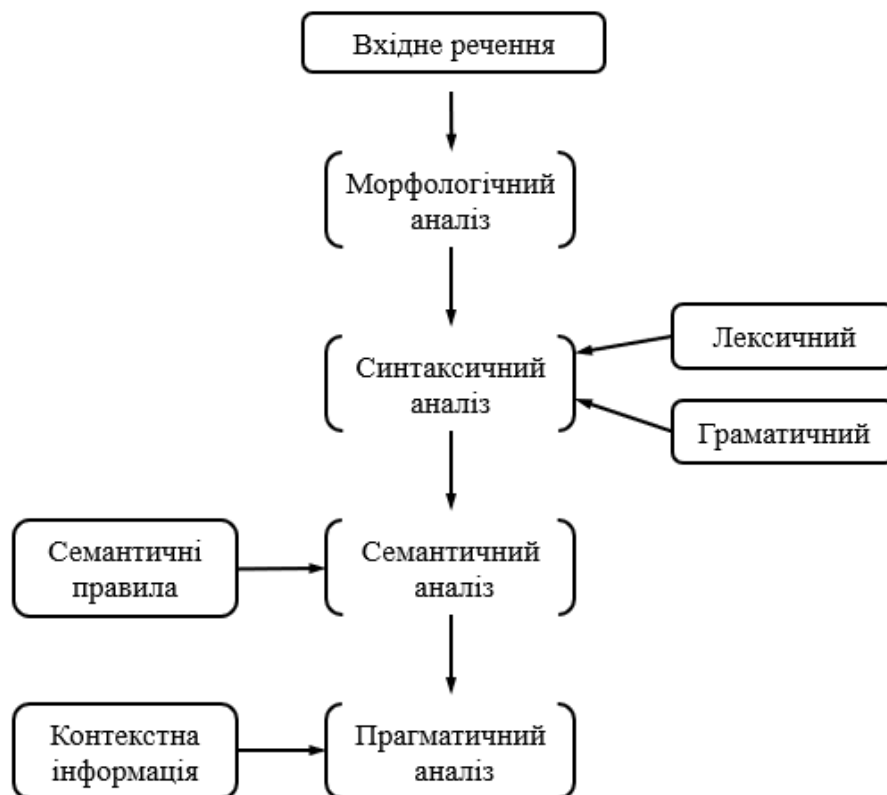


Рис.1.1. Основні компоненти обробки природних мов

Семантичний аналіз – це структура, створена синтаксичним аналізатором, який присвоює значення. Цей компонент переносить лінійні послідовності слів у структури. Він показує, як слова пов’язуються між собою. Семантика зосереджується лише на прямому значенні слів, словосполучень та речень. Це лише абстрагує значення словника чи реальне значення із заданого контексту. Структури, призначені синтаксичним аналізатором, завжди мають присвоєне значення.

Прагматичний аналіз стосується загального комунікативного та соціального змісту та його впливу на інтерпретацію. Це вміння абстрагувати або виводити змістовне використання мови в ситуаціях. У цьому аналізі основна увага завжди приділяється тому, що було сказано в переосмисленні на те, що мається на увазі.

Синтаксичний аналіз зосереджується на правильному впорядкуванні слів, що може вплинути на їх значення. Це передбачає аналіз слів у реченні шляхом дотримання граматичної структури речення. Слова перетворюються на структуру, щоб показати, як слова пов'язані між собою.

Інтеграція дискурсу – це відчуття контексту. Значення будь-якого слова, яке залежить від контексту речень у якому вживається. Може враховувати як контекст попереднього, так і наступного речення.

Система письма, що використовується для мови, є одним із вирішальних факторів у визначенні найкращого підходу до попереднього опрацювання тексту.

Системи письма можуть бути:

- логографічні: велика кількість окремих символів представляють слова;
- силабічні: окремі символи являють собою склади;
- алфавітні: окремі символи представляють звук.

Більшість систем письма використовують силабічну чи алфавітну систему [6].

Нижче наведено методи, що використовуються для процесу навчання NLP:

- машинне навчання;
- статистичні висновки.

Найпоширеніші методи нормалізації корпусу тексту в NLP:

- токенизація;
- стоп-слова;



- стемінг та лематизація;
- виділення частин мови;
- розпізнавання іменованих сутностей (PIC);
- застосування моделі «торба слів» (BOW);
- TF-IDF (term frequency - inverse document frequency);
- N-грама [7].

1.2.1. Токенізація. Токенізація – це процес ділення потоку символів на окремі значимі одиниці. Зазвичай токенізація – це перше завдання, яке виконується при обробці природніх мов. Токенізація може здійснюватися на двох рівнях: рівні слова та рівні речення.

Токенізація слів розділяє речення на окремі слова. Наприклад, для позначення речення «Я люблю подорожувати» ми отримаємо наступний набір слів: «я», «люблю» і «подорожувати».

Токенізація речень розділяє фрагмент тексту у документі на речення. Наприклад, розглянемо наступний текст. «Київ – столиця України. Він розташований на півночі України. Це прекрасне місто». Токенізація речень вищезазначеного документа поверне набір наступних речень:

S1 = «Київ – столиця України»;

S2 = «Він розташований на півночі України»;

S3 = «Це прекрасне місто».

1.2.2. Стоп-слова. Стоп-слова у природній мові – це слова, які не дають корисної інформації в заданому контексті. Для прикладу, такі слова як «і», «я» та «у» не передають ніякої інформації, пов'язаної з емоціями.

Наприклад, у реченні «Я сьогодні відчуваю себе щасливим», слово «Я» можна видалити, оскільки вони не дає жодної інформації, пов'язаної з емоціями. Однак, слово «Я» може бути важливим з інших причин, наприклад, для визначення того, хто відчуває себе щасливим. Не існує універсального

списку стоп-слів, який потрібно видалити, оскільки деякі фактично можуть надавати іншого смислового значення – це залежить, тільки від програмного забезпечення, що використовується.

В NLP кожне слово вимагає обробки. Тому зручно, що у нашому тексті є лише ті слова, які є важливими в заданому контексті. Це економить час на опрацювання природної мови і приводить до більш надійного «двигуна» для опрацювання NLP [7].

1.2.3. Стемінг та лематизація. Стемінг – це процес скорочення слів до основи, відкидаючи суфікси і закінчення. Наприклад, стемінг слів «швидко», «швидкий», «швидкі», все це призведе до «швидк», оскільки це не змінна частина слова. Наприклад, стемінг слів «вільна» призведе до «віл». Стемінг не враховує контексти слів.

З іншого боку, лематизація виконує подібне завдання, але вона враховує контекст під час зменшення слів. Лематизація є більш складною, оскільки вона виконує пошук словника для отримання точного слова, що містить семантичну інформацію. Для кожної частини мови така форма може різнитися. Наприклад, для іменників – це називний відмінок, для дієслів – інфінітивна форма дієслова.

Стемінг та лематизація дуже корисні для пошуку смислової подібності між різними текстами [5, 7].

1.2.4. Виділення частин мови. Для побудови змістовно-граматично правильного речення важливу роль відіграють частини мови. Упорядкування та спільне використання різних частин мови у реченні, роблять його граматично та семантично зрозумілим. Крім того, частини мови також є невід’ємною частиною ідентифікації контексту. Отож, нормалізація корпусу тексту з виділенням частин мови допомагає досягти цих завдань [5].

Нормалізація корпусу тексту з виділенням частин мови позначає слова відповідними частинами мови. Наприклад, «ноутбук, миша, клавіатура»

позначені як іменники. Так само «їдуть, грають» – це дієслова, а «добре» та «погано» позначаються як прикметники.

З першого погляду, процес нормалізації корпусу тексту з виділенням частин мови може здатися простим завданням, але насправді це не так. Для багатьох слів можна просто скористатися пошуком з допомогою словника, щоб віднести слова до тієї чи іншої частини мови, проте значна частина слів мають однакове написання, але різне значення і тому однакові слова можуть виступати різними частинами мови.

1.2.5. Розпізнавання іменованих сутностей. Розпізнавання іменованих сутностей (РІС) відноситься до процесу знаходження й класифікації об'єктів на заздалегідь визначені категорії, такі як особа, місцезнаходження, організація тощо.

Важливим застосуванням розпізнавання іменованих сутностей є тематичне моделювання, коли за допомогою інформації про сутності в тексті, тема документа може бути автоматично виявлена [7].

Вирішення задачі РІС часто поділяється концептуально на дві різні проблеми: виявлення імен, а також класифікація імен за типом сутності, до якої вони належать.

Перший етап, зазвичай, спрощується до вирішення проблеми сегментації: імена визначаються, як неперервні ділянки токенів без вкладеності, отож «готель Тернопіль» – це одне ім'я, не враховуючи той факт, що всередині цього імені слово «Тернопіль» є також іменем. Другий етап вимагає вибору онтології, за допомогою якої можна здійснити приналежність до того чи іншого типу сутності.

1.2.6. Модель «торба слів» (BOW). Модель «торба слів» (BOW) стосується методології, яка використовується для вилучення елементів з

текстових документів. Ці особливості можуть бути використані для виконання різних завдань, таких як алгоритмів машинного навчання.

Підхід BOW починається з побудови словникового запасу всіх унікальних слів, що зустрічаються у всіх документах навчального набору. Ця лексика слугує функцією вектора для навчального набору. Наприклад, розглянемо наступні документи:

D1 = «Київ – столиця України»;

D2 = «Він розташований на півночі України»;

D3 = «Київ – найбільше місто України».

Словник або побудова функцій векторів за допомогою наведених вище документів виглядатиме наступним чином: «Київ», «столиця», «України», «Він», «розташований», «на», «півночі», «найбільше», «місто».

Вигляд даних про навчання для вищезазначених документів зображено у табл.1.1.

Таблиця 1.1

### Особливості навчання з використанням моделі BOW

Документ	Київ	столиця	України	Він	розташований	на	півночі	найбільше	місто
D1	1	1	1	0	0	0	0	0	0
D2	0	0	1	1	1	1	1	0	0
D3	1	0	1	0	0	0	0	1	1

У навчальний набір (табл.1.1) вставляється один рядок для кожного документа. Для кожного атрибута (слова) у рядку встановлюється частота слова у відповідному документі. Якщо в документі слово не існує, то позначається 0.

Це і називається підходом BOW, оскільки послідовність слів у документів не враховується [5].

1.2.7. TF-IDF. TF-IDF є статистичним показником, який використовується для оцінки важливості слів у контексті. Обчислення значень TF-IDF полягає в тому, що тим словам, які частіше зустрічаються в одному документі і є загалом рідшими у всіх документах, слід надавати більше зважування, оскільки вони є більш важливими для класифікації. Щоб зрозуміти TF-IDF, розглянемо наступні три документи D1, D2 і D3:

D1 = «Київ – столиця України»;

D2 = «Він розташований на півночі України»;

D3 = «Київ розташований на півночі України, і є найбільшим містом України».

TF-IDF – це комбінація двох значень: TF (частота терміну) та IDF (зворотна частота документа).

Термін «частота» означає кількість разів, коли слово виникає в документі. У документі D1 термін «Київ» зустрічається один раз. Так термін «столиця» також зустрічається один раз. У D3 «України» зустрічається двічі, тому термін частоти для «України» становить 2 для D3. Тут важливо зазначити, що частота слова обчислюється на документ.

У деяких випадках, коли довжина документів змінюється, частота терміну для певного слова обчислюється, як:

$$F_T = \frac{K_C}{K_3}, \quad (1.1)$$

де  $F_T$  – частота терміну;

$K_C$  – кількість зустрічей заданого слова в документі;

$K_3$  – загальна кількість слів у документів.

Однак, для простоти, ми будемо використовувати лише кількість зустрічей слова в документі.

Зворотна частота документа для певного слова означає загальну кількість документів у наборі даних, поділену на кількість документів, у яких це слово існує. У документах D1, D2 і D3 слово «Київ» зустрічається в D1 і D3. Тож IDF для «Київ» буде 3/2.

Щоб зменшити вплив унікальності, загальноприйнятною практикою є використання логарифма значення IDF. Отож, остаточна формула для конкретного слова виглядатиме наступним чином:

$$IDF(\text{слова}) = \log\left(\frac{K_D}{K_{D.C.}}\right), \quad (1.2)$$

де  $K_D$  – загальна кількість документів;

$K_{D.C.}$  – кількість документів, що містять задане слово.

Для прикладу, спробуємо знайти значення IDF для слова «столиця». У нас є три документи: D1, D2 і D3, і слово «столиця» зустрічається лише в одному документі. Тож значення IDF згідно формули (1.2) для «столиця» буде:

$$IDF(\text{столиця}) = \log(3/1) = 0,477.$$

Величина TF-IDF – це добуток значень TF та IDF для певного терміна в документі. Для «столиця» значення TF-IDF становитиме:  $1 \cdot 0,477 = 0,477$ .

Цей вид аналізу може бути корисним для таких речей, як пошук або категоризація документів [8].

1.2.8. N-Gram. N-Gram відноситься до набору слів, що зустрічаються разом. Сутність, що стоїть за N-Gram підходом, полягає в тому, що слова, що зустрічаються разом, дають більше інформації, ніж ті, що зустрічаються

окремо. Розглянемо для прикладу наступне речення: «Я, студент ТНТУ імені Івана Пулюя».

Тут, якщо ми створимо набір функцій для цього речення з окремими словами, це буде виглядати приблизно так: «Я», «студент», «ТНТУ», «імені», «Івана», «Пулюя».

Але якщо ми подивимось на це речення, то можемо побачити, що словосполучення, «Я студент» і «студент ТНТУ» разом надають більше інформації про те, що йдеться у реченні, ніж якщо перевірять слова «Я», «студент» і «ТНТУ» окремо.

У N-Grams N позначає кількість слів, що зустрічаються. Наприклад, розглянемо речення: «Степан студент ТНТУ імені Івана Пулюя». Якщо, ми спробуємо побудувати 2-Grams з цього речення, вони виглядатимуть наступним чином:

2-Grams (S1) = [«Степан студент», «студент ТНТУ», «ТНТУ імені», «імені Івана», «Івана Пулюя»].

Тепер, якщо ми подивимось на ці N-Grams, ми можемо побачити, що щонайменше три N-Grams передають значну частину інформації про речення: «Степан студент», «студент ТНТУ», «Івана Пулюя». З цих N-Grams ми можемо зрозуміти, що зміст стосується особи Степана, який є студентом ТНТУ, що названий на честь Івана Пулюя.

Ми можемо мати будь-яку кількість N-Grams. Кількість N-Grams для речення S, що має X кількість слів, становить:

$$N\text{-gram}(S) = X - (N - 1). \quad (1.3)$$

Набір N-Grams може бути корисним для таких моделей, як автодоповнення/автокорекція та мовних моделей. Створення N-Grams з величезного корпусу тексту надає багато інформації про те, які слова зазвичай зустрічаються разом, і тому дозволяє передбачити, яке слово буде подане у реченні [5].

### 1.3. Дослідження побудови нейронних мереж для задач опрацювання природної мови

1.3.1. Загальні відомості про нейронні мережі. Нейронні мережі – це обчислювальні системи із взаємопов’язаними вузлами, які так само працюють як нейрони в мозку людини. Використовуючи алгоритми, вони можуть розпізнавати приховані шаблони та кореляції в необроблених даних, кластеризувати та класифікувати їх та постійно вчитися та вдосконалюватись.

Першу нейронну мережу створили нейрофізіолог Воррен МакКаллох та математик Волтер Пітс у 1943 р. Вони пояснили як нейрони можуть працювати, і змодельовали свої ідеї, створивши просту нейронну мережу за допомогою електричних схем.

Перша справжня багат шарова нейронна мережа була створена у 1975 році Куніхіко Фукусімою [9].

Нейронні мережі ідеально підходять, щоб допомогти людям вирішувати складні проблеми в реальних ситуаціях. Вони можуть вивчити та моделювати зв’язки між входами та виходами, які є нелінійними та складними; робити узагальнення та висновки; розкрити приховані відносини, закономірності та прогнози; а також моделювати дуже мінливі дані та відхилення, необхідні для прогнозування рідкісних подій. Складовим елементом нейронних мереж є нейрони (рис.1.2).

Функції нейрона:

- приймальна функція: синапси отримують інформацію;
- інтегративна функція: на виході нейрона формується сигнал, що несе інформацію про всі підсумовані в нейроні сигнали;
- провідна функція: по аксону проходить інформація до синапса;
- передавальна функція: імпульс, що дійшов до закінчення аксона, змушує медіатор передавати збудження наступному нейрону.



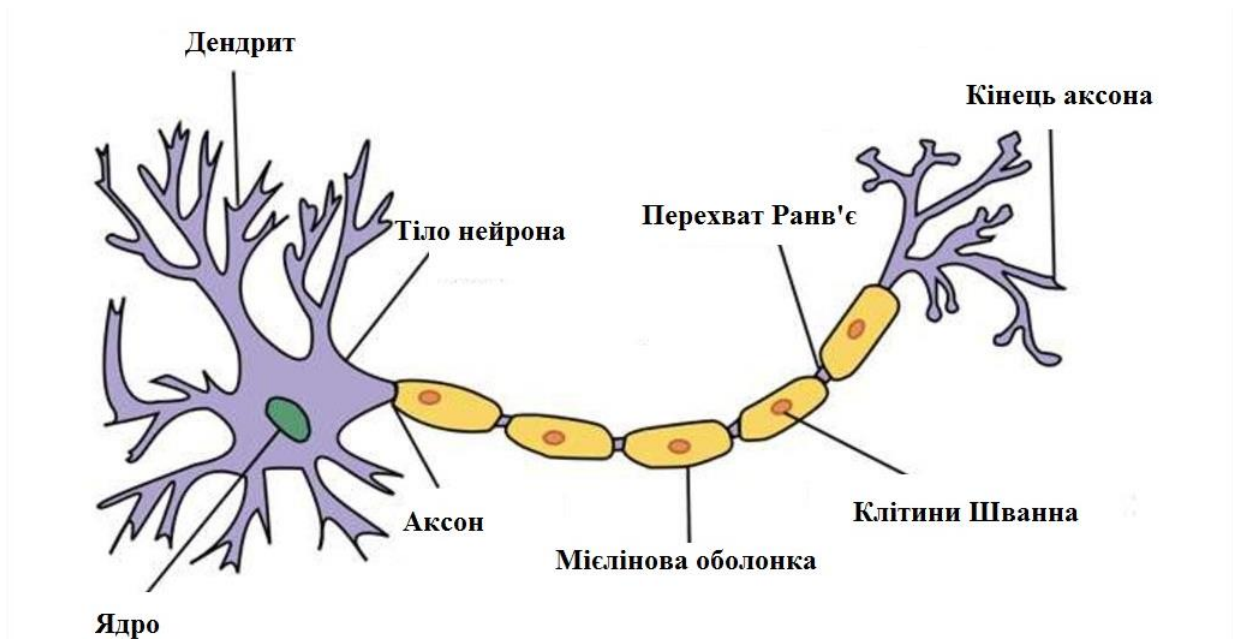


Рис.1.2. Будова нейрона

Синапс – місце функціонального контакту двох збудливих клітин одна з яких – нервова. За функціями синапси бувають збуджуючі та гальмівні. Вихід нейрона називається аксоном.

У штучної нейронної мережі штучний нейрон – це деяка нелінійна функція, аргументом якої є лінійна комбінація всіх вхідних сигналів. Така функція називається активаційною. Потім результат активаційної функції посиляється на вихід нейрона. Об'єднуючи такі нейрони з іншими, отримують штучну нейронну мережу.

Існують різні види нейронних мереж, й кожна має свої переваги та недоліки, залежно від використання.

Згорткові нейронні мережі (CNN) містять п'ять типів шарів: вхідний, згортковий, об'єднаний, повністю пов'язаний і вихідний. Кожен шар має певну мету, наприклад підбиття підсумків, з'єднання або активацію. Згорткові нейронні мережі популяризували класифікацію зображень та виявлення об'єктів. Однак CNN також застосовуються для інших областей, таких як обробка природних мов та прогнозування.

Рекурентні нейронні мережі (RNN) використовують послідовну інформацію, таку як неперервне опрацювання даних з сенсорного пристрою

або розмовне речення, що складається з послідовності доданків. На відміну від традиційних нейронних мереж, всі входи в рекурентну нейронну мережу не залежать один від одного, і вихід для кожного елемента залежить від обчислень його попередніх елементів. RNN застосовуються до таких задач, як розпізнавання несегментованого неперервного тексту та розпізнавання мовлення [10].

Людський мозок містить близько 100 мільярдів мізерних клітин, які називаються нейронами (ніхто не знає точно, скільки їх існує, і підрахунки становлять приблизно від 50 мільярдів до цілих 500 мільярдів). Кожен нейрон складається з клітинного тіла (центральної маси клітини) з низкою з'єднань, що відходять від нього: численні дендрити (вхід клітини – інформацію передає до тіла клітини) та окремий аксон (вихід клітини – передає інформація іншому нейрону) (рис.1.2).

Нейрони настільки крихітні, що ви могли зібрати близько 100 їх клітинних тіл в один міліметр. Варто також замітити, що нейрони складають лише 10 відсотків усіх клітин у мозку, решта – це гліальні клітини, які також називаються нейрогліями, які підтримують і захищають нейрони та живлять їх енергією, яка дозволяє їм працювати і рости. Всередині комп'ютера еквівалент клітині мозку є крихітний комутаційний пристрій, який називається транзистором. Останні передові мікропроцесори містять понад 2 мільярди транзисторів, навіть базовий мікропроцесор має близько 50 мільйонів транзисторів, усі упаковані в інтегральну схему розміром всього 25 мм<sup>2</sup> [10].

