УДК 004.02 **Информационные технологии**

Краснова А. Ю., *студент 4 курса напр. «Математика и компьютерные науки», ИМиМ КФУ (Институт математики и механики им. Лобачевского Казанского Федерального Университета), Россия, Казань*

МЕТОД ИССЛЕДОВАНИЯ ФОТОГРАФИЙ С ПИГМЕНТНЫМИ НОВООБРАЗОВАНИЯМИ ПРИ ПОМОЩИ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

Аннотация: В данной статье представлен один из методов исследования фотографий с раком кожи при помощи сверточных нейронных сетей. Обучение и составление программы представлено на языке Python.

Ключевые слова: сверточная нейронная сеть, модель сети, язык программирования Python.

Annotation: This article presents one of the methods for studying photos with skin cancer using convolutional neural networks. Training and programming is provided by Python.

Keywords: convolutional neural network, network model, Python.

На сегодняшний день сверточные нейронные сети активно применяются для решения большого количества задач. Одна из них, которую мы рассмотрели – это анализ фотографий рака кожи.

Цель проведения эксперимента заключалась в том, чтобы нейронная сеть обучилась на готовой базе данных и выдавала правильный результат: имеются ли пигментные новообразования на кожи или нет.

Описание базы данных

На вход нейронной сети подавались 10 015 изображений (рис. 1) из которых 9 015 использовались для обучения и 1 000 на тестовую часть.

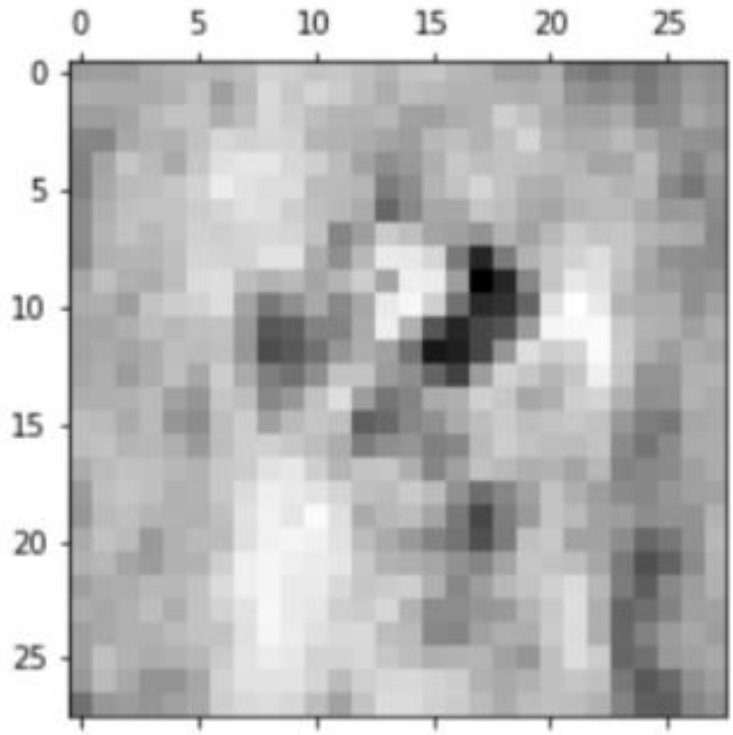

Рис.1. Пример фотографии кожи с раком

Необходимый набор данных был взят из ресурса сайта Kaggle. В наборе были собраны дерматоскопические изображения в формате jpg из разных групп населения. Предоставленный набор являлся учебным комплектом для академического машинного обучения.

В наборе присутствовали фотографии как с раком кожи, так и без. Для того, чтобы использовать данные, нужно было создать ключ API и загрузить файл kaggle.json.

Модель сверточной нейронной сети

Для создания сверточной нейронной сети нам потребовалось подключить ряд библиотек. Одна из них Pandas, которая является высокоуровневой Python библиотекой для анализа данных. Она поддерживает такие форматы хранения как: csv, excel, sql, html и другие. Для работы с нашей нейронной сетью нам понадобилось работать с файлами типа csv.

Библиотека Keras – популярная библиотека для Python с открытым кодом, которая участвует в создании и обучении нейронных сетей.

Помимо этих библиотек стоит также упомянуть о NumPy, которая предоставляет базовые методы для работы с большими матрицами и массивами, что было необходимым при работе с изображениями.

После подключения нужных библиотек, мы загрузили файл kaggle.json на Google Colab, выгрузили все имеющиеся файлы и изменили размер наших изображений.

Структура модели сверточной нейронной сети состояла из 8 слоёв. Для начала мы создали объект последовательной модели и постепенно добавляли слой за слоем. На рис. 2 мы можем увидеть из чего состоит наша нейронная сеть и каким образом чередуются слои. Рассмотрим каждый слой отдельно.

