

ОБРАБОТКА ДАННЫХ ПРИ РАСПОЗНАВАНИИ ОБЪЕКТОВ С ПОМОЩЬЮ СВЕРТОЧНЫХ НЕЙРОСЕТЕЙ

DATA PROCESSING IN RECOGNITION OF OBJECTS USING CONVOLUTIONAL NEURAL NETWORK

С. В. Ткаченко^{1,2}, Т. В. Смирнова^{1,2}, И. В. Лефанова^{1,2}
S. Tkachenko, T. Smirnova, I. Lefanova

¹Белорусский государственный университет, БГУ, г. Минск, Республика Беларусь

²Учреждение образования «Международный государственный экологический институт имени А. Д. Сахарова» Белорусского государственного университета, МГЭИ им. А. Д. Сахарова БГУ г. Минск, Республика Беларусь

¹Belarusian State University, BSU, Minsk, the Republic of Belarus

²International Sakharov Environmental Institute of Belarusian State University, ISEI BSU
Minsk, the Republic of Belarus
smirnova@iseu.by

Представлены этапы обработки данных от различных источников для задачи распознавания изображений нейронной сетью. Алгоритмы реализованы на языке Python с помощью библиотек TensorFlow и API Keras.

The stages of data processing from different sources for the task of image recognition by a neural network are presented. The algorithms are implemented in Python using the TensorFlow libraries and the Keras API.

Ключевые слова: TensorFlow, Keras, python, нейронные сети, CV, распознавание рукописных знаков.

Keywords: TensorFlow, Keras, python, neural network, CV, handwriting recognition.

<https://doi.org/10.46646/SAKH-2023-2-269-272>

Введение. Метод математического моделирования, основанный на данных (Data-Driven Modelling), широко используется в областях, где недостаточно знаний, чтобы сформулировать закономерности в виде математических форм, но накоплено много информации по изучаемым процессам и явлениям.

Растущий спрос на прогнозирование в экологии, в частности, в таких областях, как сокращение биоразнообразия, глобальные изменения климата, опасные явления природы, способствуют разработке количественных подходов в области экологии. Но для их применения необходимы другие технологии данных.

Эффективными инструментами моделирования стали методы машинного обучения. Машинное обучение (Machine Learning, ML) – это семейство вычислительных алгоритмов, предназначенных для определения закономерностей в сложных структурах данных, и создания на их основе точных прогностических моделей. ML основывается на описании сложных отношений и предсказывает параметры и доверительные интервалы. Машинное обучение, находящееся на пересечении компьютерных наук и статистики, и составляющее основу искусственного интеллекта и науки о данных, является бурно развивающейся областью знаний.

Глубокое обучение (Deep Learning, DL) – относительно новая сфера ML. Главный инструмент DL – глубокая нейронная сеть (DNN), основанная на искусственных нейронных сетях (Artificial Neural Networks, ANNs). DL представляет из себя набор методов, которые позволяют обучать большие и глубокие (много слоев) нейронные сети. Такие сети стали возможными благодаря разработке улучшенных алгоритмов для обработки данных; увеличению вычислительной мощности и доступу к наборам данных для обучения проектируемой сети. Обученные нейросети применяются в задачах анализа данных, например, в задачах распознавания и классификации изображений, а также в задачах прогнозирования.

Цель работы – разработать инновационные учебно-методические материалы по созданию и глубокому обучению нейронных сетей, способных обрабатывать большие объемы данных, включая данные от систем компьютерного зрения, для распознавания и классификации объектов.

1 Программные средства для распознавания изображений. Наиболее популярными и эффективными фреймворками и библиотеками программ, используемыми для распознавания изображений с помощью нейросетей, являются TensorFlow и Keras.

TensorFlow – это библиотека с открытым исходным кодом на Python, содержит множество различных алгоритмов и моделей, позволяющих реализовать глубокие нейронные сети для использования в таких задачах, как распознавание и классификация изображений по статичным и динамическим данным, обработка естественного языка, распознавание текста. **Keras** – это высокоуровневый API (интерфейс прикладного программирования), который может использовать более эффективно использовать функции TensorFlow.

Распознавание изображения – это передача оцифрованного изображения в нейронную сеть и присвоения ему метки. Метка, выбранная нейросетью, будет соответствовать заранее определенному классу. Изображению может быть присвоена одна или несколько меток. Если изначально была определена одна метка (и класс), это будет задача «распознавания»; если процесс предполагает поиск нескольких типов меток (и классов), то это задача «классификации».

Обнаружение объектов – это подмножество классификаций изображений, когда определенные экземпляры объектов идентифицируются как принадлежащие к определенному классу (животные, люди).

