

Московский Государственный Технический Университет имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



Отчёт по лабораторной работе № 3 по дисциплине «Проектирование интеллектуальных систем»

Исполнитель:

Саврасов П.А.

Группа ИУ5-24М

«__» _____ 2021 г.

Преподаватель:

Терехов В.И.

«__» _____ 2021 г.

Москва, 2021 г.

1. Цель работы

Познакомиться с примерами работы со свёрточными нейронными сетями (CNN) на примере набора CIFAR10.

2. Свёрточная нейронная сеть, обученная на наборе данных CIFAR10

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.python.keras.datasets import cifar10

import seaborn as sns
```

```
(xTrain, yTrain), (xTest, yTest) = cifar10.load_data()
print(xTrain.shape, yTrain.shape)
print(xTest.shape, yTest.shape)
```

```
(50000, 32, 32, 3) (50000, 1)
(10000, 32, 32, 3) (10000, 1)
```

Формат датасета имеет следующий формат:

50000 образцов 32 на 32 пикселя, каждый пиксель зашифрован RGB кодом

Для применения его как обучающей выборки необходимо сначала отмасштабировать данные

```
xTrain = xTrain.astype('float32') / 255
xTest = xTest.astype('float32') / 255
```

Создадим нейросеть со следующей архитектурой:

- Входной слой: 32 x 32 x 3
- Свёрточный слой 1
- Свёрточный слой 2
- Свёрточный слой 3
- Полносвязный слой: 10

```
model = keras.Sequential([
    layers.InputLayer(input_shape=(32, 32, 3)),
    layers.Conv2D(32, 3, activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(64, 3, activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(128, 3, activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Flatten(),
    layers.Dense(512, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(10)
])
```

```
model.compile(
    loss = keras.losses.SparseCategoricalCrossentropy(from_logits = True),
    optimizer = keras.optimizers.Adam(learning_rate = 0.001),
    metrics = ['accuracy']
)
```

```
history = model.fit(xTrain, yTrain, batch_size=128, epochs=10, verbose=1)
model.evaluate(xTest, yTest, batch_size=128, verbose=2)
```

```

Epoch 1/10
391/391 [=====] - 13s 33ms/step - loss: 1.8170 - accuracy: 0.3307
Epoch 2/10
391/391 [=====] - 13s 33ms/step - loss: 1.2392 - accuracy: 0.5580
Epoch 3/10
391/391 [=====] - 13s 33ms/step - loss: 1.0465 - accuracy: 0.6343
Epoch 4/10
391/391 [=====] - 13s 33ms/step - loss: 0.9405 - accuracy: 0.6710
Epoch 5/10
391/391 [=====] - 13s 34ms/step - loss: 0.8349 - accuracy: 0.7091
Epoch 6/10
391/391 [=====] - 13s 34ms/step - loss: 0.7593 - accuracy: 0.7328
Epoch 7/10
391/391 [=====] - 13s 33ms/step - loss: 0.7015 - accuracy: 0.7556
Epoch 8/10
391/391 [=====] - 13s 33ms/step - loss: 0.6494 - accuracy: 0.7735
Epoch 9/10
391/391 [=====] - 13s 34ms/step - loss: 0.5823 - accuracy: 0.7955
Epoch 10/10
391/391 [=====] - 13s 34ms/step - loss: 0.5286 - accuracy: 0.8118
79/79 - 1s - loss: 0.8111 - accuracy: 0.7304
[0.811121940612793, 0.730400025844574]

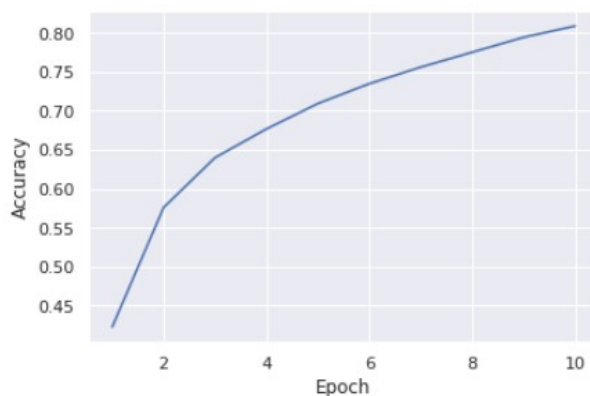
```

```

historyData = {}
historyData['Accuracy'] = history.history['accuracy']
historyData['Epoch'] = list(range(1, history.params['epochs'] + 1))
sns.lineplot(x="Epoch", y="Accuracy",
             data=historyData)

```

<AxesSubplot:xlabel='Epoch', ylabel='Accuracy'>



3. Выводы по работе:

Познакомился с примерами работы со свёрточными нейронными сетями (CNN) на примере набора CIFAR10.

4. Контрольные вопросы:

4.1. Что такое свертка?

Свёртка — это упорядоченная процедура смешивания двух источников информации. Свёртку можно описать математически. Это такая же операция, как сложение, умножение или взятие производной. Хотя сама по себе операция свёртки сложна, она может быть очень полезна для упрощения ещё более сложных выражений.

4.2. Напишите математическую операцию свертки

Свёртка функций — операция в функциональном анализе, которая при применении к двум функциям f и g , возвращает третью функцию, соответствующую взаимнокорреляционной функции $f(x)$ и $g(-x)$. Операцию свертки можно интерпретировать как «схожесть» одной функции с отражённой и сдвинутой копией другой. Понятие свёртки обобщается для функций, определённых на произвольных измеримых пространствах и может рассматриваться как особый вид интегрального преобразования. В дискретном случае свёртка соответствует сумме значений f с коэффициентами, соответствующими смещённым значениям g , то есть $(f*g)(x) = f(1)g(x-1) + f(2)g(x-2) + f(3)g(x-3) + \dots$

4.3. Какие свойства сверточного слоя?

В сверточных слоях каждый элемент связан с определенным количеством элементов в предыдущем слое. Обычно количество элементов варьируется от 1 до 9 элементов. При этом, значения весов между нейронами разделяются, т.е. значения весов i -го нейрона равны значениям весов $i + 1$ нейрона. Данное свойство значительно уменьшает требования к объему памяти.

4.4. Сколько этапов в сверточном слое? Какие?

Три этапа: (Применение фильтра, Применение нелинейности, Применение операции дискретитизации (pooling)).

4.5. Что такое регуляризация? Зачем она нужна?

Регуляризация это — метод добавления некоторых дополнительных ограничений к условию с целью решить неправильно поставленную задачу. В машинном обучении и статистике регуляризация в основном используется для обозначения ограничения оптимизации путем наложения штрафа на сложность решения в попытке предотвратить переобучение на обучающей выборке. Регуляризация чаще всего применяется путем добавления неявной информации о желаемом результате.

4.6. Какой вид регуляризации использовался в лабораторной?

Dropout. - отключает один из нейронов в слое. Добавить слой dropout в TensorFlow можно с помощью команды: `tf.keras.layers.Dropout(rate)`, где `rate` - вероятность работы нейрона в слое.