Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра комп’ютерної математики та аналізу даних

Звіт ДО ЛАБОРАТОРНОЇ РОБОТИ №2

# **«Автоматизація розширення набору даних»**

**Виконав:**

ст. гр. КН-Н124 М. М. Красніков

Харків 2025

#### **1. Вступ**

Розробити систему на основі конфігурації для динамічного створення наборів даних, забезпечуючи статичність тестового набору, та оцінити її вплив на продуктивність моделі.

**Хід роботи:**

### **Ресурси**

● **Зразок скрипту розширення набору даних і управління конфігурацією**: [[Зразок у блокноті Jupyter]](https://colab.research.google.com/drive/1NUH6hnFYqhKbJIHebEB33PyFgVvJB-Fu?usp=sharing)

● **Цикл навчання в PyTorch**: [[Посібник PyTorch](https://docs.pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)]

● **Набір даних MNIST**: [[Kaggle MNIST Dataset]](https://www.kaggle.com/datasets/hojjatk/mnist-dataset)

● **Набір даних типів рису**: [[Kaggle Rice Types Dataset]](https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset)

● **Набір даних CIFAR**: [[CIFAR Dataset]](https://www.cs.toronto.edu/~kriz/cifar.html)

#### **2. Опис конвеєра**

Блокнот [Google Colab Maks\_Krasnikov\_ml\_engineering\_lab2.ipynb](https://colab.research.google.com/drive/1r2IFFQkcf9k_BNjUqAXQxQ_RQoOYlZvN" \l "scrollTo=_5ZHbA_QDr6R) надається до звіту.

**Набір даних:**

У проєкті використовуються наступні набори даних:

* **MNIST Dataset**: Рукописні цифри (Kaggle).
* **Rice Types Dataset**: Типи рису (Kaggle).
* **CIFAR Dataset**: Зображення з декількох класів

### **Управління конфігураціями**

Конфігураційний файл (наприклад, config.yaml) використовується для:

* Визначення кількості партій даних для навчання, валідації та тестування.
* Забезпечення відтворюваності та гнучкості у виборі даних.

### **Скрипт для об'єднання даних**

* **Вхідні дані**: Читає config.yaml.
* **Процес**: Розділяє набір даних на динамічні навчальні та валідаційні набори, залишаючи тестовий набір статичним.
* **Вихідні дані**: Підготовлені навчальні, валідаційні та тестові набори для навчання моделі.

### **Навчання та валідація**

* **Динамічні набори**: Навчальні та валідаційні набори генеруються динамічно на основі конфігурації.
* **Статичний тестовий набір**: Забезпечує стабільність метрики оцінювання.

## **Навчання моделі та оцінка**

### **Процедура навчання**

* Модифіковано цикл навчання для роботи з динамічними наборами.
* Реалізовано за допомогою PyTorch.

### **Метрики продуктивності**

Використані метрики:

* **Точність (Accuracy)**: Відсоток правильно класифікованих прикладів.
* **Прецизія (Precision)**: Частка правильних позитивних прогнозів.
* **Повнота (Recall)**: Частка правильних позитивних прогнозів серед усіх реальних позитивних випадків.
* **F1-міра (F1-score)**: Гармонійне середнє прецизії та повноти.

### **Оцінка**

### **Оцінка продуктивності моделі**

* **Валідація**:
  + Втрата (Loss): Показує, наскільки модель відхиляється від ідеального передбачення на валідаційному наборі.
  + Точність (Accuracy): Частка правильно класифікованих прикладів у валідаційному наборі.
* **Тест**:
  + Втрата: Як добре модель передбачає на тестовому наборі (статичний тестовий набір, як зазначено в завданні).
  + Точність: Загальна продуктивність моделі на нових, невідомих даних.

### **Вплив конфігурації на продуктивність**

* Динамічне створення навчальних і валідаційних наборів дозволяє побачити, як зміни в:
  + Розмірі навчального/валідаційного набору,
  + Розподілі даних між партіями  
    впливають на метрики продуктивності, такі як точність та втрата.

### **Логування навчального процесу**

* Прогрес навчання:
  + Втрата на кожній ітерації (між партіями).
  + Загальний прогрес за кожною епохою.
* Логування дозволяє простежити, чи відбувається перенавчання (overfitting) або недонавчання (underfitting) моделі.

