



Министерство образования и науки Российской
Федерации

Федеральное государственное автономное
образовательное учреждение высшего образования
«Уральский федеральный университет имени
первого Президента России Б. Н. Ельцина»

Основы анализа данных с Python и SQL

Отчет

По лабораторной работе №5

Классификация изображений сверточными моделями

Студент: Манакин Андрей

г. Екатеринбург

2024 г.

Цель работы: Реализовать глубокую модель для классификации изображений из CIFAR-10

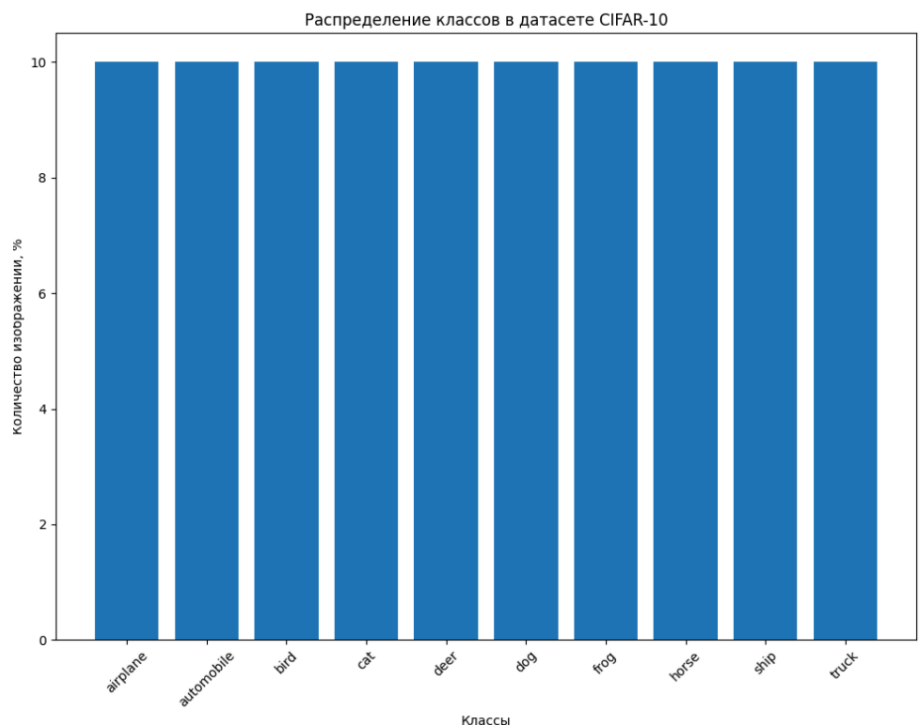
Список задач:

- Запустить Baseline-модель и убедиться в том, что все работает
- Оценить качество модели
- Создать собственную модель
- Обучить свою модель
- Достигнуть необходимый уровень Accuracy для своей модели

Ход работы:

Лабораторную работу я выполнял в среде Google Colab, т.к она предоставляет мощную GPU T4, способную обучать модели в разы быстрее моего процессора и видеокарты.

Первым делом мы загрузили данные, и убедились в том, что датасет изображений CIFAR-10 сбалансированный, и количество элементов в нём равномерно распределено между 10 классами