У типовій нейронній мережі є від кількох десятків до сотень, тисяч, а то й мільйонів штучних нейронів, які називаються одиницями, розташованими в ряд шарів, кожен з яких з'єднується з шарами по обидва боки. Деякі з них, відомі як блоки введення, призначені для отримання різних форм інформації із зовнішнього світу, про які мережа намагатиметься дізнатися, розпізнати чи іншим чином обробити. Інші містяться на протилежній стороні мережі та сигналізують, як вона реагує на інформацію, яку вона засвоїла, вони відомі як

вихідні одиниці. Між вхідними і вихідними одиницями розташовані один або кілька шарів прихованих одиниць, які разом утворюють більшість штучного мозку. Більшість нейронних мереж – повнозв’язні, це означає, що кожен прихований блок і кожен вихідний блок підключені до кожного блоку в шарах з будь-якої сторони. Зв’язки між однією одиницею та іншою представлені числом, так званою вагою, яке може бути як позитивним (якщо одна одиниця збуджує іншу), так і негативним (якщо одна одиниця пригнічує або гальмує іншу). Чим більша вага, тим більший вплив одна одиниця має на іншу. Це відповідає тому, як реальні клітини мозку спрацьовують одна на одну через крихітні прогалини, які називаються синапсами [11].

Повнозв’язна нейронна мережа складається з вхідних, прихованих та вихідних шарів, причому всі шари з’єднані з усіма іншими шарами з обох боків. Шари входу подаються ліворуч, активують приховані шари посередині, а шари виходу подають праворуч. Міцність (вага) зв’язку між будь-якими двома шарами поступово регулюється по мірі навчання мережі (рис.1.3) [10].

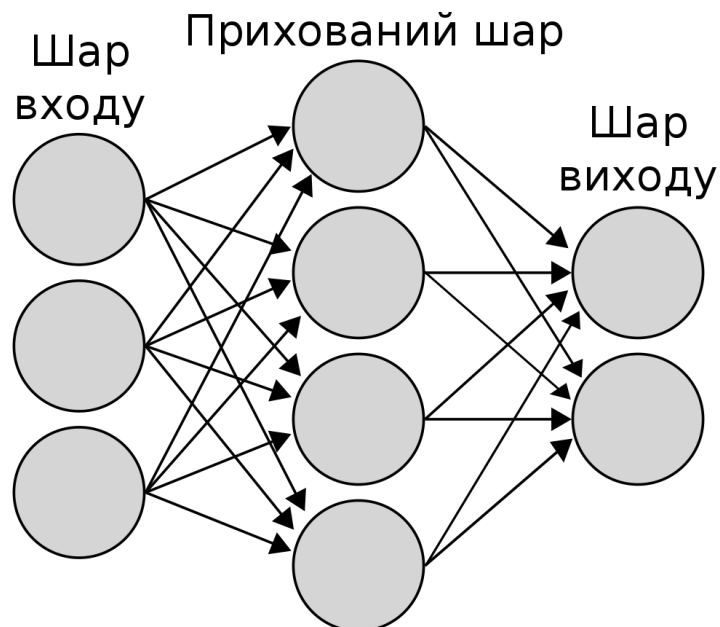


Рис.1.3. Будова повнозв’язної нейронної мережі

1.3.2. Активаційна функція нейронів. Функція активації нейрона характеризує залежність сигналу на виході нейрона від суми сигналів на його входах. Зазвичай функція є монотонно зростаючою і знаходиться в області значень  $[-1,1]$  (гіперболічний тангенс) і  $[0,1]$  (сигмоїда). Для деяких алгоритмів навчання необхідно, щоб активаційна функція була безперервно диференційованою на всій числової осі. Штучний нейрон характеризується своєю активаційною функцією [12].

Основними активаційними функціями є:

– порогова активаційна функція (функція Хевісайда). Не можна використовувати для алгоритму зворотного поширення помилки:

$$f(x) = \begin{cases} 1 & \text{if } x \geq -w_0x_0 \\ 0 & \text{else} \end{cases} ; \quad (1.4)$$

– сигмоїдальна активаційна функція:

$$\sigma(x) = \frac{1}{1 + e^{-x}} ; \quad (1.5)$$

– гіперболічний тангенс:

$$\tanh(Ax) = \frac{e^{Ax} - e^{-Ax}}{e^{Ax} + e^{-Ax}} . \quad (1.6)$$

1.3.3. Рекурентна нейронна мережа. У нейронних мережах прямого поширення використовується єдиний вхід, який повністю визначає активації всіх нейронів в шарах, що залишилися. Таку мережу неможливо навчити передбачати події. Рекурентні нейронні мережі (РНМ) можуть вирішити цю проблему. Маючи всередині цикли, рекурентна нейронна мережа дозволяє інформації зберігатися: поведінка прихованих нейронів буде визначатися не

тільки активацією в інших прихованих шарах, але і отриманими раніше активаціями самих нейронів.

РНМ може бути представлена в якості безлічі копій однієї і тієї ж нейронної мережі, де кожна копія передає повідомлення наступній копії. Тобто, маючи ланцюгоподібну структуру, як послідовність або списки, РНМ є природною архітектурою нейронної мережі, що використовується для таких даних [13].

РНМ здатні обумовлювати модель за всіма попередніми обробленими словами з корпусу текстів. На рис.1.4 прямокутник є прихованим шаром на тимчасовому кроці  $t$ .

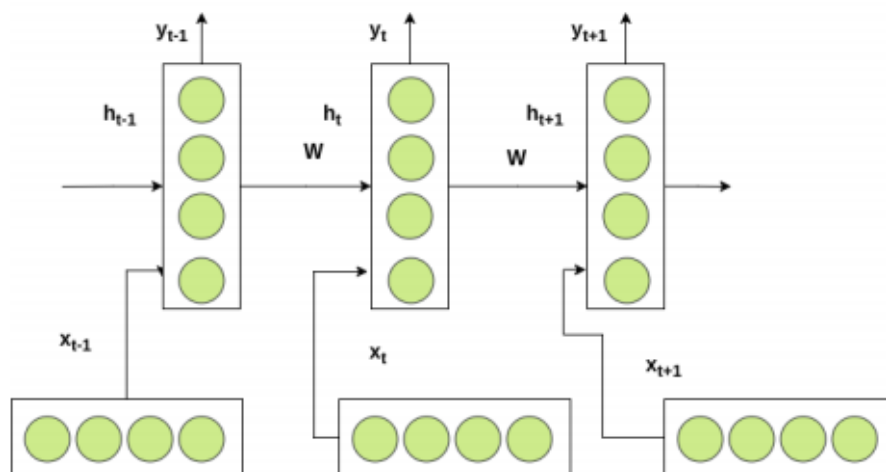


Рис.1.4. Рекурентна нейронна мережа, три тимчасових кроки

Кожен шар містить нейрони, кожен з яких виконує операцію лінійної матриці на своїх входах, за якою слідує нелінійна операція.

На кожному часовому кроці вихідні дані попереднього кроку разом з наступним вектором слова  $x_t$  тексту, представляє вхідні дані прихованого шару для створення передбачення  $\hat{y}_t$  і ознак  $h_t$ :

$$h_t = \sigma(W^{(hh)}h_{(t-1)} + W^{(hx)}x_t), \quad (1.7)$$

$$\hat{y}_t = \text{softmax}(W^{(S)}h_t). \quad (1.8)$$

- 1)  $x_1, \dots, x_t + 1, \dots, x_t$  – вектори слів, словник складається з  $T$  слів;
- 2) формула  $h_t$  дозволяє обчислити вихідні ознаки прихованого шару на

кожному тимчасовому кроці  $t$ :

- $x_t \in R_d$ : слово, подане на вхід в момент часу  $t$ ;
- $W_{hx} \in R^{Dh \times d}$ : матриця ваг для обумовлення вхідного слова  $x_t$ ;
- $W^{hh} \in R^{Dh \times Dh}$ : матриця ваг, використана для обумовлення

вихідних даних на попередньому часовому кроці  $h_{t-1}$ ;

- $h_{t-1} \in R^{Dh}$ : вихід нелінійної функції на попередньому часовому кроці.  $h_0$  – вектор ініціалізації для прихованого шару в момент часу  $t = 0$ ;
- $\sigma$  – нелінійна функція;

3)  $\hat{y}_t$ : вихідний розподіл ймовірності за словником на кожному часовому кроці  $t$ . По суті  $\hat{y}_t$  є наступним вказаним словом на основі оцінки контексту документа  $h_{t-1}$  і останнього спостережуваного вектора слова  $x_t$ .

В якості функції втрат використовується функція помилки перехресної ентропії (cross entropy error):

- на тимчасовому кроці  $t$ :

$$J^t(\theta) = - \sum_{j=1}^{|V|} y_{t,j} \times \log(\hat{y}_{t,j}); \quad (1.9)$$

- над корпусом текстів розміру  $T$ :

$$J = -\frac{1}{T} \sum_{t=1}^T \sum_{j=1}^{|V|} y_{t,j} \times \log(\hat{y}_{t,j}). \quad (1.10)$$

Обсяг пам'яті, необхідний для запуску шару РНМ, пропорційний кількості слів в корпусі текстів. Тобто речення, що складається з  $k$  слів, буде зберігатися в пам'яті як  $k$  векторів. Розмір матриці ваг  $W$  не масштабується відповідно до розміру корпусу текстів. Для рекурентної мережі, що складається з 1000 рекурентних шарів, розмір матриці завжди буде  $1000 \times 1000$ , в незалежності від розміру корпусу текстів [14].

#### 1.3.4. Висновки до розділу

У даному розділі здійснено огляд предметної області опрацювання природної мови для задач розпізнавання власних назв. Досліджено особливості опрацювання природної мови. Проаналізовано основні компоненти природної мови, а також проаналізовано методи нормалізації корпусу тексту в NLP.

Розглянуто методи, що використовуються для процесу навчання NLP. Досліджено принципи побудови нейронних мереж. Наведено загальні відомості про нейронні мережі й розглянуто основні функції нейронів.

На основі рекурентної нейронної мережі обґрунтовано математичне забезпечення побудови програмної системи для задач розпізнавання власних назв.

## РОЗДІЛ 2

### АНАЛІЗ І ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

#### 2.1. Аналіз методів опрацювання природної мови

2.1.1. Метод Word2Vec. Word2Vec – це ефективний і дієвий спосіб представлення слів у вигляді векторів. Весь корпус тексту переміщений в певний простір у якому всі вектори мають певні координати, і можна чітко визначити їх взаємозв'язок один з одним. Розподілене подання слів у вигляді векторів відкриває безліч можливостей для пошуку слів різними способами.

Word2Vec є одним з найпопулярніших методів для розпізнавання власних назв з допомогою нейронної мережі. Він був розроблений Томашем Миколовим в 2013 році в компанії Google.

CBOW (Common Bag Of Words) і Skip-Gram – два основних алгоритми навчання слів (рис.2.1).

У CBOW слова, що зустрічаються в контексті (оточуючі слова) обраного слова, використовуються в якості вхідних даних, а вибране слово як ціль. Протилежний принцип має Skip-Gram, де вибране слово намагається передбачити слова, що йдуть до і після нього.

Розглянемо текст, що складається з символів від «a» до «z» послідовно [a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z], нехай цілі числа від 0 до 25 представляють відповідні літери. Реалізуючи алгоритм CBOW, зберігши розмір вікна 2, ми бачимо, що «a» пов'язано з «b» і «c», «b» з «a», «c» і «d» і так далі. З цього випливає, що один вектор, який представляє вхідні слова, матиме розмірність [26, 1]. За допомогою моделі ми знайдемо щільний і розподілений вектор розміру [10, 1]. Розмір вектора вибирається довільно.



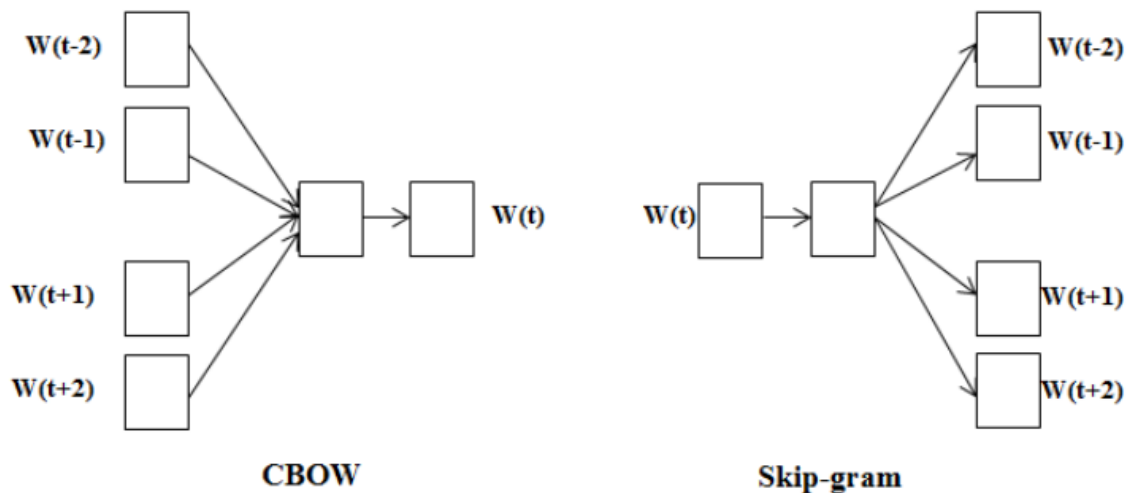


Рис.2.1. Ілюстрація алгоритмів CBOW та Skip-gram

Ключем до реалізації Word2Vec є побудова 2 додаткових вагових матриць для подання слів у якості вхідних даних і в якості контексту або цілей. У рядки першої матриці ( $w_1$ ) та стовпці другої матриці ( $w_2$ ) вбудовуються вхідні слова та цільові слова відповідно. Добуток цих двох векторів слів потім використовується для отримання ймовірностей, щоб бути цільовим словом, враховуючи вибране вхідне слово. Під час навчання ці вектори оптимізуються з використанням градієнтного спуску таким чином, що ймовірності для справжніх цілей максимальні [15]. Схематичне зображення Word2Vec моделі відображено на рис.2.2.

Принцип роботи Word2Vec розглянемо на прикладі подібних речень: Бажаю Вам гарного дня та Бажаю Вам хорошого дня. Вони навряд чи мають різні значення. Якщо побудувати словник (назвемо його  $V$ ), він буде мати:  $V = \{ \text{Бажаю, Вам, гарного, хорошого, дня} \}$ .

Для кожного з цих слів в  $V$  створюється кодований вектор. Довжина вектора дорівнює розміру  $V$  ( $= 5$ ). Принцип кодування наведено нижче.

Бажаю =  $[1,0,0,0,0]$ ; Вам =  $[0,1,0,0,0]$ ; гарного =  $[0,0,1,0,0]$ ;  
хорошого =  $[0,0,0,1,0]$ ; дня =  $[0,0,0,0,1]$ .

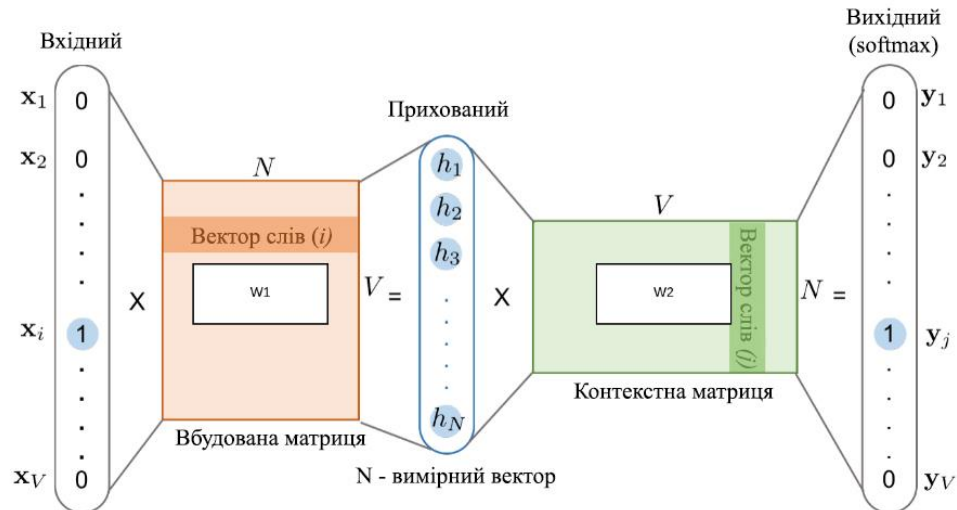


Рис.2.2. Схематичне зображення Word2Vec моделі

Якщо спробувати візуалізувати ці кодування, можна уявити собі 5-вимірний простір, де кожне слово займає певну величину і не має нічого спільного з іншими (немає проекції вздовж інших величин). Це означає, що слова «хорошого» і «гарного» так само різні, як «Бажаю» і «дня», що невірно.

Принцип роботи Word2Vec полягає в тому, щоб слова з подібним контекстом займають близькі просторові позиції. Математично косинус кута між такими векторами повинен бути близький до 1, тобто кут близький до 0. Вводиться деяка залежність одного слова від інших слів. Слова в контексті цього слова отримують велику частку цієї залежності [16].

2.1.2. Метод Glove. GloVe – це непідвладний алгоритм навчання для отримання векторних уявлень для слів. Цей спосіб був винайдений в Стенфорді. Принцип роботи GloVe дуже схожий з Word2Vec, але на відміну від Word2Vec, у якого в основі лежить модель «передбачення», GloVe побудований на основі моделі «обчислення».

GloVe працює наступним чином. Спочатку він обходить весь корпус і збирає статистику появи слів, після чого складає матрицю спільної появи слів. Матриця спільної появи слів – це матриця, кожен рядок і стовпець якої

відповідають якомусь слову з корпусу, а на перетинах рядків і стовпців стоять числа, що відповідають тому, скільки разів слово в рядку стояло поруч зі словом в стовпці. Відстань, на яку повинні стояти один від одного слова визначається параметром алгоритму.

Показники подібності, що використовуються для оцінювання найближчого сусіда, дають єдиний скаляр, який кількісно визначає спорідненість двох слів. Наприклад, чоловіка можна вважати схожим на жінку тим, що обидва слова описують людину; з іншого боку, два слова часто вважаються протилежностями, оскільки вони виділяють первинну вісь, уздовж якої люди відрізняються один від одного. Основна концепція, яка відрізняє чоловіка від жінки, тобто стать, може бути рівнозначно визначена різними іншими парами слів, такими як король і королева або брат і сестра. Якщо висловити це спостереження математично, можна очікувати, що векторні відмінності чоловік – жінка, король – королева та брат – сестра можуть бути приблизно однаковими.

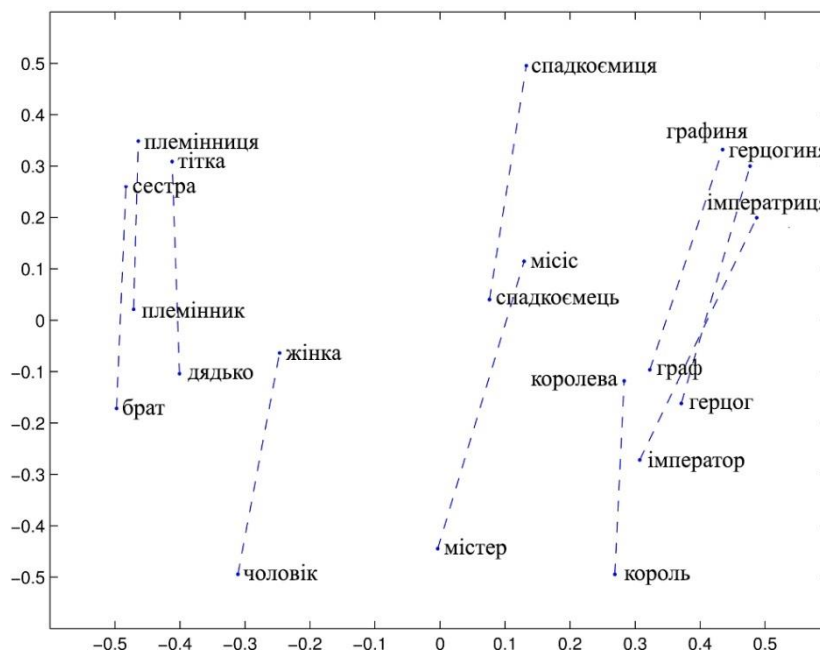


Рис.2.3. Векторне відображення слів

Для того щоб кількісно зафіксувати унікальність, що потрібна для розрізнення чоловіка від жінки, необхідно, щоб модель асоціювала до пари слів більше ніж одне число. Природним і простим кандидатом на збільшений набір дискримінаційних чисел є різниця між векторами двох слів. GloVe розроблений для того, щоб такі векторні відмінності максимально захоплювали значення, визначене поєднанням двох слів [17].

2.1.3. Переваги та недоліків методів Word2Vec та Glove. Методи Word2Vec та Glove вивчають геометричне кодування, а саме вектори слів із інформацією, як задані слова з'являються у великих текстових корпусах. Дані методи відрізняються тим, що в основі Word2Vec закладена «прогностична» модель, тоді, як у Glove – «підрахункова модель» [18].

Перевагою Word2Vec є те, що модель, яка лежить в основі даного методу є досить універсальна, вона перетворює необроблений сирий корпус тексту у мічені дані (шляхом зіставлення цільового слова до його контекстного слова) та вивчає подання слів у завданні класифікації. Також дані можуть подаватися в модель у режимі онлайн і вимагають невеликої попередньої обробки, тому потрібно мало пам'яті [19].

