

Министерство науки высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)

Факультет цифровых трансформаций

Образовательная программа Искусственный интеллект в промышленности

Направление подготовки (специальность) 09.04.02 - Информационные системы и технологии

О Т Ч Ё Т

Лабораторная работа №2

Тема задания: Применение инструментов для логирования параметров при проведении экспериментов с обучением моделей ИИ

Обучающихся *Штыкина О.С, Кузьмина А. С., группа J4151*

Преподаватель: *Старобыховская А.А.*

Санкт-Петербург
2024

СОДЕРЖАНИЕ

ЗАДАНИЕ	3
ОСНОВНЫЕ ЭТАПЫ	4
1. Анализ исходных данных	4
2. Постановка задачи	4
3. Реализация базового алгоритма	4
4. Проведение экспериментов	7
ВЫВОДЫ	10

ЗАДАНИЕ

Цель задания: Сформировать гипотезы по улучшению качества обучения модели. Освоить инструменты для логирования и визуализации обучения.

Описание предметной области: анализ параметров и метрик качества при обучении модели ИИ.

Исходные данные: датасет CIFAR-10. Ссылка: <https://www.geeksforgeeks.org/cifar10-dataset-in-keras-tensorflow-for-object-recognition/>

ОСНОВНЫЕ ЭТАПЫ

1. Анализ исходных данных

В качестве исходных данных был взят датасет CIFAR-10. Он содержит 60 000 цветных RGB изображений размером 32x32(1024) пикселя в 10 различных классах, таких как самолеты, автомобили, птицы, кошки, олени, собаки, лягушки, лошади, корабли и грузовики. Обучающий набор содержит 50 000 изображений, а тестовый 10 000. Изображения разделены на 10 классов, по 5000 изображений в каждом классе. Тестовый набор содержит по 1000 изображений на класс.



Рисунок 1 — Изображения из датасета

2. Постановка задачи

В качестве задачи классификации, модели необходимо было определить класс изображения.

(Метка) -> (Класс)

0 -> Самолет

1 -> Автомобиль

2 -> Птица

3 -> Кот

4 -> Олень

- 5 -> Собака
- 6 -> Лягушка
- 7 -> Лошадь
- 8 -> Корабль
- 9 -> Грузовик

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170498071/170498071 [=====] - 2s 0us/step



Рисунок 2 — Изображения с классификацией

3. Реализация базового алгоритма

Для решения поставленной задачи было решено реализовать простую свёрточную нейронную сеть.

Сама сеть была инициализирована со следующим набором гиперпараметров:

- Размер батча batch – 64
- Количество эпох epochs – 5

Значения метрик следующие:

- Average loss – 0,8061
- Accuracy – 72%

Для логирования параметров при обучении и удобного вывода и сравнения результатов был выбран фреймворк ClearML. Данный фреймворк

имеет ряд преимуществ по сравнению со всеми остальными аналогами, а также в нём есть возможность выполнять практически полный цикл машинного обучения. В нашей лабораторной работе мы воспользуемся такими возможностями ClearML, как логирование, сравнение экспериментов, оркестрация.























								
Open Source	✓ Apache	✓ Apache	✓ Apache	✓ Apache	✗	✗	✓ MIT	✓ MIT
Platform & language agnostic	✓	✓	✗	✗	✗	✗	✓	✓
Experiment Data Access Local / Cloud	 		 		 	 	 	 
Remote server set up for the user	✗	✗	✓	✗	✓	✓	✓	✓
Custom Visualizations	✓	✓	✓	✓	✓	✓	✗	✗
Scalable for a large number of experiments	✓	✗	✓	✗	✓	✓	✓	✓
Auto-logging	✓	✓	✓	✗	✗	✓	✓	✓
Collaboration features	✓	✗	✓	✗	✓	✓	✓	✓
Team-based access	✗	✓	✓	✗	✓	✗	✓	✓