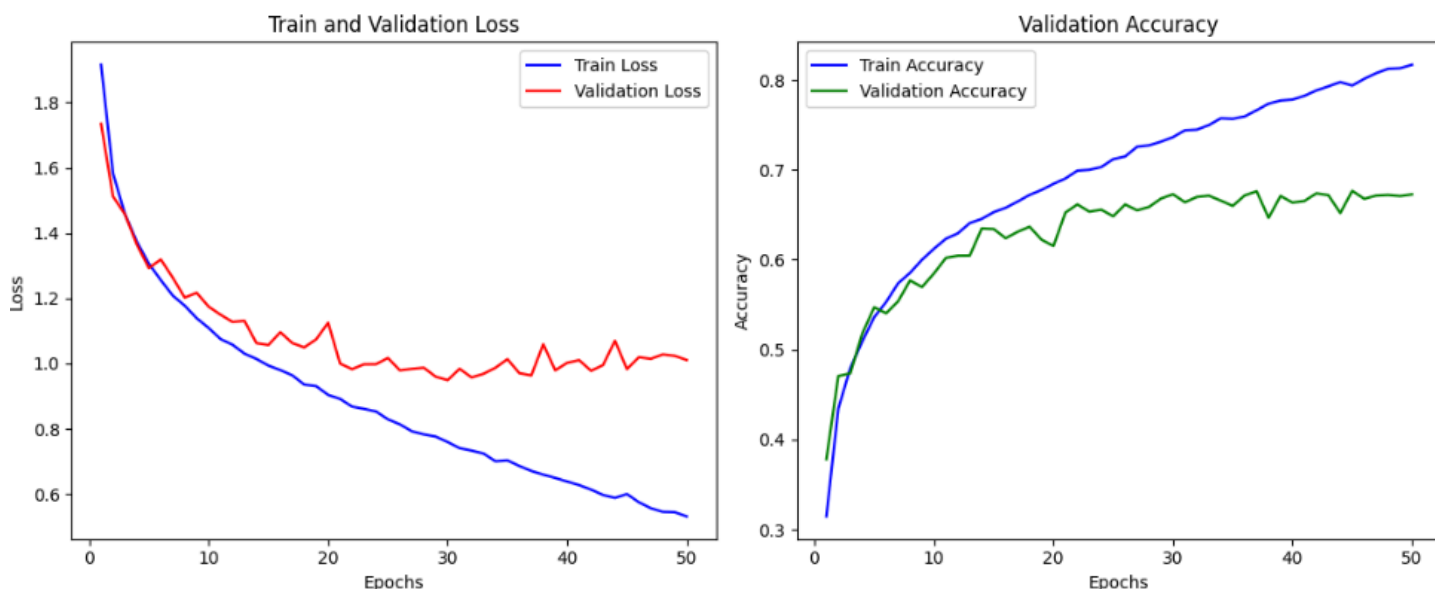


Далее были написаны функции для обучения и тестирования нашей модели: `train()` и `test()` соответственно.

После написания этих функций набор данных для обучения был разделен на 2 части: Train и Validate: Во время обучения, модель будет обучаться на данных из Train, а затем так-же, во время обучения проверять свои успехи на данных из Val.

Далее я установил гиперпараметры (лернинг рейт, кол-во эпох и размер батча), выбрал функцию оптимизации (Adam) и loss-функцию (CrossEntropyLoss) и создал загрузчики данных для тренировки, валидации и тестирования.

Далее я запустил обучение модели на 50 эпох и получил следующие графики:



Проанализировав графики я установил, что модель начинает переобучаться +- на 25 эпохе, в момент, когда Ассигасу валидации перестаёт расти, тогда как Ассигасу тренировки продолжает тянуться вверх

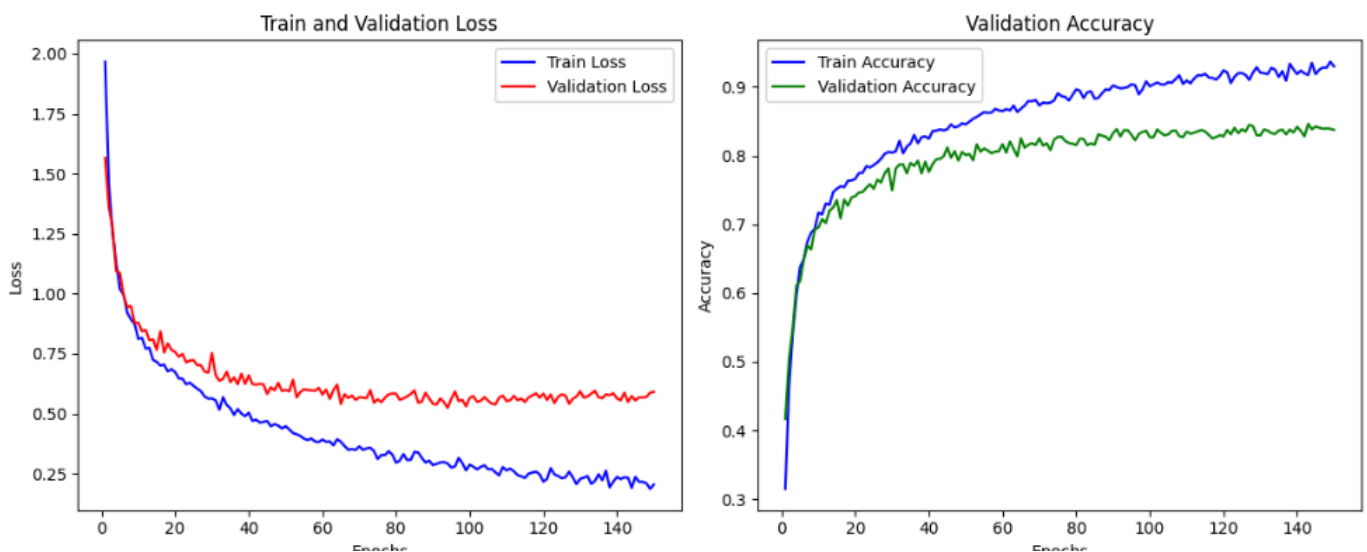
В конце я оценил тестовую метрику на тестовых данных и получил следующие результаты:

```
test_acc, test_loss = test(model, test_loader, criterion, device)
print('Test Loss: {:.4f}\nTest Accuracy: {:.4f}'.format(test_loss, test_acc))
```

```
Test Loss: 1.0276
Test Accuracy: 0.6730
```