Как нейронные сети учатся распознавать изображения

Работа нейросети состоит из процессов:

- 1) Извлечение признаков, т.е. элементов данных из одной группы изображений, которые будут обрабатываться нейронной сетью. На практике признаками являются группы пикселей (линии и точки)
- 2) Функции активации, которая получает карту признаков (массив чисел), и увеличивает их нелинейность.
- 3) Объединение слоёв. Оно «упрощает» изображение, сокращая информацию о нем.
- 4) Связка слоёв: определяется вероятность принадлежности изображения каждому из классов.

Рабочий процесс машинного обучения. Процесс обучения модели нейронной сети может быть поделен на четыре этапа.

Подготовка данных. Подготовка данных для обучения сети включает сбор изображений и их маркировку. Нужно проверить, есть ли среди данных поврежденные значения, или данные в неправильном формате, или с неправильными метками и т. д.

Создание модели. На этом этапе принимается решение о количестве слоев в модели, о размерах входного и выходного слоев, о применяемой функции активации. Создается экземпляр модели.

Обучение модели. Создается обучающая выборка, задается количество эпох обучения.

Оценка модели. Точность модели оценивается на 1) проверочном наборе данных, после чего обычно возвращаются к этапу дообучения сети; 2) на тестовом наборе данных.

3. Подготовка данных для работы с последовательной моделью Keras. В качестве набора данных для машинного обучения сети использовался набор данных MNIST, содержащий более 60000 изображений рукописных цифр от 0 до 9 в градациях серого, а также 10 000 картинок для тестирования. Изображения с цифрами представлены в виде матриц 28 x 28, где каждая ячейка содержит определенный оттенок серого.

```
import tensorflow as tf
(x_train, y_train), (x_test, y_test)=tf.keras.datasets.mnist.load_data()
```

Для обработки данных надо подключить библиотеку NumPy для математических расчетов; класс Sequential для создания слоев:

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization, Activation
```

Создание собственного набора данных. Для создания собственного набора данных потребуется выполнить поиск, загрузку и преобразование файлов с информацией об изображениях. Загрузим малый набор данных MNIST для тренировки сети и преобразования данных в изображения. Необработанный набор данных mnist_train_small.csv - это CSV-файл из 20 000 строк и 785 столбцов. Первый столбец каждой строки – метка класса; следующие 784 столбца представляют значение пикселей изображения цифры размером 28x28 в градациях серого.

Преобразуем набор данных csv в формат с одной размерностью: загрузим набор в кадр данных pandas, затем преобразуем кадр данных в ndarray NumPy, используя функцию to_numpy(). В результате получим сжатый массив numpy.ndarray размером 20000x785:

```
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
import os
from PIL import Image
import random
import shutil
df=pd.read_csv('./sample_data/mnist_train_small.csv', header=None)
image_arrays=df.to_numpy()
len(image_arrays)
```

Для автоматизации работы создаем папку images и сохраняем каждый массив изображений в формате JPEG. Имя файла представляет собой случайное 12-значное число, за которым следует метка изображения:

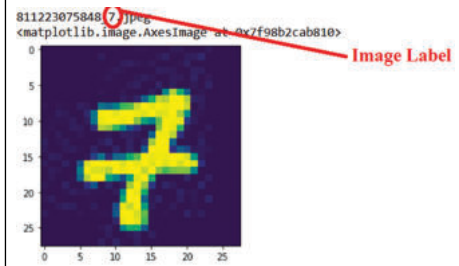
```
if os.path.exists('images'):
    shutil.rmtree('images')
os.makedirs('images')
```

```
for img in image_arrays:
filename = os.path.join('images', str(random.randint(100000000000, 99999999999))
+f'_{img[0]}.jpeg')
Image.fromarray(np.uint8(np.reshape(img[1:], (28,28)))) .save(filename)
```

Загрузим и визуализируем одно из изображений из каталога:

```
images=os.listdir('images')
n=62
print(images[n])
im=plt.imread(os.path.join('images', images[n]))
plt.imshow(im)
```

Результат показан справа



Создадим функцию для получения метки каждого изображения из имени его файла.

