

Программирование нейронных сетей в Python с использованием библиотек keras и tensorflow

Семченко Регина Викторовна

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Еровлев Павел Андреевич

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Аннотация

В данной статье описан процесс создания нейронной сети в приложении google notebook с помощью библиотек keras и tf.

Ключевые слова: TensorFlow, Keras, python, google

Neural network programming in Python using libraries keras and tensorflow

Semchenko Regina Viktorovna

*Sholom-Aleichem Priamursky State University
student*

Erovlev Pavel Andreevich

*Sholom-Aleichem Priamursky State University
student*

Abstract

This article describes the process of creating a neural network in the google notebook application using the keras and tf libraries.

Keywords: TensorFlow, Keras, python, google

Нейронные сети одна из самых актуальных тем в настоящем времени. Они применяются во множестве самых различных технологиях, начиная от бота в социальной сети и заканчивая проектированием самых сложных технических инструментов.

Алгоритмы нейронных сетей начали появляться с 1943 года, а в 1949 году уже предлагается первый алгоритм по обучению нейронной сети. Нейросеть представляет собой математическую модель, которая умеет обучаться, она запоминает множество пройденных этапов и в последующем может дать примерный ответ на вопрос основываясь на прошлых данных, она запоминает при каких обстоятельствах получаются конкретные ответы и имея эти знания она решает уже следующие задачи. Нейронная сеть применяется во множестве задач.

Цель задачи разработать нейронную сеть по определению изображения.

В статье И.И. Латкина рассматриваются примеры использования TensorFlow для решения некоторых задач машинного обучения, раскрывается понятия графа данных и основных принципов работы TensorFlow. Работа с библиотекой демонстрируется на решении задач логистической регрессии, а также нейронной сети и сравниваются результаты работы с аналогами [1]. А.А. Крыловецкий и Д.М. Сузодолов в своей работе с помощью метода вычислительного эксперимента исследовали возможность использования нейросетей глубокого обучения для распознавания изображений элементов зерновых смесей, провели сравнение результатов для трех различных архитектур, реализованных в пакетах VGG16, VGG19 и MobileNet на наборе изображений зёрен различных растений [2]. А.И. Тур и др. в своей статье рассмотрели пример повышения эффективности распознавания объекта по его изображению с помощью искусственной нейронной сети на основе TensorFlow и Keras. Представлены результаты экспериментов по выбору количества распознаваемых классов, изменениям параметров распознавания, а также модификациям обучающей выборки. На основании полученных данных сделаны выводы и предложен вариант реализации системы для данной задачи [3]. В.С. Мамедов и С.А. Заслонов в своей работе провели исследование по применению TensorFlow в задачах распознавания изображения и в качестве помощника использовали технологию Blockchain. В их работе использовалась технология GPU для оптимизации вычислений[4]. В статье Р.Р. Гатин, Р.Р. Бикмухаметов рассмотрели библиотеку Tensorflow, её команды, функции для исследования рекуррентных сетей на базе языка программирования Python. Так же уточнили способы настройки и работы с данной библиотекой [5].

Написание будет происходить на бесплатной платформе Google Colab, в которой уже установлены все необходимые библиотеки для работы с нейросетями. Для начала необходимо будет подключить эти библиотеки, чтобы они использовались в проекте (рис.1).

```
from tensorflow.keras.models import load_model #Подключение возможности загрузки модели нейросети
from google.colab import files #Подключение загрузки файлов с локального компьютера
from IPython.display import Image #Подключение просмотра картинок
from tensorflow.keras.preprocessing import image #Набор инструментов для увеличения данных в реальном времени на данных изображения
import numpy as np #Подключаем пакет для научных вычислений на Python
```

Рисунок 1 – Подключение библиотек

На рисунке описаны все подключаемые библиотеки. Они понадобятся для загрузки фотографии, уменьшении ее в размере, подключении уже обученной системы.

Далее напишем классы, которые производят разделение вещей (рис.2).

```
classes = ['футболка', 'брюки', 'свитер', 'платье', 'пальто', 'туфли', 'рубашка', 'кроссовки', 'сумка', 'ботинки']
```

Рисунок 2 – Добавление классов

Следующим шагом будет добавление уже обученной модели. В сервисе google есть возможность подключить уже обученные сети для определенных задач, обучение у них происходит на огромном количестве данных, следовательно, ответ нейросети будет максимально приближенным к истине (Рис.3).

