

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ярославский государственный университет им. П.Г. Демидова»

Кафедра дискретного анализа

«Допустить к защите»

Заведующий кафедрой,

д.ф.-м.н., профессор

_____ Бондаренко В.А.

« 13 » июня 2020 г.

Выпускная квалификационная работа бакалавра
по направлению 01.03.02 Прикладная математика и информатика

**Анализ алгоритмов глубокого машинного обучения в задачах
распознавания изображений**

Научный руководитель

к.т.н., старший преподаватель

_____ Д.В. Матвеев

« ____ » _____ 2020 г.

Студент группы ИВТ-41БО

_____ А.С. Коротков

« ____ » _____ 2020 г.

Ярославль, 2020

РЕФЕРАТ

Выпускная квалификационная работа 20 стр., 6 гл., 2рис., 0 источников

Ключевые слова: машинное обучение, глубокие нейронные сети, распознавание изображений, TensorFlow, OpenCV, Keras.

Объектом исследования являются методы на основе глубоких нейронных сетей для задач распознавания изображений.

Цель работы – изучить применение глубокого машинного обучения в задачах распознавания изображений.

В результате работы была разработана и реализована нейронная сеть для решения ????. Проведен анализ полученных результатов и сделан вывод о качестве работы нейронных сетей в задачах распознавания изображений.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1 Задача компьютерного зрения	5
1.2 Искусственные нейронные сети	5
1.2.1 Понятие искусственной нейронной сети	5
1.2.2 Активационная функция	6
1.2.3 Структура нейронной сети	7
1.2.4 Глубокие нейронные сети	8
1.2.5 Сверточные нейронные сети	8
1.2.6 Проблемы обучения нейронных сетей	8
1.3 Применение нейронных сетей в задачах распознавания изображений	8
2 ПОСТАНОВКА ЗАДАЧИ	9
3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ ДЛЯ ???	10
4 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ	11
5 ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ	12
6 АНАЛИЗ РАБОТЫ ПРОГРАММЫ	13
ЗАКЛЮЧЕНИЕ	14
СПИСОК ЛИТЕРАТУРЫ	15
СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА	16
СПИСОК ТАБЛИЧНОГО МАТЕРИАЛА	17
ПРИЛОЖЕНИЕ А	18

ВВЕДЕНИЕ

В настоящее время, в связи со стремительным развитием цифровых технологий, использование автоматизированных и роботизированных систем распространилось на множество областей как в промышленности, науке, так и в повседневной жизни. В следствие этого, возрастает необходимость в эффективной обработке информации, представленной, в частности, в формате видео и изображений.

На текущий момент изображения тесно влились в жизнь человека. Поэтому многие автоматизированные системы используют их в качестве основного источника информации. Нахождение, локализация, классификация и анализ объектов на изображении компьютером – сложная задача компьютерного зрения.

В процессе обработки информации, получаемой из глаз, человеческий мозг продельывает колоссальный объем работы. Человек без труда сможет описать что находится и что происходит на случайно взятой фотографии. Изображения могут нести в себе колоссальное количество деталей и отличаться множеством параметров, таких как: разрешение, цветность, качество, яркость, наличие шума и т.д. Объекты на изображениях также могут обладать множеством особенностей: масштаб, положение, цвет, поворот, наклон и т.д. Однако, в цифровом формате, каждое изображение представляет собой лишь массив числовых данных. Научить компьютер находить и классифицировать образы на изображении с учетом всех факторов – очень сложная алгоритмическая задача. Для её решения активно применяют технологии машинного обучения.

В данной работе был проведен анализ алгоритмов глубокого машинного обучения для решения задач распознавания изображений, а также разработана система для ???.

В первой главе проведен обзор основных тем и задач ???

Во второй главе ???

В третьей главе ???

1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Задача компьютерного зрения

Компьютерное (машинное) зрение – это совокупность программно-технических решений в сфере искусственного интеллекта (ИИ), нацеленных на считывание и получение информации из изображений, в реальном времени и без участия человека.

Большое количество информации человек получает при помощи зрения.

В основе компьютерного зрения лежит В настоящий момент, такие технологии применяются для решения таких сложных задач как:

- OCR – Optical character recognition (Оптическое распознавание символов): преобразование текста на изображении в редактируемый.
- Фотограмметрия – технология создания трехмерной модели объекта на основе фотографий, сделанных с различных ракурсов.
- Motion capture – технология, широко применяемая в киноиндустрии, позволяющая преобразовывать движения реальных людей в компьютерную анимацию.
- Дополненная реальность (AR) – технология, позволяющая в реальном времени проецировать виртуальные объекты на изображение реального окружения.
- Медицинская диагностика – обнаружение раковых клеток на ранней стадии, увеличение качества МРТ изображений, их анализ и т.д.