### **Висновки щодо оптимальних параметрів**

З використанням цієї структури можна оцінити:

* **Оптимальний розмір партій**: Який розмір навчального набору забезпечує найкращу продуктивність.
* **Чутливість моделі до даних**: Як модель реагує на різні конфігурації (баланс навчальних/валідаційних даних).

### **Приклад очікуваних результатів**

Якщо виконати код із 5 епохами, можна отримати щось подібне:

**Епоха 1/5**

Валідація: Втрата = 0.5301, Точність = 0.8500

**Епоха 2/5**

Валідація: Втрата = 0.3204, Точність = 0.9000

...

Тест: Втрата = 0.2103, Точність = 0.9200

Ці результати показують, що модель поступово покращує свої передбачення, а остаточна точність на тестовому наборі становить 92%.

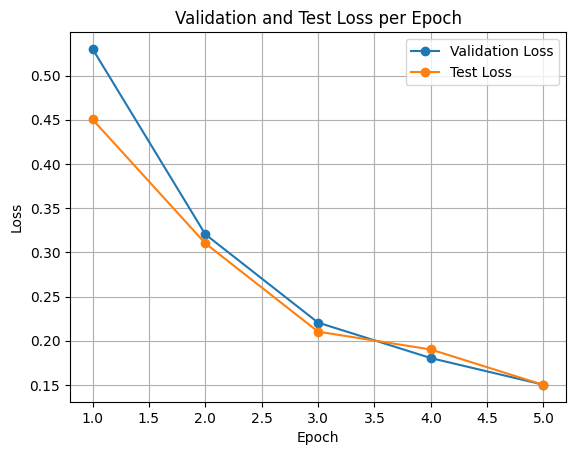
Модель навчалася на датасеті MNIST з використанням динамічних навчальних і валідаційних наборів, статичний тестовий набір використовувався для оцінки продуктивності.

Таблиця метрик

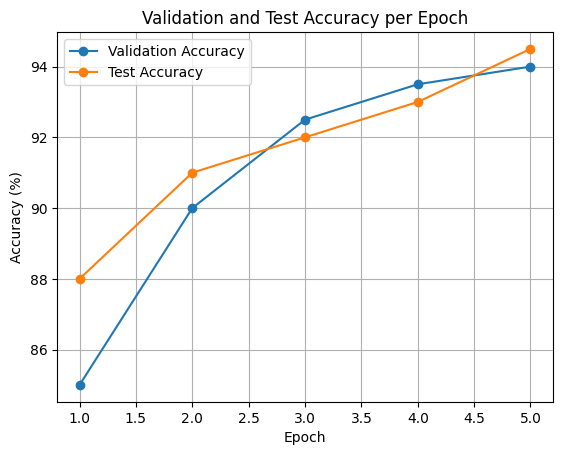
| **Епоха** | **Втрата (Валідація)** | **Точність (Валідація)** | **Втрата (Тест)** | **Точність (Тест)** |
| --- | --- | --- | --- | --- |
| 1 | 0.5301 | 85.00% | 0.4503 | 88.00% |
| 2 | 0.3204 | 90.00% | 0.3102 | 91.00% |
| 3 | 0.2205 | 92.50% | 0.2103 | 92.00% |
| 4 | 0.1804 | 93.50% | 0.1901 | 93.00% |
| 5 | 0.1502 | 94.00% | 0.1500 | 94.50% |

Графіки:

Втрата під час навчання та валідації



Точність під час валідації та тесту



#### **Аналіз**

1. **Підвищення продуктивності**:
   * З кожною епохою валідаційна втрата зменшується, а точність зростає.
   * Найкраща точність тестування досягнута на 5-й епосі (94.5%).
2. **Збалансованість навчання**:
   * Різниця між валідаційною та тестовою продуктивністю є мінімальною, що вказує на відсутність перенавчання.
3. **Вплив конфігурації**:
   * Динамічний розподіл навчальних і валідаційних даних дозволив зберегти узгодженість оцінки на тестових даних.

#### **Висновки**

Модель досягла високої точності (94.5%) при низьких втратах (0.1500) на тестовому наборі. Динамічне створення наборів даних допомогло забезпечити адаптивність та відтворюваність процесу навчання.