Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та комп’ютерних технологій

Кафедра системного проектування

Звіт

Про виконання лабораторної роботи №5

З курсу «Системи машинного навчання»

«Нейронні мережі та PyTorch/Keras»

**Виконала*:***

Студентка групи ФЕС-32

Філь Дарина

**Перевірив*:***

Доцент Колич І. І.

Львів 2024

**Мета:** навчитися будувати та навчати прості нейронні мережі для задач класифікації та регресії.

**Теоретичні відомості**

**Нейронні Мережі**

**Нейронні мережі** — це обчислювальні моделі, які натхнені біологічними нейронними системами. Вони складаються з великої кількості взаємопов’язаних обчислювальних елементів, названих нейронами, які працюють разом для вирішення конкретних задач.

**Основні Компоненти Нейронних Мереж**

1. **Вхідний шар (Input Layer):** Приймає дані, які будуть оброблятися мережею.
2. **Приховані шари (Hidden Layers):** Шари, розташовані між вхідним і вихідним шарами, кожен з яких складається з певної кількості нейронів.
3. **Вихідний шар (Output Layer):** Видає результат обробки даних мережею.

**Основні Параметри Нейронних Мереж**

1. **Кількість шарів:** Впливає на здатність нейронної мережі навчатися складних патернів.
2. **Кількість нейронів у шарі:** Кожен шар може містити різну кількість нейронів, що впливає на обчислювальну потужність та здатність до узагальнення.
3. **Функції активації:** Визначають нелінійність, яку нейронна мережа може моделювати. Популярні функції активації включають ReLU, сигмоїдальну функцію та тангенс.

**PyTorch та Keras**

PyTorch — це відкрита бібліотека для машинного навчання, розроблена Facebook, яка є потужним інструментом для створення та навчання нейронних мереж.

Keras — це високорівнева бібліотека, яка може використовуватися з різними бекендами, включаючи TensorFlow та Theano. У даному прикладі ми будемо використовувати PyTorch як бекенд для Keras.

**Хід роботи**

**Завдання**

**1. Підготовка середовища**

**1.1 Встановіть необхідні бібліотеки, якщо вони ще не встановлені: PyTorch, Keras, Matplotlib**

**1.2 Імпортуйте необхідні бібліотеки в Python.**

**2. Створення простої нейронної мережі, що відмінна від прикладу**

**2.1 Створіть та налаштуйте модель нейронної мережі за допомогою Keras (або PyTorch):**

* + Використовуйте послідовну модель (Sequential).
  + Додайте шари до моделі (Dense). Кількість шарів має бути іншою, ніж у прикладі
  + Вкажіть кількість нейронів у кожному шарі та функції активації.

**2.2 Сконфігуруйте модель для навчання (в прикладі позначено ):**

* + Виберіть алгоритм оптимізації (наприклад, sgd).
  + Вкажіть функцію втрат (наприклад, categorical\_crossentropy для задач класифікації).
  + Вкажіть метрику для оцінки моделі (наприклад, accuracy).

**3. Навчання нейронної мережі на наборі даних**

**3.1 Завантажте та підготуйте дані:**

* + Використайте набір даних, наприклад, Fashion-MNIST.
  + Розділіть дані на тренувальну та тестову вибірки.
  + Нормалізуйте дані, якщо необхідно.
  1. **Навчіть модель на тренувальних даних:**
  + Вкажіть кількість епох.
  + Вкажіть розмір пакета (batch size).
  + Збережіть історію навчання для подальшої візуалізації.

**4. Оцінка та візуалізація результатів**

**4.1 Оцініть модель на тестових даних**:

* + Використайте метод оцінки (evaluate) для отримання метрик якості.

**4.2 Візуалізуйте результати навчання та оцінки:**

* + Побудуйте графіки втрат та точності на тренувальних та тестових даних.
  + Візуалізуйте помилково класифіковані приклади.