1.2 Искусственные нейронные сети

1.2.1 Понятие искусственной нейронной сети

Машинное обучение – раздел исследований в сфере ИИ, в основе которых лежат методы разработки систем способных к обучению. ???

Искусственная нейронная сеть (ИНС) – компьютерная модель, в основе которой лежат принципы работы биологической нейронной сети - совокупности связанных между собой нервных клеток - нейронов. Каждый нейрон имеет набор входных связей - синапсов, по которым он получает информацию, представленную в виде импульсов, от других нейронов. По полученным данным нейрон формирует своё состояние и, с помощью аксона, сообщает его другим нейронам, обеспечивая функционирование системы. В про-

цессе формирования системы одни нейронные связи укрепляются, а другие ослабляются, обеспечивая обучаемость сети.

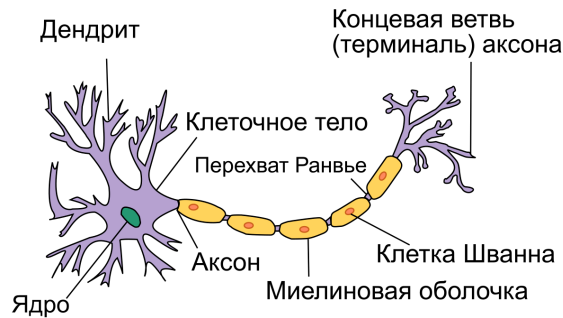


Рисунок 1.1 – Типичная структура нейрона

Искусственный нейрон представляет собой упрощенную модель биологического нейрона. На вход подаются n -мерные вектор значений $X = (x_1, \dots, x_n)$ и вектор весов $W = (w_1, \dots, w_n)$. Входные значения несут информацию, а веса указывают на "силу" межнейронных связей. Процесс подбора весов называется обучением нейронной сети. Значение выхода нейрона вычисляется по формуле:

$$out(x) = \sigma\left(\sum_{i=1}^n x_i w_i\right)$$

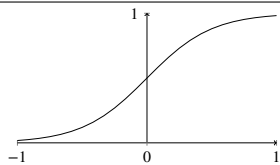
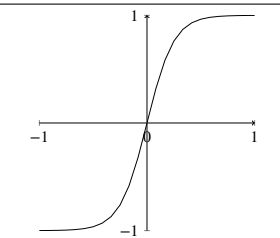
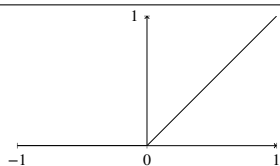
Где σ - функция активации.

1.2.2 Активационная функция

При вычислении взвешенной суммы входов нейрона, результат может принимать абсолютно любое значение $x \in (-\infty; +\infty)$, что препятствует дальнейшим вычислениям. Активационная функция нейрона обеспечивает нормализацию посчитанной суммы, таким образом, что значение выхода нейрона всегда принадлежит некоторому, заранее заданному, диапазону, часто: $(0; 1)$ или $(-1; 1)$. Для многих моделей нейронных сетей также требуется, чтобы активационная функция была нелинейной, монотонной и непрерывно-дифференцируемой на всей области определения.

Существует большое количество функций активации. Наиболее распространенные из них представлены в табл. 1.1

Таблица 1.1 – Нелинейные активационные функции

Название	Функция	Вид
Сигмоидная	$\sigma(x) = \frac{1}{1+e^{-x}}$	
Гиперболический тангенс	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
ReLU	$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0. \end{cases}$	

1.2.3 Структура нейронной сети

Множества нейронов формируют слои, слои в свою очередь формируют нейронную сеть. Входной слой получает данные, обрабатывает и передает нейронам скрытого слоя. Аналогично сработает каждый последующий слой вплоть до выходного.

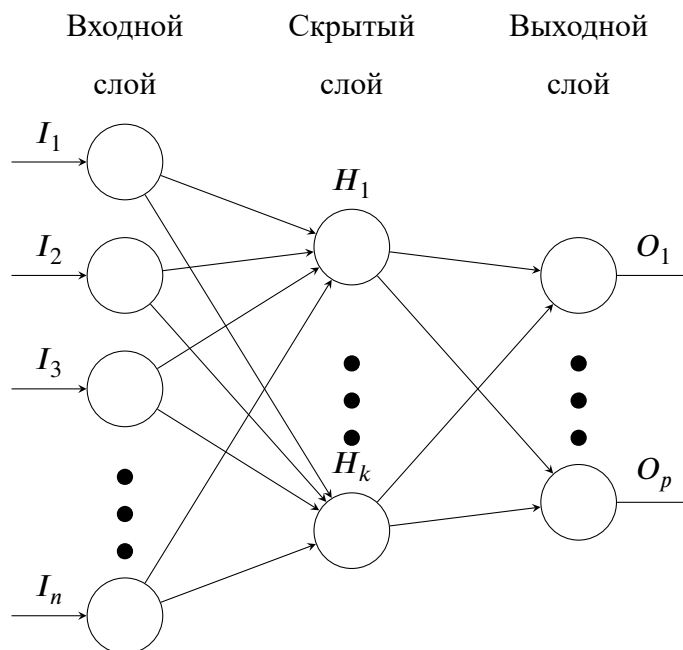


Рисунок 1.2 – Схема простой нейронной сети

1.2.4 Глубокие нейронные сети

Под глубокими нейронными сетями понимают сети с большим количеством скрытых слоев.

