Лабораторная работа №8

**Исследуем датасет MNIST с использованием разных фреймфорков для работы с нейронными сетями**

База данных **MNIST** (сокращение от «Modified National Institute of Standards and Technology») —база данных образцов рукописного написания цифр. База данных является стандартом, предложенным Национальным институтом стандартов и технологий США с целью калибрации и сопоставления методов распознавания изображений с помощью машинного обучения в первую очередь на основе нейронных сетей.

База данных была создана после переработки оригинального набора чёрно-белых образцов размером 20x20 пикселей NIST. Создатели базы данных NIST, в свою очередь, использовали набор образцов из Бюро переписи населения США, к которому были добавлены ещё тестовые образцы, написанные студентами американских университетов. Образцы из набора NIST были нормализированы, прошли сглаживание и приведены к серому полутоновому изображению размером 28x28 пикселей.

База данных MNIST содержит 60000 изображений для обучения и 10000 изображений для тестирования.



Поддержка датасета MNIST есть во многих крупных фреймворках для работы с нейронными сетями. Например, MNIST есть среди данных библиотеки Keras. Чтобы добавить датасет к программе, нужно просто указать в начале кода строчку:

from keras.datasets import mnist

Библиотека Keras работает на основе фреймворка для машинного обучения Tensor Flow — тот тоже поддерживает MNIST. PyTorch также в своем модуле torchvision содержит набор данных MNIST.

**Примеры работы с MNIST используя Tensor Flow:**

<https://webtort.ru/решаем-задачу-mnist-в-keras-или-учим-нейросеть-р/>

<https://www.kaggle.com/code/prashant111/mnist-deep-neural-network-with-keras>

**Примеры работы с MNIST используя PyTorch :**

<https://h1ros.github.io/posts/train-the-image-classifier-using-pytorch/>

<https://www.kaggle.com/code/cgurkan/mnist-with-pytorch-cnn>

**Задания к работе:**

Установить Tensor Flow и PyTorch если вы работаете на локальной системе

Часть 1. Распознавание данных MNIST используя многослойный персептрон (MLP)

Используемая конфигурация сети – 3 скрытых слоя по 100 нейронов плюс выходной слой из 10 нейронов. Функции активации выбрать самостоятельно. Реализовать обучение **одинаковой**! конфигурации сети используя Tensor Flow и PyTorch. Вывести метрики классификации. Сравнить время обучения и полученные результаты.

Часть 2. Распознавание данных MNIST используя сверточную сеть(CNN)

Используемая конфигурация сети – 2 набора слоев свертка+пуллинг (использовать свертку с размером ядра 5 и пулинг с размером 2). Один полносвязный слой на 500 узлов и выходной слой на 10. Остальные параметры выбрать самостоятельно. Внимательно изучите как связываются слои между собой! Реализовать обучение **одинаковой**! конфигурации сети используя Tensor Flow и PyTorch. Вывести метрики классификации. Сравнить время обучения и полученные результаты, а также сравнить с результатами первой части.

Сделать общий вывод. Какой из фреймворков вам понравился больше и почему. Отчет должен содержать 3 файла: 1 – с реализацией на PyTorch, второй - Tensor Flow. Отчет оформить отдельным файлом (третьим)