

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

**ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ  
РАСПОЗНАВАНИЯ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 248 группы

направления 09.04.03 — Прикладная информатика

механико-математического факультета

Карахтанова Андрея Александровича

Научный руководитель

к. ф.-м. н., доцент

\_\_\_\_\_

С. В. Тышкевич

Заведующий кафедрой

д. ф.-м. н., доцент

\_\_\_\_\_

С. П. Сидоров

Саратов 2021

## ВВЕДЕНИЕ

**Актуальность темы.** Данная работа посвящена использованию нейронных сетей для распознавания математических выражений. На данный момент большинство документов в основном содержат текст, печатный или рукописный. Для быстрого и удобного перевода рукописных документов в память компьютера, необходимы системы, способные преобразовать физический элемент в цифровой.

Распознавание математических выражений это сфера исследования, относящаяся к компьютерным технологиям. В отличие от текстов, печатные математические выражения имеют двумерный характер, а их распознавание подразумевает восстановление структуры формулы и содержащихся в ней символов.

**Целью бакалаврской работы** является изучение основ нейронных сетей и создание программы для анализа изображений математических текстов и распознавания отдельных символов.

**Объект исследования** — анализ и считывание текстового изображения.

**Предмет исследования** — разработка программы для анализа изображений математических текстов.

Для достижения поставленных целей в работе необходимо решить следующие **задачи**:

- определить основные технологии, необходимые для разработки программы;
- изучить основы нейронных сетей и методы объектно-ориентированного программирования;
- изучить подходящий для разработки язык программирования;
- построить модель нейронной сети;
- описать анализ изображения для поиска отдельных элементов и структур;
- реализовать программный продукт.

**Практическая значимость.** Данная работа представляет интерес, поскольку нейронные сети на данный момент является одной из самых пер-

спективных технологий, а проблемы анализа текстовых изображений достаточно часто возникает при работе с печатными документами и книгами.

**Структура и содержание магистерской работы.** Работа состоит из введения, двух разделов, заключения, списка использованных источников, содержащего 25 наименования, и двух приложений. Общий объем работы составляет более 50 страниц.

## **Аппробация**

Работа прошла апробацию на следующих конференциях:

- IX Международная научно-практическая конференция «Математическое и компьютерное моделирование в экономике, страховании и управлении рисками», 26 - 27 ноября 2020 года, Саратов;
- Студенческая научная конференция «Математика. Механика», 12 - 24 апреля 2021, Саратовский государственный университет, Саратов.

Результаты ВКР опубликованы в статьях <http://risk2020.sgu.ru/> и <https://www.sgu.ru/conference/nkmm-2021>.

## Основное содержание работы

Во **введении** обосновывается актуальность темы работы, формулируются цель и задачи работы, отмечается практическая значимость полученных результатов.

В **первом** разделе рассматриваются основы теории нейронных сетей, используемые технологии анализа изображений и разработки приложения по распознаванию текста.

Изучение и использование искусственных нейронных сетей, началось ещё в начале XX века, но по настоящему широкую известность они получили намного позже. Связано это, в первую очередь, с тем, что стали появляться продвинутое (для того времени) вычислительные устройства, мощности которых были достаточно велики для работы с искусственными нейронными сетями. По сути, на данный момент можно легко смоделировать нейронную сеть средней сложности на любом персональном компьютере.

Нейрон представляет из себя элемент, который вычисляет выходной сигнал (по определенному правилу) из совокупности входных сигналов. То есть основная последовательность действий одного нейрона такая:

- Прием сигналов от предыдущих элементов сети,
- Комбинирование входных сигналов,
- Вычисление выходного сигнала,
- Передача выходного сигнала следующим элементам нейронной сети.

Между собой нейроны могут быть соединены абсолютно по-разному, это определяется структурой конкретной сети. Но суть работы нейронной сети остается всегда одной и той же. По совокупности поступающих на вход сети сигналов на выходе формируется выходной сигнал (или несколько выходных сигналов). То есть нейронную сеть упрощенно можно представить в виде черного ящика, у которого есть входы и выходы. А внутри этого ящика сидит огромное количество нейронов.

Поскольку к каждому нейрону могут приходить несколько входных сигналов, то при моделировании нейронной сети необходимо задать определенное правило комбинирования всех этих сигналов. И довольно-таки часто используется правило суммирования взвешенных значений связей

Каждую связь в сети нейронов можно полностью охарактеризовать при

помощи трех факторов:

- первый – элемент, от которого исходит связь;
- второй – элемент, к которому связь направлена;
- третий – вес связи.

Вес связи определяет, будет ли усилен или ослаблен сигнал, передаваемый по данной связи. А если сигналов много, то они все суммируются. В итоге на входе нейрона мы получаем следующее:

$$net_j = \sum_{i=1}^N x_i * w_{ij} \quad (1)$$

В этой формуле  $net_j$  – это результат комбинирования всех входных сигналов для нейрона  $j$  (комбинированный ввод нейрона).  $N$  – количество элементов, передающих свои выходные сигналы на вход сигнала  $j$ . А  $w_{ij}$  – вес связи, соединяющей нейрон  $i$  с нейроном  $j$ . Суммируя все взвешенные входные сигналы, мы получаем комбинированный ввод элемента сети. Чаще всего структура связей между нейронами представляется в виде матрицы  $W$ , которую называют весовой матрицей. Элемент матрицы  $w_{ij}$ , как и в формуле, определяет вес связи, идущей от элемента  $i$  к элементу  $j$ .

