МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

КОНТРОЛЬНАЯ РАБОТА

по дисциплине «Программирование на языке Python»

на тему: «Реализация программы для распознавания рукописных цифр»

Студент _____ Евдокимов Н.А.

Шифр 170576

Институт приборостроения, автоматизации и информационных технологий

Направление подготовки 09.03.04 «Программная инженерия»

Группа 71ПГ

Руководитель _____ Захарова О.В.

Оценка: «_____» Дата _____

Орел 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время наблюдается повсеместное использование машинного обучения в различных сферах жизни человека. Согласно источнику [1], оно является классом методов искусственного интеллекта, главной характеристикой которых является не прямое вычисление, а обучение на наборе примеров уже решенных задач. Машинное обучение использует методы теории вероятностей, численных методов, математической статистики и теории данных. В связи с распространением удобных в использовании фреймворков для машинного обучения, его легко использовать для решение прикладных задач.

Целью контрольной работы является написание программы, позволяющей осуществлять определение рукописных цифр. На вход программы должны поступать изображения, на выходе она должна предсказывать, что за цифра на нем изображена. Такая разработка может использоваться совместно с машинным зрением для обработки рукописных бланков, номеров автомобилей, анкет и т.д.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Выполнить анализ поставленной цели.
2. Выработать функциональные требования.
3. Спроектировать разрабатываемую систему.
4. Провести анализ и выбор из инструментов для реализации.
5. Реализовать, провести тестирование, отладить программу.

1 АНАЛИЗ ЗАДАЧИ

Исходя из информации, приведенной в источнике [2], наиболее характерным для поставленной задачи решением является использование машинного обучения. Для успешного использования методов машинного обучения необходимо определить тип поставленной задачи, так как это оказывает влияние на архитектуру системы и на то, какие методики следует применять.

Существует пять типов задач машинного обучения:

1. Задачи регрессии.
2. Задачи классификации.
3. Задачи кластеризации.
4. Задачи уменьшения размерности.
5. Задача выявления аномалий.

Задача регрессии представляет собой предсказание на основе статистической выборки объектов с различными признаками. Примерами таких задачи являются: предсказание цены автомобиля, получаемая прибыль, количество сотрудников, которые уйдут в отпуск и т.д.

Задача классификации сводится к получению категориального ответа по какому-либо набору признаков. Например, изображен ли на картинке фрукт или на каком языке написан текст.

Задачи кластеризации являются задачами разделения объектов на группы по определенным признакам. Решение таких задач очень востребовано в астрономии: разделить фото с телескопа по тому, является ли запечатленный объект звездой, кометой, планетой и т.д.

Задачи уменьшения размерности представляют собой отсечение неважных для задачи признаков объекта или генерация нового меньшего набора признаков на основе старого. Примером такой задачи является генерация набора признаков на основе больничной истории пациента для определения того, какому заболеванию он подвержен.

Задачи выявления аномалий заключаются в определении отклонения от стандартного случая. Может показаться, что этот тип задачи совпадает с задачами классификации. Отличием является то, что количество примеров для обучения крайне мало, либо их нет вовсе. В связи с этим подход аналогичный тому, что используется в методах классификации, с задачами выявления аномалий не работает.

Так как множество выходных значений жестко определено (цифры от нуля до девяти), то поставленная задача является задачей классификации.

2 ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

Главным функциональным требованием к разрабатываемой системе является следующее – программа должна по некоторому изображению определять, какая цифра на нем изображена.

Реализация должна осуществляться с использованием методов машинного обучения и демонстрировать основные его принципы.

Система должна проходить обучение на основе заранее заготовленных данных (собранных самостоятельно или из открытых источников).

На данный модуль не должна распространяться ответственность за информирование о том, что на изображении нет цифры, т.к. это является задачей классификации другого рода и требует подготовки отдельного набора данных.

Помимо того, система не должна работать с числами, т.к. числа можно разбить на составляющие их цифры и по отдельности отправить на вход в разрабатываемый модуль, что удовлетворяет принципу единой ответственности. Помимо того, числа могут быть бесконечно большими. В силу невозможности подготовки данных в виде бесконечно больших чисел, разработать систему, которая должна работать с любым числом без разбиения на цифры, невозможно.

Для демонстрации работы системы должен быть разработан удобный пользовательский интерфейс, дающий пользователю, не являющемуся разработчиком, возможность ее использования.