Закачиваем файл с обученной моделью

```
[ ] !wget https://github.com/sozykin/dlpython_course/raw/master/introduction/fashion_mnist_dense.h5 -O fashion_mnist_dense.h5 #Загрузка обученной модели
```

Рисунок 3 – Загрузка модели

Далее напишем код, чтобы модель загружалась в память системы и сделаем вывод всех объектов(Рис.4).

```
[ ] model = load_model('fashion_mnist_dense.h5') #Загрузка модели в память  
▶ model.summary() #Расписывает количество объектов
```

Рисунок 4 – Загрузка модели в память

Теперь добавим возможность загрузки файлов с локального компьютера, для того, чтобы загрузить выбранное изображение и получить ответ от нейросети, что мы загрузили. Так же выведем само полученное изображение (Рис.5).

```
[ ] f = files.upload() #Загрузка файлов с компьютера  
[ ] !ls #Выводит содержимое загрузки  
[ ] img_path = 'bag.jpg' #Имя загруженного файла  
[ ] Image(img_path, width=150, height=150) #Вывод загруженного изображения
```

Рисунок 5 – Обработка изображения

Далее преобразуем изображение в другое разрешение, для лучшего считывания программой. Преобразуется загруженное изображение в размер 28 на 28 пикселей (Рис.6).

```
[ ] img = image.load_img(img_path, target_size=(28, 28), color_mode = "grayscale")  
#Загрузка изображения из файла(Имя файла, Размер в пикселях, цвет)
```

Рисунок 6 – Загрузка изображения

Следующим шагом будет преобразование изображения в массив, так как нейросети проще будет работать с числами. Этим самым мы нормализуем изображение на «понятный язык» для системы (Рис.7).

```
[ ] # Преобразуем картинку в массив
x = image.img_to_array(img)
# Меняем форму массива в плоский вектор
x = x.reshape(1, 784)
# Инвертируем изображение
x = 255 - x
# Нормализуем изображение
x /= 255
```

Рисунок 7 – Преобразование изображения

Так как сеть у нас уже обучена, и имеет возможность полноценно работать, то запустим предсказание. Она считает полученное изображение и сделает свое предсказание, что она видит (Рис.8-9).

```
[ ] prediction = model.predict(x) #Запускаем предсказание
```

Рисунок 8 – Запуск предсказания

```
[ ] prediction #Выводим предсказание
```

Рисунок 9 – Вывод предсказания

Теперь для лучшего видения ответа, сделаем красивый вывод. Выводим из всех предсказаний один с максимальным значением, и печатаем в вывод к какому классу принадлежит картинка и что на ней нарисовано (Рис.10).

```
prediction = np.argmax(prediction)
print("Номер класса:", prediction)
print("Название класса:", classes[prediction])
#Выводим предсказание с максимально точным значением, номер его класса и название
```

```
Номер класса: 8
Название класса: сумка
```

Рисунок 10 – Результат

В данной работе была разработана нейросеть в бесплатном приложении от google с помощью библиотек keras и tensorflow. Итоговым решением является рабочая нейросеть по определению изображения.

Библиографический список

1. Латкина И.И. Обзор возможностей tensorflow для решения задач машинного обучения // Инноватика. 2015. №1. С. 33-42.
2. Крыловецкий А.А., Сузодолов Д.М. Распознавание изображений элементов зерновых смесей методами глубокого обучения с использованием библиотек keras и tensorflow // Научные труды Кубанского государственного технологического университета. 2015. №S6. С. 97-100.
3. Тур А.И., Кокоулин А.Н., Южаков А.А., Лукичев А.Н. Подготовка системы распознавания объектов на базе tensorflow и keras // Современные проблемы науки и образования. 2015. №4 URL: <https://www.science-education.ru/ru/article/view?id=14142>.
4. Мамедов В.С., Заслонов С.А. Применение tensorflow в задачах распознавания изображений с помощью технологии blockchain // Современные материалы, техника и технологии. 2015. №3 (3) . С. 64-68.
5. Гатин Р.Р., Бикмухаметов Р.Р. Функции и возможности библиотеки tensorflow для программирования рекуррентных сетей с помощью языка python // Студенческая наука для развития информационного общества, сборник материалов 6 всероссийской научно-технической конференции. Ставрополь: Северо-Кавказский федеральный университет, 2017. С. 330-331.