

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Искусственные нейронные сети»
Тема: Распознавание объектов на фотографиях

Студент гр. 8382

Янкин Д.О.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Распознавание объектов на фотографиях (Object Recognition in Photographs) CIFAR-10 (классификация небольших изображений по десяти классам: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль и грузовик).

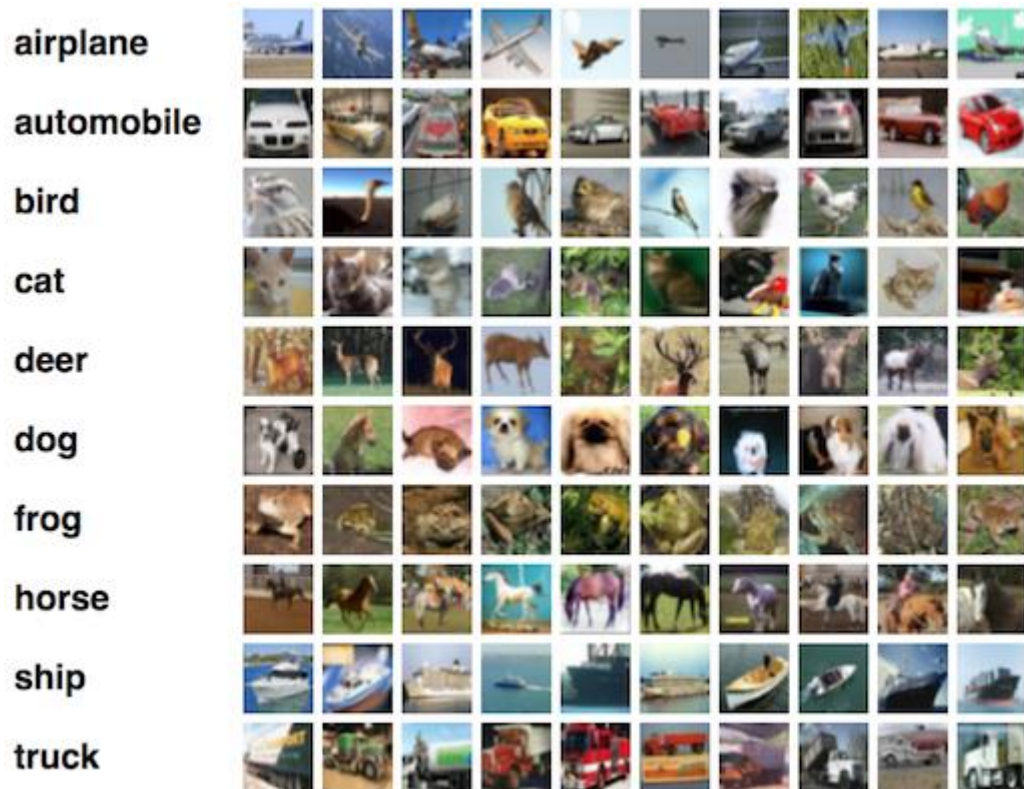


Рисунок 1. Пример входных данных

Постановка задачи.

- Ознакомиться со сверточными нейронными сетями
- Изучить построение модели в Keras в функциональном виде
- Изучить работу слоя разреживания (Dropout)

Выполнение работы.

Из библиотеки keras импортирован датасет CIFAR-10, из него загружены и нормализованы обучающие и тестовые данные:

```
from keras.datasets import cifar10
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
num_train, depth, height, width = X_train.shape
```

```

num_test = X_test.shape[0]
num_classes = np.unique(y_train).shape[0]
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= np.max(X_train)
X_test /= np.max(X_train)
Y_train = np_utils.to_categorical(y_train, num_classes)
Y_test = np_utils.to_categorical(y_test, num_classes)

```

Задана модель сети:

```

inp = Input(shape=(depth, height, width))
conv_1 = Convolution2D(conv_depth_1, kernel_size, kernel_size,
border_mode='same', activation='relu')(inp)
conv_2 = Convolution2D(conv_depth_1, kernel_size, kernel_size,
border_mode='same', activation='relu')(conv_1)
pool_1 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_2)
drop_1 = Dropout(drop_prob_1)(pool_1)
conv_3 = Convolution2D(conv_depth_2, kernel_size, kernel_size,
border_mode='same', activation='relu')(drop_1)
conv_4 = Convolution2D(conv_depth_2, kernel_size, kernel_size,
border_mode='same', activation='relu')(conv_3)
pool_2 = MaxPooling2D(pool_size=(pool_size, pool_size))(conv_4)
drop_2 = Dropout(drop_prob_1)(pool_2)
flat = Flatten()(drop_2)
hidden = Dense(hidden_size, activation='relu')(flat)
drop_3 = Dropout(drop_prob_2)(hidden)
out = Dense(num_classes, activation='softmax')(drop_3)
model = Model(input=inp, output=out)

```

Проведено обучение сети в течении 200 эпох с батчами по 32 с ядром 3x3:

Training accuracy	Validation accuracy	Test accuracy
0.9323	0.7992	0.5797

Обучение длится слишком долго, количество эпох сокращено до 80, результаты обучения отличаются не сильно, а времени затрачивается значительно меньше:

Training accuracy	Validation accuracy	Test accuracy
0.9144	0.7900	0.5519

Исследуется влияние слоя Dropout на обучение сети. Работа Dropout заключается в том, что он проходит по всем нейронам слоя и с вероятностью p исключает их на время итерации, это позволяет бороться с переобучением. Из модели сети удалены слои Dropout, проведено обучение:

Training accuracy	Validation accuracy	Test accuracy
0.9773	0.6316	0.5269

У сети возросла точность на тренировочных данных и уменьшилась на валидационных и тестовых, что соответствует теории.

Исследуется влияние размера ядра на результаты обучения. Выбран размер ядра 5x5:

Training accuracy	Validation accuracy	Test accuracy
0.8814	0.7660	0.5575

Точность на обучающих и валидационных данных несколько снизилась, на тестовых осталась на том же уровне.

Выбран размер ядра 7x7:

Training accuracy	Validation accuracy	Test accuracy
0.8100	0.7134	0.4887

В данном случае точность снизилась на всех трех наборах данных по сравнению с прошлыми запусками.

Выбран размер ядра 2x2:

Training accuracy	Validation accuracy	Test accuracy
0.7732	0.6386	0.3350

Уменьшение размера ядра также ухудшило результат обучения. Оптимальным размером является изначальный вариант – 3x3.

Выводы.

В ходе лабораторной работы на датасете CIFAR-10 была обучена сверточная нейронная сеть для классификации объектов на фотографиях по десяти классам. Было проверено влияние на результат обучения слоя Dropout и разных размеров ядра свертки.