**5. Оптимізація гіперпараметрів**

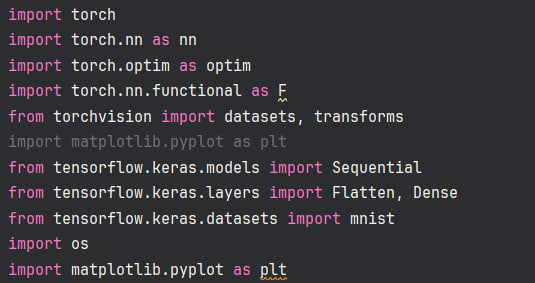
**5.1 Експериментуйте з різними параметрами моделі, щоб підвищити її точність:**

* + Змініть кількість шарів та нейронів.
  + Використайте різні функції активації.
  + Спробуйте різні алгоритми оптимізації.

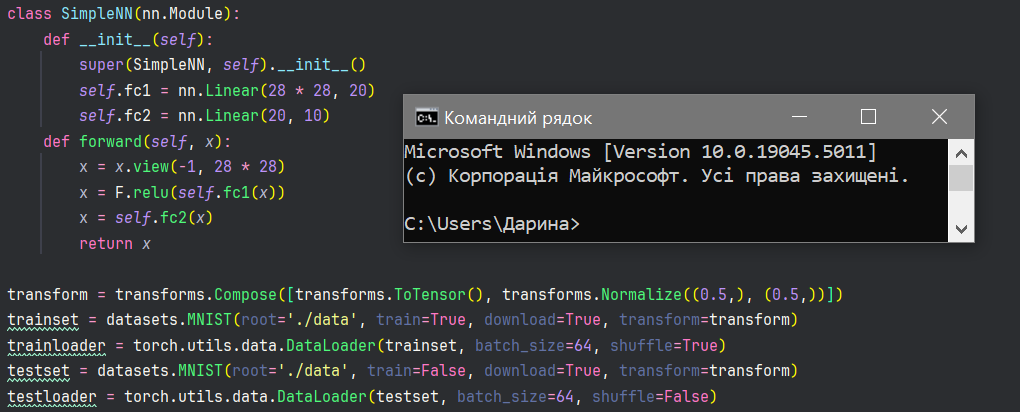
**5.2 Порівняйте результати:**

* + Запишіть результати експериментів.
  + Проаналізуйте, як зміна параметрів впливає на якість моделі.