3 МЕТОДЫ РЕАЛИЗАЦИИ ПРОГРАММЫ

Основной структурой данных в контексте машинного обучения является модель. Модель представляет собой набор операций, которые необходимо совершить над входными данными и дающий на выходе результат их вычисления.

Для получения модели необходимо спроектировать ее структуру, реализовать и обучить. Допускаются различные преобразования в целях оптимизации или возможности использования модели на платформах отличных от той, на которой она реализовывалась, но, как правило, эти задачи выполняют автоматизированные инструменты разработчика.

Для использования модели необходимо написать модули, определяющие ее структуру, содержащие код для обучения модели и модули для ее использования. Ниже сказано о том, какая структура должна быть у самой модели для достижения поставленных целей.

Классическим инструментом, который будет использован и для выполнения курсовой работы, для решения задач классификации, связанных с машинным зрением, являются сверточные нейронные сети (далее СНС). Принцип их функционирования аналогичен принципу работы зрительного центра коры головного мозга. В связи с этим далее будут приводиться аналогии с человеческим восприятием зрительной информации.

Когда человек рождается, он не способен классифицировать увиденное им, например, видит ли он собаку или же кошку. По мере взросления, он при помощи примеров (устного объяснения, различных картинок и т.д.) выявляет определенные характеристики, свойственные тем или иным классам предметов. Так, если он видит, что у предмета его наблюдения есть четыре лапы и хвост, то, скорее всего, это животное.

На основе выявления обобщенных характеристик работают и СНС. На первый слой СНС подается изображение. В случае, если это цветная картинка 32x32 пикселя, то его можно представить как матрицу

размерностью 32x32x3, каждым значением которой является интенсивность цвета в одном пикселе.

Данный слой обрабатывается фильтром – другой матрицей со своими значениями и размерностью меньше размерности исходного изображения. Фильтр можно представить как фонарик, свет от которого проходит по изображению.

Свет от фонарика, покрывающий некоторую часть изображения, называют рецептивным полем. Значения матрицы фильтра перемножаются со значениями матрицы и складываются вместе, результатом чего является единственное число. Так, если в фильтр заложены значения, характерные изогнутым линиям, а в рецептивном поле будет что-то похожее на изогнутую линию, то результатом операции будет большое значение, а в обратном – маленькое. Иными словами, высокие значения в определенной части изображения для конкретного фильтра (их может быть несколько) говорят о высокой вероятности нахождения в этой области объекта, на нахождение которого настроен этот фильтр (в данном случае говорят об активации фильтра). При прохождении изображения через фильтр формируется карта признаков (вероятность нахождения определенного признака в некоторой части изображения). Визуализация работы фильтра изображена на рисунке 1.

Например, при размерности первого слоя 32x32x3 и фильтре размером 5x5x3 размерность выходной матрицы будет 28x28x1, т.е. 784. Такое значение получается исходя из того факта, что есть 784 позиции, которые могут пройти через фильтр 5x5x3 изображения размерностью 32x32x3.

Первый слой СНС, описанный выше, позволяет выделить характеристики базового уровня (кривые, прямые и т.д.). При подаче выходных значений первого слоя на вход последующих слоев могут быть выявлены характеристики более высокого уровня, такие как: квадраты (четыре прямых), кольца (две окружности) и т.д.

Чем глубже проходит изображения по слоям СНС, тем более абстрактные и сложные характеристики возможно обнаружить. В последних

слоях СНС могут быть фильтры, активирующиеся, например, при обнаружении пони, или розовых шариков.

