Кафедра \_\_\_\_\_\_\_\_\_ електронних обчислювальних машин\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до бакалаврської кваліфікаційної роботи на тему**

\_\_\_\_\_Програмна платформа створення штучних нейронних мереж\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_A software platform for creating artificial neural networks\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент групи \_\_\_\_\_\_\_\_\_КІ-406 Ярмола Ю. Ю.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр, прізвище та ініціали)

|  |  |  |  |
| --- | --- | --- | --- |
| **Керівник роботи** |  |  | ( Олексів М. В. ) |
| **Консультанти** |  |  | ( ) |
|  |  |  | ( ) |
|  |  |  | ( ) |
|  |  |  | ( ) |
|  |  |  | ( ) |
|  |  |  | ( ) |

Завідувач кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ІКТА\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра\_електронних обчислювальних машин\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність\_123 «Комп’ютерна інженерія»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«ЗАТВЕРДЖУЮ»

Завідувач кафедри ЕОМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 р.

**ЗАВДАННЯ**

**на кваліфікаційну роботу (проект) студента групи \_КІ-406\_\_ОР \_\_\_бакалавр**

\_\_\_\_\_\_\_\_\_\_\_\_Ярмола Юрій Юрійович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*прізвище,* *ім’я,* *по батькові*)

1. Тема роботи (проекту)

\_\_\_\_ Програмна платформа створення штучних нейронних мереж \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_A software platform for creating artificial neural networks \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(у разі виконання комплексної роботи в дужках вказується “комплексна робота (проект)”)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

затверджена наказом по університету від «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 р. № \_\_\_\_\_\_\_\_\_\_\_\_

1. Термін подання студентом закінченої роботи (проекту) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Вихідні дані для роботи (проекту)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Зміст розрахунково-пояснювальної записки (перелік питань, які належить розробити)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Перелік графічного матеріалу

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Перелік програмних продуктів, які належить використати в процесі розроблення роботи (проекту) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. Консультування роботи (проекту), із зазначенням розділів роботи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Розділ | Консультант | Завдання видав | | Завдання прийняв | |
| підпис | дата | підпис | дата |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

8. Дата, коли видано завдання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Керівник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
|  |  |  | (*підпис*) |  |
|  | Завдання прийняв до виконання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
|  |  |  | (*підпис*) |  |
|  | КАЛЕНДАРНИЙ ПЛАН | | |  |
|  |  |  |  |  |
| №  з/п | Назва етапів роботи (проекту) | | Термін виконання  етапів роботи (проекту) | Примітка |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*підпис*)

Керівник\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(*підпис*)

# Розширена анотація

# Зміст

[Вступ 7](#_Toc194868415)

[Розділ 1. АНАЛІТИЧНИЙ ОГЛЯД ТА ОБГРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ 8](#_Toc194868416)

[Розділ 2. ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ 9](#_Toc194868417)

[Розділ 3. РЕАЛІЗАЦІЯ АЛГОРИТМУ РОЗВ’ЯЗАННЯ ЗАДАЧІ 10](#_Toc194868418)

[Економічна частина 11](#_Toc194868419)

[Висновки 12](#_Toc194868420)

[Список літератури 13](#_Toc194868421)

[Додатки 14](#_Toc194868422)

# Вступ

# Розділ 1. АНАЛІТИЧНИЙ ОГЛЯД ТА ОБГРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ

За останні кілька років у світі різко зріс інтерес до штучних нейронних мереж (ШНМ). Це можна пояснити виходом зрозумілим для більшості людей нейронної мережі у вигляді чату, що відповідає на будь-які запитання, генерує та розпізнає зображення, виконує як прості так і складні задачі. Цей інтерес також спровокував розвиток і інших спеціалізованих нейронних мереж, які здатні виконувати добре одну вузьку задачу, наприклад генерація або розпізнавання зображень. Проте у всіх цих ШНМ у основі роботи лежить принцип імітації нейрону нервової клітини людського мозку, який має різні реалізації, продуктивність та інші параметри які впливають на навчання та роботу готової мережі.

## 1.1 Аналіз сучасних підходів до створення штучних нейронних мереж

Основні принципи створення штучних нейронних мереж базуються на моделюванні подібних на біологічні нейрони мереж, що дають змогу обробляти великі обсяги даних та знаходити складні залежності у них. Штучні нейронні мережі мають різні архітектури для моделювання нейронів, ось найпопулярніші з них:

* **Багатошарові перцептрони**

Основний блок для багатьох інших нейронних мереж. Використовується для бінарної класифікації.

* **Згорткові нейронні мережі**

Ефективні для обробки зображень і використовують згорткові шари для виявлення патернів у зображеннях.

* **Рекурентні нейронні мережі**

Призначені для обробки послідовних даних, таких як текст або часові ряди.

* **Мережі асоціативної пам’яті**

Використовуються для розв'язання завдань асоціативної пам'яті і оптимізації.

* **Трансформери**

Використовуються для обробки послідовних даних та роботи з прикладами з різних контекстів.

Методи навчання включають підконтрольне навчання для прогнозів, безконтрольне для кластеризації та навчання з підкріпленням для прийняття рішень. Для оптимізації використовуються алгоритми, такі як градієнтний спуск та його модифікації (Adam, RMSprop), що дозволяють зменшити функцію втрат і покращити точність моделі.

Популярні платформи для реалізації ШНМ, як TensorFlow, PyTorch та Keras, забезпечують інструменти для швидкого створення, навчання і тестування моделей. Водночас вони мають обмеження, зокрема потребу у великих обчислювальних ресурсах і залежність від якості даних.

