**Лабораторна робота 2.2:** Багатокласова класифікація з використанням нейронних мереж

**Мета лабораторної роботи**

У цій лабораторній роботі ви:

* Побудуєте та навчите нейронну мережу для розв'язання задач багатокласової класифікації
* Застосуєте різні архітектури нейронних мереж та порівняєте їх ефективність
* Використаєте функцію активації softmax та функцію втрат categorical cross-entropy
* Створите інтерактивний інтерфейс за допомогою Gradio для демонстрації вашої моделі
* Розгорнете навчену модель на платформі HuggingFace для публічного доступу

### Варіант 2: Fashion MNIST

Опис: Класифікація зображень одягу та аксесуарів за 10 категоріями.

Джерело даних: Доступний через TensorFlow/Keras (tf.keras.datasets.fashion\_mnist).

Вимоги до моделі:

* Використання згорткових шарів (CNN)
* Реалізація техніки аугментації даних
* Досягнення точності не менше 90% на тестовому наборі

**Завантаження та підготовка даних**

Використали вбудований датасет Fashion MNIST, який містить 60 000 зображень для тренування і 10 000 для тестування. Провели нормалізацію пікселів (ділення на 255.0). Змінили форму даних для подачі в CNN (розмір 28x28x1). Перетворили мітки у one-hot вектор за допомогою to\_categorical.

**2. Аугментація даних**

* Використали ImageDataGenerator для штучного розширення тренувального набору за допомогою:
  + обертання зображень;
  + зсувів по ширині/висоті;
  + масштабування (zoom).

**3. Побудова та навчання моделі**

Побудували CNN-модель за допомогою Sequential, яка включає:

* + два згорткових шари (Conv2D);
  + pooling (MaxPooling2D);
  + шари Dropout для запобігання перенавчанню;
  + шар Flatten і два повнозв’язні шари (Dense), останній з softmax-активацією для багатокласової класифікації.

Скомпілювали модель з оптимізатором adam і функцією втрат categorical\_crossentropy.

Додали callback EarlyStopping для зупинки навчання при відсутності покращень.

**4. Збереження та завантаження моделі**

* Якщо файл моделі (fashion\_model.h5) не існує, модель тренується і зберігається.
* Якщо файл вже існує — модель просто завантажується.

**5. Створення інтерфейсу користувача (Gradio)**

Реалізували функцію classify\_image, яка:

* + обробляє зображення з інтерфейсу (перетворює в відтінки сірого, масштабує, нормалізує, подає у модель);
  + повертає ймовірності належності до кожного класу.

Створили інтерфейс за допомогою gr.Interface, де:

* + користувач може завантажити або намалювати зображення одягу;
  + модель виводить три найімовірніші класи (наприклад, "Сорочка — 87%", "Светр — 8%", тощо).

**Результат:**

1. Отримано точну модель для класифікації одягу.
2. Реалізовано зручний веб-інтерфейс для взаємодії з користувачем.
3. Застосовано повний pipeline машинного навчання: від обробки даних — до розгортання інтерфейсу.