Во второй части работы я усложнил модель, добавив в неё несколько фильтров в сверточном слое, дропаут, для предотвращения переобучения, слой аугментации данных, и слои нормализации батча для стабилизации обучения.

Все эти улучшения позволили мне увеличить Accuracy модели:



```
test_acc, test_loss = test(model, test_loader, criterion, device)
print('Test Loss: {:.4f}\nTest Accuracy: {:.4f}'.format(test_loss, test_acc))
```

```
Test Loss: 0.4418
Test Accuracy: 0.8731
```

Ответы на вопросы из ноутбука:

Сделайте вывод о сбалансированности датасета:

Датасет можно назвать сбалансированным, потому что в него входит 10 классов, по 6 тыс. изображений в каждом и они распределены равномерно, исходя из гистограммы выше

Исходя из этого, какие метрики будем применять для оценки качества модели
Т.к датасет имеет сбалансированное распределение, для оценки качества модели можно использовать такие стандартные метрики как: Accuracy и Cross-Entropy Loss

Сделайте разделение на train и val, зачем так делать?

Разделение на Train и Val необходимо для того, чтобы оценить как модель работает с незнакомыми ей данными, на трейне модель обучается, затем проверяется на валидейта, если аккуратность для трейна растёт, а для валидейта стоит на месте, значит модель переобучается

Создайте Dataloader'ы и настройте гипер параметры, оптимизатор и функцию потерь, какой оптимизатор будете использовать и почему, а какую loss-функцию?

В качестве оптимизатора я выберу Adam, так как он эффективно справляется с регуляризацией. loss-функцию буду использовать CrossEntropyLoss(), так как она хорошо подходит для задач многоклассовой классификации

Сделайте выводы о качестве обучения

Примерно с 25-ой эпохи точность валидации перестала расти, а местами начала уменьшаться, в то время, как точность на тренируемой выборке продолжает расти, это говорит о том, что модель переобучается и с этим что-то нужно делать.

Новая модель:

Для своей новой модели я добавил слой аугментации, увеличил количество фильтров в сверточных слоях для извлечения бОльшего кол-во признаков из

изображений, добавил дропауты, для предотвращения переобучения, а также слои нормализации батча, стабилизирующие и ускоряющие процесс обучения, так же я увеличил количество эпох до 150, что позволило моей модели получить необходимую величину Ассигасы для тестового набора данных

Вывод

В ходе выполнения лабораторной работы была реализована глубокая модель для классификации изображений из набора данных CIFAR-10. В процессе работы были выполнены следующие задачи:

1. Запуск и оценка базовой модели.
2. Создание и обучение собственной модели.

Анализ результатов показал, что базовая модель начинает переобучаться примерно на 25-й эпохе. Для улучшения модели были добавлены слои аугментации данных, увеличено количество фильтров в сверточных слоях, добавлены слои дропаута и нормализации батча. Эти улучшения позволили повысить точность модели.

В результате, улучшенная модель достигла необходимого уровня точности на тестовом наборе данных, что подтверждает эффективность примененных методов и подходов.