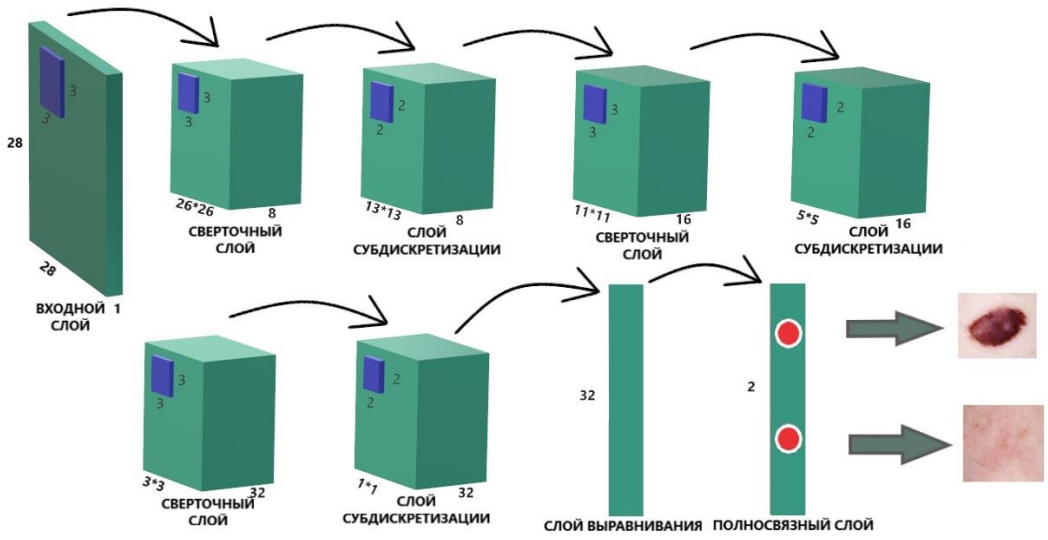

Рис. 2. Визуальное представление СНС

Первым слоем послужил сверточный. Ему на вход подавалась информация об изображениях: $input\_shape = (28, 28, 1)$. В данном случае это рисунок размера 28*28 пикселей в оттенках серого цвета. В этом слое происходит поэлементное умножение значений фильтра на исходные значения пикселей и затем они суммируются. На выходе слоя мы получили размеры карт признака.

Вторым слоем был слой субдискретизации (рис.3), который позволил уменьшить дискретизацию пространственных размеров (ширину и высоту). Таким образом произошло уплотнение карты признаков.

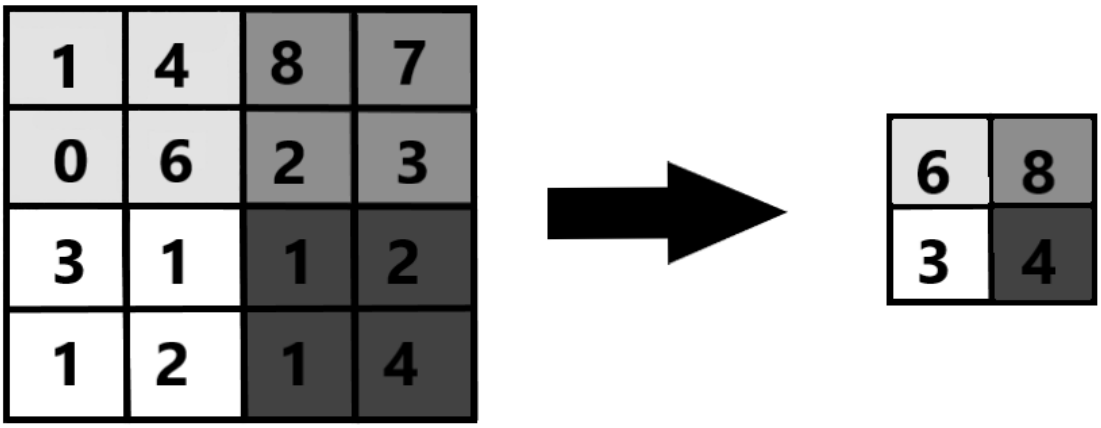

Рис. 3. Пример работы слоя субдискретизации (MaxPolling2D)

После чего мы еще два раза применили слой свертки и субдискретизации, соответственно чередуя и удваивая количество ядер вдвое.

На седьмом слое – выравнивание (рис. 4), мы произвели преобразование 2D-данных в 1D-данные.