## 1.2 Огляд інструментів та платформ для побудови, навчання та оцінки моделей

Розробка, навчання та оцінка моделей є складною задачею, яка вимагає спеціалізованих інструментів чи платформ, які забезпечують високу продуктивність роботи з даними, підготовка чи створення даних та гнучкість у розробці чи дослідженні ШНМ. Найпопулярніші платформи розраховані зазвичай на програмістів, що мають досвід у розробці ШНМ, хоча є і платформи, які розраховані на дослідження чи базові налаштування через зручний GUI. Вибір відповідного інструменту залежить від типу задачі, обсягу даних та вимог до обчислювальних ресурсів.

* TensorFlow

Відкрита програмна бібліотека для машинного навчання цілій низці задач, розроблена компанією Google для задоволення її потреб у системах, здатних будувати та тренувати нейронні мережі для виявляння та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди.

* PyTorch

Відкрита бібліотека машинного навчання на основі бібліотеки Torch, що застосовується для задач комп'ютерного бачення та обробки природної мови.

* Keras

Відкрита нейромережна бібліотека, написана мовою Python. Вона здатна працювати поверх TensorFlow, Microsoft Cognitive Toolkit, R, Theano та PlaidML. Спроєктовану для уможливлення швидких експериментів з мережами глибокого навчання, її зосереджено на тому, щоби вона була зручною в користуванні, модульною та розширюваною.

* Transformers

Бібліотека попередньо навчених моделей обробки природної мови, комп'ютерного зору, аудіо та мультимодальних моделей для логічного висновку та навчання.

* Roboflow

Комерційна платформа для розпізнавання об’єктів на відео та експериментів з навчанням моделей комп’ютерного зору. Особливо платформа корисна для роботи з власними наборами зображень

## 1.3 Виявлення недоліків існуючих рішень

Кожна розроблена платформа має свої сильні та слабкі сторони, які потрібно враховувати при проектуванні платформи розробки ШНМ. Нижче наведені недоліки уже реалізованих рішень.

**TensorFlow**

TensorFlow є однією з найпотужніших відкритих бібліотек для реалізації нейронних мереж, розробленою компанією Google. Проте, незважаючи на високу гнучкість та широкі можливості, бібліотека характеризується наступними недоліками:

* Висока складність засвоєння, що ускладнює використання її користувачами без відповідної підготовки;
* Відсутність вбудованих інструментів для анотації зображень та управління наборами даних;
* Необхідність інтеграції з додатковими рішеннями для забезпечення повного циклу розробки моделей комп’ютерного зору.
* Не можливість використовувати повні можливості через CUDA на операційній системі Windows.

**PyTorch**

PyTorch — ще одна популярна бібліотека для розробки моделей машинного навчання, яка вирізняється підтримкою динамічного графа обчислень та активно використовується у наукових дослідженнях. Однак, її використання супроводжується такими обмеженнями:

* Необхідність глибоких знань програмування для реалізації більшості завдань;
* Відсутність інструментів для створення, анотації та попередньої обробки наборів зображень у вбудованому вигляді;
* Значні трудові витрати на початкових етапах проєкту.

**Keras**

Keras є високорівневою нейронною бібліотекою, що працює поверх інших фреймворків, таких як TensorFlow чи Theano. Основною перевагою Keras є простота у використанні. Водночас, до недоліків можна віднести:

* Орієнтацію виключно на етап побудови та навчання моделей, без підтримки інструментів для керування наборами зображень;
* Обмежену функціональність у контексті розв’язання прикладних задач комп’ютерного зору без сторонніх засобів.

**Transformers**

Бібліотека Transformers, створена компанією Hugging Face, містить велику кількість попередньо навчених моделей для задач обробки природної мови, комп’ютерного зору, аудіо тощо. Незважаючи на переваги, бібліотека має наступні недоліки:

* Основний фокус зосереджено на обробці текстової інформації, тоді як можливості для комп’ютерного зору є обмеженими;
* Відсутність засобів для повноцінної роботи з власними зображеннями без попередньої підготовки даних.

**Roboflow**

Roboflow є сучасною хмарною платформою для реалізації проєктів у галузі комп’ютерного зору. Її головна перевага — це зручність роботи з наборами зображень, включно з їх анотацією, попередньою обробкою та генерацією нових даних. Однак, навіть при значних перевагах, платформа має низку обмежень:

* Обмеження функціональності у безкоштовній версії, що ускладнює масштабування проєкту;
* Тренування моделей відбувається на сторонніх серверах, що може створити загрози безпеці при роботі з конфіденційною інформацією;
* Обмежена гнучкість налаштування моделей порівняно з відкритими бібліотеками низького рівня.

Зважаючи на виявлені недоліки існуючих рішень, оптимальним варіантом для розробки платформи дослідження штучних нейронних мереж є використання бібліотеки PyTorch у поєднанні з міні-платформою для генерації даних для навчання та попередньо підготовленою програмою для навчання моделі. Такий підхід дозволяє:

* швидко змінювати параметри моделі;
* спостерігати вплив цих змін на результати;
* ефективно порівнювати різні реалізації в рамках єдиного середовища.

## 1.4 Вибір математичної моделі для навчання нейронних мереж

## 1.5 Обґрунтування доцільності створення програмної платформи

## 1.6 Вибір та аналіз алгоритмів розв’язання задачі

### 1.6.1 Опис алгоритму для створення датасету

### 1.6.2 Алгоритм навчання моделі на створеному датасеті

### 1.6.3 Алгоритм перевірки моделі на основі навчання

## 1.7 Основні режими функціонування платформи

## Висновки до розділу 1

# Розділ 2. ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ

# Розділ 3. РЕАЛІЗАЦІЯ АЛГОРИТМУ РОЗВ’ЯЗАННЯ ЗАДАЧІ

# Економічна частина

# Висновки

# Список літератури

# Додатки