

Распознавание изображений на примере предметов одежды с помощью нейронной сети

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматривается применение нейронной сети для распознавания предметов одежды с набора одинаковых чёрно-белых изображений. Программа разработана с помощью языка программирования python и готового набора данных для обучения сети. Результатом исследования станет подробное описание и исходный код программы.

Ключевые слова: машинное обучение, нейронные сети, распознавание изображений, задача классификации, keras, tensorflow, MNIST.

Image recognition on the example of clothing items using a neural network

Vikhlyaev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the use of a neural network to recognize clothing items from a set of identical black-and-white images. The program is developed using the python programming language and a ready-made data set for network training. The result of the study will be a detailed description and source code of the program.

Keywords: machine learning, neural networks, image recognition, classification task, keras, tensorflow, MNIST.

1 Введение

1.1 Актуальность

Сегодня создание искусственных нейронных сетей широко применяются для решения различных задач в реальной жизни. Распознавание речи или изображений, теперь решают нейронные сети с высокой точностью. На основе этих задач решаются более сложные задачи регрессии и классификации. Одним из распространенных способов классификации является способ описаний объектов с использованием признаков, в котором каждый объект характеризуется набором числовых или нечисловых признаков. Однако для множества типов данных открытые признаки не дают необходимой точности классификации, примером послужит цвет пикселей изображений или сигнал цифрового звука. Классификация изображений человека, одежды, животных, природы является простой для распознавания

человеком, но очень сложной для машины. Поэтому по сей день является актуальным разработка алгоритмов и методов оптимизации и нормализации данных для увеличения скорости и результата обучения нейронных сетей.

1.2 Обзор исследований

В.А.Буланов, Т.М.Волосатова разработали программный комплекс предварительной обработки изображений для обнаружения и распознавания изображений [1]. З.У.Зейдан выделила виды современных обработок биомедицинских изображений и распознавание изображений [2]. Д.Ю.Працюк, С.А.Клестов показали применение нейросетей в распознавании изображений [3]. В.В.Зиядинов, П.С.Курочкин, М.В.Терешонок оптимизировали обучение сверточных нейронных сетей при распознавании изображений с низкой плотностью точек [4]. Е.Ю.Кузнецова описал методы голосования для распознавания изображений [5].

1.3 Цель исследования

Цель исследования – создать и обучить нейронную сеть распознавать предметы одежды с изображений.

2 Материалы и методы

В качестве набора данных используется Fashion MNIST, в котором содержатся 60 000 изображений предметов одежды, относящиеся к 10 классам [8]. Для создания и обучение нейронной сети применяется библиотека Keras в составе TensorFlow.

3 Результаты и обсуждения

Чтобы тестировать работу нейронной сети, набор данных разбит на файлы с изображениями и метками классов. Изображения в бинарном виде записаны в один файл. Размер изображения 28x28 пикселей. Каждый пиксель имеет интенсивность в 1 байт, варьируясь оттенками серого цвета. Такой набор данных очень удобен для обучения нейронной сети и не требует больших вычислительных ресурсов.

База данных MNIST (модифицированная база данных Национального института стандартов и технологий) обычно используется для обучения различных систем обработки изображений. Одна из архитектур нейронной сети хорошо показавшей себя в распознавании рукописных цифр данной базы данных, включает 2 слоя объединенных в полносвязную сеть. Входной слой имеет 800 нейронов. Выходной слой 10 нейронов. Таким образом, на вход поступают все пиксели изображения, а на выходе вероятность на промежутке от 0 до 1, что изображён данный предмет одежды.

Добавить в проект базу данных «Fashion MNIST» можно одной строчкой кода. В «Keras» уже есть модуль для автоматической загрузки данного набора.

Модуль «Sequential» предназначен для представления нейронной сети, в которой все слои идут последовательно.

Для создания полносвязной нейронной сети, импортируется модуль «Dense».

Так же необходимо подключить утилиты для перевода входных данных в формат, с которым работает «Keras»(рис.1).

```
from tensorflow.keras.datasets import fashion_mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras import utils
```

Рис. 1. Подключаемые модули

Загрузка данных осуществляется с помощью функции «load_data()». Набор данных разбивается на 2 картера для обучения и тестирования, в каждом из которых массив изображений и правильных ответов.

Полносвязная сеть не может работать с большими данными, поэтому двухмерные изображения преобразуются в вектор. С помощью метода «reshape()» меняется размерность изображений, первым параметром является количество картинок, вторым количество пикселей.

