

Машинное обучение и интеллектуальный анализ данных

Семинар 10 Лабораторная работа 8

**Г.А. Ососков*, О.И. Стрельцова*, Д.И. Пряхина*,
Д.В. Подгайный*, А.В. Стадник*, Ю.А. Бутенко***

Государственный университет «Дубна»

***Лаборатория информационных технологий, ОИЯИ
Дубна, Россия**

Государственный университет «Дубна»

Постановка задачи

Сверточные нейронные сети

Решение задачи классификации изображений с помощью сверточной нейронной сети.

Исходные данные: DataSet CIFAR-10
(<https://www.cs.toronto.edu/~kriz/cifar.html>)

```
from keras.datasets import cifar10
```

Набор данных содержит 60 000 цветных изображений размером 32x32.
Все изображения делятся на 10 классов (по 6 000 изображений в каждом).

Классы изображений: самолеты, автомобили, птицы, коты, олени, собаки, лягушки, лошади, корабли, грузовики.

Набор данных для обучения содержит 50 000 картинок (5 комплектов по 10 000 изображений).

Набор данных для проверки содержит 10 000 картинок (по 1 000 случайно выбранных изображений каждого из 10 класса).

Алгоритм решения задачи

1. Загрузка данных из CIFAR-10 `(x_train, y_train), (x_test, y_test) = cifar10.load_data()`

2. Визуализация данных

Необходимо визуализировать 10 случайных изображений из набора данных.

3. Преобразование данных для *Keras*

- Необходимо создать и векторизовать метки классов.

4. Нормализация данных

* Необходимо привести значения яркости каждого пикселя исходных изображений из диапазона [0, 255] в диапазон [0, 1].

Алгоритм решения задачи

5. Построение модели сверточной нейронной сети

```
model = Sequential()  
model.add(Conv2D(32, (3, 3), padding='same', input_shape=x_train.shape[1:]))  
model.add(Activation('relu'))  
model.add(Conv2D(32, (3, 3)))  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
  
model.add(Conv2D(64, (3, 3), padding='same'))  
model.add(Activation('relu'))  
model.add(Conv2D(64, (3, 3)))  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Dropout(0.25))  
  
model.add(Flatten()) # преобразует (6, 6, 64) в вектор размером 6*6*64=2304  
model.add(Dense(512))  
model.add(Activation('relu'))  
model.add(Dropout(0.5))  
model.add(Dense(num_classes))  
model.add(Activation('softmax'))
```

Алгоритм решения задачи

6. Общая информация о созданной модели. Визуализация модели

7. Компиляция модели

```
sgd = SGD(lr = 0.1, decay=1e-6, momentum=0.9, nesterov=True)  
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

8. Обучение модели. Визуализация процесса обучения

9. Проверка модели

Содержание отчета

1. Постановка задачи.
2. Описание исходных данных.
3. Подготовка данных для работы с нейронной сетью.
4. Построение модели нейронной сети и ее визуализация.
5. Визуализация процесса обучения модели.
6. Проверка модели на тестовом наборе данных. Оценить точность.
7. Можно ли улучшить результат, если изменить число эпох? Для ответа на этот вопрос необходимо изменить количество эпох, оценить точность полученного результата, сделать выводы, найти оптимальное количество эпох.
8. Построение модели нейронной сети, состоящей из 6 сверточных слоев.
9. Проверка модели на тестовом наборе данных. Оценить точность.
10. Можно ли улучшить результат, если изменить число эпох? Для ответа на этот вопрос необходимо изменить количество эпох, оценить точность полученного результата, сделать выводы, найти оптимальное количество эпох.
11. Список литературы.

Домашнее задание (!!!)

Подготовить набор данных: рукописные ч/б цифры 28x28 (*Paint*). Имя файла: цифра.png
Каждый студент из группы должен подготовить 10 картинок с цифрами (от 0 до 9).
Все изображения собрать в одну директорию и загрузить на jhub.