

Московский государственный технический университет имени Н.Э.Баумана

Кафедра «Системы обработки информации и управления»

ОТЧЕТ

Лабораторная работа №3  
по дисциплине  
«Проектирование интеллектуальных систем»  
на тему  
«Сверточные нейронные сети»

Выполнил:  
Студент ИУ5-24М  
Гаврилюк А.Г.

Москва, 2020

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
```

## Загружаем датасет и нормализуем его, чтобы значения были от 0 до 1

In [2]:

```
(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()

train_images, test_images = train_images / 255.0, test_images / 255.0
```

## Создаем модель и добавляем сверточные слои

In [7]:

```
model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

In [8]:

```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dropout(0.3))
model.add(layers.Dense(10))
```

## Описание модели

In [9]:

```
model.summary()
```

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
=====		
conv2d_3 (Conv2D)	(None, 30, 30, 32)	896
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_4 (Conv2D)	(None, 13, 13, 64)	18496
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 64)	36928
flatten_1 (Flatten)	(None, 1024)	0
dense_2 (Dense)	(None, 64)	65600
dropout_1 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 10)	650
=====		
Total params: 122,570		
Trainable params: 122,570		
Non-trainable params: 0		

## Обучение модели

In [10]:

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=10,
                   validation_data=(test_images, test_labels))
```

Train on 50000 samples, validate on 10000 samples

```
Epoch 1/10
50000/50000 [=====] - 25s 498us/sample - loss: 1.6499 - acc: 0.3898 - val
_loss: 1.3538 - val_acc: 0.5178
Epoch 2/10
50000/50000 [=====] - 26s 510us/sample - loss: 1.3148 - acc: 0.5301 - val
_loss: 1.1534 - val_acc: 0.5861
Epoch 3/10
50000/50000 [=====] - 30s 593us/sample - loss: 1.1582 - acc: 0.5900 - val
_loss: 1.0853 - val_acc: 0.6043
Epoch 4/10
50000/50000 [=====] - 29s 587us/sample - loss: 1.0599 - acc: 0.6281 - val
_loss: 1.0102 - val_acc: 0.6407
Epoch 5/10
50000/50000 [=====] - 27s 543us/sample - loss: 0.9906 - acc: 0.6539 - val
_loss: 0.9633 - val_acc: 0.6622
Epoch 6/10
50000/50000 [=====] - 26s 526us/sample - loss: 0.9258 - acc: 0.6748 - val
_loss: 0.9111 - val_acc: 0.6811
Epoch 7/10
50000/50000 [=====] - 26s 525us/sample - loss: 0.8799 - acc: 0.6920 - val
_loss: 0.9030 - val_acc: 0.6939
Epoch 8/10
50000/50000 [=====] - 26s 522us/sample - loss: 0.8357 - acc: 0.7044 - val
_loss: 0.9137 - val_acc: 0.6840
Epoch 9/10
50000/50000 [=====] - 26s 520us/sample - loss: 0.7963 - acc: 0.7198 - val
_loss: 0.8880 - val_acc: 0.6954
Epoch 10/10
50000/50000 [=====] - 26s 525us/sample - loss: 0.7667 - acc: 0.7301 - val
_loss: 0.8658 - val_acc: 0.7050
```

Точность модели - 70%

## Контрольные вопросы

### 1) Что такое свертка?

Свертка - это специализированный вид линейной операции. Сверточные сети - это нейронные сети, которые используют свертку вместо общего матричного умножения, по крайней мере, в одном из своих слоев.

### 2) Напишите математическую операцию свертки.

Свертка для двумерных данных:  $S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$

### 3) Какие свойства сверточного слоя?

Сверточные слои реализуют три важных свойства: разреженные взаимодействия, разделение параметров и эквивариантные изменения. Более того, свертка обеспечивает средства для работы с входами переменного размера.

### 4) Сколько этапов в сверточном слое? Какие?

1. Применение фильтра
2. Применение нелинейности
3. Применение операции дискретизации (pooling)

#### **5) Что такое регуляризация? Зачем она нужна?**

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.

#### **6) Как вид регуляризации использовался в лабораторной?**

В данной лабораторной используется метод регуляризации - дропаут. Данный слой с определенной вероятностью "выключает" один из нейронов в слое. При обучении с дропаутом можно представлять себе процесс обучения ансамбля нейронных сетей. Данный слой используется только во время обучения. На этапе тестирования вероятность присваивается единице, и слой не влияет на качество работы.