Ще один цікавий момент Word2Vec полягає в тому, що подібні слова розташовані разом у векторному просторі, а арифметичні операції над векторами слів можуть створювати семантичні чи синтаксичні зв'язки, для прикладу: «король» – «чоловік» + «жінка» = «королева», або «ліпше» – «добре» + «погано» = «гірше» [20].

Недоліком методу Word2Vec є те, що модель може бути дуже важко навчити, якщо використовувати функцію softmax, так, як кількість категорій буде занадто багато (розмір словника). Модель також не вирішує проблему багатозначних слів. Word2Vec вивчає вектори лише повних слів, знайдених у корпусі.

Перевага Glove в тому, що модель надає певного практичного значення векторам слів, розглядаючи зв'язки між парою слів, а не словом і словом.

Також Glove надає меншу вагу для дуже частих пар слів, щоб «стоп-слова» не домінували під час процесу навчання. Даний метод створює глобальну матрицю спільного виникнення, оцінюючи можливість того, що дане слово буде співпадати з іншими словами.

Недоліком Glove є те, що модель навчається на матриці спільного виникнення слів, яка займає багато пам'яті для зберігання. Тим більше, якщо ви змінюєте гіпер-параметри, пов'язані з матрицею спільного виникнення, вам доведеться знову реконструювати матрицю, що забирає дуже багато часу.

Проте, є завдання які методи Word2Vec та Glove не в змозі вирішити. Для прикладу, як розділити пари протилежних за значенням слів. Наприклад, слова «добре» і «погано» зазвичай розташовані дуже близько один до одного у векторному просторі [19].

## 2.2. Дослідження математичного забезпечення методів роботи нейронних мереж для задач опрацювання природної мови

### 2.2.1. Дослідження математичного забезпечення методу Word2Vec.

Застосування методів машинного навчання до текстів на природній мові може дати багато цікавих і корисних результатів, наприклад – автоматичне сортування текстів за темами (завдання класифікації), пошук схожих текстів (задача кластеризації), розпізнавання власних назв, автоматичний переклад, тощо. Для того, щоб застосувати математичні методи до текстів необхідно певним чином формалізувати дані. У разі класифікатора текстів формалізація виконується з допомогою частотного аналізу ( $g$ ), у цьому випадку кожному тексту  $T$  ставимо у відповідність точку в просторі ознак  $X \subset R^n$ .

$$g: T \rightarrow X \quad (2.1)$$

Цей метод досить добре працює для текстів середнього розміру, однак для коротких повідомлень частотна характеристика може виявитися

неінформативною. Не підходить цей метод і для завдань машинного перекладу, де цікавою є не загальна характеристика тексту, а кожне слово окремо і послідовності зі слів. Таким чином, виникає необхідність побудувати ефективний метод кодування окремих слів [21, 22].

Слова можна кодувати різними методами. Напевно найпростіший спосіб це їх пронумерувати, тобто складаємо повний словник з тексту, збираємо всі можливі форми слів, використані в тексті, і нумеруємо всі ці слова. Але такий спосіб кодування не несе ніякого смислового навантаження, тобто за кодом не можна сказати наскільки близькими за змістом є слова, наприклад, номер 7 та номер 457.

Для побудови «осмисленого» простору для слів був розроблений метод Word2Vec, який відображає слова  $W$  у векторному просторі  $V \subset R^n$ .

$$\text{word2vec}: W \rightarrow V \quad (2.2)$$

При цьому, спільно вживанні в тексті  $T$  слова з  $W$  відображаються близько (в сенсі евклідової метрики) у точках простору  $V$ . До того ж, над точками  $V$  можна виконувати операції, що мають сенс у  $W$ . Наведемо приклад нижче:

$$\text{word2vec}[\text{king}] - \text{word2vec}[\text{man}] + \text{word2vec}[\text{woman}] \approx \text{word2vec}[\text{queen}] \quad (2.3)$$

Результатом роботи Word2Vec є набір векторів (матриця) – кодів слів, яку отримуємо в результаті навчання певної нейронної мережі на деякому тексті (впорядкованій множині слів), й перше, що потрібно зробити, це підготувати набір навчальних даних [23].

Набір навчальних даних для нейронної мережі Word2Vec створюємо наступним чином:

- фільтруємо вхідний текст  $T$  від зайвих символів (розділові знаки, тощо);

- з очищеного тексту  $T$  формуємо словник  $W$ ;
- для кожного слова  $w_i \in T$  створюємо контекст, тобто набір слів  $C_i \subset T$ , віддалених від  $w_i$  не більше ніж на  $s$  позицій у послідовності слів  $T$ :

$$C_i = \{w_j \in T: (i - s) \leq j \leq (i + s), j \neq i\}; \quad (2.4)$$

- виконуємо унітарне кодування словника  $W$ , тобто кожному слову  $w_i \in W$  ставиться у відповідність вектор  $u_i \in U$  з нулів і однієї одиниці, довжина вектора  $u_i$  дорівнює розміру словника  $W$ , позиція одиниці у векторі  $u_i$  відповідає номеру слова у словнику  $W$ ;
- замінюємо слова в тексті  $T$  і контекстах  $C$  відповідними кодами  $P$  і  $Q$  з  $U$ .

Таким чином, отримуємо дві множини кодованій текст  $P$  і набори кодованих слів контексту  $Q$  (рис.2.4).

```

      Pi:
    0 0 1 0 0

      Qi:
    0 1 0 0 0
    0 0 0 0 1
    0 0 0 1 0
    1 0 0 0 0
    0 0 0 1 0
  
```

Рис.2.4. Приклад отриманих множин (слово і контекст)

Нейронна мережа Word2Vec має наступний вигляд (рис.2.5).

Вона складається з трьох шарів: розмір першого шару  $X$  дорівнює розміру словника  $W$  (довжини векторів  $U$ ), прихований шар  $H$  має лінійну функцію активації, вихідний шар  $Y$  має функцію активації *softmax*, його розмір дорівнює розміру вхідного шару  $X$ :

$$H = X \cdot V_i \quad (2.5)$$

$$Y = \text{softmax}(H \cdot V_o) \quad (2.6)$$

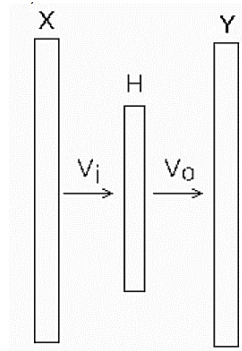


Рис.2.5. Схема нейронної мережі Word2Vec

Кінцевий результат – це матриця внутрішніх представлень  $V_i$ , й щоб її отримати необхідно навчити мережу.

Для здійснення навчання мережі Word2Vec застосовують метод простого градієнтного спуску й один із двох алгоритмів – CBOW або Skip-Gram.

- Skip-Gram – алгоритм, що надає змогу відновити контекст за допомогою слова.
- CBOW (Continuous Bag of Words) – алгоритм, який по контексту відновлює слово.

Основна ідея методу Skip-Gram навчання нейромережі Word2Vec полягає у відновленні слів контексту за поточним словом (рис.2.6) [24].

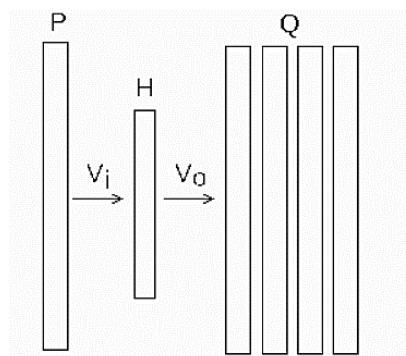


Рис.2.6. Схема нейронної мережі Word2Vec (Skip-Gram)



Обчислення зміни ваг за методом градієнтного спуску в даному випадку здійснюється наступним чином:

– на вхід мережі подається код слова  $P$ , обчислюємо стан прихованого шару  $H$ :

$$H = P \cdot V_i; \quad (2.7)$$

– далі обчислюємо вихід мережі  $O$ :

$$U = H \cdot V_o, \quad (2.8)$$

$$O = \text{softmax}(U); \quad (2.9)$$

– для кожного слова контексту  $Q_j$  та входу  $P$ , обчислюємо помилку  $D$  на виході мережі  $O$  і зміну ваг мережі  $\Delta V_o$ ,  $\Delta V_i$ :

$$D = O - Q_j, \quad (2.10)$$

$$\Delta V_{oj} = H^T \cdot D, \quad (2.11)$$

$$\Delta V_{ij} = D^T \cdot P \cdot V_o^T; \quad (2.12)$$

– обчислюємо сумарну зміну ваг мережі  $\Delta V_o$ ,  $\Delta V_i$ :

$$\Delta V_o = \sum_j \Delta V_{oj}, \quad (2.13)$$

$$\Delta V_i = \sum_j \Delta V_{ij}. \quad (2.14)$$

Функція втрати на навчальному прикладі  $i$  виглядає наступним чином:

$$E_i = \left| \log \sum \exp(U_i) - \sum_j \sum (U_i * Q_{ij}) \right|. \quad (2.15)$$

Альтернативою Skip-Gram слугувати алгоритм CBOW, тут ми по контексту намагаємося відновити слово (рис. 2.7) [25].

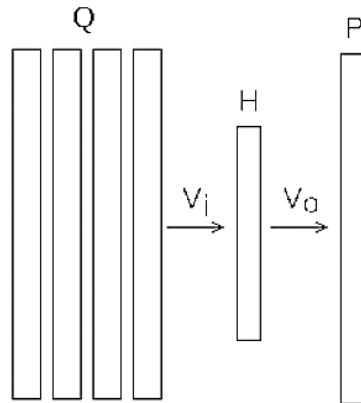


Рис.2.7. Схема нейронної мережі Word2Vec (CBOW)

Обчислення зміни ваг за методом градієнтного спуску в даному випадку здійснюється наступним чином:

– на вхід мережі подається середнє значення контексту  $Q$ , обчислюємо стан прихованого шару  $H$ :

$$H = \frac{1}{c} \sum_{j=1}^c Q_j \cdot V_i; \quad (2.16)$$

– далі обчислюємо вихід мережі  $O$ :

$$U = H \cdot V_o, \quad (2.17)$$

$$O = \text{softmax}(U); \quad (2.18)$$

– для кожного слова контексту  $Q_i$  та коду слова  $P$ , обчислюємо помилку  $D$  на виході мережі  $O$  і зміну ваг мережі  $\Delta V_o$ ,  $\Delta V_i$ .

$$D = O - P, \quad (2.19)$$

$$\Delta V_o = H^T \cdot D, \quad (2.20)$$

$$\Delta V_i = \sum_j D^T \cdot Q_j \cdot V_o^T. \quad (2.21)$$

Функція втрати на навчальному прикладі  $i$  виглядає наступним чином:

$$E_i = \left| \log \sum \exp(U_i) - \sum (U_i * P_i) \right|. \quad (2.22)$$

Реалізація Word2Vec складається з трьох частин:

- кодування – на вхід надходить відфільтрований текст, на виході отримуємо кодовані набори слів тексту  $P$  і контекстів  $Q$ ;
- навчання мережі – на вхід надходить навчальний набір  $P, Q$ , на виході маємо матрицю уявлень  $V_i$ ;
- тест результату – на вхід надходить словник і матриця уявлень  $V_i$ , на виході отримуємо випадково вибрані слова зі словника і найбільш близькі до них слова Word2Vec [26].

2.2.2. Дослідження математичного забезпечення методу Glove. Модель GloVe навчається на ненульових записах глобальної матриці спільного виникнення слів, в якій відображається таблиця того, як часто слова зустрічаються між собою в даному корпусі. Для заповнення цієї матриці необхідний один прохід по всьому корпусу для збору статистичних даних. Для великих матриць цей прохід може бути обчислювально дорогим, але це разова авансова вартість. Подальші навчальні ітерації набагато швидші, оскільки кількість ненульових матричних записів, як правило, набагато менша, ніж загальна кількість слів у корпусі.

Розглянемо простий приклад, який демонструє, як певні аспекти значення можуть бути витягнуті безпосередньо з сукупності можливих випадків. Розглянемо два слова  $i$  та  $j$ , які виявляють певний аспект інтересу.

Наприклад, нас цікавить концепція термодинамічної фази, для якої ми можемо взяти  $i = \text{лід}$  та  $j = \text{пара}$  [17].

Зв'язок цих слів можна дослідити, вивчивши співвідношення ймовірностей їх спільної появи з різними словами  $k$ . Для слів  $k$ , пов'язаних з льодом, але не парою, скажімо,  $k = \text{«твердий»}$ , ми очікуємо, що співвідношення  $P_{ik}/P_{jk}$  буде великим. Аналогічно, для слів  $k$ , пов'язаних з парою, але не з льодом, скажімо,  $k = \text{«газ»}$ , співвідношення має бути невеликим. Для слів  $k$ , як вода або мода, які пов'язані або з льодом, і з парою, або з жодним, співвідношення має бути близьким до одиниці. На рис.2.8 наведені ці ймовірності та їх співвідношення для великого корпусу, і числа, що підтверджують ці очікування.

Співвідношення ймовірностей	$k = \text{твердий}$	$k = \text{газ}$	$k = \text{вода}$	$k = \text{мода}$
$P(k   \text{лід})$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k   \text{пара})$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k   \text{лід}) / P(k   \text{пара})$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

Рис.2.8. Співвідношення ймовірності спільної появи слів

Як можна було б очікувати, лід зустрічається частіше з твердим, ніж з газом, тоді як пар зустрічається частіше з газом, ніж з твердим. Обидва слова часто зустрічаються із спільною появою води, і обидва спільно рідко трапляються з модою. Таким чином, співвідношення ймовірностей кодує деяку грубу форму значення, пов'язану з абстрактним поняттям термодинамічної фази.

Вищенаведений аргумент говорить про те, що відповідне початкове місце для вивчення векторів слів повинно бути із співвідношеннями ймовірностей спільної появи, а не самих ймовірностей. Зазначаючи, що співвідношення  $P_{ik}/P_{jk}$  залежить від трьох слів  $i$ ,  $j$  і  $k$ , найзагальніша модель приймає форму:

$$F(w_i, w_j, w_k) = \frac{P_{ik}}{P_{jk}}, \quad (2.23)$$

де  $w \in \mathbb{R}^d$  – вектори слова,

$w \in \mathbb{R}^d$  – вектори контекстних слів.

Далі відбувається факторизація матриці. Схожа операція відбувається і всередині алгоритму Word2Vec. В результаті GloVe потрібно мінімізувати наступний функціонал:

$$J = \sum_{i=1}^V \sum_{j=1}^V f(X_{ij})(\omega_i^T \omega_j + b_i + b_j - \log(X_{ij}))^2, \quad (2.24)$$

де  $V$  – величина словника,

$w_i$  – вектор головного слова,

$w_j$  – вектор контекстного слова,

$b_i, b_j$  – скалярні зміщення,

$f(X_{ij})$  – зважуюча функція, яка запобігає перенаванчання на часто повторюваних парах:

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{X_{MAX}}\right)^a, & \text{при } X_{ij} < X_{MAX} \\ 1, & \text{в інших випадках} \end{cases}. \quad (2.25)$$

2.3. Аналіз апаратних засобів опрацювання природної мови з використанням нейронних мереж

2.3.1. Тензорний процесор. Тензорний процесор (TPU) є інтегральною схемою специфічного застосування (ASIC) (рис. 2.9), розроблений компанією

Google й призначений прискорювати розрахунки для задач штучного інтелекту [27].

Свою назву тензорний процесор отримав від бібліотеки TensorFlow. Оскільки, призначення TPU полягає в пришвидшенні алгоритмів штучного інтелекту, з використанням бібліотек вільного програмного забезпечення.

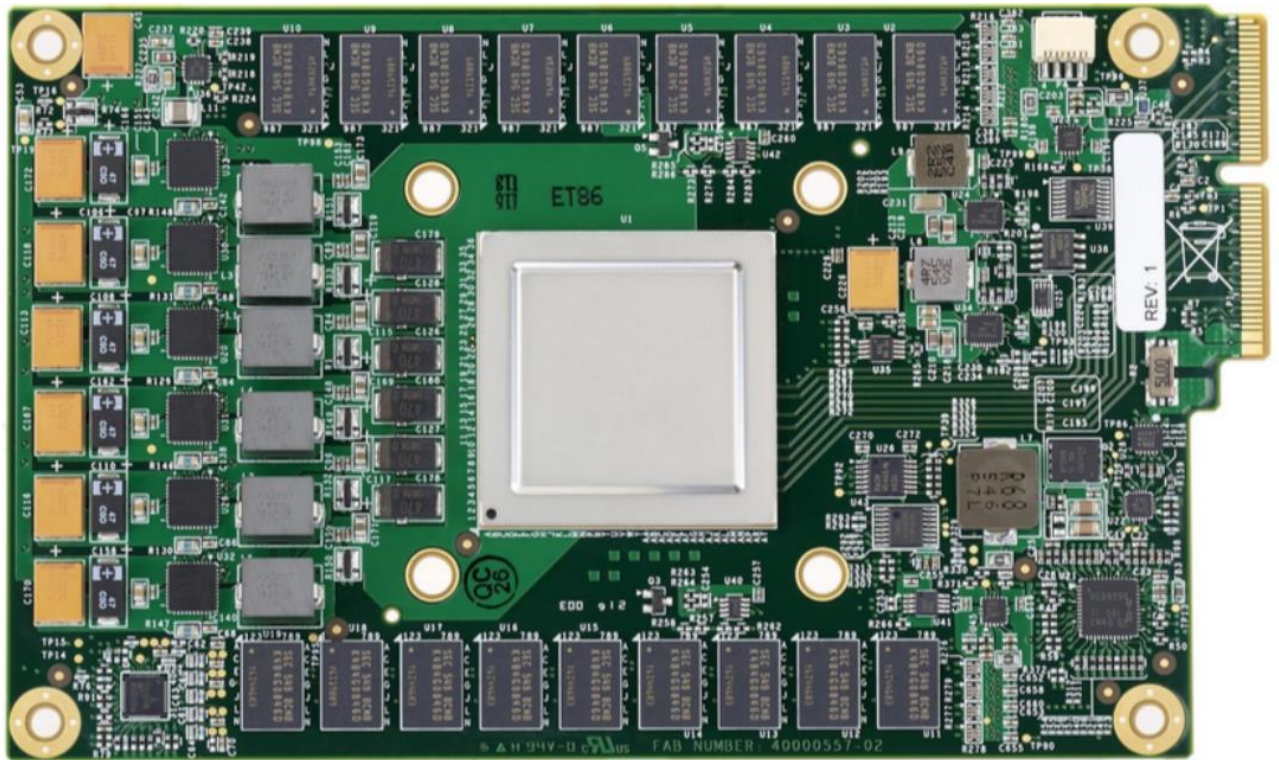


Рис. 2.9. Загальний вигляд друкованої плати TPU

Компанія Google у сфері штучного інтелекту досягла значних висот. І одним із факторів досягнення такого успіху є те, що Google розробила спеціалізовані апаратні засоби, які працюють з нейронними мережами набагато ефективніше, ніж традиційні центральні (CPU) та графічні (GPU) процесори.

В середньому тензорний процесор в 15-30 разів швидше здійснює обчислення, в порівнянні із традиційними серверними CPU і GPU. Продуктивність у розрахунку на ватт у TPU у 25-80 разів вища, ніж у центрального і графічного чіпів [28].

2.3.2. Структура тензорного процесора. TPU ASIC побудований на 28 нм процесі, працює на частоті 700 МГц і споживає 40 Вт під час роботи. Процесор представлений в якості зовнішньої прискорювальної карти, яка поміщається в слот для жорсткого диска SATA для встановлення. TPU підключений до свого хосту через шину PCIe Gen3 x16, яка забезпечує ефективну пропускну здатність 12.5 GB/s.

Програмованість була ще однією важливою метою дизайну для TPU. TPU не призначений для запуску тільки одного типу моделі нейронної мережі. Замість цього він розроблений таким чином, щоб бути достатньо гнучким для прискорення обчислень, необхідних для запуску багатьох різних моделей нейронних мереж.

Більшість сучасних CPU побудовані з використанням архітектури Reduced Instruction Set Computer (RISC). У RISC основна увага приділяється визначенню простих інструкцій (наприклад, завантаження, зберігання, додавання та множення), які зазвичай використовуються більшістю додатків, а потім виконують ці інструкції якомога швидше. Архітектура Complex Instruction Set Computer (CISC) була обрана як основа набору інструкцій TPU. Архітектура CISC фокусується на реалізації високорівневих інструкцій, які виконують більш складні завдання (такі як обчислення багаторазового множення і додавання) з кожною інструкцією.

TPU включає наступні обчислювальні ресурси:

- матричний множник (MXU): 65,536 8-бітових одиниць множення та додавання для операцій матриці;
- єдиний буфер (UB): 24 МБ SRAM, які працюють як регістри;
- активаційний блок (AU): функції активації [27, 28].

Типові процесори RISC містять інструкції для простих обчислень, таких як множення або додавання чисел. Це так звані скалярні процесори, оскільки вони обробляють одну операцію (скалярну операцію) з кожною інструкцією.