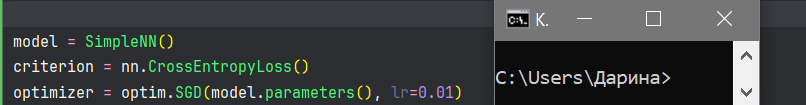
**6. Оформити звіт**



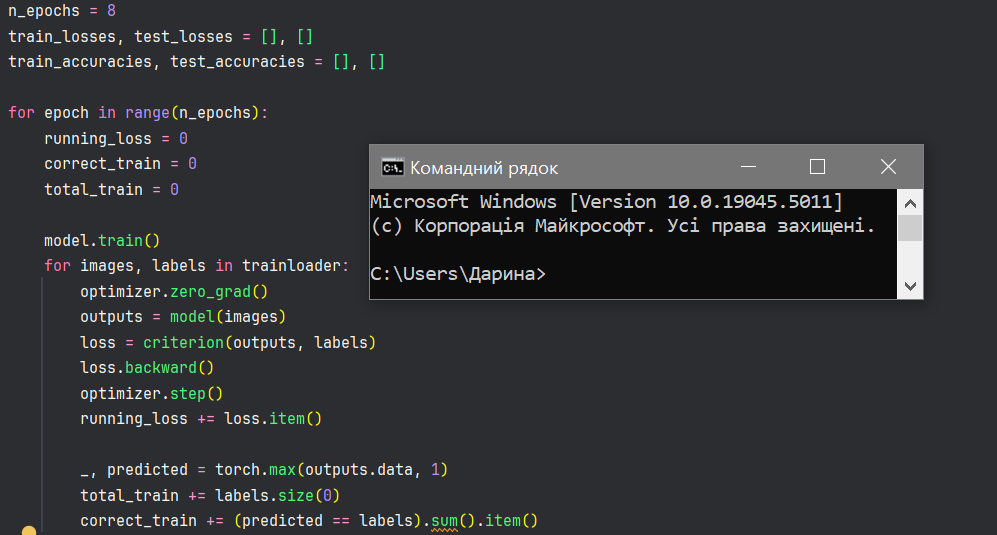
**Рис.1** Завантаження бібліотек



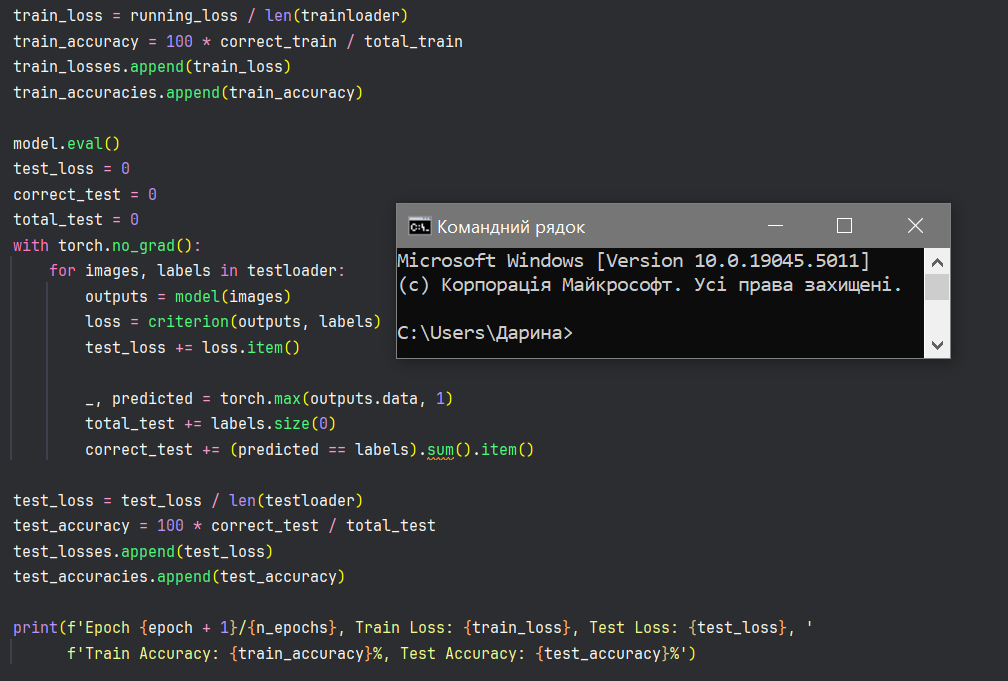
**Рис.2** Розробка класу для простої нейронної мережі, завантаження набору даних Fashion-MNIST, його нормалізація та поділ на тренувальну і тестову вибірки



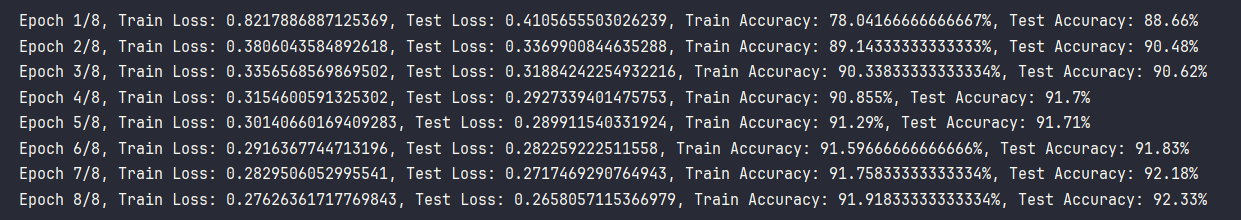
**Рис.3** Вибір для моделі простої нейронної мережі, функції для класифікації та оптимізатора