Рисунок 3 – Сравнение ClearML и аналогов

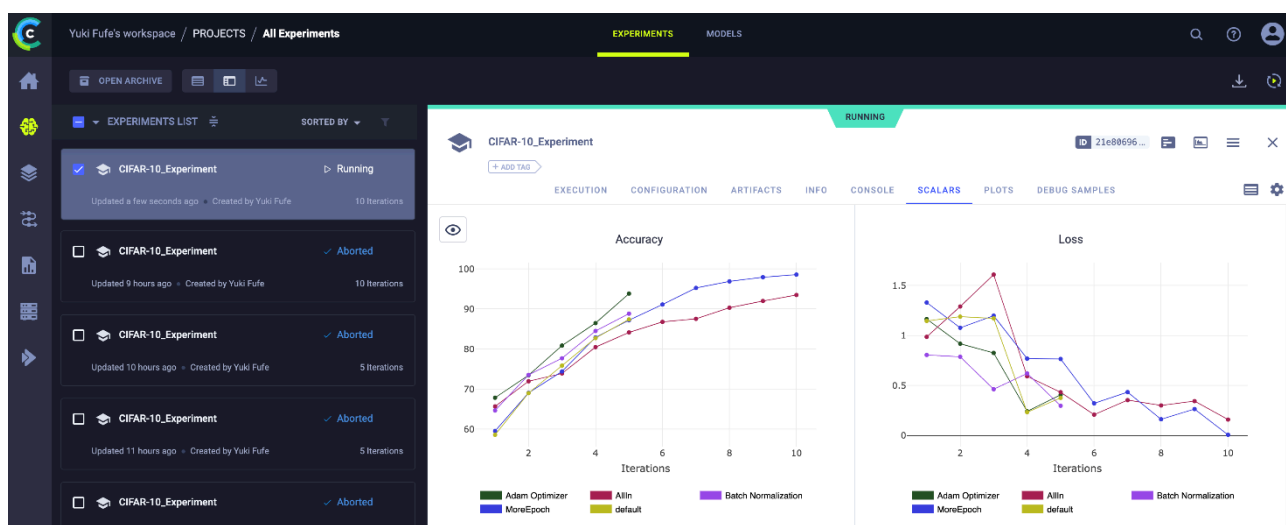
