

Лабораторная 5 - image

[github classroom task](#)

Порядок сдачи и защиты лабы описан [тут](#)

Дедлайн указан в [таблице с баллами](#)

Данные

Соберите датасет изображений. Можно спарсить данные с сайта или получить через api

Идеи задач

Использовать конкретно эти - уже не так интересно. Но для вдохновения подойдет

- регрессия по количеству лайков на меме
- поиск похожих лиц с помощью [сиамской сети](#)
- определение персонажей на странице манги
- бинарная классификация того, сгенерирована ли картинка
- решение капч
- вид ремонта на фото квартиры: дизайнерский/бабушкин/отсутствует

План действий

1. Соберите датасет картинок
2. (если нет разметки и она нужна) Разметьте данные
Используйте gpt vision/любую opensource мультимодальную LLM для разметки целого датасета. Если нет подходящей видеокарты, используйте google colab/kaggle
Либо можете использовать [активное обучение](#), чтобы разметить данные полуавтоматически. Не забывайте проверять качество авторазметки
3. Реализуйте и обучите CNN модель
Следите, чтобы модель не переобучилась. Логируйте метрики обучения и валидации в tensorboard
4. Создайте эмбеддинги картинок своего датасета
Используйте эмбеддинг из ~~последнего сверточного~~ предпоследнего линейного слоя вашей модели. Полученные эмбеддинги отобразите в 2d с помощью tsne/umap. Выделите цветом целевую переменную
5. Реализуйте transfer learning модель
Используйте предобученную модель в качестве backbone. Например, ResNet из torchvision. Добавьте несколько своих линейных слоев и обучите модель на тех же данных
6. Сравните длительность обучения и метрики двух получившихся моделей

Рекомендации

- не стесняйтесь задавать вопросы в чат курса

- рекомендуемый размер датасета - более 1000 картинок на класс в случае классификации. Остальные случаи сложно оценить вне контекста задачи
- используйте `pytorch.nn.Sequential` для слоев модели
- если не получается использовать `torch.device("cuda")` на своей машине, то лучше использовать google colab. Учтите, что он может медленно работать с большим количеством маленьких файлов, таким как датасет картинок

Для реализации не обязательно, но рекомендуются эти python библиотеки:

- `pytorch`
- `torchvision`
- [huggingface](#) как банк предобученных моделей и кода для них
- `tensorboard`

Популярные замечания

- используйте конфиг файл для констант
- не коммитить `.idea`
- функции длиннее 20 строк - плохо. Их можно декомпозировать
- не показано, что модель не переобучена. Сравните метрики на `train` и `test`
- в модели нет `dropout/batchnorm` слоев. Она быстро переобучается из-за этого
- обучайте модель больше 30 эпох. Deep learning модель с 4 эпохами - почти рандомная модель
- не забудьте указать `.requires_grad=False` для backbone модели, чтобы не тратить время на ее оптимизацию
- CNN из 2 сверточных слоев - скорее всего слишком маленькая модель
- CNN из более чем 100млн параметров - скорее всего слишком большая модель

Теормин

В каждом вопросе про метрики, формулы и тп нужно знать как это рассчитывается и почему формулы именно такие. Зубрить сами формулы не нужно

В вопросах про модели подразумевается: Как устроена, как обучается, какие гиперпараметры, какие требования к данным и к чему уязвима, какие сценарии применения

Преподаватель может задать доп вопрос об использованном вами термине, чтобы понять, что вы понимаете его смысл

1. датасет `mnist` и `fashion mnist`: из чего состоит, зачем нужен. Что такое свертка [точки зрения мат анализа](#) (на примере свертки импульса)
2. что такое сверточный слой с точки зрения глубокого обучения. Какие гиперпараметры, на что они влияют, как слой обучается, какие признаки извлекает на каком слое сверточной сети

3. как операция свертки меняет размер изображения? От каких параметров зависит то, насколько изображение станет меньше? Что такое padding и stride в свертке? Неплохая [визуализация](#)
4. карта признаков (feature map). Как рассчитывается, является ли слоем, какие признаки хранит на разных слоях сети, разное ли количество карт признаков в разных слоях
5. ядро свертки применяется к одному или к нескольким слоям изображения? К одному или нескольким картам признаков?
6. 2d pooling. Как работает, обучается ли слой, зачем нужен, какие гиперпараметры и на что влияют
7. batchnorm2d. Как работает, обучается ли слой, зачем нужен, какие гиперпараметры и на что влияют
8. как сверточная сеть преобразует картинку в эмбеддинг картинки. Что делать с эмбеддингом
9. архитектура resNet. Как улучшает поток градиентов через модель. Как использовать для transfer learning