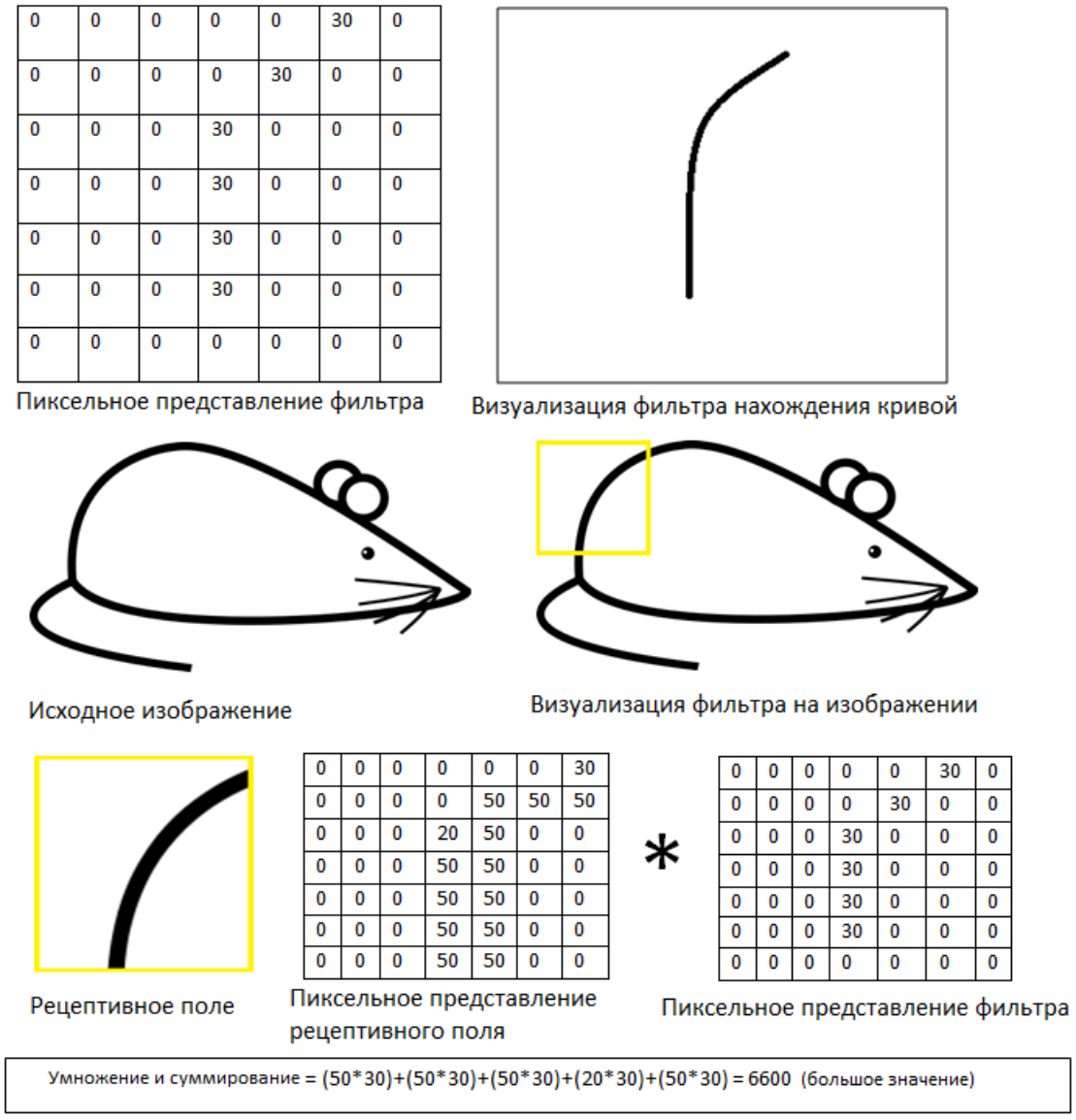

Рисунок 1 – Принцип работы фильтра

Последним слоем в случае задачи классификации должен быть полносвязный слой. На основе выходных данных предпоследнего слоя (наличие характеристик высокого уровня) он определяет свойства наиболее характерные для какого-либо класса объектов. На выходе у него вектор вероятностей принадлежности объекта с изображения к какому-либо классу.

Ответом на вопрос о том, как в фильтрах слоев появляются необходимые значения и о том, как полносвязный слой определяет, какие

характеристики свойственны классам объектов, является обучение нейронной сети.

В исходном состоянии модели значений фильтров случайны. При обработке изображения выходные значения, с большой долей вероятности, будут сильно отличаться от тех, что должны быть на самом деле. Для определения такой ситуации используется функция потери. Как правило. ее определяют как среднеквадратическую ошибку полученных результатов относительно реальных значений. Сведение функции потери к минимуму – главная задача обучения.

С каждым обработанным изображением значения фильтров (весов) модели немного изменяются, что оказывает влияние на выходное значение. Программа должна выполнять этот процесс для каждого тренировочного изображения.