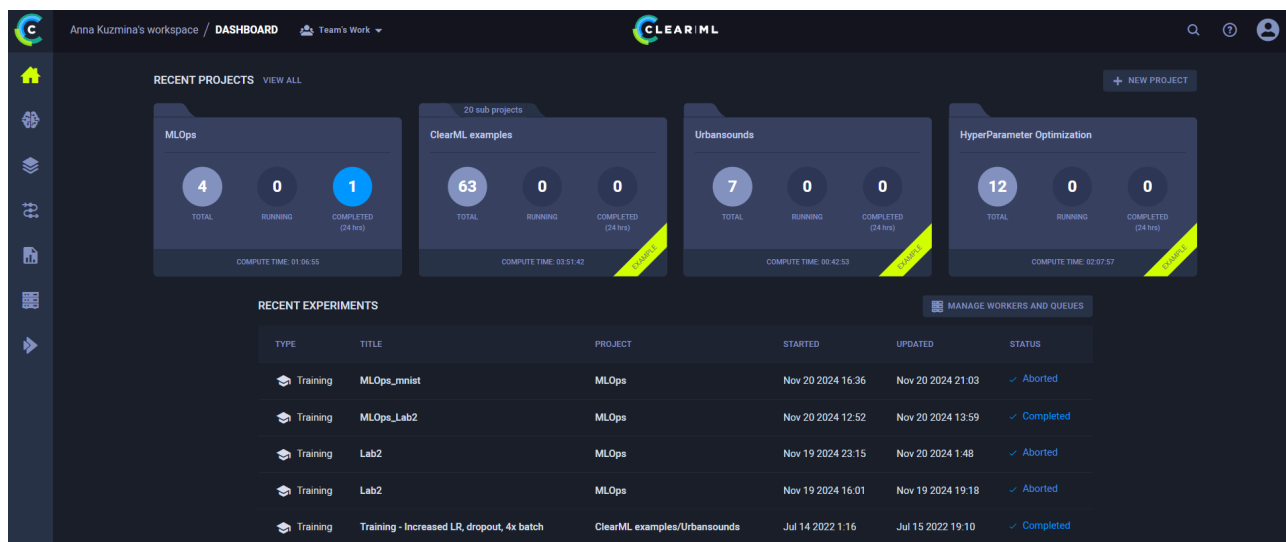
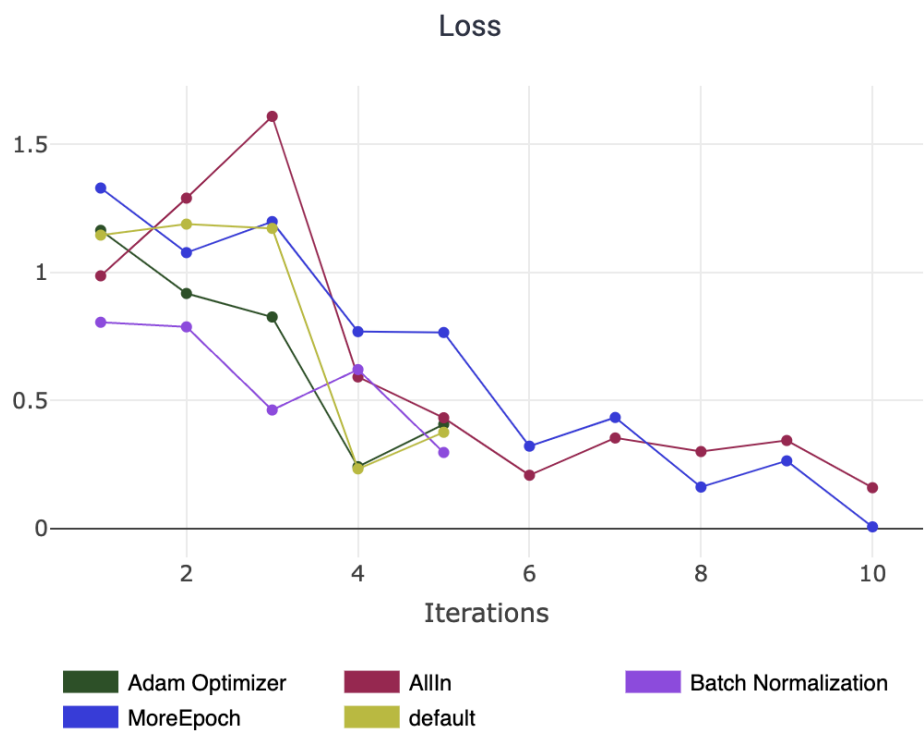
