

1. Состав команды

Лаптева Арина Александровна Б03-013 (тимлид):

- реализация модели ProtoNet (задания 2-3)
- обучение и логирование модели (задание 4)

Скитская Александра Владимировна Б03-014 (разработчик):

- чтение и аугментация исходных данных (задание 1)
- визуализация графиков обучения с методом simCLR и без (задание 6)

Прошкина Надежда Алексеевна Б03-014 (разработчик):

- визуализация результатов обучения ProtoNet (задание 5)
- реализация классификатора и обучение с методом simCLR и без (задание 6)
- сравнение результатов с предобученным энкодером с методом simCLR и без (задание 6)

2. Графики обучения и метрики

Отчёт по графикам лоса и метрикам на выборке во время обучения, реализованной модели ProtoNet:

[homework4 | hw4 – Weights & Biases \(wandb.ai\)](#)

Метрика и функция потери на тестовой выборке:

- acc_loss_test.json - метрика и лосс на тестовой выборке модели ProtoNet

3. Веса обученной модели ProtoNet хранятся в файле model_omniglot.pt, [скачать файл](#)

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

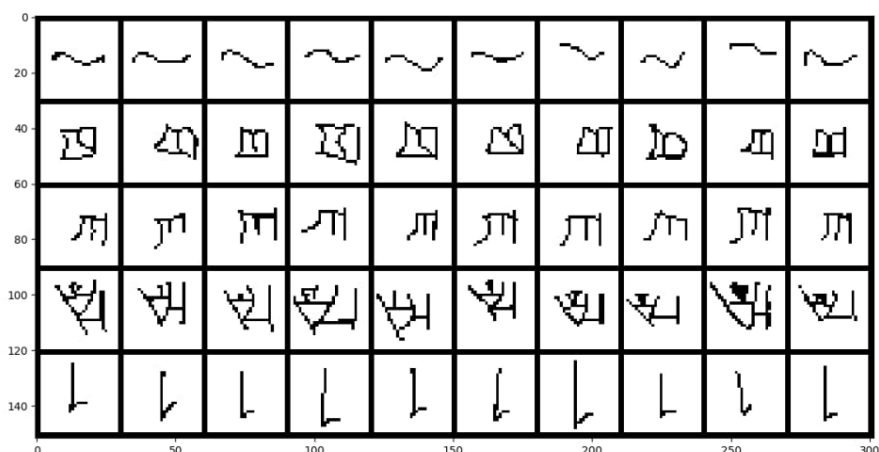
- для скачивания данных запустить файл load_data.py
- для начала обучения запустить файл train.py
- для подсчёта метрики запустить файл test.py

5. Визуализация предсказаний

Функция для **визуализации предсказания** с тестирования реализована в файле visualization.py. Для получения визуализации работы модели и визуализации эмбедингов выборки нужно запустить этот файл. Основа для разработки функции взята из семинара 7, функция plot_features().

Пример визуализации:

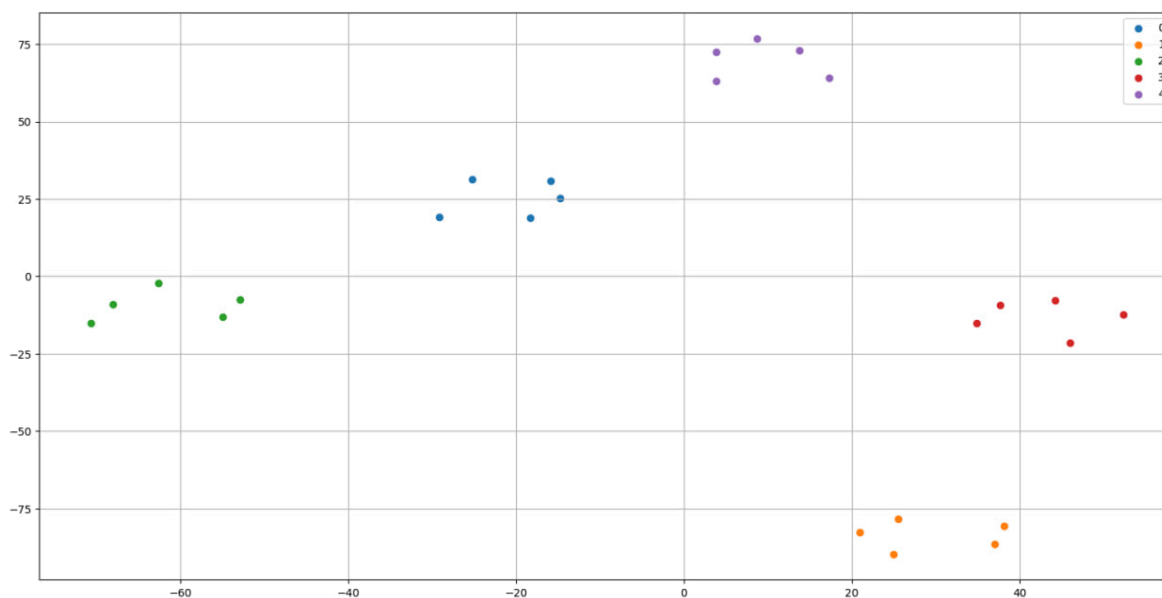
Сэмпл:



Предсказания:

```
{'loss': -0.0, 'acc': 1.0, 'y_hat': tensor([[0, 0, 0, 0, 0],  
      [1, 1, 1, 1, 1],  
      [2, 2, 2, 2, 2],  
      [3, 3, 3, 3, 3],  
      [4, 4, 4, 4, 4]], device='cuda:0')}
```

Результат:



6. Задание 6

Все файлы для **задания 6** находятся в папке task_6. Задание реализовано в файле downstream.py. Для запуска в директории должен находиться файл из семинара 7 cifar.py, а так же папка add_data с двумя файлами: load.py и simCLR.py (материалы взяты из семинара 7 и разбиты по файлам). В целом файл сделан по аналогии семинара 7.

Картинки по визуализации лосс функции и accuracy_score модели с обученным энкодером методом simCLR и без этого метода приложены в папке plot_metrics. Название каждой картинки обозначает какая модель (с или без simCLR), на вертикальной оси указано какая метрика визуализируется, график подписан по какой выборке идет визуализации (val или train). Из графиков accuracy score можно сделать вывод, что модель с энкодером обученным методом simCLR показывает результаты лучше.

В файлах hyp_params_clsif.yaml и hyp_params.yaml указаны гиперпараметры для моделей, метрики которых сравниваем (с simCLR и без).