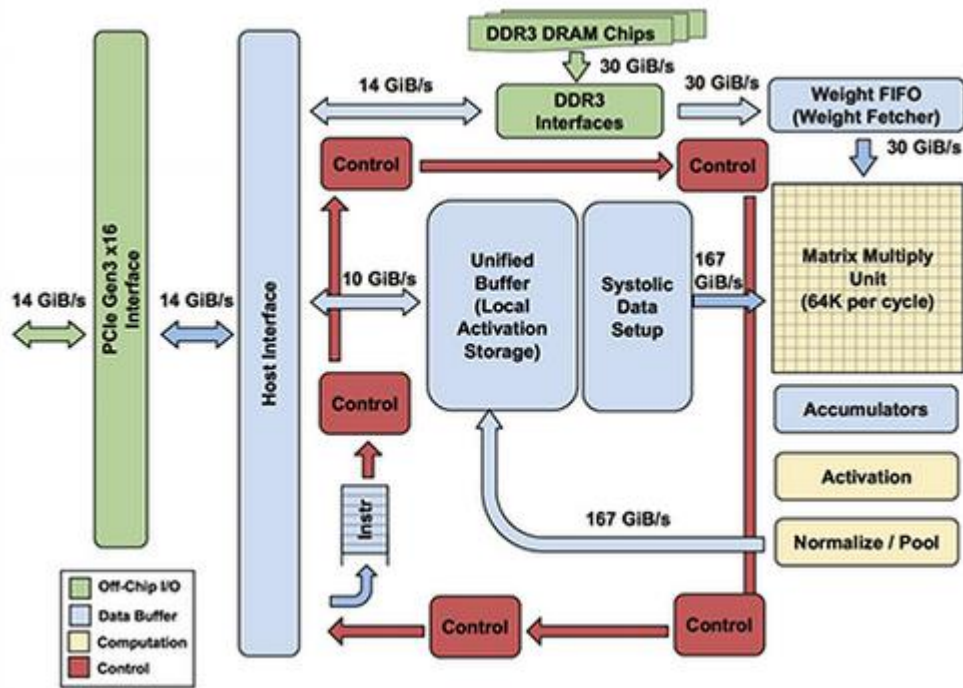


Рис. 2.10. Блокова діаграма TPU

Навіть незважаючи на те, що процесори працюють на тактовій частоті в гігагерцовому діапазоні, для виконання великих матричних операцій за допомогою скалярних операцій може знадобитися багато часу. Одним з ефективних і відомих способів поліпшення продуктивності таких великих матричних операцій є векторна обробка, де одна і та ж операція виконується одночасно з великою кількістю елементів даних. Мультипроцесори (SM) поточкових графічних процесорів є ефективними векторними процесорами. Машини з підтримкою векторної обробки можуть обробляти сотні до тисяч операцій за один цикл.

У випадку TPU, Google розробив свій матричний процесор MXU, який обробляє сотні тисяч операцій (матричних операцій) в одному тактовому циклі.

Для реалізації такого масштабного матричного процесора MXU має іншу архітектуру, ніж типові центральні та графічні процесори, що називається систолічним масивом. Дані процесори розроблені для виконання практично будь-яких розрахунків. Це комп'ютери загального призначення.



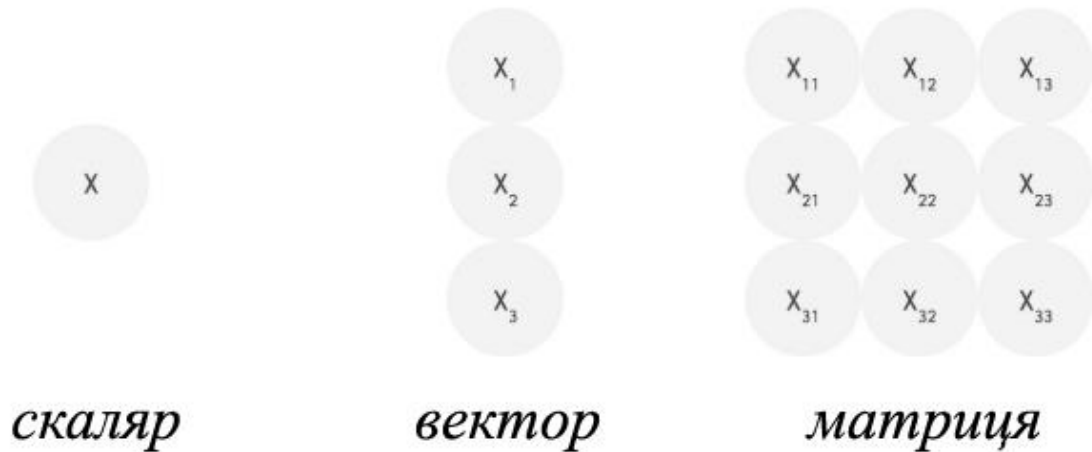


Рис. 2.11. Порівняння скалярних, векторних і матричних операцій

Щоб реалізувати цю властивість, процесори зберігають значення в регістрах, а програма повідомляє арифметичним логічним одиницям (ALUs), які регістри читати, яку операцію слід виконати (наприклад, додавання, множення) і регістр, в який помістити результат. Програма складається з послідовності операцій читання, роботи і запису. Всі функції, які підтримують спільність (регістри, ALUs і запрограмований контроль) мають менші витрати з точки зору потужності.

Для MXU матричне множення повторно використовує обидва входи багато разів як частину виведення. Ми можемо прочитати кожне вхідне значення один раз, але використовувати його для багатьох різних операцій, не зберігаючи його назад в регістр. ALUs виконують лише множення і доповнення в фіксованих моделях, що спрощує їх проектування.

Особливий вид систолічного масиву в MXU оптимізований при виконанні множення матриць, і не дуже добре підходить для обчислень загального призначення. Це робить інженерний компроміс: обмеження регістрів, контроль і експлуатаційної гнучкості в обмін на ефективність і набагато більш високу щільність роботи [28].

Модуль множення TPU має систолічний механізм масиву, який містить  $256 \times 256 = 65,536$  ALUs. Це означає, що TPU може обробити 65,536 операцій

множення та додавання для 8-бітових цілих чисел кожного циклу. Оскільки TPU працює на частоті 700 МГц, TPU може обчислити  $65536 \times 700000000 = 46 \times 10^{12}$  операцій множення і додавання або 92 TeraOps в секунду ( $92 \times 10^{12}$ ) в матричному блоці.

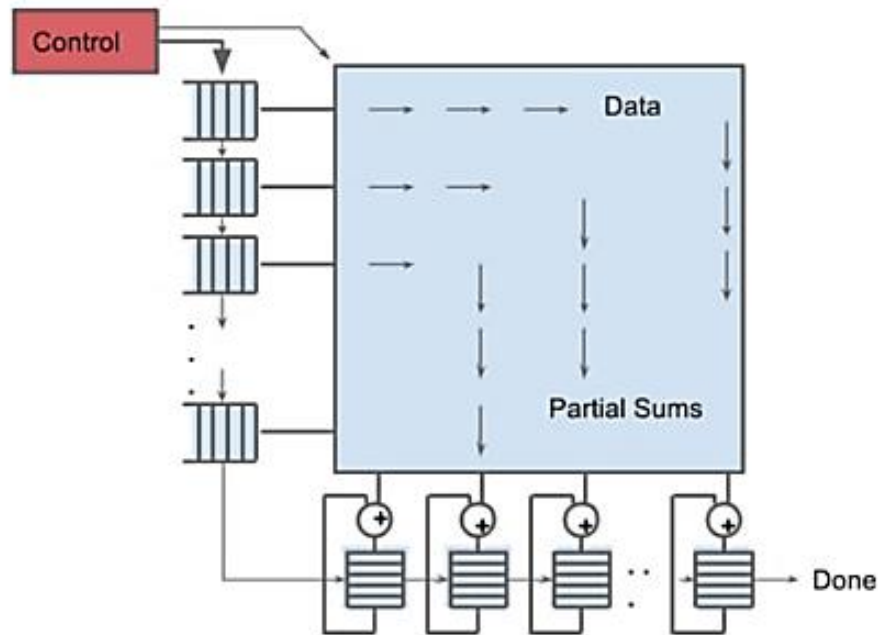


Рис.2.12. Блок помножувача матриці (MXU) TPU

Для порівняння, типовий процесор RISC без векторного розширення може виконувати тільки одну або дві арифметичні операції за інструкцію, а графічні процесори можуть виконувати тисячі операцій за інструкцію. З TPU єдиний цикл інструкції MatrixMultiply може викликати сотні тисяч операцій.

Під час виконання MXU, всі проміжні результати передаються між 64К ALUs без будь-якого доступу до пам'яті, значно знижуючи споживання енергії та збільшуючи пропускну здатність. Як результат, дизайн матричних процесорів на базі CISC забезпечує співвідношення продуктивності на ватт коефіцієнт: TPU забезпечує в 80 рази краще співвідношення в порівнянні з сучасними процесорами і в 29 разів більше співвідношення, ніж сучасні графічні процесори (рис. 2.5).

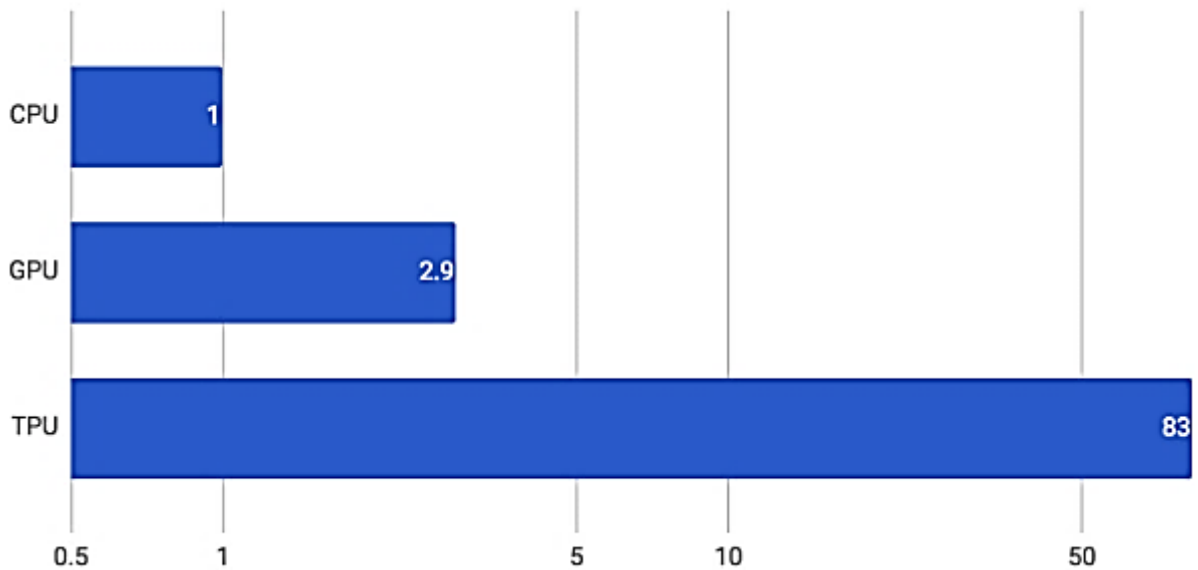


Рис. 2.13. Порівняння співвідношення продуктивності на ватт  
затраченої енергії CPU, GPU і TPU

Оскільки процесори загального призначення, такі як CPU і GPU, повинні забезпечувати хорошу продуктивність у широкому діапазоні застосувань, вони мають безліч складних, орієнтованих на продуктивність механізмів. Як побічний ефект, поведінка цих процесорів може бути важко передбачувана, що ускладнює гарантію певного обмеження затримки на опрацювання нейронних мереж. На відміну від цього, дизайн TPU є строго мінімальним і детермінованим, оскільки він повинен виконувати лише одне завдання за один раз: прогнозування нейронної мережі.

Незважаючи на те, що у CPU і GPU мають набагато більше арифметичних одиниць і велику кількість оперативної пам'яті, чіп TPU вдвічі менший за розмір інших чіпів.

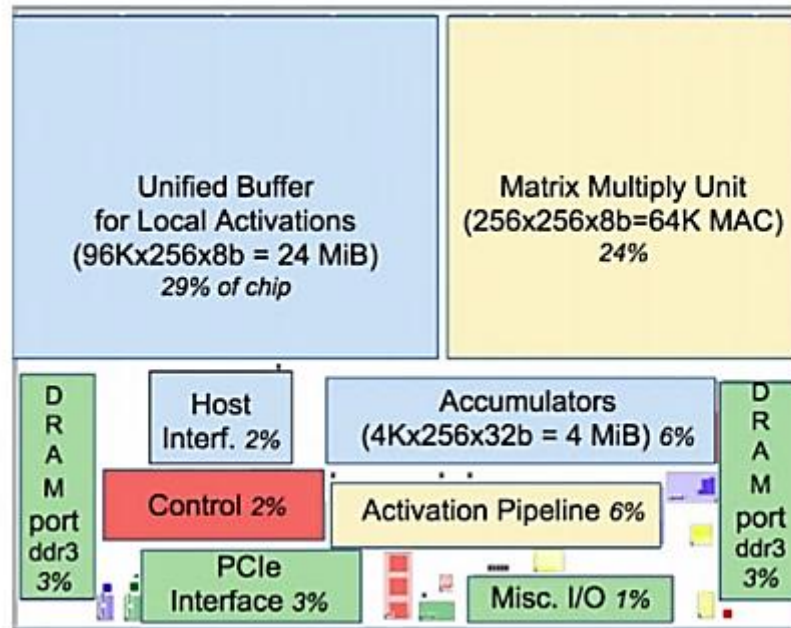


Рис. 2.14. Розміщення елементів TPU

З TPU можемо легко оцінити скільки часу потрібно для запуску нейронної мережі та прогнозування. Це дозволяє працювати з максимальною пропускнуою спроможністю даного чіпа [28].

### 2.3. Висновки до розділу

У даному розділі досліджено ефективні і дієві методи для представлення слів у вигляді векторів Word2Vec та Glove. Встановлено переваги та недоліки кожного з проаналізованих методів. Розглянуто два основних алгоритми навчання слів – CBOW і Skip-Gram. Також досліджено математичне забезпечення методів Word2Vec та Glove.

Здійснено аналіз апаратних засобів опрацювання природної мови з використанням нейронних мереж. Досліджено ефективність використання тензорних процесорів, в порівнянні із традиційними серверними CPU і GPU. Також, досліджено структуру тензорного процесора, який призначений для задач навчання нейронних мереж.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ З МЕТОЮ РОЗПІЗНАВАННЯ ВЛАСНИХ НАЗВ

#### 3.1. TensorFlow

В даний час найвідомішою бібліотекою глибокого навчання в світі є Tensorflow від Google.

TensorFlow – це бібліотека, розроблена командою Google Brain для прискорення машинного навчання і досліджень нейронних мереж. Вона була побудована для роботи на декількох процесорах або графічних процесорах і навіть мобільних операційних системах.

Tensorflow вперше представлена публіці в кінці 2015 року, а перша стабільна версія з'явилася в 2017 році. Tensorflow є відкритою програмною бібліотекою, що розповсюджується за ліцензією Apache Open Source.

Архітектура Tensorflow працює у наступні 3 етапи: попередня обробка даних, побудова моделі, навчання та оцінювання моделі.

Важливою особливістю TensorFlow є TensorBoard. TensorBoard дозволяє графічно та візуально відстежувати, що робить TensorFlow.

Ім'я Tensorflow безпосередньо походить від її основної структури: тензора. У Tensorflow всі обчислення включають тензори. Тензор – це вектор або матриця n-вимірна, що представляє всі типи даних. Всі значення в тензорі містять ідентичний тип даних з відомою (або частково відомою) формою. Форма даних – це розмірність матриці або масиву.

Тензор може бути створений з вхідних даних або результату обчислення. У TensorFlow всі операції здійснюються всередині графу. Граф – це набір обчислень, які відбуваються послідовно. Кожна операція називається операцією вузлів, що з'єднуються між собою. Граф показує операції і зв'язки

між вузлами. Однак він не відображає значення. Край вузлів – це тензор, тобто спосіб заповнити операцію даними.

Бібліотека Tensorflow включає в себе різні API для побудов архітектур глибокого навчання таких, як CNN або RNN. TensorFlow заснована на графових обчисленнях, що дозволяє розробнику візуалізувати побудову нейронної мережі за допомогою Tensorboard. Цей інструмент корисний для налагодження програми [29].

У 2017 році з'явився проект під назвою Deeplearn.js, який мав на меті включити ML/DL в JavaScript без API. Але були питання щодо швидкості. Дуже добре було відомо, що JavaScript – код не може працювати з GPU. Для вирішення цієї проблеми було введено WebGL. Це інтерфейс браузера до OpenGL. WebGL здійснює виконання JavaScript коду на GPU.

Навесні 2018 року команда DeepLearn.js об'єдналася в команду TensorFlow у Google і була перейменована в TensorFlow.js [30].

TensorFlow.js – це JavaScript бібліотека, розроблена Google для навчання та використання моделей машинного навчання (ML) в браузері. Це супутня бібліотека для TensorFlow, популярної бібліотеки ML для Python.

Як і в TensorFlow, основною структурою даних TensorFlow.js є тензор. API TensorFlow.js забезпечує методи створення тензорів з масивів JS, а також математичні функції, які працюють на тензорах [31].

Розроблену в компанії Google бібліотеку машинного навчання TensorFlow з відкритим вихідним кодом перевели на JavaScript. Нова бібліотека — Tensorflow.js працює безпосередньо в браузері і використовує можливості прискорення обчислень засобами WebGL (рис.3.1). Вона може працювати і в серверному варіанті на платформі Node.js. Перший варіант цієї бібліотеки Google випустила із назвою Deeplearn.js. Функціональність Deeplearn.js повністю включена в Tensorflow.js. Моделі для TensorFlow.js можна будувати з використанням або низькорівневих функцій лінійної алгебри в JavaScript, або інтерфейсів більш високого рівня.

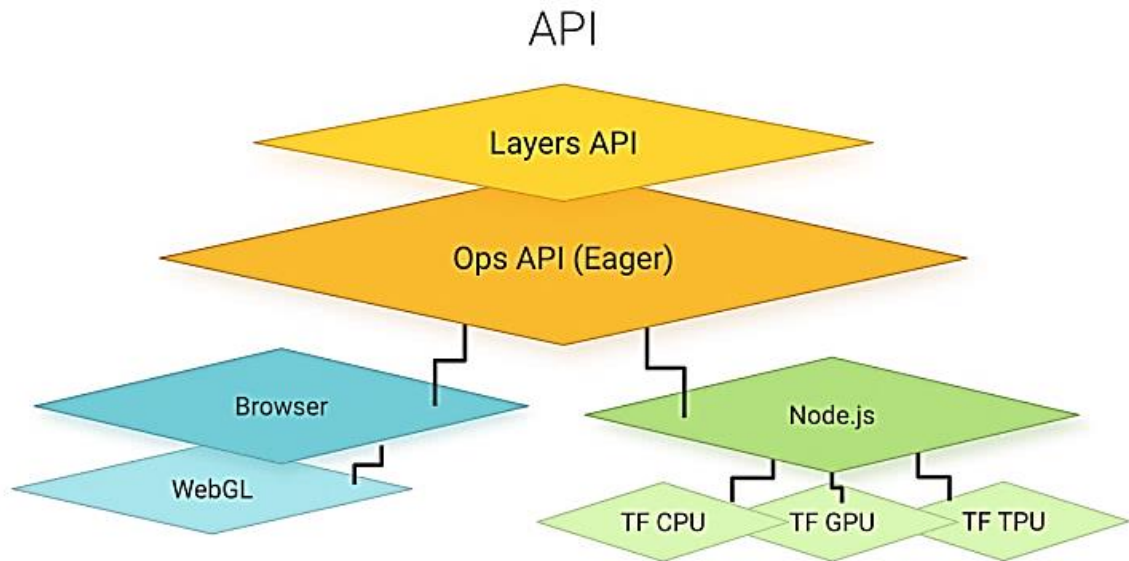


Рис.3.1. Архітектура Tensorflow.js

Інтерфейс TensorFlow.js побудований за зразком інтерфейсу до TensorFlow на мові Python, але поки не підтримує всієї його функціональності, а продуктивність Tensorflow.JS з WebGL становить 50-60% від продуктивності TensorFlow з інтерфейсом на мові Python і бібліотекою AVX [32].

### 3.2. Keras

Keras – це високорівневий API нейронних мереж, написаний на Python і здатний працювати поверх TensorFlow, CNTK або Theano. Він був розроблений з акцентом на забезпечення швидкого експериментування. Здатність перейти від ідеї до результату з найменшою затримкою є успіхом для проведення хороших досліджень.

Keras – це бібліотека глибокого навчання, що:

- надає можливість створювати прототипи швидко і легко (за рахунок зручності для користувача, модульності і розширюваності);
- має підтримку згорткових і рекурентних нейронних мереж, а також їх можливі комбінації;

– працює без проблем на CPU, GPU і TPU.

Keras має більш сильне прийняття як у промисловості, так і в дослідницькому співтоваристві, ніж будь-яка інша бібліотека машинного навчання, крім самого TensorFlow (Keras API є офіційним інтерфейсом TensorFlow, через tf.keras модуль).

Основна структура даних Keras – це модель, спосіб організації шарів. Найпростішим типом моделі є Sequential модель, що представляє собою лінійний стек шарів. Для більш складніших архітектур слід використовувати функціональний API Keras, що дозволяє будувати довільні графи шарів.

Основний принцип Keras полягає в тому, щоб зробити речі простішими, надаючи користувачеві можливість повністю контролювати їх, коли це необхідно (кінцевим результатом є легка розширюваність вихідного коду) [33].

### 3.3. Опис навчання та функціонування розроблюваного програмного забезпечення

В розроблюваній програмі використовується TensorFlow.js – бібліотека для розробки й навчання нейронної мережі з використанням мови програмування JavaScript, а також для розгортання в браузері або на Node.js.