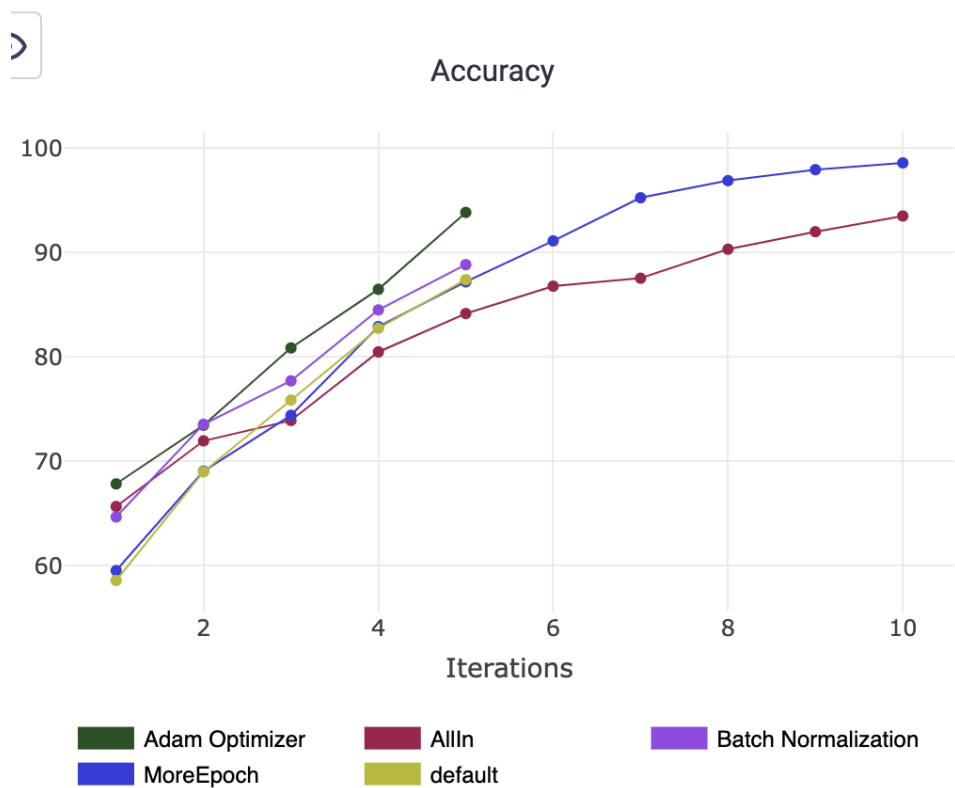
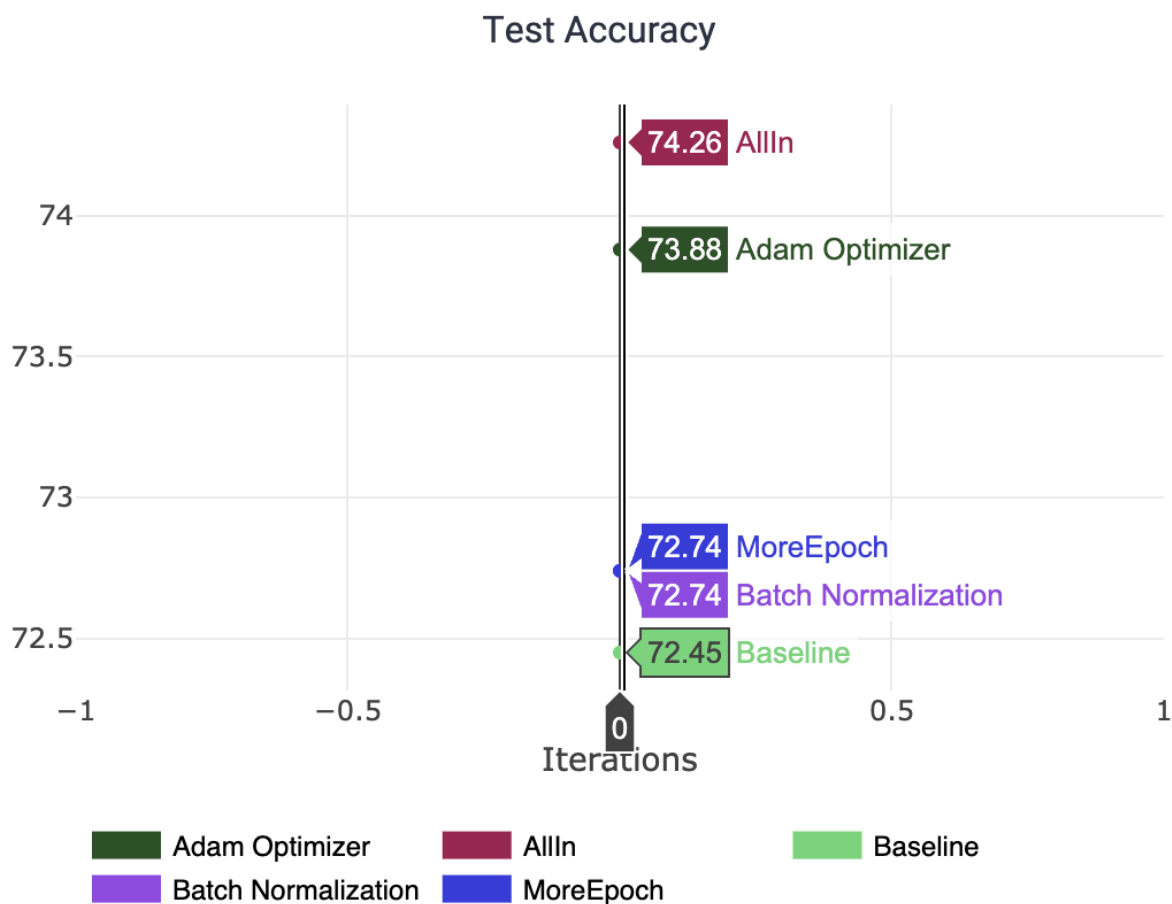


