

1. Состав команды

Лаптева Арина Александровна Б03-013 (тимлид):

- предобработка датасета LFW
- реализация модели SiameseNetwork
- обучение и логирование обучения модели
- подбор гиперпараметров и исследование функций ошибки

Скитская Александра Владимировна Б03-014 (разработчик):

- реализация ArcFaceLoss
- подбор аугментации изображений
- выбор модели для обучения SiameseNetwork

Прошкина Надежда Алексеевна Б03-014 (разработчик):

- выбор датасета для обучения LFW
- реализация SiameseDataset
- реализация чтения данных для датасета LFW и для нашего датасета
- сбор и предобработка датасета из своих фотографий

2. Модель и датасет

Для реализации этого домашнего задания был выбран Вариант 3 и Вариант 4 из предложенных, а именно:

- Обучите Siamese Neural Network с помощью PyTorch, чтобы изучить функцию сходства между двумя изображениями для задач верификации лиц. Используйте набор данных Labeled Faces in the Wild (LFW) для обучения и оценки модели, и оптимизируйте ее, используя техники регуляризации и настройки гиперпараметров.
- Исследуйте потери ArcFace, функцию потерь, специально разработанную для задач распознавания лиц, с помощью PyTorch. Обучите модель, используя потери ArcFace, и сравните ее производительность с моделью, обученной с помощью традиционных потерь softmax. Оптимизируйте модель, используя техники дообучения и передачи обучения.

3. Предобработка датасета LFW

На [сайте](#) LFW помимо датасета есть файлы pairsDevTest.txt и pairsDevTrain.txt, в которых прописаны пары изображений и соответствующая метка: 1 - один и тот же человек, 0 - разные.

В качестве предобработки использовалась следующая аугментация: случайный поворот относительно горизонтальной оси, случайный поворот относительно вертикальной оси, случайный поворот на 30 градусов в положительном или отрицательном направлении.

Размер изображений был оставлен как в датасете, 250 на 250 пикселей.

4. Исследование ArcFaceLoss

Графики обучения и их описание приведены в wandb-отчёте по ссылке: [homework5 | hw5 – Weights & Biases \(wandb.ai\)](#)

Для начала был использован ArcFaceLoss. И на 20, и на 40 эпохах сама функция потерь хорошо падала, но метрика ассурасу на тесте составила максимум 0.51, что отражено в файле metrics.json. Поэтому в окончательном варианте мы

использовали BCELoss. Метрика accuracy на тесте составила 0.73 и приведена в файле metrics_BCE.json.

5. Демонстрация решения на наших изображениях

Датасет с нашими фотографиями приведен в папке data/our_dataset, результат работы модели приведен в data/result. В качестве опорного изображения (изображение, для которого мы точно знаем кто это) берется первое изображение в папке data/our_dataset/{name}. Для каждого человека есть минимум 6 случаев, для которых верно определено, что это не тот человек. Для каждого человека есть минимум 3 случая, для которых верно определено, что это тот человек. Метрика accuracy и лосс приведены в папки для каждого человека в папке data/result/{name}.

Для воспроизведение процесса необходимо запустить файл example.py и в консоли ввести имя, для которого будут сохранены результаты (Arina, Sasha или Nadya).

6. Структура проекта

Веса обученной модели хранятся в файле model_siamese_BCE.pt, который можно скачать по [ссылке](#).

Исходный код обученной модели хранится в папке src. Для того, чтобы воспроизвести процесс загрузки данных, обучения и подсчёта метрики необходимо:

- предварительно [скачать данные](#) и текстовые файлы описания пар для [обучающей](#) и [тестовой](#) выборки в папку data
- для предобработки данных и начала обучения запустить файл train.py
- для подсчёта метрики на тесте запустить файл infer.py