Розроблювальна програма здійснює пошук власних назв в неструктурованому тексті й поділяє їх по заздалегідь визначених категоріях.

Спочатку необхідно підготувати навчальний набір даних. Оскільки, створювати набір даних розміченого тексту є досить дорогавартісним завданням, тому на сьогодні доступних корпусів тексту з розміткою є досить мало. Проте, існують популярні конференції, де люди змагаються у рішенні задач NER. Однією з таких конференцій є CoNLL, для якої створюється набір підготовлених даних, що містить близько 10 тисяч сутностей та близько 300 тисяч одиниць розміченого тексту. Даний навчальний набір будемо



використовувати у нашій системі для розпізнавання власних назв. У цьому наборі містяться дані, що поділені на чотири колонки (рис.3.2).

capital	NN	I-NP	O
in	IN	B-PP	O
early	JJ	I-NP	O
July	NNP	I-NP	O
Alexander	NNP	I-NP	B-PER
have	VB	I-VP	O
Ukraine	NNP	I-NP	B-LOC
Czech	NNP	B-NP	B-LOC
Republic	NNP	I-NP	I-LOC
done	VBN	I-VP	O
it	PRP	I-NP	O
but	CC	O	O
when	WRB	I-ADVP	O
after	IN	I-PP	O

Рис.3.2. Фрагмент навчального набору даних

У першій колонці міститься власне саме слово, у другій – позначення належності даного слова до певної частини мови, що визначається як по контексті, так і по визначенні цього слова (NN – іменник; IN – сполучник, прийменник; JJ – прикметник; NNP – іменник, власне ім'я; VB – основна форма дієслова; VBN – дієслово, дієприкметник минулого часу; PRP – особовий займенник; CC – сполучник; WRB – wh-прислівник і т.д.). У третій колонці вказується приналежність слова до певної фрази (I-TYPE, B-TYPE, O). I-TYPE означає, що слово знаходиться в середині фрази типу TYPE. Якщо дві фрази одного типу слідуєть один за одним, перше слово матиме тег B-TYPE, щоб показати, що воно починає нову фразу. Слова з тегом O – не є частиною мови. І у четвертій колонці вказує приналежність даної сутності до певної категорії (люди (PER), організації (ORG), місця (LOC) й інше (O)).

Для того, щоб програма могла знаходити й класифікувати іменовані сутності в неструктурованому тексті, спершу необхідно завантажити вхідні дані.

Як відомо, нейронні мережі не вміють працювати із самими словами, тільки із числами. Тому необхідно здійснити перетворення кожного слова у вигляді числа. Це є не складним завданням. Необхідно перерахувати всі унікальні слова й записати номер кожного даного слова замість нього самого. Для цієї задачі Keras містить Tokenizer клас.

Щоб зберігати цифри і слова складемо словник. Словник також має підтримувати невідомі слова (UNK) в ситуаціях, коли будемо здійснювати передбачення для нового рядка із словами, яких не має у словнику. Також словник повинен містити слово, щоб позначати доповнення (PAD). Для нейронної мережі всі рядки повинні мати однаковий розмір, тому коли одна стрічка буде меншою за іншу, необхідно заповнити різницю цим словом. Для цієї задачі Keras має вбудований метод pad\_sequence (рис.3.3).

```
trainData = pad_sequences(
    trainData["wordsSequences"],
    (maxlen = maxSequenceLength),
    (padding = "post"),
    (truncating = "post"),
    (value = padId)
);

validData = pad_sequences(
    validData["wordsSequences"],
    (maxlen = maxSequenceLength),
    (padding = "post"),
    (truncating = "post"),
    (value = padId)
);
```

Рис.3.3. Лістинг виклику методу pad\_sequence

Метод pad\_sequences приймає наступні параметрами:

- sequences – список послідовностей;
- maxlen – максимальна довжина всіх послідовностей;
- padding – вказує, яким чином буде доповнюватись послідовність, щоб досягти довжини maxlen, аргумент «pre» – вказує, що доповнення буде здійснюватися перед послідовністю, «post» – після послідовності;
- truncating – здійснює видалення значень з послідовності, якщо послідовність більша за maxlen й приймає аргументи аналогічні, як в padding;
- value – значення слова, яке позначає доповнення.

Для розв'язання задачі векторного представлення слів використовується метод Glove. Побудова моделі нейронної мережі потребує правильної конфігурації кожного шару і подальшої компіляції моделі.

Нейронна мережа складається з чотирьох основних об'єктів:

- шари (layers) – все навчання відбувається в шарах. Є три шари: вхідний, прихований і вихідний;
- функція втрат (loss function) – використовується для оцінки ефективності фази навчання;
- оптимізатор (optimizer) – покращує навчання за допомогою оновлення знань у мережі, що оснований на вхідних даних і функції втрат;
- метрики (metrics) – використовуються для моніторингу тренування і тестування моделі.

Алгоритм навчання нейронної мережі зображено на рис.3.4.

Далі необхідно створити модель нейронної мережі. Keras надає можливість використовувати два типи моделей: послідовну і з функціональним API, яку й будемо використовувати.

Як відомо базовими будівельними блоками нейронної мережі є шари. Основна частина нейронного навчання складається з послідовності простих шарів. Тому необхідно додати вхідні, приховані й вихідні шари. В якості функції активації для прихованих шарів використовуватимемо функцію активації tanh (рис.3.4).

```
hidden_layer = Dense(64, (activation = "tanh"))(outputs);
hidden_layer = Dropout(0.25)(hidden_layer);
```

Рис.3.4. Лістинг прихованих шарів

Для запобігання перенавчання використовуватимемо між шарами функцію dropout з коефіцієнтом 25%. Для кожного шару застосовується функція dense для з'єднання шарів один з одним. Функція dense приймає параметри, які можна налаштувати під час навчання (рис.3.5).

```
l1 = Dense(num_hidden_units)(encoding);
l1 = Dropout(0.25)(l1);
l2 = Dense(tags_size)(l1);
out = Softmax((name = "Softmax"))(l2);
model = Model((inputs = words_input), (outputs = out));
```

Рис.3.5. Лістинг з'єднання шарів один з одним

Перед тим як модель буде готова для навчання, нам потрібно вказати ще декілька параметрів таких, як: метрики, оптимізатор і функцію втрат, вони додаються на кроці компіляції моделі.

Після побудови моделі, необхідно скомпілювати її, навчити й зберегти результати (рис.3.6).

В якості мінімізації функції втрат, будемо використовувати перехресну ентропію (categorical crossentropy):

$$F_B = - \sum_{i=1}^n (x_i \cdot \log(y_i)) . \quad (3.1)$$

Оптимізаційний алгоритм – adaptive moment estimation (Adam) й метрику

categorical accuracy, яка перевіряє чи індекс максимального істинного значення дорівнює індексу максимального передбаченого значення.

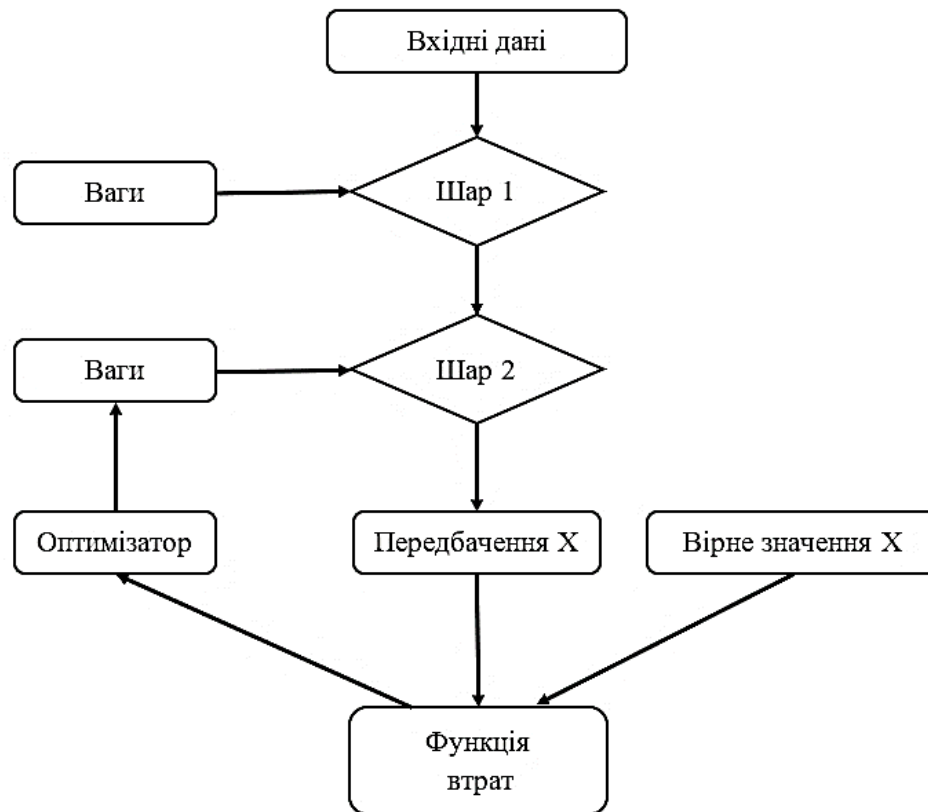


Рис.3.6. Алгоритм навчання нейронної мережі

Тренування моделі здійснюється з допомогою виклику методу `model.fit` (рис.3.7). Даний метод приймає наступні параметри:

- `trainData` – вектор або масив навчальних даних;
- `trainTags` – вектор або масив цільових атрибутів даних;
- `epochs` – вказує кількість навчальних ітерацій;
- `batch_size` – кількість навчальних даних, що обробляються одночасно за одну ітерацію алгоритму;
- `validation_data` – дані на основі яких, можна здійснити оцінку втрат навчання моделі в кінці кожної епохи, також варто додати, що модель не навчається за цими даними.

```

model.compile(
    (loss = "categorical_crossentropy"),
    (optimizer = "Adam"),
    (metrics = ["categorical_accuracy"])
);
model.fit(
    trainData,
    trainTags,
    (epochs = args.epochs),
    (batch_size = args.batch_size),
    (validation_data = (validData, validTags))
);

```

Рис.3.7. Лістинг методів компіляції й тренування моделі

Після того, як модель навчена, нам необхідно завантажити модель з допомогою методу `tf.loadLayersModel`.

Далі, необхідно забезпечити передачу даних із простого текстового формату рядка у формат TF – тензор й створити масив `tf.tensor`. Використовуючи метод `predict` (рис.3.8) здійснюємо передбачення й виводимо спрогнозовані дані у браузер.

```

predictions = model.predict(data);

```

Рис.3.8. Лістинг використанням методу `predict`

У програмі реалізована задача обробки природної мови (NLP) з метою розпізнавання власних назв.

Програма намагається знайти і класифікувати власні назви у неструктурованому тексті, що введений користувачем в заздалегідь визначені категорії такі, як: люди (B-PER, I-PER), місця (B-LOC, I-LOC), й організації (B-ORG, I-ORG). Результати роботи програми зображені на рис.3.9 – 3.10.

Enter sentence =)

I am a student of Ternopil Ivan Puluj National Technical University

Search Entities

I am a student of Ternopil **B-LOC** Ivan **B-PER** Puluj **I-PER** National Technical University **I-ORG**

Рис.3.9. Результат обробки інформації введеної користувачем

Enter sentence =)

NOW I AM IN THE TERNOPIL BUT NOT IN LVIV

Search Entities

NOW I AM IN THE TERNOPIL **B-LOC** BUT NOT IN LVIV **B-LOC**

Рис.3.10. Результат роботи програми

### 3.4. Оцінка якості розробленого програмного забезпечення

На конференції CoNLL (Conference on Computational Natural Language Learning) було запропонований простий і ефективний спосіб оцінки: іменована сутність виділена системою вірно, якщо її клас і границі, позначені системою співпадають із класом і границями, розміченими в корпусі, в іншому випадку сутність виділена не вірною.

Одним із критеріїв якості є точність (precision,  $P$ ). Тобто в нашому випадку це відношення кількості власних назв, які система розпізнала вірно до загальної кількості сутностей, що були виділені системою:

$$\text{Точність } (P) = \frac{\text{кількість вірно виділених власних назв}}{\text{загальна кількість сутностей виділених системою}} . \quad (3.2)$$

Іншим критерієм є повнота (recall,  $R$ ) – відображає відношення кількості власних назв, які система правильно розпізнала до загальної кількості власних назв, що містяться в корпусі тексту:

$$\text{Повнота } (R) = \frac{\text{кількість вірно виділених власних назв}}{\text{загальна кількість власних назв в корпусі тексту}} . \quad (3.3)$$

Приклад досить точного, але не повного класифікатора для розпізнавання власних назв, це класифікатор який виділяє в тексті одну правильну сутність і більше нічого. Також можливий досить повний, але не точний класифікатор – класифікатор, що виділяє власні назви на будь якому відрізьку тексту, таким чином, окрім усіх еталонних сутностей, що містяться в навчальному наборі, класифікатор виділятиме також надлишковий текст.

Тому основним критерієм для задачі розпізнавання власних назв є F-міра. F-міра являє собою спільну оцінку точності і повноти системи. F-міра дозволяє отримати більш збалансовану характеристику системи, чим попередні критерії й визначається:

$$\text{F-міра} = \frac{2PR}{P + R} . \quad (3.4)$$

Оцінювання якості системи проводимо наступним чином, три різних корпуси тексту в яких міститься по 100 сутностей подамо на вхід системи,



після чого здійснюємо підрахунок кількості вірно виділених сутностей й загальної кількості виділених сутностей. Відповідно до отриманих значень визначаємо ефективність розробленого програмного забезпечення за значеннями точності, повноти і F - міри (табл.3.1).

Таблиця 3.1

### Результати оцінювання якості

№ тесту	Точність (P)	Повнота (R)	F-міра
1	0,77	0,79	0,78
2	0,68	0,72	0,7
3	0,74	0,85	0,79
Середнє значення	0,73	0,79	0,76

Як бачимо з табл.3.1 розроблена система здійснює коректне розпізнавання власних назв із попередньо поданого на вхід корпусу текст із точністю 76%.

### 3.5. Висновки до розділу

В даному розділі розглянуто бібліотеку TensorFlow, що використовується для прискорення машинного навчання і дослідження нейронних мереж й її супутню бібліотеку TensorFlow.js, яка призначена для навчання та використання моделей машинного навчання в браузері. Розглянуто високорівневу API Keras, що здатна працювати поверх TensorFlow.

Здійснено опис навчання та функціонування розробленого програмного забезпечення для опрацювання природної мови з метою розпізнавання власних назв. Також, розглянуто принцип навчання нейронної мережі й наведено результати роботи реалізованої програми. Здійснено оцінку якості розробленого програмного забезпечення на основі критеріїв таких, як точність, повнота і F-міра.

## РОЗДІЛ 4

### ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

#### 4.1. Визначення стадій технологічного процесу та загальної тривалості проведення НДР

Економічне обґрунтування дипломної роботи магістра є суттю даного розділу, оскільки, дозволяє встановити доцільність проведення науково-дослідних робіт і економічно обґрунтувати доцільність застосування тих чи інших засобів.

Метою дипломної роботи магістра є дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв.

Як відомо, розробка надійної і ефективною системи вимагає значних затрат часу. Слід зауважити, що затрати часу залежать від кваліфікації розробника і його можливостей. Розробник повинен у достатній мірі володіти навиками програмування, вміти адекватно застосовувати математичний апарат, бути добре обізнаним з об'єктом дослідження.

Розробку даної системи можна поділити на такі етапи:

- 1) постановка задачі;
- 2) збір інформації по тематиці роботи;
- 3) аналіз методів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв;
- 4) розробка алгоритму програми для розпізнавання власних назв;
- 5) налаштування середовища розробки і роботи вже готової програми;
- 6) написання програми;
- 7) тестування розробленої програми;
- 8) написання і оформлення документації.

Для оцінки тривалості виконання окремих робіт використовують нормативи часу або попередній досвід. До таких нормативів відносять тривалість написання операцій (команд), які в деяких підприємствах становлять: для однієї операції - 0,5-1,6 год та 8 годин для п'яти операцій (тривалість зміни).

У разі їх відсутності звертаються до експертних оцінок по встановленню тривалості кожного етапу (стадії):

при трьох оцінках

$$T_{ec} = (t_{min} + 4t_{н.й} + t_{max}) / 6, \quad (4.1)$$

при двох оцінках

$$T_{ec} = (3t_{min} + 2t_{max}) / 5, \quad (4.2)$$

де  $T_{ec}$  – очікуване (середнє) значення тривалості виконання етапу (стадії);

$t_{min.}, t_{н.й.}, t_{max}$  – відповідно мінімальна, найбільш імовірна і максимальна оцінки тривалості виконання етапу (стадії).

Для визначення загальної тривалості проведення НДР (розробки програмного продукту) доцільно дані витрат часу на виконання окремих стадій (етапів) звести у табл.4.1.

Витрати часу наукового керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

Таблиця 4.1

## Основні етапи і час їх виконання у НДР

№ з/п	Етап	Середній час виконання етапу, год	
		інженер	керівник
1	2	3	10
1	постановка задачі	3	1
2	збір потрібної інформації і наступне її опрацювання	13	5
3	аналіз методів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв	21	8
4	розробка алгоритму програми для розпізнавання власних назв	13	5
5	налаштування середовища розробки і роботи вже готової програми	5	3
6	написання програми	89	13
7	тестування розробленої програми	13	8
8	написання і оформлення документації	21	13
разом		175	56

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України «Про оплату праці» заробітна плата –

це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% – 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати – 21 дні/міс., що відповідає 168 год./міс. Розмір місячних окладів керівника та інженерів слід приймати згідно існуючих на даний час норм. Основна заробітна плата розраховується за формулою:

$$Z_{осн} = T_c \cdot K_z, \quad (4.3)$$

де  $T_c$  – тарифна ставка, грн.;

$K_z$  - кількість відпрацьованих годин.

Посадові оклади (тарифні ставки) за розрядами Єдиної тарифної сітки визначаються шляхом множення окладу (ставки) працівника 1 тарифного розряду на відповідний тарифний коефіцієнт. У разі коли посадовий оклад (тарифна ставка) визначені у гривнях з копійками, цифри до

0,5 відкидаються, від 0,5 і вище - заокруглюються до однієї гривні. У 2018 році посадові оклади (тарифні ставки) розраховуються згідно з Законом України «Про Державний бюджет України на 2019 рік».

Мінімальна зарплата в 2019 р. складає 4173 грн., в погодинному розмірі 25,13 грн., прийmemo 80 грн. для інженера, для керівника – 130 грн.

Тарифні ставки: керівник проекту – 130 грн./год., інженер – 80,0 грн./год.

Основна заробітна плата становитиме:

$$Z_{осн} = T_{осн} \cdot K_{ГОД}. \quad (4.4)$$

Керівник проекту:

$$Z_{осн} = 130 \cdot 56 = 7\,280 \text{ грн.}$$

Інженер:

$$Z_{осн} = 80 \cdot 175 = 14\,000 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15% від суми основної заробітної плати:

$$Z_{дод} = Z_{осн} \cdot K_{додл}, \quad (4.5)$$

де  $K_{додл}$  – коефіцієнт додаткових виплат працівникам 0,1.

Керівник проекту:

$$Z_{дод} = 7\,280 \cdot 0.15 = 1\,092 \text{ грн.}$$

Інженер:

$$Z_{\text{од}} = 14\,000 \cdot 0.15 = 2\,100 \text{ грн.}$$

Звідси загальні витрати на оплату праці ( $V_{\text{оп}}$ ) визначаються за формулою (4.6) і становлять:

$$V_{\text{оп}} = Z_{\text{осн}} + Z_{\text{од}}. \quad (4.6)$$

Керівник проекту:

$$V_{\text{оп}} = 7\,280 + 1\,092 = 8\,372 \text{ грн.}$$

Інженер:

$$V_{\text{оп}} = 14\,000 + 2\,100 = 16\,100 \text{ грн.}$$

Таким чином загальна сума становить 24 472 грн. Крім того, слід визначити відрахування на соціальні заходи:

- податок на доходи фізичних осіб: 18%;
- військовий збір 1,5%;
- єдиний внесок 22%.

У сумі зазначені відрахування становлять 41,5%. Отже, загальна сума відрахувань на соціальні заходи становитиме:

$$V_{\text{с.з.}} = \text{ФОП} \cdot 0,415, \quad (4.7)$$

$$V_{\text{с.з.}} = 24\,472 \cdot 0,415 = 10\,155,88 \text{ грн.,}$$

де ФОП – фонд оплати праці, грн.

Проведені розрахунки витрат на оплату праці зведемо у табл.4.2.

Таблиця 4.2

## Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарах. на ФОП, грн.	Всього витрати на оплату праці, грн. 8=5+6+7
		Тарифна ставка, грн.	К-сть відпрацьов. год.	Фактично нарах. з/пл., грн.			
1	2	3	4	5	6	7	8
1.	Керівник проекту	130	56	7 280	1 092	3 474,38	11 846,38
2.	Інженер	80	175	14 000	2 100	6 681,5	22 781,5
Разом				21 280	3 192	10 155,88	34 627,88

## 4.3. Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_E = W \cdot T \cdot S, \quad (4.8)$$

де  $W$  – необхідна потужність, кВт;

$T$  – кількість годин роботи обладнання;

$S$  – вартість кіловат-години електроенергії.

Вартість роздрібного тарифу на електричну енергію ВАТ «Тернопільобленерго», що відпускається для споживачів 2 класу напруги становить 241,548 коп./кВт.год.

