Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 2

Версионирование данных и моделей
Тема

Преподаватель _____ Е. О. Пересунько
Подпись, дата Инициалы, Фамилия

Студент КИ19-17/1Б, №031939174 _____ А. К. Никитин
Номер группы, зачетной книжки Подпись, дата Инициалы, Фамилия

Красноярск 2022

1 Цель

Знакомство с инструментами организации процесса MLOps, получение практических навыков по работе с версионированием экспериментов и автоматизации пайплайнов машинного обучения.

2 Задачи

1. Создать удаленный репозиторий для проекта.
2. Разработать решение задачи в соответствии с вариантом.
   а. Построить автоматический ML-пайплайн, состоящий из всех необходимых шагов для решения задачи (предобработка данных, обучение моделей, оценка качества и т.д.).
   б. Провести серию экспериментов, используя как минимум три различные модели машинного обучения для решения поставленной задачи.
3. Выполнить аугментацию, сгенерировав дополнительные данные с помощью различных преобразований, и сохранить новый датасет.
4. Повторить пункт 2, используя все данные (исходные и аугментированные).
5. В отчете привести сравнение результатов, получаемых при различных условиях экспериментов на разных наборах данных (с аугментацией и без).

3 Ход работы

3.1 Инициализация проекта

В качестве удаленного репозитория был выбран Gitlab, в качестве программного обеспечения, проводящего версионирование данных для машинного обучения был выбран инструмент dvc.

Был создан каталог и проведена инициализация git и dvc проекта.

Был выбран датасет “A Large Scale Fish Dataset” для предстоящего обучения модели (режим доступа – https://www.kaggle.com/datasets/crowww/a-large-scale-fish-dataset?select=Fish_Dataset).

Датасет был локально размещен на компьютере в корне проекта в каталоге data и добавлен в .gitignore.

3.2 Пайплайны

3.2.1 Предобработка

Набор изображений был перемещен в папку prepared и разбит на выборки train, valid и test в соотношении 0.7 : 0.15 : 0.15.

Также размер исходной выборки был сокращен с 9000 изображений до 1000 тысячи в целях оптимизации работы dvc и обучения нейросети.

3.2.2 Обучение и оценка моделей

Задача машинного обучения – небинарная классификация изображений, вследствие чего логичным решением является использование сверточных нейронных сетей с softmax выходным слоем и категориальной кросс-энтропией в качестве loss-функции.

Используемым фреймворком для конструирования и обучения нейросети является keras.

Обучение проводится на 6 эпохах и с размером батча равным 50.

Структура первой итерации нейронной сети представлена на рисунке 1.

```
model = keras.Sequential([
    tf.keras.layers.Conv2D(30, (3, 3), padding='same', activation='relu', input_shape=IMAGE_SIZE + (1,)),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(60, activation='relu'),
    tf.keras.layers.Dense(9, activation='softmax'),
])
```

Рисунок 1 – Структура первой нейронной сети

Метрики первой нейронной сети представлены на рисунке 2.

Path	test_acc	train_acc	validation_acc
metrics\train.json	-	0.82232	0.75776
metrics\test.json	0.45223	0.82232	-

Рисунок 2 – Метрики первой нейронной сети

При очевидно неудовлетворительном результате было принято решение усложнить структуру модели. Полученный результат представлен на рисунке 3.

```
model = keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=IMAGE_SIZE + (1,)),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D((2, 2), strides=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(200, activation='relu'),
    tf.keras.layers.Dense(9, activation='softmax'),
])
```

Рисунок 3 – Структура второй нейронной сети

Метрики второй нейронной сети представлены на рисунке 4.

Path	test_acc	train_acc	validation_acc
metrics\train.json	-	0.99285	0.95918
metrics\test.json	0.81699	0.99285	-

Рисунок 4 – Процессы хранилища данных

Результат все еще тяжело назвать удовлетворительным, поэтому было принято решение подавать на вход модели трехканальное изображение заместо одноканального. Получившийся результат представлен на рисунке 5.

```
Path                test_acc    train_acc    validation_acc
metrics\train.json  -           0.99411      0.96226
metrics\test.json   0.88194     0.98701      -
```

Рисунок 5 – Метрики третьей нейронной сети

Налицо явное переобучение. Из этого факта следует, что необходимо расширить датасет и разнообразить данные. Для этих целей была использована аугментация размытия. Метрики второй и первой модели на аугментированных данных не включены в отчет, чтобы сэкономить место.

Получившиеся после применения аугментации метрики представлены на рисунке 6.

```
Path                test_acc    train_acc    validation_acc
metrics\train.json  -           1.0          0.96541
metrics\test.json   0.93789     1.0          -
```

Рисунок 6 – Метрики аугментированной третьей нейронной сети

4 Вывод

В результате работы были изучены и применены инструменты MLOps для автоматизации ведения работы по улучшению и подбору оптимальной модели; была применена технология пайплайнов для облегчения версионирования ML-моделей.