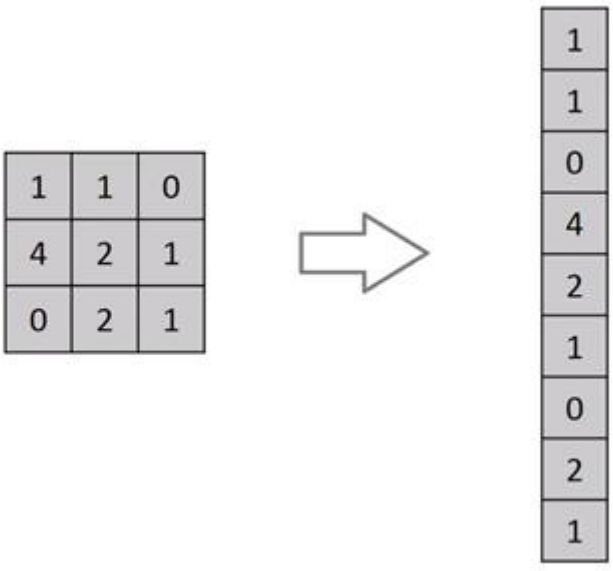

Рис. 4. Слой выравнивания (Flatten)

На восьмом слое – полносвязном, который выполнил сортировку полученных данных и вывел n-мерный вектор. В данной задачи – это двумерный.

После сформирования модели мы ее скомпилировали и передали три параметра: оптимизатор, функцию потерь и метрику. Рассмотрим данные параметры подробнее.

В качестве оптимизатора мы использовали «adam» [1]. Он является одним из самых популярных алгоритмов в области глубокого обучения, поскольку быстро достигает высоких результатов. С помощью оптимизатора мы контролируем скорость обучения сети. Другими словами, скорость обучения определяет быстроту расчета оптимальных весов для модели, чтобы на обучение затрачивалось как можно меньше времени.

Для функции потерь мы использовали категориальную кросс-энтропию («$categorical\_crossentropy$») [2], которая используется для большинства нейронных сетей, обученных выполнять классификацию. Стоит отметить, что данная функция потерь вычисляется по следующей формуле:

$$-\sum_{i=1}^{n}(x_i \cdot \log(y_i)),$$

где x_i – истинное значение вектора, y_i – прогнозируемое значение вектора и n – размер вектора x_i(или y_i).

Для упрощения интерпретации мы использовали такую метрику как «точность» («$accuracy$»), чтобы получить точность оценки при валидации во время обучения нашей модели.

Обучение созданной нейронной сети происходило при помощи метода fit, в который передавались введенные ранее параметры и данные для обучения и тестирования, а также количество эпох – 17.

Результат эксперимента

В ходе обучения нейронной сети мы получили, что точность вычисления составляет 87.5% на тестовом множестве.

Для того, чтобы убедиться в корректной работе, проверим нейронную сеть на тестовом множестве. Возьмем, к примеру, изображение №12 (рис. 5) на котором имеется рак.

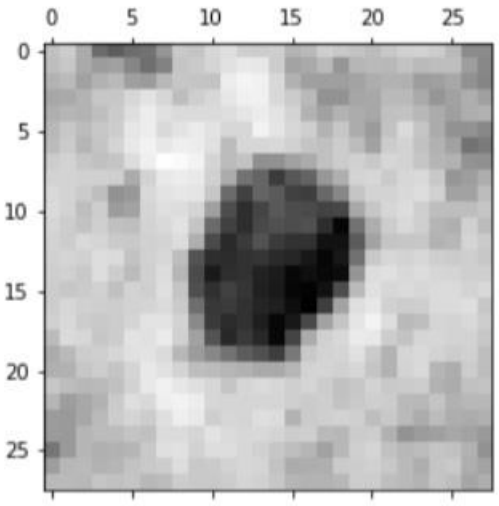

Рис. 5. Изображение №12 из тестового файла

С помощью метода *predict* мы отправили на вход обученной сети это изображение. В итоге получили вектор: [[0.8623163, 0.1376837]]. Получается, что нейронная сеть предложила, что на фотографии рак с точностью 86.2% и нет рака – 13.7%. Если выведем вектор тестовый базы, то убедимся, что на фотографии действительно рак и наша нейронная сеть показала хорошую точность.

Следовательно, мы можем утверждать, что сеть прошла обучение успешно.

Заключение

Таким образом, мы создали собственную модель сверточной нейронной сети, позволяющую определять по фотографиям наличие новообразований на коже. Для этого мы воспользовались рядом функций, которые предлагают следующие библиотеки: NumPy, Keros, Pandas. Модель сети является последовательной и содержит 8 слоев для ее обучения.

В ходе эксперимента выяснилось, что точность нейронной сети составляет 87.5% - это является неплохим показателем для обучения.

Библиографический список:

1. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. – СПб.: Питер, 2018. – 480 с.

2. Создаем сверточную нейронную сеть (CNN) в Keras [Электронный ресурс]. – Режим доступа - URL: https://robo-hunter.com/article/sozdaem-svertochnuy-neironnuy-set-cnn-v-keras (дата обращения 20.05.2020).