```
def get_label(filename):
return filename.split('.')[0].split('_')[1]
```

Все изображения из созданного каталога images переместим в подкаталог, соответствующий классу, к которому оно принадлежит. Нужно создать 10 подкаталогов для каждой цифры от 0 до 9:

```
path = 'images'
for image in images:
if not os.path.exists(os.path.join(path, get_label(image))):
os.makedirs(os.path.join(path, get_label(image)))
shutil.move(os.path.join(path, image), os.path.join(path, get_label(image), image))
```

В результате перемещения всех изображений в соответствующий подкаталог класса, сам каталог изображений будет иметь линейную структуру вида:



Фрагмент кода на python для выборки набора данных для любого изображения представлен ниже:

```
tf.keras.preprocessing.image_dataset_from_directory(directory,
labels='inferred', label_mode='int', class_names=None,
color_mode='rgb', batch_size=32, image_size=(256, 256),
shuffle=True, seed=None,
validation_split=None, subset=None,
interpolation='bilinear',
follow_links=False,
crop_to_aspect_ratio=False,
**kwargs )
```

Зададим размер пакета при обновлении весов и требуемую высоту и ширину изображения

```
batch_size = 32
img_height = 180
img_width = 180
```

Используем 80 % изображений для обучения и 20 % для проверки

```
train_ds = tf.keras.utils.image_dataset_from_directory( path,
validation_split=0.2, subset='training',
seed=123,
image_size=(img_height, img_width),
batch_size=batch_size)
```

Имена классов в наборах данных можно найти в атрибуте class_names.

```
class_names = train_ds.class_names
print(class_names)
```

Визуализируем данные из обучающего набора. Результат для первых шести изображений от train_ds

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 10))
```

```
for images, labels in train_ds.take(1):
    for i in range(6):
        ax = plt.subplot(2, 3, i + 1)
        plt.imshow(images[i].numpy().astype('uint8'))
        plt.title(class_names[labels[i]])
        plt.axis('off')
```



Далее на этих наборах данных модель обучается, используя метод `Model.fit()`. С помощью метода `evaluate()` оцениваем качество обученной модели. На тестовом наборе показатель достигал 72%

Заключение. Реализация алгоритмов распознавания изображений с помощью нейросетей требует большого объема расчетов. Основная трудность заключается в обработке исходной неструктурированной информации с изображениями различных объектов и подготовке наборов данных. В работе рассмотрен и реализован алгоритм по преобразованию входной информации к виду, пригодному для визуализации.

ЛИТЕРАТУРА

1. *Раица, С.* Python и машинное обучение: [Электронный ресурс] – Москва: ДМК Пресс, 2017. – 418 с. Режим доступа: <http://znanium.com/catalog> – Дата доступа: 28.01.2023.

ИННОВАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ РАЗРАБОТКИ СИСТЕМЫ ОНЛАЙН-МОНИТОРИНГА СОСТОЯНИЯ КОМПОНЕНТОВ ОКРУЖАЮЩЕЙ СРЕДЫ РЕСПУБЛИКИ БЕЛАРУСЬ

INNOVATIVE TECHNOLOGIES TO DEVELOP THE ONLINE MONITORING SYSTEM OF THE STATE OF ENVIRONMENTAL COMPONENTS OF THE REPUBLIC OF BELARUS

В. В. Журавков^{1,2}, Н. Д. Урбанович^{1,2}
Zhuravkov V.V.^{1,2}, Urbanovich N.D.^{1,2}

¹Белорусский государственный университет, БГУ, г. Минск, Республика Беларусь

²Учреждение образования «Международный государственный экологический институт имени А. Д. Сахарова» Белорусского государственного университета, МГЭИ им. А. Д. Сахарова БГУ, г. Минск, Республика Беларусь
nick.urbanovich@mail.ru

¹Belarusian State University, BSU, Minsk, the Republic of Belarus

²International Sakharov Environmental Institute of Belarusian State University, ISEI BSU, Minsk, the Republic of Belarus

В статье представлены инновационные технологии для разработки системы онлайн-мониторинга состояния компонентов окружающей среды Республики Беларусь. Прототипом для разработки является система онлайн-мониторинга состояния компонентов окружающей среды города Орши и Оршанского района.

The article presents the innovative technologies used to develop the system of online monitoring of the state of environmental components of the Republic of Belarus. The prototype for the development of the above system is the system for online monitoring of the state of environmental components of the city of Orsha and Orsha district.

Ключевые слова: информационно-аналитическая система, php, javascript, leaflet, openstreetmap, карты google, drupal.

Keywords: information analytical system, php, javascript, leaflet, openstreetmap, google maps, drupal.

<https://doi.org/10.46646/SAKH-2023-2-272-275>

Система онлайн-мониторинга состояния компонентов окружающей среды на г. Орши и Оршанского района содержит такие данные, как: мониторинг атмосферного воздуха, мониторинг сточных и поверхностных вод, мониторинг подземных вод, мониторинг почв, радиационный мониторинг.