Потужність комп'ютера – 400 Вт з підключеним маршрутизатором, кількість годин роботи обладнання згідно табл.4.1 – 231 година.



$$Z_E = 0,4 \cdot 231 \cdot 2.4155 = 223,19 \text{ грн.}$$

#### 4.4. Розрахунок витрат на матеріали

Результати розрахунку затрат на матеріали зводяться у табл.4.3.

Таблиця 4.3

#### Визначення величини затрат на матеріал

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Затрати матеріалів, грн	Транспортно-заготівельні витрати, грн	Загальна сума витрат на матеріали, грн
Папір А4-80	пачка	1	100	100	–	100
Ватман	шт.	10	10	100	–	100
Чорнила для принтера	шт.	1	80	80	–	80
Разом						280

#### 4.5. Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B \cdot H_A}{100}, \quad (4.9)$$

де  $A$  – амортизаційні відрахування за звітний період, грн.,

$B_B$  – балансова вартість комп'ютера, на початок звітного періоду, грн.,

$H_A$  – норма амортизації, %.

$$A = \frac{24\,000 \cdot 15\%}{100\%} = 3\,600.$$

#### 4.6. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

Накладні витрати можуть становити 20% від суми основної та додаткової заробітної плати працівників:

$$H_B = V_{O.P.} \cdot 0,2, \quad (4.10)$$

$$H_B = 24\,472 \cdot 0,2 = 4\,894,4 \text{ грн.}$$

#### 4.7. Складання кошторису витрат та визначення собівартості НДР

Результати проведених вище розрахунків зведемо у табл.4.4. Собівартість ( $C_B$ ) НДР розрахуємо за формулою:

$$C_B = B_{O.п.} + B_{C.з.} + Z_E + Z_B + T_B + A + H_B, \quad (4.11)$$

$$C_B = 24\,472 + 10\,155,88 + 223,19 + 280 + 3\,600 + 4\,894,4 = 43\,625,47 \text{ грн.}$$

Таблиця 4.4

#### Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
1	2	3
Витрати на оплату праці (основну і додаткову заробітну плату)	24 472	56,1
Відрахування на соціальні заходи	10 155,88	23,28
Матеріальні витрати	280	0,64
Витрати на електроенергію	223,19	0,51
Амортизаційні відрахування	3 600	8,25
Накладні витрати	4 894,4	11,22
Собівартість	43 625,47	100

#### 4.8. Розрахунок ціни НДР

Ціну НДР можна визначити за формулою:

$$\Pi = \frac{C_B \cdot (1 + P_{\text{рен}}) + K \cdot B_{\text{н.і.}}}{K}, \quad (4.12)$$

$P_{\text{рен.}}$  – рівень рентабельності, 30 %;

$K$  – кількість замовлень;

$B_{\text{н.і.}}$  – вартість носія інформації, грн.

$$\Pi = \frac{43\,625,47 \cdot (1 + 0,3) + 1 \cdot 150}{1} = 56\,863,11 \text{ грн.}$$

Таким чином ціна рівна 56 863,11 грн.

Визначимо величину прибутку:

$$\Pi = \Pi - C_B. \quad (4.13)$$

Згідно формули (4.13) отримаємо 13 237,64 грн.

4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність ( $E_p$ ) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \Pi / C_B, \quad (4.14)$$

де  $\Pi$  – прибуток;

$C_B$  – собівартість.

$$E_p = 13\,237,64 / 43\,625,14 = 0,3.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень ( $T_p$ ):

$$T_p = E_p, \quad (4.15)$$

$$T_p = 1 / 0,3 = 3,33 \text{ р.}$$

Про доцільність розробки програми можна сказати при врахуванні наступних критеріїв:

Таблиця 4.5

#### Техніко-економічні показники НДР

№ п/п	Показник	Значення
1	Собівартість, грн	43 625,47
2	Плановий прибуток, грн	13 237,64
3	Ціна, грн	56 863,11
4	Економічна ефективність	0,3
5	Термін окупності, рік	3,33

У результаті проведення розрахунків можна зробити висновок: розробка матиме оптимальну економічну ефективність 0,3 і термін окупності становитиме 3,33 року.

Варто зазначити, що дані розрахунки носять номінальний характер і основна їх мета оцінити приблизну вартість дослідження та створення даного продукту. Номінальний характер розрахунків зумовлений тим, що даний програмний продукт має дослідницьке призначення.

## РОЗДІЛ 5

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 5.1. Охорона праці

Дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв здійснювалося з дотриманням правил та норм охорони праці і вимог техніки безпеки, що є невід'ємною частиною виконання всіх видів робіт при проведенні даного дослідження.

Зокрема, при дослідженні методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв були враховані:

1. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018 № 207 (набули чинності 18 травня 2018 року) [34].

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10 грудня 1998 року № 7 [35].

3. Санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, затверджених постановою Головного державного санітарного лікаря України від 01 грудня 1999 року № 37 [36].

4. Закон України «Про охорону праці» [37].

5. Правила безпечної експлуатації електроустановок споживачів, затвердженим наказом Комітету по нагляду за охороною праці Міністерства праці та соціальної політики України від 09 січня 1998 року № 4,

zareєстрованим в Міністерстві юстиції України 10 лютого 1998 року за № 93/2533 [38].

6. Правила пожежної безпеки в Україні, затверджені наказом Міністерства внутрішніх справ України від 30 грудня 2014 року № 1417 [39] та інші нормативні документи.

Приміщення, де проводилося дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв є достатньо просторим і добре провітрюваним, що забезпечується припливно-витяжною вентиляцією відповідно до ДБН В.2.5-56:2014 [40]. Приміщення, також оснащено аптечкою першої медичної допомоги.

Площа приміщення на один комп'ютер становить  $18 \text{ м}^2$ , а об'єм –  $69 \text{ м}^3$ . Віконні прорізи приміщення обладнані жалюзями. Для внутрішнього оздоблення приміщення використано дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі світло-жовтого кольору 0,75 й для стін світло-бежевого кольору 0,6. Покриття підлоги матове з коефіцієнтом відбиття 0,48. Поверхня підлоги є рівною з антистатичними властивостями. Робоче місце розташоване по відношенню до віконних прорізів так, що природне світло падає зліва. Приміщення, також обладнане шафами й полицями з урахування вимог до площі приміщення [35].

Приміщення, де проводилось дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв, відповідно до ДСанПН 3.3.2.007-98 [35] забезпечує оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря. Рівні електромагнітного випромінювання та магнітних полів регламентовані нормами [36] та не перевищують допустимі рівні. Рівні звукового тиску, рівні звуку та еквівалентні рівні звуку на робочих місцях, відповідають вимогам згідно з ДСН 3.3.6.037-99 [36].

Для забезпечення нормованих рівнів шуму у приміщенні використовуються шумопоглинаючі засоби. Значення характеристик вібрації на робочому місці не перевищує допустимі рівні відповідно до ДСанПІН 3.3.2.007-98 [35].

Оскільки, основне навантаження під час проведення дослідження припадало на зорову систему, тому приміщення також забезпечене природним та штучним освітленням. Природне світло проникає через бічні світлопрорізи, зорієнтовані, на північ, і забезпечує коефіцієнт природної освітленості (КПО) не нижче 1,5%. Штучне освітлення в приміщенні здійснюється системою загального рівномірного освітлення. Як джерела світла штучного освітлення застосовуються люмінесцентні лампи типу ЛБ. Значення освітленості на поверхні робочого столу в зоні розміщення документів становить 420 лк. При цьому яскравість бліків на екрані ВДТ не перевищує 40 кд/м<sup>2</sup>.

Робоче місце, де проводилось дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв, згідно із санітарними нормами [35] та [36], забезпечено наступними вимогами.

Конструкція робочого місця забезпечує підтримання оптимальної робочої пози. Висота робочого столу становить 710 мм, а ширина і глибина – 1400 мм і 800 мм відповідно. Робочий стілець був вибраний підйомно-поворотним, регульованим за висотою, за кутом і за нахилом сидіння та спинки. Екран ВДТ розташовується на відстані 600 мм від очей, що забезпечує зручність зорового спостереження у вертикальній площині під кутом +30° від нормальної лінії погляду [35].

Все обладнання, яке використовувалося під час дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв було сертифіковане для використання на території України.



У процесі роботи над дослідженням було дотримано правильний режим праці та відпочинку. Впродовж дня передбачалися: перерви для відпочинку і вживання їжі (обідні перерви), а також перерви для особистих потреб [35].

Електробезпека на робочому місці, відповідає вимогам Правил безпечної експлуатації електроустановок споживачів, затверджених наказом Міністерства праці та соціальної політики від 10.02.98 р. № 93/2533 [38].

Особливу увагу приділено заходам дотримання протипожежної безпеки в приміщенні, де проводилось дослідження відповідно до правил пожежної безпеки в Україні, затверджені наказом Міністерства внутрішніх справ України від 30 грудня 2014 року № 1417 [39].

Приміщення обладнане системою автоматичної пожежної сигналізації із димовими пожежними сповіщувачами з розрахунку 2 шт. на кожні 20 м<sup>2</sup> площі приміщення, а також оснащене переносними вуглекислотними вогнегасниками з розрахунку 1 на 50 м<sup>2</sup>. У всьому приміщенні лінії електромережі забезпечені від виникнення короткого замикання, а також від перепадів напруги, що може викликати збої в роботі електронно-обчислювальної техніки [39].

В дипломній роботі магістра проведено аналіз норм праці, шкідливих та небезпечних чинників при дослідження методів та засобів опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв, а також описано параметри і характеристики приміщення, заходи, які були виконані для забезпечення належних умов роботи.

5.2. Аварія на радіаційно небезпечних підприємствах із джерелом іонізуючого випромінювання, їх несанкціоноване перевезення, зберігання, використання та виявлення в непристосованих місцях

В національній економіці України є багато об'єктів, які з тою чи іншою метою у своїй виробничій діяльності використовують потужні джерела іонізуючого випромінювання (ДІВ), що може призвести до виникнення радіаційних аварій з досить серйозними наслідками, як для персоналу цих об'єктів, так і для населення, що проживає на прилеглій території.

Радіаційні аварії (РА) є найбільш небезпечними, оскільки вплив радіоактивних речовин (РР) та інших джерел іонізуючого випромінювання (ДІВ) людина не може відчутти за допомогою своїх органів чуттів, а виявляє тільки за допомогою спеціальних приладів. Крім того, радіоактивне забруднення, яке може бути заподіяне радіаційною аварією, залишається на тривалий час, а дезактивація має обмежені можливості.

Недопустима недбалість і безграмотність при поводженні з радіоактивними матеріалами лежать в основі багатьох випадків радіаційних аварій. За даними Міжнародної агенції з атомної енергії (МАГАТЕ), в світі щорічно реєструється близько 200 випадків вкрадених, загублених або кинутих напризволяще радіонуклідних джерел. Такі джерела можуть потрапляти в металобрухт, створюючи аварійну ситуацію при зборі металобрухту і його переробці, або ж бути використані для здійснення актів радіаційного тероризму [41].

Правові засади провадження діяльності з використання ДІВ в Україні базуються на основних принципах радіаційного захисту (ст. 4 Закону України «Про використання ядерної енергії та радіаційну безпеку»), регулюються законами України, нормативно-правовими актами Уряду, правилами та вимогами безпеки, розробленими органами державного регулювання у сфері ядерної та радіаційної безпеки [42].

Джерелами іонізуючого випромінювання (ДІВ) називають матеріали, що містять РР, або технічні пристрої, що випромінюють або здатні випромінювати іонізуюче випромінювання (рентгенівські установки, прискорювачі заряджених частинок, атомні реактори).

Розрізняють закриті і відкриті джерела іонізуючого випромінювання, що містять РР.

Закритим джерелом називають радіонуклідні джерела, поміщені в тверду неактивну оболонку, яка за нормальних умов експлуатації запобігає потраплянню РР до навколишнього середовища.

Відкритим джерелом називають радіонуклідні джерела (рідкі, газоподібні, пилоподібні), при звичайному використанні яких можливе потрапляння РР до навколишнього середовища. Основною фізичною величиною, що визначає ступінь радіаційного впливу, є доза опромінення.

Під дозою опромінення розуміють енергію випромінювання, поглинену в одиниці маси речовини за певний час дії випромінювання.

У будь-якому виробничому процесі, де використовуються ДІВ, може статися надзвичайна ситуація (НС).

НС – це порушення нормальних умов життя та діяльності людей на об'єкті або території, що спричинено аварією, катастрофою, стихійним лихом або іншою небезпечною подією, яка призвела або може призвести до неможливості проживання населення на території чи знаходження на об'єкті, здійснення там господарської діяльності, загибелі людей та/або значних матеріальних втрат.

Усі надзвичайні ситуації, які можуть виникати в процесі поводження з ДІВ, доцільно поділити на дві групи:

- радіаційний інцидент, який, на думку МАГАТЕ, являє собою будь-яку ненавмисну подію, в тому числі помилки під час експлуатації, відмови устаткування та інші несправності, або несанкціоновані дії, реальні або потенційні наслідки яких не можуть ігноруватись з точки зору захисту або безпеки. Іншими словами, це втрата регулюючого контролю за ДІВ.

– радіаційна аварія - це подія, внаслідок якої втрачено контроль за ДІВ або ядерною установкою, і яка призводить або може призвести до радіаційного впливу на людей та навколишнє природне середовище, що перевищує допустимі межі, встановлені нормами, санітарними правилами та стандартами безпеки. Розрізняють осередок радіаційної аварії і зону радіоактивного забруднення.

Усі РА поділяються на дві великі групи:

– перша група – аварії, що не супроводжуються радіоактивним забрудненням виробничих приміщень, промайданчика об'єкту та навколишнього середовища;

– друга група – аварії, внаслідок яких відбувається радіоактивне забруднення виробничих приміщень, промайданчика об'єкта та навколишнього середовища.

За масштабами, тобто розміром території та можливістю опромінення персоналу та населення, РА поділяються на два класи: промислові і комунальні.

При промислових РА радіоактивне забруднення не виходить за межі промислових приміщень і території промайданчика. Аварійного опромінення може зазнавати тільки персонал підприємства.

До промислових РА з радіонуклідним джерелом належать випадки, при яких відбуваються:

– крадіжка або втрата джерела випромінювання або захисного блоку з джерелом, якщо джерело не потрапило за межі підприємства;

– випадіння джерела із захисного блоку або падіння блоку з джерелом із місця кріплення;

– розгерметизація джерела, якщо вчасно прийняті заходи по недопущенню розповсюдження радіоактивного забруднення за межі виробничого приміщення;

- руйнування або погіршення якості радіаційного захисту захисного блоку з джерелом;
- виявлення не облікованого раніше джерела;
- опромінення персоналу дозою, що перевищує діючі нормативи, внаслідок виникнення критичної події;
- радіоактивне забруднення устаткування та території підприємства.
- руйнування радіонуклідного джерела при його транспортуванні.

Будь-які аварії, наслідки яких розповсюдилися за межі підприємства і обумовили підвищене опромінення населення, вважаються комунальними аваріями.

Заходи, що спрямовані на попередження промислових РА, є комплексом технічних та організаційних умов, яких необхідно дотримуватись при експлуатації ДІВ на підприємстві. Запобіганню РА сприяє правильне розміщення радіологічного об'єкту відносно населеного пункту, а також приміщень, де працюють з ДІВ відносно інших приміщень, безпосередньо не пов'язаних з використанням ДІВ. Важливим моментом є дотримання умов експлуатації джерел, передбачених технічними умовами - температури, вологості, заповишеності, механічного та хімічного впливу та ін. Необхідно дотримуватися встановлених термінів експлуатації ДІВ та термінів їх заміни згідно з паспортом на дане джерело.

Всі роботи, пов'язані з виготовленням, застосуванням, транспортуванням і похованням РР, повинні проводитись при наявності санітарного паспорту і ліцензії та у відповідності з інструкцією з радіаційної безпеки, узгодженої з територіальним органом Держсанепідслужби. При роботах з РР, там де це необхідно, повинні використовуватися засоби індивідуального захисту персоналу згідно вимог охорони праці [41].

Ті радіоактивні джерела, які не перебувають або ніколи не були під регулюючим контролем, або були залишені без нагляду, загублені, поміщені в

неналежне місце, передані без належного державного дозволу або викрадені називаються «покинутими джерелами».

Проблема «покинутих джерел» полягає в їх потенційній небезпеці для здоров'я населення та складності їх виявлення. Потрапляння «покинутого джерела» до рук пересічного громадянина може призвести до катастрофічних наслідків.

Беручи до уваги вищезгадану інформацію, закономірним є необхідність попередження появи «покинутих джерел». Одними з основних заходів попередження є: забезпечення дозвільного принципу використання джерел (тобто використовувати радіоактивне джерело можуть тільки особи або підприємства, які довели державі, що можуть забезпечити безпеку та захищеність джерела), облік та контроль радіоактивних джерел.

Відповідно, в Україні використання ядерних установок та джерел іонізуючого випромінювання в Україні ґрунтується на дозвільному принципі. Державна інспекція ядерного регулювання України уповноважена видавати дозвіл на кожен окремий вид діяльності, що пов'язаний з використанням радіоактивного матеріалу, зокрема – ліцензію на використання джерела іонізуючого випромінювання (ДІВ).

Іншим методом попередження є протидія потенційній загрозі, що існуючі ДІВ можуть перейти у розряд «покинутих». Джерела, які можуть стати покинутими визначають як вразливі. Регулюючий орган України обмежує терміни зберігання таких джерел та вимагає їх передачі на спеціалізовані підприємства, де забезпечено високий рівень безпеки та захищеності таких відпрацьованих джерел.

Особливу небезпеку представляють ті джерела, які належали підприємствам, що стали банкрутами або фінансово не стабільні, таким чином не в змозі самостійно вирішити питання передачі таких джерел на спеціалізовані підприємства. Для вирішення даної проблеми існують міжнародні програми з американськими та німецькими партнерами.

Але є певна кількість джерел, що знаходиться поза межами контролю. Одна з причин подібної ситуації полягає у відсутності необхідності контролю певних джерел іонізуючого випромінювання в минулому та існування у минулому менш жорстких вимоги щодо контролю над певними джерелами. На сьогодні, даний підхід був докорінно змінений, і ті джерела, що знаходилися поза межами контролю, тепер є об'єктами обов'язкових реєстрації/ліцензування. Інша причина – втрата контролю над певними ДІВ по наступним причинам: джерело було загублено; джерело було вкрадено; джерело було розміщене не у відповідному місці. Такі джерела необхідно виявляти (шукати).

Питання пошуку «покинутих» ДІВ в Україні займає важливе місце. В Україні існує два методи пошуку вже «покинутих» джерел: адміністративний та фізичний. В даному контексті адміністративний метод відіграє важливу роль. Його застосування передбачає пошук інформації щодо «покинутих» джерел та проведення опитування, що дає можливість на початковій стадії створення системи по виявленню «покинутих» джерел для її ефективного функціонування зібрати необхідну інформацію.

В Україні більше уваги приділяють фізичному пошуку, тобто пошуку джерел по їх фізичним характеристикам: зовнішній вигляд та випромінювання. Даний вид пошуку займає особливе місце на шляху до вирішення проблем з «покинутими» джерелами, оскільки ДІВ, в основному, знаходяться в металевих контейнерах, тому часто опиняються на металобрухті. Більше того, існує велика вірогідність їх переплавлення, що може призвести до значного радіаційного забруднення [42].

5.3. Умови і фактори виникнення пожеж та суцільних пожеж. Заходи попередження пожеж та суцільних пожеж

Пожежа – неконтрольований процес горіння, що супроводжується знищенням матеріальних цінностей і створює небезпеку для життя людей.

Вторинними наслідками пожеж можуть бути вибухи і витоки отруйних або забруднювальних речовин у навколишнє середовище; крім того, великих збитків приміщенням і предметам, яких не торкнувся вогонь, може завдати вода, яка застосовується для гасіння пожежі.

За масштабами та інтенсивністю пожежі підрозділяються на окремі, суцільні, масові і вогняні шторми.

Окрема пожежа – пожежа, що виникла в окремому будинку чи споруді. Пересування людей і техніки забудованою територією між окремими пожежами можливе без засобів захисту від теплового впливу.

Суцільна пожежа – одночасне інтенсивне горіння переважної кількості будинків і споруд на певній ділянці забудови. Пересування людей і техніки через ділянку суцільної пожежі неможливе без засобів захисту від теплового випромінювання.

Масова пожежа – сукупність окремих і суцільних пожеж.

Вогняний шторм – особлива форма суцільної пожежі, яка швидко поширюється і характерними ознаками якої є: наявність висхідного потоку продуктів згоряння і нагрітого повітря, приплив свіжого повітря з усіх боків зі швидкістю не меншою за 50 км/год у напрямку до межі вогняного шторму.

Інтенсивність пожежі багато в чому залежить від вогнестійкості об'єктів та їх складових частин, а також від пожежної безпеки технологічних процесів виробництва в місці її виникнення [43].

Процес горіння виникає за одночасної наявності трьох основних складових: окиснювача, горючої речовини, джерела займання:

1. Горючої речовини. Тверда, рідка або газоподібна речовина, яка здатна окислюватись з виділенням тепла та випромінюванням світла.

2. Окиснювача. Кисень, хлор, фтор, сірка та інші речовини які при нагріванні або ударі мають здатність розкладатись з виділенням кисню.

3. Джерела займання. Вплив на горючу речовину та окиснювач, що може викликати загоряння. Джерела займання діляться на відкриті і приховані.

При відсутності одного з трьох факторів горіння не виникає [44].