Рисунок 4 – Интерфейс фреймворка

После обучения модели было выявлено, что модель недообучается, т.к. график loss (default линия на графиках) не приходит в стабильное положение.





4. Проведение экспериментов

Для улучшения метрик классификации изображений были сформулированы три гипотезы, пытающиеся решить основные проблемы текущей модели:

I. Недообучение модели, увеличение количества эпох: т.к. модель слишком мало раз проходила по всему датасету, возникла такая проблема как недообучение. Данную проблему можно решить увеличением количества эпох. В базовой модели было 5 эпох, в эксперименте количество эпох увеличено до 10.

Эксперимент по увеличению количества эпох показал такие результаты:
Train set: На 10 эпохе Loss достиг значения 0,005, accuracy 98,57%
Test set: Average Loss достиг значения 1,3022, accuracy 73%

Т.к. точность на тренировочных данных очень высокая, а на тестовых значительно ниже, можно сделать вывод о переобучении модели.

II. Невысокая точность, применение Batch Normalization: Batch Normalization нормализует активации внутри сети, что может привести к улучшению обобщающей способности модели, что приведёт к повышению точности.

Эксперимент по Batch Normalization показал такие результаты:
Train set: На 5ой эпохе Loss достиг значения 0,296, accuracy 88,83%
Test set: Average Loss достиг значения 0,7654, accuracy 75%

По сравнению с увеличением эпох Batch Normalization показал лучше результаты и не привёл к переобучению. Также можно заметить уменьшение колебаний loss, что говорит о лучшей обобщающей способности модели.

III. Невысокая точность, применение Adam Optimizer: В базовой модели используется SGD алгоритм оптимизации (стохастический градиентный спуск). Наша гипотеза заключается в том, чтобы использовать Adam Optimizer вместо SGD, т.к. он является более продвинутым.

Эксперимент с применением Adam Optimizer показал такие результаты:

Train set: На 5ой эпохе Loss достиг значения 0,4, accuracy 93,83%
Test set: Average Loss достиг значения 0,8415, accuracy 74%

Применение всех гипотез в одном эксперименте показало следующие результаты:

Train set: На 10ой эпохе Loss достиг значения 0.158, а accuracy 93.49%
Test set: Average loss достиг значения 0,9429, accuracy: 74%

ВЫВОДЫ

Анализируя базовую модель были выявлены основные проблемы и были предложены гипотезы по их решению.

1. **Увеличение количества эпох:** данная гипотеза привела к переобучению модели, что является плохим показателем.
2. **Применение Batch Normalization:** в целом данная гипотеза помогла улучшить точность, но незначительно. Это может быть связано с тем, что CIFAR-10 - относительно простой датасет, и BN не так критичен, как для более сложных задач.
3. **Применение Adam Optimizer:** Использование Adam привело к наилучшей точности на тренировочном наборе и неплохому результату на тестовом. Adam часто сходится быстрее SGD, что может объяснить более высокую точность за то же количество эпох.
4. Применение всех гипотез в одном эксперименте не привело к значительному улучшению модели. Вероятно, при применении Batch Normalization и Adam Optimizer без увеличения количества эпох (либо при небольшом увеличении) можно добиться результатов лучше.