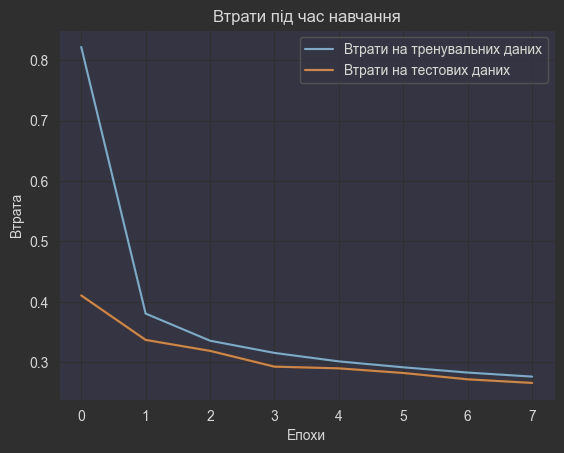


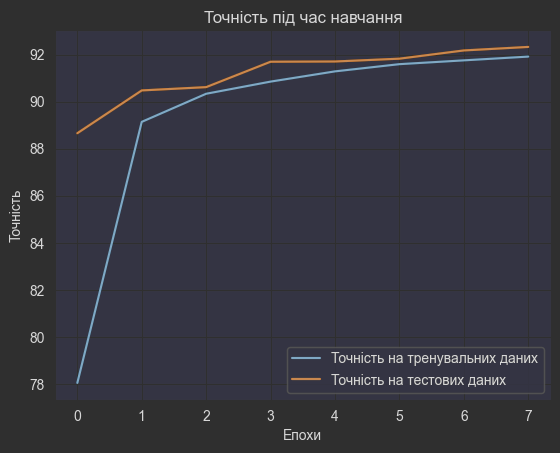
**Рис. 4** Тренування моделі простої нейронної мережі, з кількістю епох = 8.



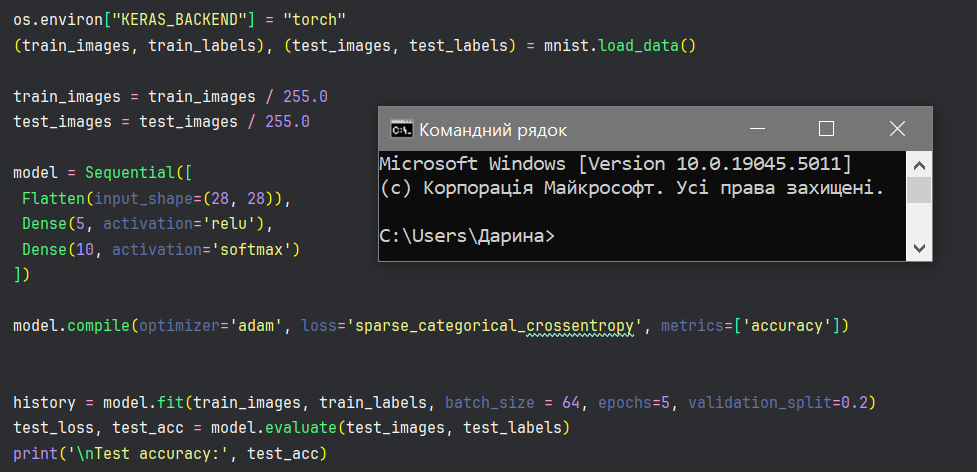
**Рис.5** Код для оцінювання нашої моделі.



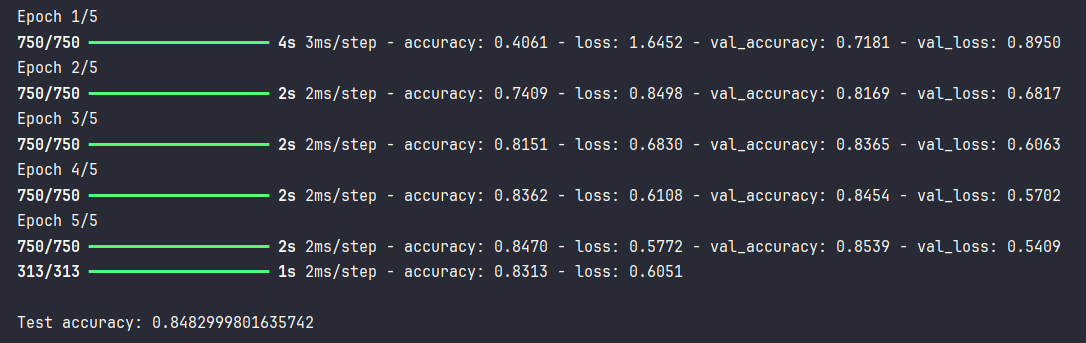
**Рис.6** Результати навчання моделі за епохами показують, що з підвищенням кількості епох точність моделі збільшується. Однак видно, що з 7-ї до 8-ї епохи точність майже не змінилася