4 АНАЛИЗ И ВЫБОР ИНСТРУМЕНТОВ РЕАЛИЗАЦИИ

На данный момент существует множество готовых инструментариев для работы с машинным обучением, у каждого из которых имеется ряд особенностей. Так, на данный момент, наиболее популярными фреймворками машинного обучения являются:

1. Tensorflow.
2. PyTorch.
3. Keras.
4. CNTK.
5. ONNX.

Согласно источнику [3], Tensorflow – проект от компании Google, заключающий в себе гибкую экосистему инструментов, библиотек и ресурсов для изучения. Особенностью является легкость развертывания моделей в любой среде, будь то браузер, облако или физическое устройство, не зависимо от используемого языка. Получаемая в процессе работы модель имеет статическую структуру.

PyTorch – это разработка Facebook. В отличие от Tensorflow оперирует динамическим графом. По умолчанию поддерживает модель распределенного обучения и параллелизма вычислений. Недостатком является сложность портирования модели на различные платформы.

Keras – фреймворк, дающий возможность свести к минимуму усилия по обучению и прототипированию модели, т.к. эти задачи выполняются путем написания однострочных выражений. Таким образом, это лучший выбор для начинающих разработчиков. Помимо того, в новых версиях Tensorflow имеется интеграция с Keras.

CNTK – фреймворк машинного обучения от Microsoft. Поддерживает работу с C++, Python, C# и Java. Содержит множество готовых решений для распознавания речи и изображений. Недостатком является то, что CNTK, как

и остальные продукты от Microsoft, ориентирован на использование на Windows.

Исходя из источника [4], ONNX – продукт сотрудничества Facebook и Microsoft. Направлен на облегчение портирование моделей между различными инструментами машинного обучения. Минусом является сложность использования.

В связи с тем, что Keras получила поддержку Google, которая выражается в том, что Tensorflow фактически будет работать на Keras, что дает возможность использования полученного продукта в любой среде, а также из-за простоты использования, данный фреймворк будет использован для выполнения контрольной работы.

5 РЕАЛИЗАЦИЯ, ТЕСТИРОВАНИЕ, ОТЛАДКА

Так как архитектура модели уже была выбрана в пункте 3 контрольной работы, следующее, что необходимо сделать – это подготовить данные, на которых будет происходить обучение. Keras предоставляет доступ к набору данных MNIST (сокращение от «Modified National Institute of Standards and Technology»). Согласно [5], MNIST – это объемный набор образцов рукописного написания цифр, состоящий из пар изображение – ярлык. Ярлык в данном случае означает находящуюся на изображении цифру.

Код, написанный для получения данных из MNIST, представлен ниже.

```
import keras
from keras.datasets import mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data('mnist_data')
```

Модель, реализующая описанную в пункте 3 архитектуру, была построена путем написания следующего кода:

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), input_shape=(28, 28, 1)))
model.add(Conv2D(64, (3, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128))
model.add(Dropout(0.5))
model.add(Dense(num_classes))
```

После формирования модели, было проведено ее обучение на полученных тестовых данных и сохранение ее структуры и весов в файлы на жесткий диск.

Далее был написан класс Inference, осуществляющий загрузку модели с диска при инициализации. У этого класса есть метод, который необходимо вызвать для предсказания. Этот метод в качестве аргумента принимает изображение, а возвращает цифру, которую предсказала модель.

При каждом предсказании происходит препроцессинг изображения. Оно сжимается до рамеров 28x28 пикселей, а цвета инвертируются. При этом

определяются границы цифры, и она переносится в центр. Эти действия необходимы, т.к. для получения адекватных результатов на вход модели нужно подавать данные в том же формате, в каком были обучающие данные (в MNIST изображения 28x28 пикселей, с инвертированными цветами, а цифры центрированы).

Для использования полученного программного модуля средствами PyQt5 был реализован пользовательский интерфейс. Он содержит поле для рукописного ввода цифры, а также две кнопки: отчистить и предсказать. При нажатии на кнопку отчистки происходит отчистка поля для рукописного ввода, а при нажатии на кнопку предсказания происходит отчистка вышеупомянутого поля и вывод в него цифры, предсказанной моделью. Пользовательский интерфейс программы изображен на рисунках 2 и 3.

Рисунок 2 – Пользовательский с рукописным вводом

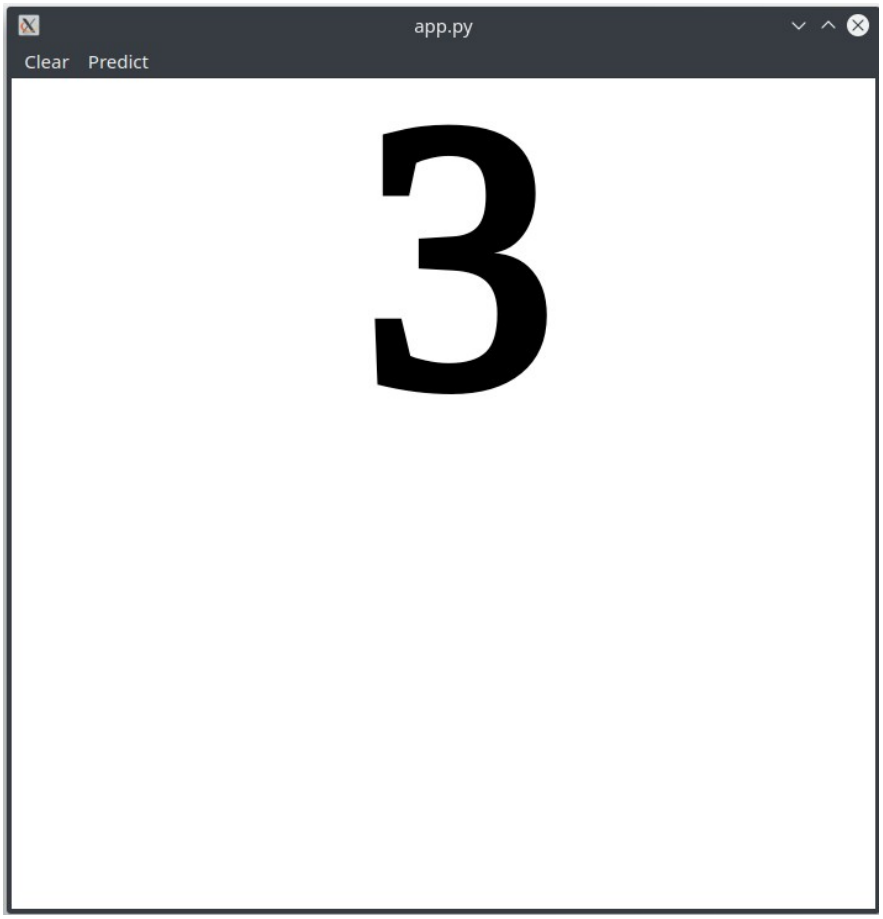

Рисунок 3 – Пользовательский интерфейс с предсказанием модели

Тестирование и отладка осуществлялись стандартными средствами, согласно методологии «водопад» (по мере написания программы). Данная методология была выбрана из-за того, что объем работы, необходимый для написания описанного программного средства не велик. Следовательно, перегружать маленький проект сложной в использовании методологии нерационально.

ЗАКЛЮЧЕНИЕ

Результатом выполнения контрольной работы является программа, реализующая использование обученной модели машинного обучения. Для получения данного решения был проведен анализ поставленной цели, выработаны требования, было проведено проектирование программной системы, выбраны и использованы инструменты реализации.

Так как написание вышеописанной программы соответствует поставленной цели, а все необходимые для достижения цели задачи выполнены, можно сделать вывод, что контрольная работа выполнена успешно.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. MachineLearning.ru [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Machine_Learning, свободный (дата обращения 10.11.2019).

2. Salman K. A Guide to Convolutional Neural Networks for Computer Vision (Synthesis Lectures on Computer Vision) [Текст]/ K. Salman, H. Rahmani, Syed Afaq Ali Shah – Morgan & Claypool, 2018. – 185с.

3. Tensorflow [Электронный ресурс]. – Режим доступа: https://www.tensorflow.org/ свободный (дата обращения 10.11.2019).

4. ONNX [Электронный ресурс]. – Режим доступа: https://onnx.ai/, свободный (дата обращения 10.11.2019).

5. MNIST [Электронный ресурс]. – Режим доступа: http://yann.lecun.com/exdb/mnist/, свободный (дата обращения 10.11.2019).