Горіння становить необоротний процес, тому що з продуктів горіння та термічного розкладу неможливо поновити речовину, яка згоріла.

Здатність матеріалу витримувати без руйнування вплив високих температур і відкритого полум'я називається горючістю.

За горючістю всі матеріали поділяються на негорючі та горючі.

До негорючих належать так звані «мінеральні» матеріали: природне каміння, метали, бетон, скло, кераміка. Горючі – це матеріали на основі органічних, рослинних компонентів. До них належать матеріали з волокон деревини, синтетичні матеріали, нафтопродукти, пластмаси.

Деякі органічні матеріали під впливом вогню не горять відкритим полум'ям, але спікаються, плавляться, виділяючи при цьому їдкий небезпечний дим.

До явищ, що спричиняють пожежі з горінням твердих речовин, належать полум'я, теплове випромінювання, розжарені матеріали.

Коли тверда речовина зазнає впливу полум'я, її температура підвищується, що може спричинити пожежу. Ймовірність виникнення пожежі залежить від таких факторів:

- характеру твердої речовини, яка може бути горючою або негорючою;
- маси твердої речовини – невелика кількість матеріалу не здатна виділити достатню кількість теплової енергії для поширення пожежі;
- стану твердої речовини – легко запалити за допомогою сірника деревну стружку або окремі аркуші паперу, оскільки в цих матеріалів більша площа поверхні, відкритої для доступу повітря, а отже, висока швидкість окиснення, тоді як для займання колоди або щільного стосу паперу потрібне потужніше джерело займання;
- способу, в який запалюється тверда горюча речовина – якщо предмет із цієї речовини розміщений над вогнем вертикально, він загориться швидше, ніж у разі горизонтального розміщення.

Не завжди для виникнення горіння необхідно, щоб горючі речовини вступали в безпосередній контакт із полум'ям або дуже нагрітими матеріалами. Всі джерела тепла випромінюють видимі та інфрачервоні промені, тобто електромагнітні хвилі. Коли ці хвилі зустрічають перешкоду (в нашому випадку – горючу речовину), вони передають йому свою енергію, яка перетворюється на тепло. Таким чином, тіло, що опромінюється, нагрівається й за недостатнього охолодження може загорітися. Нагрітий до високої температури матеріал (наприклад, розжарений до червоного метал) може спричинити загоряння в разі стикання з горючим твердим тілом.

Тління (жевріння) визначається як процес безполуменового горіння твердого матеріалу (речовини), який виникає за умови недостатнього припливу до матеріалу, що горить, кисню й тепла, та часто супроводжується виділенням диму.

Пожежі, що розвиваються в режимі тління, пов'язані з низкою проблем. До них належать: складність виявлення на початковій стадії, прогрівання поверхонь, що огорожують осередок тління; перехід до полуменового горіння; труднощі гасіння заглиблених осередків пожежі; висока ймовірність повторних загорянь тліючих осередків.

Знизити рівень пожежної небезпеки можна шляхом застосування негорючих матеріалів, обмеження маси та об'єму горючих матеріалів і речовин, ізолюванням горючого середовища (наприклад, застосуванням ізолюваних негорючих відсіків, кабін), вогнезахистом.

До основних причин виникнення пожеж належать:

- необережне поводження з вогнем;
- порушення правил обладнання та експлуатації електроустановок;
- порушення правил обладнання та експлуатації печей;
- пустощі дітей з вогнем;
- підпали;
- несправність виробничого обладнання.

Більшість пожеж в Україні (50-60%) виникає внаслідок необережного поводження з вогнем.

Переважає більшість пожеж виникає безпосередньо з вини людини через необізнаність її з правилами та вимогами пожежної безпеки або внаслідок несвідомого, поверхового чи просто недбалого ставлення до них [45].

З метою зменшення матеріальних збитків, шкоди життю і здоров'ю громадян, інтересам суспільства і держави реалізується комплекс профілактичних заходів, спрямованих на попередження та (або) усунення чинників ризику пожежі.

Профілактичні заходи – найважливіша складова системи захисту від пожеж, спрямованої на формування у населення мотивації на забезпечення пожежної безпеки в майбутньому.

Пожежна профілактика – це комплекс попереджувальних і регламентують соціально-економічних, організаційних, технічних заходів, мета яких забезпечення стану захищеності життєво важливих інтересів особистості, суспільства і держави від внутрішніх і зовнішніх загроз.

Робота з попередження пожеж на різних об'єктах включає:

- пожежно-технічне обстеження з метою виявити справжній стан пожежобезпечної системи об'єктів в цілому і окремих їх частин представниками пожежного нагляду з подальшим врученням приписів і наказів;
- контроль своєчасного виконання призначених для об'єкта заходів;
- постійний контроль над виконанням протипожежних робіт;
- перевірку справності первинних засобів пожежогасіння;
- установку систем пожежної автоматики;
- проведення навчальних робіт з персоналом організації для визначення часу евакуації людей при пожежі.

Критерієм оцінки (самооцінки) якості роботи щодо попередження пожеж на різних об'єктах є відповідність результатів виконаних робіт нормативним документам з пожежної безпеки - національним стандартам, нормам і правилам, технічним регламентам, стандартам, інструкціям і іншим документам, що містить вимоги пожежної безпеки.

Обмеження розповсюдження та розвитку пожежі, загалом, забезпечується:

- потрібною вогнестійкістю будівель та споруд;
- використанням негорючих матеріалів для внутрішнього оздоблення приміщень;
- використанням антипіренів і вогнегасних сумішей;
- улаштуванням протипожежних відстаней між будівлями та спорудами;
- улаштуванням протипожежних перешкод;
- встановленням гранично допустимих за техніко-економічними розрахунками площ поверхів виробничих будівель та поверховості будівель і споруд, улаштуванням протипожежних відсіків та секцій;
- улаштуванням аварійного відключення та перемикання установок і комунікацій;
- використанням засобів, що запобігають або обмежують розлив і розтікання пожежонебезпечної рідини під час пожежі;
- використанням вогнеперешкоджуючих пристроїв в устаткуванні;
- локалізацією пожежі вогнегасними речовинами, автоматичними установками пожежогасіння, а також шляхом утворення розривів горючого середовища вибуховими речовинами, тощо [46].

У розділі розглянуто особливості радіаційних аварій, які можуть виникати на радіаційно небезпечних підприємствах, що у своїй виробничій діяльності використовують джерела іонізуючого випромінювання. Проаналізовано особливості робіт пов'язаних з виготовленням, зберіганням,

використанням, транспортуванням та похованням об'єктів – джерел іонізуючого випромінювання. Також розглянуто умови і фактори виникнення пожеж та суцільних пожеж. Описано комплекс профілактичних заходів, що спрямований на попередження та (або) усунення чинників виникнення пожеж та суцільних пожеж.

## РОЗДІЛ 6

### ЕКОЛОГІЯ

#### 6.1. Застосування екологічних знань у різних галузях соціально-політичного життя

Пізнаючи природні закономірності та використовуючи її потенціал у своїх цілях, людина не завжди діяла раціонально, наносила шкоду природі і собі. У зв'язку з цим стало актуальним вести мову про розвиток навколишнього середовища за новими критеріями його якості, заснованими на понятті «здоров'я» (середовища, людини, нації, міста, екосистеми) [47].

Ситуація, що склалася у загальносвітовому масштабі, потребує докорінної зміни цілого ряду пріоритетів суспільного розвитку, в тому числі й зміни політичних орієнтацій як на внутрішній арені, так і на зовнішній.

Соціоекологічна політика як соціальний феномен є частиною зовнішньої та внутрішньої політики кожної держави. Головними завданнями якої є попередження загрози глобальної екологічної катастрофи та пошук шляхів та засобів виходу людства з соціально-екологічної кризи [48].

Державна соціально-екологічна політика враховує екологічний стан території тої чи іншої держави, включно з водною акваторією і повітряним простором, наявність в її межах природних ресурсів, подальші перспективи розвитку господарства та зміни чисельності населення.

Зважаючи на реалії сьогодення, сформувались основні завдання соціально-екологічної політики на державному рівні, а саме: стати пріоритетом у загальнодержавній політиці; максимально наблизити технологічні цикли до вимог природи; використовувати повністю ресурси, які закладені у переробку, а відходи направити у природні біолого-хімічні цикли; сформувати систему моральних цінностей в українців та змінити системи освіти і виховання.

До основних напрямків соціально-екологічної політики належать [49]:

- розробка і реалізація програм раціонального природокористування, ресурсо - та енергозбереження;
- впровадження замкнених, маловідходних та екологічно чистих технологій, виробництво екологічно чистих продуктів харчування;
- інформаційність та гласність екологічних експертиз, доступність екологічної інформації для всіх громадян держави;
- формування екологічної свідомості, екологічної культури та екологічної відповідальності у всіх членів суспільства;
- міжнародне співробітництво з екологічних проблем.

Держава для реалізації соціально-екологічної політики впроваджує такі основні механізми [49]:

- виробничо-технічний потенціал засобів суспільного виробництва;
- науково-інформаційне забезпечення;
- організаційно-правові та економічні важелі управління виробничою діяльністю.

До основних інструментів соціально-екологічної політики належать:

- забезпечення державою заходів для збереження природи;
- державно-економічне регулювання і підтримка соціально-екологічної діяльності приватного сектора;
- державні заходи, спрямовані на збереження навколишнього середовища.

Україна має потужний науково-виробничий потенціал. Але Україна – молода країна, в якій політичне, економічне, інституціональне становлення державності тільки відбувається, до того ж здійснюється соціально-економічна трансформація [49, 50].

Соціальна переорієнтація економічної політики України здійснюється з огляду на пріоритети гармонійного розвитку. Це пріоритети економічної, соціальної та екологічної сфер діяльності.

#### Пріоритети економічної сфери:

- затвердження методів, що забезпечать стабілізацію економіки та її швидкого зростання;
- впровадження технологічно-наукового та інноваційного розвитку України;
- розвиток базових галузей економіки;
- розвиток аграрного сектора.

#### Пріоритети соціальної сфери:

- зростання інтелектуального потенціалу населення;
- попередження збіднення населення (шляхом реформи системи оплати праці, зниження безробіття, вдосконалення механізму нарахування державної соціальної допомоги, запровадження порядку загальнообов'язкового державного страхування;
- соціальна підтримка сім'ї, жінок, дітей та молоді;
- покращення сфери охорони здоров'я;
- зміцнення позицій середнього класу.

#### Пріоритети екологічної сфери:

- розвиток екологічної освіти в Україні;
- гарантування екологічної безпеки ядерних об'єктів і радіаційного захисту населення та довкілля;
- програма екологічного оздоровлення стану річок та басейну Дніпра, поліпшення якості питної води;
- запобігання забруднення Чорного та Азовського морів;
- формування збалансованої системи природокористування;
- збереження біологічного та ландшафтного різноманіття, розвиток природно-заповідного фонду [51].

Людство має усвідомити необхідність утвердження нової філософії життя, головним правилом якої є екологічний гуманізм.



## 6.2. Вимоги до моніторів (ВДТ) і ПЕОМ

Використання комп'ютерної техніки сильно підвищує продуктивність праці користувачів відеотерміналу (ВДТ) персональної електронно-обчислювальної машини (ПЕОМ). Проте активне впровадження в роботу ПЕОМ має і негативний характер. Тому, до ВДТ висувають наступні вимоги.

1) Різкість і чіткість зображення. В першу чергу на ці характеристики впливає роздільна здатність дисплея. Якщо роздільна здатність висока, то зображення на дисплеї буде виглядати точніше та чіткіше.

На чіткість зображення також впливає крок люмінофора (dot pitch) – мінімальна відстань між люмінофорними елементами однакового кольору в сусідніх рядках. Відстань між сусідніми точками люмінофора в якісному моніторі становить 0,25-0,41 мм. При роботі з графікою з роздільною здатністю вище 1024 x 768 ця відстань становить 0,25-0,26 мм. При роботі в домашніх чи офісних умовах, де використовується роздільна здатність 1024 x 768 або нижче, достатньо 0,27-0,28 мм.

Велику роль для чіткості зображення відіграє також якість фокусування електронних променів [52].

2) Усунення мерехтіння зображення. Людське око сприймає будь-яку зміну зображень як рухоме зображення, якщо частота зміни зображень не нижча за 20-25 Гц. Але за таких частот не достатньо для роботи з монітором на близькій відстані. Тому, оптимальним вибором частоти кадрів є 60 Гц або 75 Гц [35].

3) Яскравість та контрастність. Яскравість монітора – це характеристика середнього сприйняття яскравості зображення на екрані в контрольованих умовах зовнішнього освітлення.

На моніторах мають бути кнопки регулювання яскравості і контрасту, щоб забезпечити можливість регулювання цих параметрів від мінімальних до максимальних значень. Яскравість знака або тла має становити від 35 до

120 кд/м<sup>2</sup>, зовнішня освітленість екрана від 100 до 250 лк, контраст – від 3 : 1 до 1,5 : 1.

4) Відсутність відблисків. Відблиски погіршують розбірливість зображень і роблять екранну інформацію нерозбірливою. А різкі відблиски відволікають увагу від поставлених завдань. Інтенсивність екранних відблисків, що визначає зорову працездатність і міру прийнятності цих відблисків, залежить від співвідношення між параметрами освітлення, параметрами дисплейного екрану і взаєморозташуванням спостерігача, дисплейного екрану і джерел світла.

У сучасних моніторах, для зменшення віддзеркалення, використовують темне або тоноване скло, проводять спеціальну хімічну обробку лицьової поверхні ; застосовують циліндрові і плоскі прямокутні екрани, а також використовують захисні фільтри [52].

5) Безпечні рівні випромінювань від моніторів. Рівні електромагнітних випромінювань моніторів регламентуються нормами MPR II 1990:10 Шведського національного комітету з вимірювань та досліджень, які вважаються базовими, а також більш жорсткими нормами TCO 92 Шведської конференції профспілок (табл.6.1) [52].

Таблиця 6.1

#### Допустимі рівні випромінювань ПК

Види поля	TCO 92	MPR II
Змінне електричне поле: 5 Гц - 2 кГц 2 кГц - 400 кГц	10 В/м 1 В/м на відстані 0,3 м від центру екрана і 0,5м навколо монітора	25 В/м 2,5 В/м на відстані 0,5 м навколо монітора
Змінне магнітне поле: 5 Гц - 2 кГц 2 кГц - 400 кГц	250 нТл            200 мА/м 25 нТл             20 мА/м на відстані 0,3 м від центру екрана і 0,5м навколо монітора	250 нТл, 200 мА/м 25 нТл, 20 мА/м на відстані 0,5 м навколо монітора

## ВИСНОВКИ

У даній магістерській роботі досліджено методи та засоби опрацювання природної мови з використанням нейронних мереж з метою розпізнавання власних назв. Основні результати та висновки проведених досліджень такі:

- досліджено сучасні методи та засоби опрацювання природної мови з метою розпізнавання власних назв та встановлено їх переваги та недоліки;

- проаналізовано методи векторного представлення тексту, що використовуються для задач опрацювання природної мови з метою розпізнавання власних назв, що дало змогу обґрунтувати математичне забезпечення програмної системи для розпізнавання власних назв з використанням методів та сучасних засобів рекурентних нейронних мереж;

- на основі попередньо обґрунтованого математичного забезпечення розроблено архітектуру програмної системи для розпізнавання власних назв;

- обґрунтовано бібліотеку TensorFlow.js та API Keras, як засобів розробки програмної системи для розпізнавання власних назв, що уможливило ефективну її реалізацію;

- здійснено оцінку якості роботи програмного забезпечення за такими параметрами, як: точність, повнота та F-міра.

Проведено економічні розрахунки, які спрямовані на визначення економічної ефективності та вартості проведення дослідження.

Здійснено опис вимог з охорони праці й техніки безпеки відповідно до нормативних документів щодо: організації робочого місця, електробезпеки, шуму та вібрації, освітленості, мікроклімату та пожежної безпеки.

Розглянуто питання екології, що стосуються магістерської роботи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Konkol M. Named Entity Recognition PhD Study Report, University of West Bohemia in Pilsen, Czech Republic, 2012. 40 p.
2. Nadeau D., Sekine S. A survey of named entity recognition and classification, National Research Council Canada / New York University, 2007. 20 p.
3. Balasuriya D., Ringland N., Nothman J., Murphy T., Curran J. Named entity recognition in wikipedia. School of Information Technologies University of Sydney, Australia, 2009. 9 p.
4. Erik F. Tjong Kim Sang, Fien De Meulder Introduction to the conll-2003 shared task: language-independent named entity recognition. CNTS - Language Technology Group, University of Antwerp, 2003. 6 p.
5. Robinson S. What is Natural Language Processing? URL: <https://stackabuse.com/what-is-natural-language-processing/> (Last accessed: 09.09.2019).
6. NLP (Natural Language Processing) Tutorial: What is, History, Example. URL: <https://www.guru99.com/nlp-tutorial.html> (Last accessed: 13.09.2019).
7. Sharma M. NLP CORE. URL: <https://medium.com/hackernoon/nlp-core-4c16f379ced0> (Last accessed: 14.09.2019).
8. Panchal A. Text Summarization using NLTK: TF-IDF Algorithm. URL: <https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3> (Last accessed: 14.09.2019).
9. Kriesel D. A Brief Introduction to Neural Networks. 2009. 286 p.
10. A Complete Guide To Artificial Neural Network In Machine Learning. URL: <https://www.softwaretestinghelp.com/artificial-neural-network/> (Last accessed: 16.09.2019).

11. Woodford C. How neural networks work. URL: <https://www.explainthatstuff.com/introduction-to-neural-networks.html> (Last accessed: 16.09.2019).
12. Sharma A. Understanding Activation Functions in Neural Networks. URL: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0> (Last accessed: 20.09.2019).
13. Allam A., Nagy M., Thoma G., Krauthammer M. Neural networks versus Logistic regression for 30 days all-cause readmission prediction. URL: <https://arxiv.org/pdf/1812.09549.pdf> (Last accessed: 21.09.2019).
14. Mohammadi M., Mundra R., Socher R. CS 224D: Deep Learning for NLP. URL: [https://cs224d.stanford.edu/lecture\\_notes/notes4.pdf](https://cs224d.stanford.edu/lecture_notes/notes4.pdf) (Last accessed: 25.09.2019).
15. Lakhey M. Word2vec Made Easy. URL: <https://towardsdatascience.com/word2vec-made-easy-139a31a4b8ae> (Last accessed: 25.09.2019).
16. Karani D. Introduction to Word Embedding and Word2Vec. URL: <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa> (Last accessed: 27.09.2019).
17. Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation. Computer Science Department, Stanford University, Stanford, 2014. 12 p.
18. How is GloVe different from word2vec? URL: <https://www.quora.com/How-is-GloVe-different-from-word2vec> (Last accessed: 28.09.2019).
19. What are the advantages and disadvantages of Word2vec and GloVe? URL: <https://www.quora.com/What-are-the-advantages-and-disadvantages-of-Word2vec-and-GloVe> (Last accessed: 28.09.2019).
20. Jiao W. How is GloVe different from word2vec? URL: <https://www.quora.com/How-is-GloVe-different-from-word2vec> (Last accessed: 29.09.2019).

21. Борисов Е. Автоматизированная обработка текстов на естественном языке, с использованием инструментов языка Python. URL: <http://mechanoid.kiev.ua/ml-text-proc.html> (дата звернення: 30.09.2019).
22. Борисов Е. методах обучения многослойных нейронных сетей прямого распространения. URL: <http://mechanoid.kiev.ua/neural-net-backprop.html> (дата звернення: 30.09.2019).
23. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed Representations of Words and Phrases and their Compositionality. URL: <https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf> (Last accessed: 01.10.2019).
24. Minnaar A. Word2Vec Tutorial Part I: The Skip-Gram Model. URL: [http://mccormickml.com/assets/word2vec/Alex\\_Minnaar\\_Word2Vec\\_Tutorial\\_Part\\_I\\_The\\_Skip-Gram\\_Model.pdf](http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf) (Last accessed: 03.10.2019).
25. Minnaar A. Word2Vec Tutorial Part II: The Continuous Bag-of-Words Model. URL: [http://mccormickml.com/assets/word2vec/Alex\\_Minnaar\\_Word2Vec\\_Tutorial\\_Part\\_II\\_The\\_Continuous\\_Bag-of-Words\\_Model.pdf](http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_II_The_Continuous_Bag-of-Words_Model.pdf) (Last accessed: 03.10.2019).
26. Rong X. Word2vec Parameter Learning Explained. URL: <http://arxiv.org/abs/1411.2738> (Last accessed: 06.10.2019).
27. Cloud Tensor Processing Units (TPUs). URL: <https://cloud.google.com/tpu/docs/tpus> (Last accessed: 06.10.2019).
28. Sato K., Young C., Patterson D. An in-depth look at Google's first Tensor Processing Unit (TPU). URL: <https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu> (Last accessed: 09.10.2019).
29. What is TensorFlow? Introduction, Architecture & Example. URL: <https://www.guru99.com/what-is-tensorflow.html> (Last accessed: 11.10.2019).
30. Get to know TensorFlow.js in 7 minutes. URL: <https://www.freecodecamp.org/news/get-to-know-tensorflow-js-in-7-minutes-afcd0dfd3d2f/> (Last accessed: 12.10.2019).