****

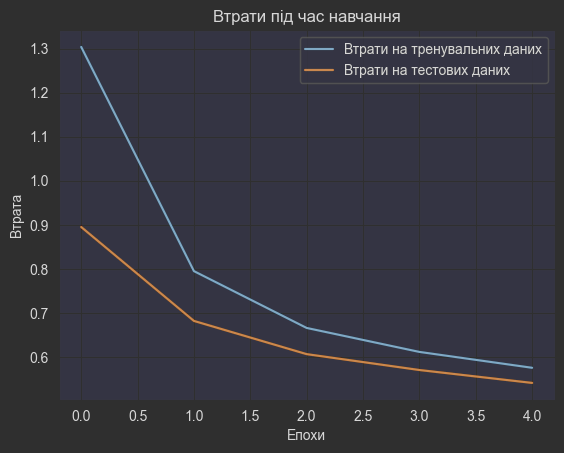
**Рис.7-8** Графіки, що відображають точність та втрати відносно епохи тренування простої нейронної мережі.

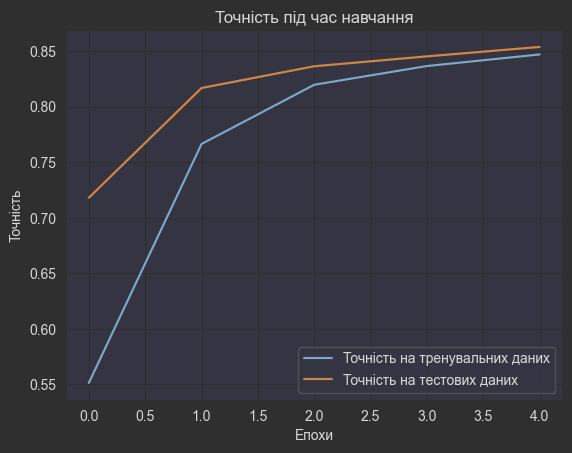


**Рис.9** Створення та навчання нейронної мережі з послідовними шарами, де як бекенд обрано "torch", хоча це вже застаріла версія; наразі частіше використовується tensorflow. Для оптимізації застосовано оптимізатор adam, а кількість нейронів становить 5