Далее выполняется нормализация пикселей, интенсивность которых равна 255, чтобы данные на входе были в диапазоне от 0 до 1. Данное действие способствует оптимизации алгоритмов используемых для обучения нейронных сетей (рис.2).

```
(img_train, answer_train), (img_test, answer_test) = fashion_mnist.load_data()

img_train = img_train.reshape(60000, 784)
img_train = img_train / 255
```

Рис. 2. Загрузка и нормализация данных

В наборе данных «Fashion MNIST» правильные ответы содержат номер класса, который находится на изображении. Создаваемая нейронная сеть выдаёт 10 значений (по количеству нейронов) с вероятностями, что на картинке принадлежит номеру класса. Для корректного обучения нейронной сети, нужно преобразовать данные правильных ответов из номеров классов в формат быстрого кодирования (рис.3,4).

```
answer_train = utils.to_categorical(answer_train, 10)
classes = [
    'футболка',
    'брюки',
    'свитер',
    'платье',
    'пальто',
    'туфли',
    'рубашка',
    'кроссовки',
    'сумка',
    'ботинки'
]
```

Рис. 3. Преобразование правильных ответов в формат быстрого кодирования

```

0 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
1 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
2 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
3 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
4 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
5 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
6 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
7 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
8 [0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
9 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]

```

Рис. 4. Сопоставление номеров классов и соответственных кодов, выдаваемых на выходе нейронной сетью

Далее создаётся последовательная модель нейронной сети с помощью метода «Sequential()» и добавляются полносвязные слои через метод Dense().

Входной слой содержит 800 нейронов, и принимает 784 входения в каждый нейрон. Функция активации – ReLu(rectified linear unit). Если входное значение меньше или равно 0, то функция выдаёт 0, а если больше 0, то выдаёт само это значение.

Выходной слой имеет 10 нейронов, количество входений можно не указывать, так как каждый нейрон текущего слоя соединен с каждым нейроном предыдущего слоя. Функция активации – SoftMax(Нормализованная экспоненциальная функция), позволяет получить суммарное значение всех нейронов на выходе из слоя равное 1. Это используется для представления вероятности (рис.5).

```

model = Sequential()

model.add(Dense(800, input_dim=784, activation="relu"))

model.add(Dense(10, activation="softmax"))

```

Рис. 5. Создание модели нейронной сети

Перед обучением модели её нужно скомпилировать. В параметре функция ошибки используется категориальная перекрёстная энтропия, данная функция хорошо подходит для задач классификации, если классов больше 2. Тип оптимизатора SGD(Стохастический градиентный спуск). В метрике качества обучения нейронной сети указывается доля правильных ответов (рис.6).

```
model.compile(  
    loss="categorical_crossentropy",  
    optimizer="SGD",  
    metrics=["accuracy"]  
)  
model.summary()
```

Рис. 6. Компиляция модели нейронной сети

Обучение нейронной сети с учителем в «Keras», осуществляется с помощью метода «fit()», который принимает набор данных для обучения и правильные ответы.

Для обучения применён метод Стохастического градиентного спуска с мини выборкой. На вход принимается 200 изображений, по ним рассчитывается функция ошибки и градиент, после чего изменяются веса нейронной сети. Так повторяется пока не пройдут все изображения. Количество повторений обучения, указано в параметре «epochs». Параметр «verbose» выводит данные прогресса сети.

На выходе обучения нейронная сеть выдаёт свыше 98% правильных ответов (рис. 7,8).

```
model.fit(img_train, answer_train,  
        batch_size=200,  
        epochs=100,  
        verbose=1)
```

Рис. 7. Обучение нейронной сети

```
Epoch 93/100  
300/300 [=====] - 6s 20ms/step - loss: 0.3110 - accuracy: 0.8916  
Epoch 94/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3101 - accuracy: 0.8924  
Epoch 95/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3091 - accuracy: 0.8924  
Epoch 96/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3085 - accuracy: 0.8927  
Epoch 97/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3073 - accuracy: 0.8927  
Epoch 98/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3066 - accuracy: 0.8939  
Epoch 99/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3058 - accuracy: 0.8932  
Epoch 100/100  
300/300 [=====] - 5s 15ms/step - loss: 0.3051 - accuracy: 0.8936
```

Рис. 8. Процесс обучения нейронной сети

Используя метод «predict()» применено распознавания предметов одежды на входных данных. В каждом из результатов распознавания приведено 10 значений, из которых все кроме одного приближены к нулю.