31. Gubler B. TensorFlow.js: An intro and analysis with use cases. URL: <https://blog.logrocket.com/tensorflow-js-an-intro-and-analysis-with-use-cases-8e1f9a973183/> (Last accessed: 18.10.2019).
32. TensorFlow.js: машинное обучение на JavaScript с доставкой в браузер. URL: <https://proglib.io/p/tensorflow-js/> (дата звернення: 20.10.2019).
33. Keras Documentation. URL: <https://keras.io> (Last accessed: 25.12.2019).
34. Наказ Міністерства соціальної політики України «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» № 207, 2018. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18> (дата звернення: 07.11.2019).
35. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98, 1998. URL: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення: 07.11.2019).
36. Санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99, 1999. URL: <https://zakon.rada.gov.ua/rada/show/va037282-99> (дата звернення: 07.11.2019).
37. Закон України «Про охорону праці» № 49, 1992. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 07.11.2019).
38. Наказ Міністерства праці та соціальної політики України «Про затвердження Правил безпечної експлуатації електроустановок споживачів» № 93/2533, 1998. URL: <https://zakon.rada.gov.ua/rada/show/z0093-98> (дата звернення: 09.11.2019).
39. Наказ Міністерства внутрішніх справ України «Про затвердження Правил пожежної безпеки в Україні» № 1417, 2014. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15> (дата звернення: 09.11.2019).
40. Наказ Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України «Про затвердження

ДБН В.2.5-56:2014 Системи протипожежного захисту» № 312, 2014. URL: <https://zakon.rada.gov.ua/rada/show/v0312858-14> (дата звернення: 09.11.2019).

41. Мурашко В. О., Костенецький М. І., Рушак Л. В. Промислові радіаційні аварії з джерелами іонізуючого випромінювання, запобігання та порядок їх розслідування. Київ, 2013. 82 с.

42. Державний комітет ядерного регулювання України. Проект від 01.03.2008 р. Консультації щодо підвищення безпеки джерел іонізуючого випромінювання в Україні. Київ, 2008. 24 с.

43. Васійчук В.О., Гончарук В.Є., Качан С.І., Мохняк С.М. Основи цивільного захисту: Навч. Посібник. Львів: Видавництво Національного університету «Львівська політехніка», 2010. 417 с.

44. Гурін О. А. Індивідуальне завдання з оцінки пожежної обстановки. Д.: Національний гірничий університет, 2013. 10 с.

45. Абрамова С. В. Безпека життєдіяльності. М.: видавництво Юрайт, 2019. 399 с.

46. Основні заходи пожежної профілактики на галузевих об'єктах. Київ, 2014, 17 с. URL: <https://opcb.kpi.ua/wp-content/uploads/2014/09/Лекція-9.pdf> (дата звернення: 27.11.2019).

47. Яблоков А. В. Необходимость перемен в мировой экологической политике // Екологічний вісник / А. В. Яблоков, – 2004. – № 1(21). – 19-22 с.

48. Назарук М. М. Основи екології та соціоекології. Львів, 2000. 256 с.

49. Салтовський О. І. Основи соціальної екології. Київ, 1997. 168 с.

50. Білявський Г.О, Бутченко Л.І., Навроцький В.М. Основи екології: Теорія і практикум: Навч. Посібник. Київ, 2002. 352 с.

51. Маринець О. М., Мозговий А. М., Романовська Н. Г. Соціальна екологія: теоретичні аспекти: Методичні вказівки. Миколаїв, 2007. 28 с.

52. Касьянов М. А., Васильчук М. В. Охорона праці користувачів ПК. Луганськ, видавництво СНУ ім. В. Даля. 2009. 101 с.



ДОДАТОК А  
ТЕЗИ КОНФЕРЕНЦІЙ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний технічний університет імені Івана Пулюя (Україна)  
Національна академія наук України  
Університет імені П'єра і Марії Кюрі (Франція)  
Маріборський університет (Словенія)  
Технічний університет у Кошице (Словаччина)  
Вільнюський технічний університет ім. Гедимінаса (Литва)  
Шауляйська державна колегія (Литва)  
Жешувський політехнічний університет ім. Лукасевича (Польща)  
Білоруський національний технічний університет (Республіка Білорусь)  
Міжнародний університет цивільної авіації (Марокко)  
Національний університет біоресурсів і природокористування України (Україна)  
Наукове товариство ім. Шевченка  
ГО «Асоціація випускників Тернопільського національного технічного університету імені Івана Пулюя»

**АКТУАЛЬНІ ЗАДАЧІ  
СУЧАСНИХ ТЕХНОЛОГІЙ**

**Збірник**

тез доповідей

**Том II**

**VIII Міжнародної науково-технічної  
конференції молодих учених та студентів**

27-28 листопада 2019 року



**УКРАЇНА  
ТЕРНОПІЛЬ – 2019**

38. **Д.Є. Костенко, В.В. Гавриш, В.І. Фрінцко, В.В. Саснко**  
ЗАСТОСУВАННЯ СПЕЦІАЛІЗОВАНИХ ТЕХНОЛОГІЙ ДЛЯ  
БЕЗПЕЧНОГО ПОШУКУ ТА ОТРИМАННЯ ІНФОРМАЦІЇ 49
39. **D.Y. Kostenko, V.V. Gavrysh, V.I. Frintsko, V.V. Sayenko**  
USING THE SPECIALIZED TECHNOLOGIES FOR SAFE SEARCH AND  
OBTAINING INFORMATION 49
40. **Ю.Р. Криль, В.І. Кашеба, В.А. Нестеренко**  
НЕЙРОМЕРЕЖЕВІ МЕТОДИ ВИЯВЛЕННЯ ТА АНАЛІЗУ ЗОБРАЖЕНЬ 50
41. **В.В.Крючков, М.О.Стешик**  
АНАЛІЗ СТРУКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ  
ДИСПЕТЧЕРИЗАЦІЇ ТЕПЛОПОСТАЧАННЯ 51
42. **А.О. Кукуруза, Д.П. Павлюк, В.В. Сенник, Б.Ю. Шутко**  
МЕТОДИ ОПТИМІЗАЦІЇ ПРОГРАМИ 53
43. **Т.П. Лавренюк, Р.Б. Трезбач**  
ДОСЛІДЖЕННЯ АВТОМАТИЗОВАНОГО УЛЬТРАЗВУКОВОГО  
ПРИСТРОЮ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ТА ЖИРНОСТІ  
МОЛОКА 55
44. **О.Б. Лішук, Є.В. Тиш**  
МЕТОДИ ТА ЗАСОБИ РЕЗЕРВУВАННЯ ТА АГРЕГАЦІЇ КАНАЛІВ  
КОМП'ЮТЕРНИХ МЕРЕЖ 57
45. **Н.В. Луб'янецький, Г.П. Хизмич, Ю.А.Умзар**  
КЕРОВАНИЙ ХВИЛЕВІДНИЙ ФАЗОПОВЕРТАЧ НВЧ ДІАПАЗОНУ 58
46. **С.А. Лупенко, Б.А. Яворський**  
АРХІТЕКТУРА РОЗПОДІЛЕНОЇ КОМП'ЮТЕРНОЇ СИСТЕМИ ЗБОРУ ТА  
УПРАВЛІННЯ ДАНИМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ 59
47. **С.А. Лупенко, В.О. Васьков**  
АНАЛІЗ МЕТОДІВ ДЛЯ ЗАДАЧ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ 60
48. **А.М. Лушків, Н.М. Попович, Х.Б. Юркевич**  
БІБЛІОТЕКИ ОБРОБКИ ПРИРОДНИХ МОВ У ПРЕДМЕТНІЙ ОБЛАСТІ  
ВЕЛИКИХ ДАНИХ 62
49. **А.М. Лушків, І.А. Форись**  
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ  
СИСТЕМ КЕРУВАННЯ ТРАНСПОРТНИМ ЗАСОВОМ 64
50. **Ю.М. Миколок, І.В. Бойко**  
РОЗРОБКА ІНФОРМАЦІЙНО-ЕЛЕКТРОННОЇ СИСТЕМИ ДЛЯ  
КОНТРОЛЮ ВІДВІДУВАНОСТІ ТА УСПІШНОСТІ СТУДЕНТІВ 65

УДК 004.021

С.А. Лупенко, докт. техн. наук, професор, В.О. Васьков

Тернопільський національний технічний університет імені Івана Пулюя, Україна

### АНАЛІЗ МЕТОДІВ ДЛЯ ЗАДАЧ ОПРАЦЮВАННЯ ПРИРОДНОЇ МОВИ

S.A. Lupenko, Dr., Prof., V.O. Vaskov

### ANALYSIS OF METHODS FOR NATURAL LANGUAGE PROCESSING TASKS

Застосування методів машинного навчання до текстів на природній мові може дати багато цікавих і корисних результатів, наприклад – автоматичне сортування текстів за темами (завдання класифікації), пошук схожих текстів (задача кластеризації), автоматичний переклад та ін. Для того, щоб застосувати математичні методи до текстів необхідно певним чином формалізувати дані. У разі класифікатора текстів формалізація виконується з допомогою частотного аналізу (g), у цьому випадку кожному тексту T ставимо у відповідність точку в просторі ознак  $X \subset R^n$  [1].

Цей метод досить добре працює для текстів середнього розміру, однак для коротких повідомлень частотна характеристика може виявитися неінформативною. Не підходить цей метод і для завдань машинного перекладу, де цікавою є не загальна характеристика тексту, а кожне слово окремо і послідовності зі слів. Таким чином, виникає необхідність побудувати ефективний метод кодування окремих слів.

Слова можна кодувати різними методами. Напевно найпростіший спосіб це їх пронумерувати, тобто складаємо повний словник з тексту, збираємо всі можливі форми слів, використані в тексті, і нумеруємо всі ці слова. Але такий спосіб кодування не несе ніякого смислового навантаження, тобто за кодом не можна сказати наскільки близькими за змістом є слова.

В 2013 році команда дослідників компанії Google розробила метод побудови «осмисленого» простору для слів – Word2Vec [2].

Word2Vec ґрунтується на тому, що слова, які мають подібний контекст, мають також подібні смислові значення. В основі технології Word2Vec лежить представлення слів у вигляді векторів заданої розмірності, розташовуючи схожі слова близько один до одного [3].

Для створення бази відповідей «слово – вектор», алгоритм спочатку переглядає весь виданий йому текст, формулюючи, таким чином, «словник», який в наступних етапах роботи алгоритму, буде використаний для визначення відповідних векторів.

Результатом роботи Word2Vec є набір векторів (матриця) – кодів слів, яку отримуємо в результаті навчання певної нейронної мережі на деякому тексті (впорядкованій множині слів), й перше, що потрібно зробити, це підготувати набір навчальних даних.

Для навчання мережі Word2Vec застосовують два основних методи: CBOW (Continuous Bag of Words) і Skip-gram (рис. 1).

CBOW – модельна архітектура, яка передбачає поточне слово, виходячи з навколишнього його контексту. Архітектура типу Skip-gram діє інакше: вона використовує поточне слово, щоб передбачати оточуючі його слова [4].

Реалізація Word2Vec складається з трьох частин:

- кодування – на вхід надходить відфільтрований текст, на виході отримуємо кодовані набори слів тексту P і контекстів Q;
- навчання мережі – на вхід надходить навчальний набір P, Q, на виході

маємо матрицю уявлень  $V_i$ ;

– тест результату – на вхід надходить словник і матриця уявлень  $V_i$ , на виході отримуємо випадково вибрані слова зі словника і найбільш близькі до них слова Word2Vec.

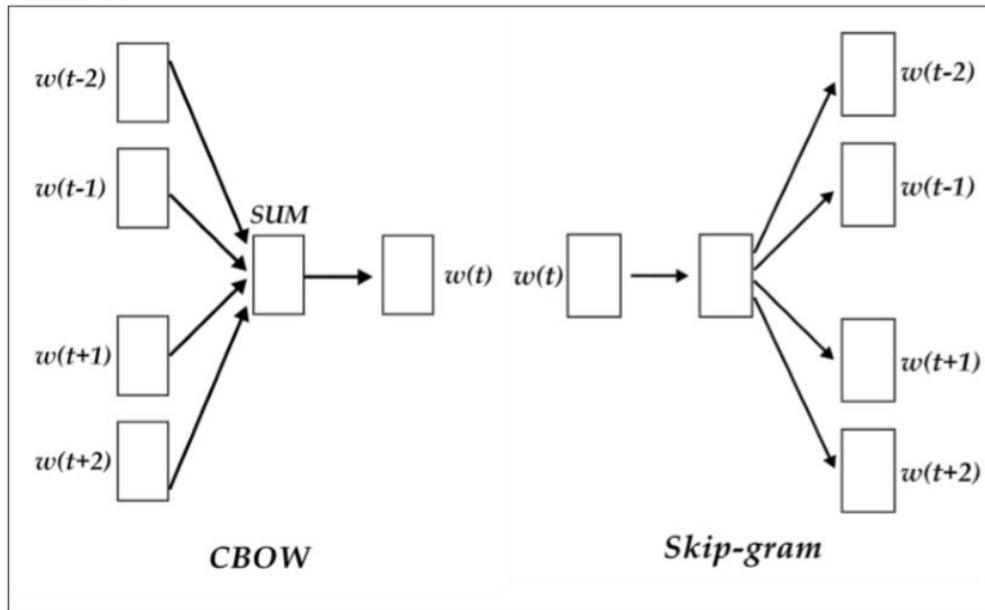


Рисунок 1. Структура моделей CBOW та Skip-gram

#### Література

1. Rong X. word2vec Parameter Learning Explained [Електронний ресурс] / Xin Rong. – 2016. – Режим доступу до ресурсу: [https://www.researchgate.net/publication/268226652\\_word2vec\\_Parameter\\_Learning\\_Explained](https://www.researchgate.net/publication/268226652_word2vec_Parameter_Learning_Explained).
2. Lakhey M. Word2vec Made Easy [Електронний ресурс] / Munesh Lakhey. – 2016. – Режи доступу до ресурсу: <https://towardsdatascience.com/word2vec-made-easy-139a31a4b8ae>.
3. Nayak M. An Intuitive Introduction of Word2Vec by Building a Word2Vec From Scratch [Електронний ресурс] / Manish Nayak. – 2019. – Режим доступу до ресурсу: <https://medium.com/towards-artificial-intelligence/an-intuitive-introduction-of-word2vec-by-building-a-word2vec-from-scratch-a1647e1c266c>.
4. Karani D. Introduction to Word Embedding and Word2Vec [Електронний ресурс] / Dhruvil Karani. – 2018. – Режим доступу до ресурсу: <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa>.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ**

**МАТЕРІАЛИ**

**VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



**11–12 грудня 2019 року**

**ТЕРНОПІЛЬ  
2019**

## СЕКЦІЯ 2. ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

<b>Л. Андріюк, С. Уніят, В. Хвостівський</b> ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОБРОБКИ ЕЛЕКТРОНЕЙРОМІОСИГНАЛУ	19
<b>Д. Антонюк, Н. Бабій, Б. Годованець, В. Марусяк</b> СУЧАСНЕ ВИЗНАЧЕННЯ «РОЗУМНОГО МІСТА»	20
<b>М. Баранський</b> РОЗРОБКА СИСТЕМИ МАШИННОГО ПЕРЕКЛАДУ НА ОСНОВІ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ ВЕКТОРА МЕТРИК ЯКОСТІ	21
<b>Я. Бачинський</b> АНАЛІЗ СПОСОБІВ ЗВ'ЯЗКУ РОБОТІВ НА БАЗІ МІКРОКОНТРОЛЛЕ- РІВ ARDUINO	22
<b>А. Бельма, О. Кареліна</b> ВИЯВЛЕННЯ ЗАГРОЗ ДЛЯ ІОТ-ПРИСТРОЇВ ЗАСОБАМИ HONEY- POTS	23
<b>Ю. Брезмен, Н. Кунанець</b> ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА ДІАГНОСТУВАННЯ ПСИХІЧНОГО СТАНУ ЛЮДИНИ	24
<b>Р. Буцій</b> СИСТЕМА ЗБОРУ ТА ВІЗУАЛІЗАЦІЇ ДАНИХ МІКРОКЛІМАТУ РО- ЗУМНОГО БУДИНКУ З ВИКОРИСТАННЯМ LoRa MESH- ТЕХНОЛОГІЙ	25
<b>А. Вапляк, П. Пронів, В. Дозорський</b> ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ БІОМЕТРИЧНОЇ АВТЕН- ТИФІКАЦІЇ ЛЮДИНИ ЗА ВІДБИТКАМИ ПАЛЬЦІВ	26
<b>В. Васьков, С. Лупенко</b> ПЕРЕВАГИ ВИКОРИСТАННЯ ТЕНЗОРНОГО ПРОЦЕСОРА ДЛЯ РО- БОТИ З НЕЙРОННИМИ МЕРЕЖАМИ	27
<b>В. Веселовська, Л. Дмитроца</b> СТАТИСТИЧНИЙ БАГАТОМОВНИЙ ПЕРЕКЛАД ЗАПИТІВ ПРИ ІН- ФОРМАЦІЙНОМУ ПОШУКУ	28
<b>В. Вівчарик</b> ОСОБЛИВОСТІ МЕТОДІВ АНАЛІЗУ ТЕКСТІВ ДЛЯ ІДЕНТИФІКАЦІЇ АВТОРСТВА ДОКУМЕНТУ	29
<b>Р. Волянський</b> ЗАСОБИ ПЕРЕДАВАННЯ ІНФОРМАЦІЇ В СИСТЕМІ «РОЗУМНИЙ ПІШОХІДНИЙ ПЕРЕХІД»	30
<b>Р. Гаврилів, Н. Кунанець</b> АВТОМАТИЗАЦІЯ СТОМАТОЛОГІЧНОЇ КЛІНІКИ	31
<b>Р. Галаз, Н. Кунанець</b> ІНТЕЛЕКТУАЛЬНА СИСТЕМА ВИЗНАЧЕННЯ ГАРМОНІЇ МУЗИЧ- НОГО ТВОРУ	32
<b>О. Голояд, А. Шурхай, І. Дедів</b> ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ІМПУЛЬСНИХ ПЕРЕТВОРЮВАЧІВ ПОСТІЙНОГО СТРУМУ	33
<b>М. Горалечко, С. Метохір</b> СТВОРЕННЯ МНОЖИНИ АЛЬТЕРНАТИВНИХ АРХІТЕКТУР ПРО- ГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
<b>Ю. Гулка</b> КРИТЕРІЇ ПОРІВНЯННЯ СТЕГANOГРАФІЧНИХ МЕТОДІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЗОБРАЖЕННЯХ	36

УДК 004.31

**В. Васьков, С. Лупенко**

(Тернопільський національний технічний університет імені Івана Пулюя)

### **ПЕРЕВАГИ ВИКОРИСТАННЯ ТЕНЗОРНОГО ПРОЦЕСОРА ДЛЯ РОБОТИ З НЕЙРОННИМИ МЕРЕЖАМИ**

UDC 004.31

**V. Vaskov, S. Lupenko**

(Ternopil Ivan Puluj National Technical University, Ukraine)

### **BENEFITS OF USING TENSOR PROCESSOR TO WORK WITH NEURAL NETWORKS**

Тензорний процесор (TPU) є інтегральною схемою специфічного застосування (ASIC), що призначений для прискорення розрахунків штучного інтелекту, й був розроблений компанією Google для машинного навчання нейронних мереж [1].

Свою назву процесори отримали від бібліотеки програмного забезпечення TensorFlow. Основне призначення TPU полягає в прискоренні алгоритмів штучного інтелекту.

TPU ASIC побудований на 28 нм процесі, працює на частоті 700 МГц і споживає 40 Вт під час роботи. TPU підключається до слоту через шину PCIe Gen3 x16, яка забезпечує ефективну пропускну здатність 12.5 GB/s.

В середньому тензорний процесор в 15-30 разів швидше здійснює обчислення, в порівнянні із традиційними серверними CPU і GPU. Продуктивність у розрахунку на ватт у TPU у 25-80 разів вища, ніж у центрального і графічного чіпів [2].

Програмованість була ще однією важливою метою дизайну для TPU. TPU не призначений для запуску тільки одного типу моделі нейронної мережі. Замість цього він розроблений таким чином, щоб бути достатньо гнучким для прискорення обчислень, необхідних для запуску багатьох різних моделей нейронних мереж.

Більшість сучасних CPU побудовані з використанням архітектури Reduced Instruction Set Computer (RISC). У RISC основна увага приділяється визначенню простих інструкцій (наприклад, завантаження, зберігання, додавання та множення), які зазвичай використовуються більшістю додатків, а потім виконують ці інструкції якомога швидше. Архітектура Complex Instruction Set Computer (CISC) була обрана як основа набору інструкцій TPU. Архітектура CISC фокусується на реалізації високорівневих інструкцій, які виконують більш складні завдання (такі як обчислення багаторазового множення і додавання) з кожною інструкцією.

TPU включає наступні обчислювальні ресурси:

–матричний множник (MXU): 65,536 8-бітових одиниць множення та додавання для операцій матриці;

–єдиний буфер (UB): 24 МБ SRAM, які працюють як регістри;

–активаційний блок (AU): функції активації [2].

Дизайн TPU є строго мінімальним і детермінованим, оскільки він повинен виконувати лише одне завдання за один раз: прогнозування нейронної мережі.

З TPU, можемо легко оцінити, скільки часу потрібно для запуску нейронної мережі та прогнозування. Це дозволяє працювати з максимальною пропускнуною спроможністю даного чіпа [1].

#### **Література**

1. Cloud Tensor Processing Units (TPUs) [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://cloud.google.com/tpu/docs/tpus>.
2. Sato K. An in-depth look at Google's first Tensor Processing Unit (TPU) [Електронний ресурс] / K. Sato, C. Young, D. Patterson // Google Cloud Blog. – 2017. – Режим доступу до ресурсу: <https://cloud.google.com/blog/products/gcp/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu>.