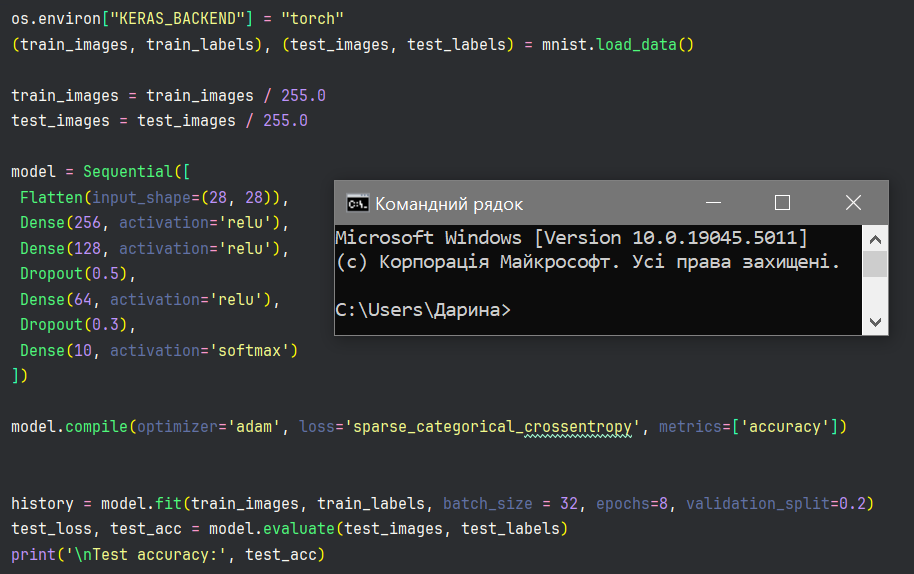


**Рис.10** Результат тренування моделі на 5 епохах

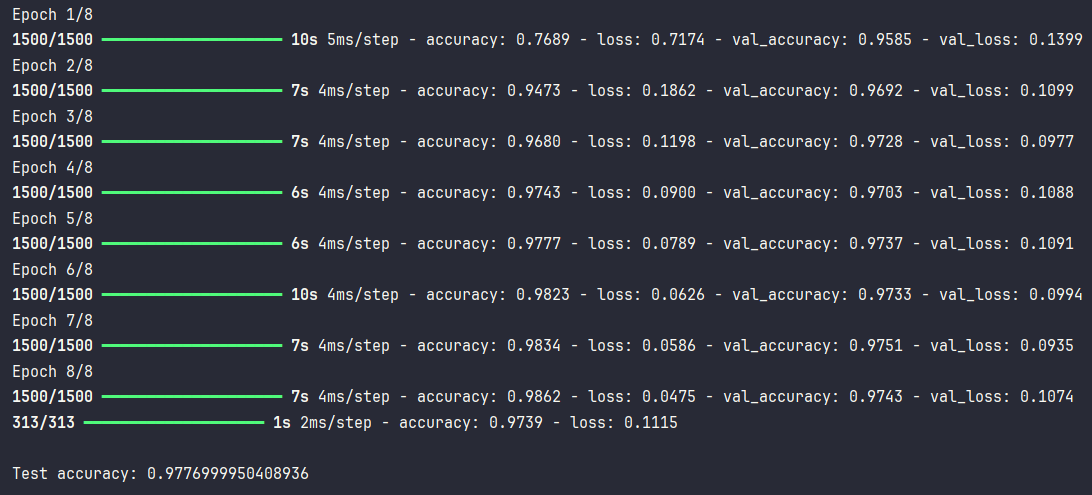




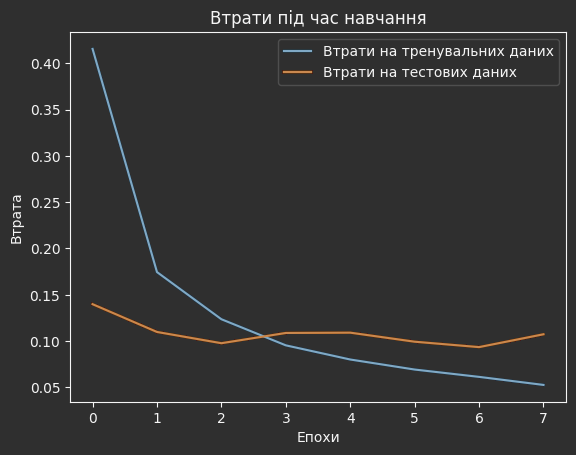
**Рис.11-12** Графіки, що відображають точність та втрати відносно епохи тренування послідовної нейронної мережі.

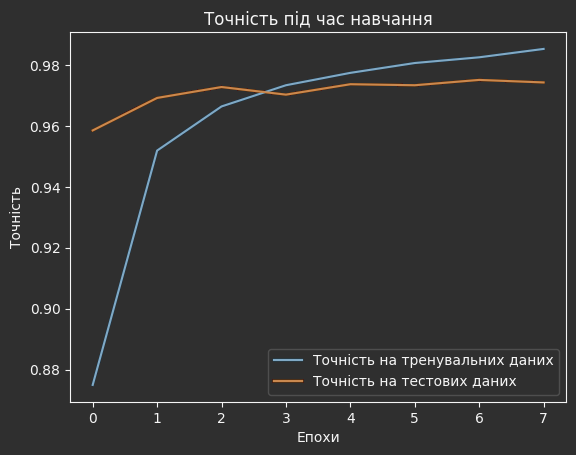


**Рис.13** Збільшимо кількість нейронів, замінимо оптимізатор на rmsprop, зменшимо `batch\_size` удвічі, додамо додаткові шари та застосуємо dropout з регуляризацією L2 для покращення узагальнювальної здатності моделі

