
ДЗ №3 (Сравнение MLP и CNN на MNIST):

Домашнее задание: Сравнение архитектур MLP и CNN на задаче классификации (MNIST)

Цель:

Исследовать эффективность MLP (Multi-Layer Perceptron) и CNN (Convolutional Neural Network) на датасете MNIST. Проанализировать различия в производительности, сложности моделей и интерпретируемости результатов.

Задание:

1. Реализация моделей:

- **MLP:**
 - Создайте модель с минимум 2 скрытыми полносвязными слоями (например, $512 \rightarrow 256$ нейронов).
 - Используйте активации ReLU и Dropout для регуляризации.
- **CNN:**
 - Создайте модель с минимум 2 свёрточными слоями (например, $3 \times 3 \rightarrow 3 \times 3$) и MaxPooling.
 - Добавьте полносвязные слои после свёрточных.
- Для обеих моделей:
 - Используйте оптимизатор Adam и кросс-энтропийную функцию потерь.
 - Обучайте на 10 эпохах с одинаковым размером батча (например, 64).

2. Метрики качества:

- Точность (Accuracy) на тестовом наборе.
- F1-score (среднее по классам).
- Матрица ошибок (Confusion Matrix).
- Время обучения модели.
- Количество обучаемых параметров в каждой архитектуре.

3. Требуемые результаты:

- CNN должна достичь точности **не ниже 98%** на тестовых данных.

- MLP должен показать точность **не ниже 95%**.
-

Вопросы для анализа (5-7 предложений):

1. Архитектурные компромиссы:

- На примере ваших моделей объясните:
 - Почему CNN достигла более высокой точности, **несмотря на меньшее количество параметров?**
 - Как структура MLP (полносвязные слои) влияет на его способность улавливать пространственные зависимости в пикселях?
- *Задание:* Визуализируйте 3 примера изображений MNIST, где CNN справилась, а MLP ошибся. Какие визуальные особенности (например, наклон цифры, толщина линий) могли привести к ошибке MLP?

2. Интерпретация ошибок:

- Проанализируйте матрицу ошибок для CNN:
 - Какие два класса цифр чаще всего путает модель (например, 4/9, 3/8)?
 - Предложите **конкретную модификацию архитектуры** (добавить слой, изменить размер фильтра и т.д.), которая может снизить эти ошибки. Обоснуйте выбор.

3. Эксперимент с регуляризацией:

- Уберите Dropout из MLP и обучите модель заново. Сравните:
 - Как изменилась точность на тесте?
 - Появились ли признаки переобучения (разрыв между ассигасу на трейне и тесте)?
- *Почему* CNN менее чувствительна к отсутствию Dropout, чем MLP? Свяжите ответ с природой свёрточных операций.

4. На размашление:

- Какие **три ключевых урока** из этого эксперимента вы бы взяли для классификации более сложных изображений (например, цветных снимков животных)?
 - В каком реальном сценарии MLP для изображений всё ещё может быть предпочтительнее CNN? Приведите пример и аргументы.
-

Рекомендации:

- Используйте библиотеки: `PyTorch`, `Matplotlib` для визуализаций.
- Для анализа матрицы ошибок воспользуйтесь `sklearn.metrics.confusion_matrix`.

- Экспериментируйте с гиперпараметрами (например, размер батча, learning rate), чтобы улучшить метрики.
-

Система оценивания:

Основная часть (7 баллов):

1. Реализация моделей (2 балла):

- Корректность архитектур MLP и CNN (слои, активации, регуляризация).
- Использование оптимизатора Adam и кросс-энтропийной функции потерь.
- Стабильное обучение на 10 эпохах (для полного обучения можете использовать больше).

2. Метрики качества (2 балла):

- Достижение точности $\geq 95\%$ для MLP и $\geq 98\%$ для CNN.
- Расчет F1-score, матрицы ошибок (confussion matrix), времени обучения.

3. Анализ результатов (3 балла):

- Ответы на обязательные вопросы с примерами из эксперимента (визуализации ошибок, сравнение параметров).
 - Интерпретация матриц ошибок и графиков обучения.
-

Критерии строгости:

- -1 балл за каждую недостигнутую целевую метрику (если accuracy MLP $< 95\%$ или CNN $< 98\%$).
-

Дополнительная часть (+4 балла):

Выполните 2 задания из списка (по 2 балла каждое):

4. Оптимизация для embedded-устройств:

- Рассчитайте объем памяти для MLP и CNN.
- Упростите CNN (уменьшите фильтры/слои) и проверьте, как падает точность.
- Нарисуйте схему изменённой CNN.

5. Эксперименты с гиперпараметрами:

- Измените глубину MLP. Сравните accuracy и переобучение.
- Уберите MaxPooling в CNN.

6. Анализ ошибок через градиенты:

- Визуализируйте градиенты для ошибочных предсказаний MLP и CNN (например, с помощью Grad-CAM для CNN).
- Сравните, какие части изображения "важны" для каждой модели.

7. Перенос на другой датасет:

- Протестируйте CNN на Fashion-MNIST.
 - Предложите 3 изменения в архитектуре для улучшения результатов.
-

Критерии строгости:

- **+2 балла** за каждое доп. задание:
 - Требуются **численные результаты** (например, точность до/после изменений).
 - **Визуализации** (схемы, heatmap градиентов, примеры изображений).
 - **Выводы** с обоснованием причин наблюдаемых эффектов.
 - **-1 балл** за поверхностный анализ (например, отсутствие сравнения "было/стало").
-

Срок сдачи 2 недели. Удачи!