

Модуль 2

Разработка нейросетей для систем машинного зрения

Формы: решение кейсов

Диагностические инструменты

Пример кейса: разработка и обучение свёрточной нейронной сети (CNN) для классификации изображений

Вам необходимо разработать (можно взять за основу небольшие рабочие архитектуры, такие как LeNet, AlexNet) свёрточную сеть и обучить ее для классификации изображений на выбранном наборе данных с небольшими размерами изображений (например, MNIST, FashionMNIST, CIFAR-10 и подобные).

Задачи:

1. Разработайте архитектуру свёрточной нейронной сети для задачи классификации изображений. Опишите ваши соображения при выборе слоев (тип, количество, последовательность) и гиперпараметров (размер фильтров, их количества, функции активации и другие).
2. Подготовьте все необходимые элементы для процесса обучения из инструментария pytorch: загрузчики данных, необходимую функцию потерь. Загрузить данные для обучения (предобработка данных: перевод в тензорный вид, при необходимости — изменение размеров изображений — `resize`).
3. Обучите модель на подготовленном наборе данных.
4. Оцените качество обучения модели на тестовой выборке (по функции потерь — `validation loss` и точности — `accuracy`).
5. Реализуйте один (на выбор) из следующих методов борьбы с переобучением:
 - слой `Dropout` (если не было в разработанной архитектуре)
 - слой `BatchNorm` (если не было в разработанной архитектуре)
 - регуляризация функции потерь,
 - техники `data augmentation` (дополнение данных — модификация копий изображений в обучающей выборке)
6. Подготовьте краткий отчет (в свободной форме) с описанием выполненных шагов, архитектуры модели, процесса обучения, полученных результатов. Сделайте краткие выводы. Приложите к отчету визуализации (графики метрик обучения — `loss`, `accuracy`, примеры предсказаний сети).

Критерии оценивания

10 баллов:

- Архитектура разработана полностью с нуля, порядок слоев корректен, даны развернутые комментарии по выбору слоев, их параметров.
- Данные полностью подготовлены для подачи в модель, прямой проход осуществляется без ошибок (код запускается). Корректно реализован один или более тип аугментации данных (`transforms`), комментарии по выбранным типам аугментации.
- Есть эффект обучения т.е. функция потерь на проверочной выборке (`validation loss`) снижается, точность (`accuracy`) растет (графики). Представлены примеры предсказаний сети.
- Отчет полный и подробный, шаги и результаты работы описаны и подкреплены иллюстрациями. Описаны кратко выводы.

8-9 баллов:

- Архитектура разработана полностью — как модификация нескольких компонентов известной хорошо работающей архитектуры (AlexNet, LeNet и др) порядок слоев корректен, краткие комментарии по выбору слоев, их параметров.

- Данные полностью подготовлены для подачи в модель, прямой проход осуществляется без ошибок (код запускается). Корректно реализован хотя бы один тип аугментации данных (transforms), комментарием по выбору данного типа.
- Есть эффект обучения т.е. функция потерь на проверочной выборке (validation loss) снижается, точность (ассигасу) растет (графики).
- Отчет в целом полный, не все шаги и результаты работы описаны и подкреплены иллюстрациями. Описаны кратко выводы.

6-7 баллов:

- Архитектура разработана полностью — как модификация какого-либо компонента известной хорошо работающей архитектуры (AlexNet, LeNet и др) порядок слоев корректен.
- Данные полностью подготовлены для подачи в модель, прямой проход осуществляется без ошибок (код запускается).
- Эффект обучения слабый или отсутствует: (функция потерь) на проверочной выборке (validation loss) НЕ снижается или увеличивается, но обучение нормально идет на обучающей выборке (train loss падает, train ассигасу растет).
- В отчете не хватает некоторых разделов, не все шаги и результаты работы описаны, отсутствуют иллюстрации по существу. Описаны кратко выводы.

4-5 баллов:

- Архитектура разработана частично, порядок слоев корректен.
- Данные полностью подготовлены для подачи в модель, прямой проход осуществляется без ошибок (код запускается, проверено на сети другой архитектуры для той же задачи).
- Эффект обучения слабый или отсутствует: (функция потерь) на проверочной выборке (validation loss) НЕ снижается или увеличивается, но обучение на обучающей выборке не идет (train loss НЕ падает, train ассигасу НЕ растет).
- Отчет поверхностный. Описаны кратко выводы.

2-3 балла:

- Архитектура НЕ разработана, реализован отдельный слой. Краткие комментарии по данному слою, его параметрам.
- Данные подготовлены для прохождения через слой, прямой проход осуществляется без ошибок (код запускается). Краткие комментарии полученного результата.
- Отчет содержит краткую характеристику выбранного слоя, характеристику полученных результатов и предложения для дальнейшей разработки (где и на каком месте этот слой мог бы эффективно работать и т.д.).

Общая оценка: среднее по 4 указанным пунктам:

- архитектура,
- данные,
- обучение (в последней градации отсутствует),
- отчет

Шкала оценивания: зачтено (средняя 3,5 и больше) / не зачтено

ВАЖНО: никто на данном этапе не будет оценивать насколько хорошо вы обучили сеть!

Make it work → Make it right → Make it fast → Make more cool features! **Мы почти полностью на первом этапе.**

Предполагается возможность выполнения задания без доступа к GPU (ускорителю).