

Московский государственный университет имени М. В.
Ломоносова

Факультет вычислительной математики и кибернетики

Отчет

по заданию на тему:

“Разработка и обучение нейронной сети для классификации
изображений”

Выполнил:
Студент 2 курса, 215 группы
Клепиков Г. Д.

Москва 2025

1. Введение

В данной задаче была поставлена цель бинарной классификации изображений человеческих лиц по признаку нахождения на их лице солнцезащитных очков с помощью нейронной сети. Метрикой качества является Accuracy.

2. Подготовка данных

- Для обучения модели использовались 378 изображений формата 30x32, grayscale, jpg. Соответственно был использован файл train_tags.csv для сопоставления фотографиям класса изображения: sunglasses/open.
- Нормализация: для стабилизации работы модели во время обучения была выполнена нормализация входных изображений / 255.0

3. Описание архитектуры нейронной сети

Разработанная модель представляет собой сверточную нейронную сеть для бинарной классификации изображений. Основные компоненты архитектуры:

- Слой аугментации данных (data_augmentation):
 - Случайное горизонтальное отражение (RandomFlip).
 - Случайный поворот изображения (RandomRotation).
 - Случайное увеличение (RandomZoom).
- Сверточные и пулинговые слои:
 - Conv2D (32 фильтра, размер ядра 3x3, ReLU-активация) – начальное извлечение признаков.
 - MaxPooling2D (размер окна 2x2) – снижение размерности карт признаков.
 - Conv2D (64 фильтра, размер ядра 3x3, ReLU-активация) – более сложные признаки.
 - MaxPooling2D (размер окна 2x2) – дальнейшее уменьшение размерности.
- Полносвязные слои:
 - Flatten – преобразование двумерных карт признаков в одномерный вектор.

- Dense (64 нейрона, ReLU-активация) – извлечение высокоуровневых признаков.
 - Dropout (50%) – предотвращение переобучения.
 - Dense (1 нейрон, сигмоидная активация) – выходной слой для бинарной классификации.
 - Оптимизатор и функция потерь:
 - Оптимизатор: Adam.
 - Функция потерь: binary_crossentropy.
 - Метрика: accuracy.
4. Что было сделано/исследовано для получения конечного результата
- Загружены и предобработаны данные: изображения были приведены к черно-белому формату, изменены до фиксированного размера (30x32) и нормализованы.
 - Реализована аугментация данных, чтобы улучшить способность модели к обобщению.
 - Выбрана архитектура с двумя сверточными слоями и пулингом для обработки изображений малого размера.
 - Добавлен Dropout, чтобы уменьшить переобучение.
 - Модель обучалась 20 эпох с разбиением на обучающую и валидационную выборку (80/20).
 - Проведено тестирование на отдельном наборе изображений, предсказания сохранены в файл Gleb_Klepikov.csv.
5. Почему был выбран именно этот вариант?
- Компактность модели
 - Из-за небольшого размера изображений и простоты задачи использовалась относительно легкая архитектура с двумя сверточными слоями.
 - Добавление большего числа слоев не дало бы значительного прироста качества, но усложнило бы обучение.
 - Бинарная классификация
 - Использование сигмоидной активации и binary_crossentropy позволяет эффективно обучать модель для задачи sunglasses / open.

- Предотвращение переобучения
 - Применение Dropout и data_augmentation позволяет добиться более высокой обобщающей способности модели.
- Скорость обучения
 - Использование оптимизатора Adam и разумного числа эпох (20) позволяет достичь приемлемой точности без долгого обучения.
- В итоге получена нейросеть, эффективно классифицирующая изображения с очками и без них, демонстрирующая хорошую способность к обобщению при разумных вычислительных затратах.

6. Результаты обучения

Валидационная точность сначала колебалась, но к 20-й эпохе достигла 82.89%, что сопоставимо с точностью обучения (80.62%). Функция потерь уменьшалась равномерно и на обучающем, и на валидационном наборе, что говорит об отсутствии переобучения.

7. Заключение

Разработана и обучена сверточная нейронная сеть для бинарной классификации изображений. Модель продемонстрировала устойчивый рост точности, достигнув ~83% на валидационном наборе. Анализ обучения показал отсутствие переобучения, что подтверждается схожими значениями функции потерь и точности на обучающей и валидационной выборках.

8. Вспомогательные материалы

<https://stepik.org/course/50352/syllabus?auth=login>

<https://konfuzio.com/en/cv2/>

https://coderzcolumn.com/tutorials/artificial-intelligence/create-simple-neural-networks-in-python-using-keras#google_vignette