1.2.5 Сверточные нейронные сети

???

1.2.6 Проблемы обучения нейронных сетей

???

1.3 Применение нейронных сетей в задачах распознавания изображений

???

2 ПОСТАНОВКА ЗАДАЧИ

3 ПРОЕКТИРОВАНИЕ СИСТЕМЫ ДЛЯ ???

4 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ

5 ПРИМЕРЫ РАБОТЫ ПРОГРАММЫ

6 АНАЛИЗ РАБОТЫ ПРОГРАММЫ

ЗАКЛЮЧЕНИЕ

СПИСОК ЛИТЕРАТУРЫ

СПИСОК ИЛЛЮСТРАТИВНОГО МАТЕРИАЛА

Рисунок 1.1	Типичная структура нейрона	6
Рисунок 1.2	Схема простой нейронной сети	7

СПИСОК ТАБЛИЧНОГО МАТЕРИАЛА

Таблица 1.1	Нелинейные активационные функции	7
-------------	--	---

ПРИЛОЖЕНИЕ А

Листинг 1

```
1 from __future__ import absolute_import, division, print_function,
   unicode_literals
2
3 import numpy as np
4 import cv2
5 from tensorflow.keras.callbacks import ModelCheckpoint
6 from tensorflow.keras.layers import Conv2D, Flatten, MaxPooling2D,
   Dense, Dropout
7 from tensorflow.keras.models import Sequential
8 from tensorflow.keras.preprocessing.image import ImageDataGenerator,
   img_to_array, load_img, array_to_img
9 import random, os, glob
10 import matplotlib.pyplot as plt
11
12 DIR_PATH = '/home/alexandr/dev/datasets/garbage-classification/
   garbage_classification/Garbage_classification'
13 img_list = glob.glob(os.path.join(DIR_PATH, '*/*.jpg'))
14
15
16 def load_data():
17     train=ImageDataGenerator(
18         horizontal_flip=True, vertical_flip=True, validation_split
19         =0.1, rescale=1./255,
20         shear_range=0.1,
21         zoom_range=0.1,
22         width_shift_range=0.1,
23         height_shift_range=0.1)
24     test=ImageDataGenerator(rescale=1/255, validation_split=0.1)
25     train_generator=train.flow_from_directory(DIR_PATH,target_size
        =(300,300),batch_size=32,
        class_mode='categorical',
        subset='training')
```

```

26     test_generator=test.flow_from_directory(DIR_PATH,target_size
      =(300,300),batch_size=32,
27
      class_mode='categorical',
      subset='validation')
28
29     labels = (train_generator.class_indices)
30
31     labels = dict((v,k) for k,v in labels.items())
32
33     print(labels)
34
35     model=Sequential()
36
37     model.add(Conv2D(32,(3,3), padding='same',input_shape=(300,300,3)
      ,activation='relu'))
38
39     model.add(MaxPooling2D(pool_size=2))
40
41     model.add(Conv2D(64,(3,3), padding='same',activation='relu'))
42
43     model.add(MaxPooling2D(pool_size=2))
44
45     model.add(Conv2D(32,(3,3), padding='same',activation='relu'))
46
47     model.add(MaxPooling2D(pool_size=2))
48
49     model.add(Flatten())
50
51     model.add(Dense(64,activation='relu'))
52
53     model.add(Dense(6,activation='softmax'))
54
55     filepath="trained_model.h5"
56
57     checkpoint1 = ModelCheckpoint(filepath, monitor='val_acc',
      verbose=1, save_best_only=True, mode='max')
58
59     callbacks_list = [checkpoint1]
60
61     model.summary()
62
63     model.compile(loss='categorical_crossentropy', optimizer='adam',
      metrics=['acc'])
64
65     model.fit(train_generator, epochs=150, steps_per_epoch=2276//32,
      validation_data=test_generator,
66
      validation_steps=251//32,callbacks=callbacks_list
      )
67
68     model.save("model.h5")
69
70 if __name__ == '__main__':

```

```
55     os.environ['TF_FORCE_GPU_ALLOW_GROWTH'] = 'true'
56     load_data()
```