Для того чтобы из выходных значений нейрона получить номер класса можно воспользоваться функцией «argmax» из библиотеки «numpy». Таким же способ проверяется качество работы сети, используя тот же элемент уже из набора тестируемых данных (рис. 9,10,11).

```
predictions = model.predict(img_train)
```

```
print(predictions[0])
```

```
[7.8125897e-09 5.8385474e-10 7.7549966e-10 1.0317907e-09 4.6871385e-10
 1.4143994e-04 7.1602194e-08 7.9085137e-04 4.8235202e-06 9.9906272e-01]
```

```
np.argmax(predictions[0])
```

9

```
np.argmax(answer_train[0])
```

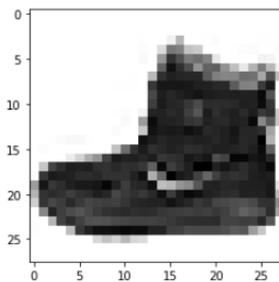
9

Рис. 9. Проверка работы нейронной сети



Рис. 10. Изображения и метки классов из набора «Fashion MNIST»

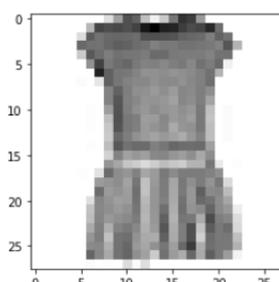
```
n = 0
plt.imshow(img_train[n].reshape(28, 28), cmap=plt.cm.binary)
plt.show()
```



```
classes[np.argmax(predictions[n])]
```

'ботинки'

```
n = 3
plt.imshow(x_train[n].reshape(28, 28), cmap=plt.cm.binary)
plt.show()
```



```
classes[np.argmax(predictions[n])]
```

'платье'

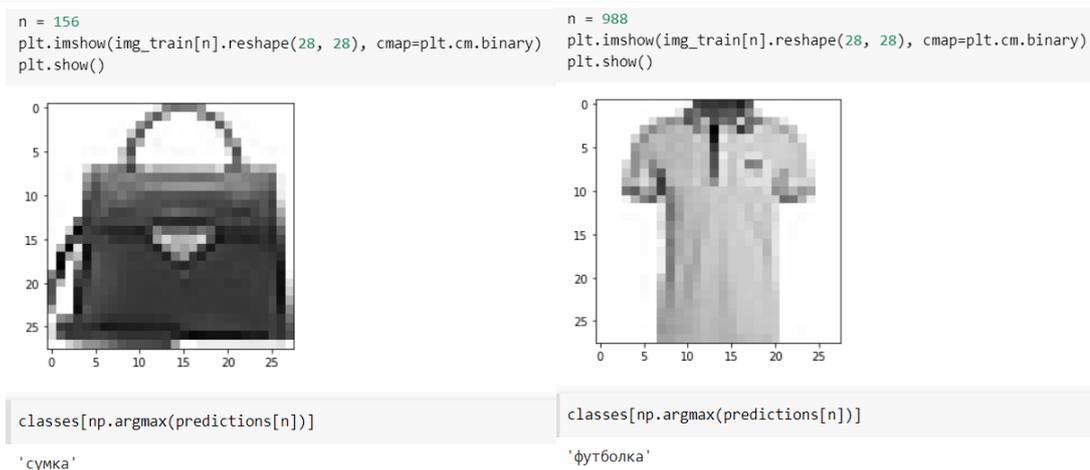


Рис. 11. Сопоставление изображений из набора данных и классами указанными нейронной сетью

В данной статье было рассмотрено решение задачи классификации на примере распознавания предметов одежды в изображениях с помощью библиотеки «keras». Описана загрузка, и нормализация входных данных. Показано создание модели для полносвязной нейронной сети и результат работы программы.

Библиографический список

1. Буланов В.А., Волосатова Т.М. Программный комплекс предварительной обработки изображений для обнаружения и распознавания изображений // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2014. № 4. С. 321-338.
2. Зейдан З.У. Виды современных обработок биомедицинских изображений и распознавание изображений // В сборнике: Исторические, философские, методологические проблемы современной науки. Сборник статей 1-й Международной научной конференции молодых ученых. Ответственный редактор А.А. Горохов. 2018. С. 200-204.
3. Працюк Д.Ю., Клестов С.А. Применение нейросетей в распознавании изображений // В сборнике: ИННОВАТИКА-2022. Сборник материалов XVIII международной школы-конференции студентов, аспирантов и молодых ученых. Томск, 2022. С. 437-440.
4. Зиядинов В.В., Курочкин П.С., Терешонок М.В. Оптимизация обучения сверточных нейронных сетей при распознавании изображений с низкой плотностью точек // Радиотехника и электроника. 2021. Т. 66. № 12. С. 1207-1215.
5. Кузнецова Е.Ю. Методы голосования для распознавания изображений // В книге: МНСК-2017: Информационные технологии. Материалы 55-й Международной научной студенческой конференции. 2017. С. 162.
6. Кривенко М.П. Байесовский подход в распознавании фрагментов изображения, имеющих различные размеры // Обзорение прикладной и промышленной математики. 2007. Т. 14. № 3. С. 538а-539.

7. Мрizaи X. Выбор начальных условий обучения нейронных сетей с многомерными входными данными // В сборнике: Информатика: проблемы, методология, технологии. материалы XV международной научно-методической конференции. 2015. С. 170-176.
8. <https://github.com/zalandoresearch/fashion-mnist>