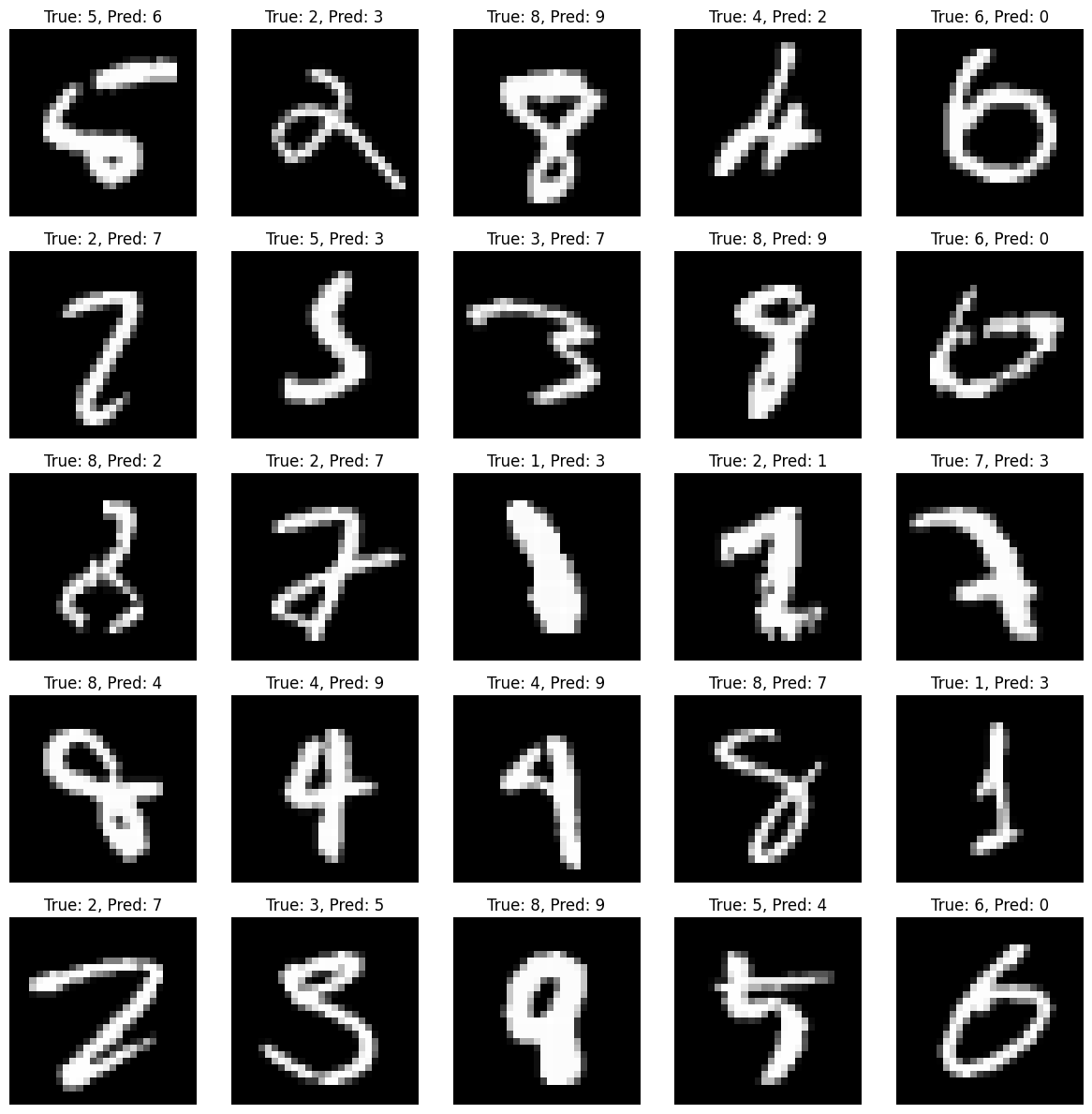


**Рис.14** Результат тренування



****

**Рис.15-16** Графіки, що відображають точність та втрати відносно епохи тренування послідовної нейронної мережі.



**Рис.17** Неправильно класифіковані значення в моделі послідовних шарів

**Висновок:** у цій лабораторній роботі я навчилася тренувати прості та послідовні нейронні мережі й налаштовувати їх. Під час виконання роботи помітила, що точність моделі залежить від кількості епох (хоча при великій їх кількості може виникнути перенавчання), а також від обраного оптимізатора та кількості нейронів.