Нейронные сети, использующие обучение с учителем. Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Далее веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемого уровня.

Нейронные сети, использующие обучение без учителя. Обучение без учителя является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей.

Классификация нейронных сетей по типу настройки весов делит их на:

- сети с фиксированными связями – весовые коэффициенты нейронной

сети выбираются сразу, исходя из условий задачи;

- сети с динамическими связями – для них в процессе обучения происходит настройка синаптических весов.

Классификация нейронных сетей по типу входной информации делит их на:

- аналоговые – входная информация представлена в форме действительных чисел;
- двоичные – вся входная информация в таких сетях представляется в виде нулей и единиц.

В полносвязных нейронных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

В многослойных (слоистых) нейронных сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях. В общем случае сеть состоит из слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов входного слоя (его часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько скрытых слоев. Связи от выходов нейронов некоторого слоя  $q$  к входам нейронов следующего слоя ( $q + 1$ ) называются последовательными.

Объектно-ориентированное программирование (ООП) — это способ организации кода в программе путем его группировки в виде объектов, то есть отдельных элементов, которые включают информацию (значения данных) и функции. Объектно-ориентированный подход к организации программы позволяет группировать определенные фрагменты информации (например, информацию об отеле, такую как ФИО клиента или характеристику номера) вместе с часто используемыми функциями или действиями, связанными с этой информацией (например, "добавить гостя в базу данных" или "обновить информацию о номере отеля"). Сейчас объектно-ориентированное про-

граммирование стало настолько распространенным подходом в мире, что понимание его основных концепций стало необходимо для всех, кто собирается заняться программированием.

Для анализа структуры математического выражения одни методы основаны на пространственных соображениях, таких как базовые линии. Другие используют системы на основе правил грамматики, а потом анализируют выражение, чтобы найти его интерпретацию. В нескольких алгоритмах учитываются знания о математических символах и операторах и их пространственное отношение. В очень немногих документах используются методы машинного обучения, хотя есть некоторые работы с использованием нечеткой логики.

Python – это высокоуровневый универсальный язык программирования, ориентированный на повышение производительности разработчика и читаемости кода. Для решения широкого круга задач в поставку Python входит достаточно обширная стандартная библиотека. Для ещё большего расширения возможностей языка в интернете доступны другие разнообразные качественные библиотеки по различным предметным областям: инструменты для создания приложений, средства обработки текстов и изображений, механизмы работы с базами данных, библиотеки построения графического интерфейса, пакеты для научных вычислений, и т.п. Python портирован и работает практически на большинстве известных платформ — от КПК до мейнфреймов.

PyQt5 – это набор Python-связей для фреймворка Qt5 от Digia. Набор PyQt5 доступен для Python 2.x и 3.x. PyQt5 разработан компанией Riverbank Computing. PyQt5 реализован как комплект Python-модулей. Он включает в себя около 620 классов и 6000 функций и методов. Это мульти-платформенный инструментарий, который запускается на большинстве операционных систем, среди которых Unix, Windows и MacOS. PyQt5 реализован под двумя лицензиями.

Библиотека очень проста в использовании и идеально подходит для разработки весьма серьезных оконных приложений. Пользуясь исключительно ее средствами, мы можем выводить на экран графику практически любой сложности, работать с базами данных наиболее распространенных форма-



тов, воспроизводить мультимедийные файлы, выводить документы на печать и экспортировать их в популярный формат Adobe PDF. В комплект поставки PyQt 5 входит программа Qt Designer, предназначенная для визуального создания окон, — буквально путем рисования их мышью. Результатом станет XML-файл, который можно непосредственно использовать для формирования окна или преобразовать в создающий это же окно Python-код.

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку программного обеспечения для компьютерного зрения с открытым исходным кодом и компьютерного обучения. Он был построен для различных целей, таких как машинное обучение, компьютерное зрение, алгоритм, математические операции, захват видео, обработка изображений и т. Д. На протяжении многих лет он стал очень популярен среди исследователей и разработчиков как для поддержки на разных платформах (Windows, Linux, android, ios). Также он имеет обертку на разных известных языках программирования. В соответствии с лицензионным соглашением у него есть доступ к компаниям для использования и изменения кода. Поскольку OpenCV распространяется под BSD лицензией, любой желающий может использоваться её как в исследовательских целях, так и в коммерческих.

Keras — открытая нейросетевая библиотека, написанная на языке Python. Она является надстройкой над фреймворками DeepLearning4j, TensorFlow и Theano. Нацелена на оперативную работу с сетями глубинного обучения, при этом спроектирована так, чтобы быть компактной, модульной и расширяемой. Она была создана как часть исследовательских усилий проекта ONEIROS (англ. Open-ended Neuro-Electronic Intelligent Robot Operating System), а ее основным автором и поддерживающим является Франсуа Шолле, инженер Google. С помощью Keras определяются высокоуровневые нейронные сети, работающие поверх библиотеки TensorFlow или Theano. Можно выделить основные положительные стороны в использовании:

- Удобство для пользователя.
- Модульность.
- Легкая масштабируемость.
- Работа с языком Python.

Процесс обучения не является моментальным, поэтому хранение дан-

ных и вычисления эффективнее проходят с использованием графического процессора. Обучение сети производилось на видеокарте NVIDIA с использованием CUDA 10.0. CUDA является архитектурой параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU (графических процессоров). Выполнение расчётов на GPU показывает отличные результаты в алгоритмах, использующих параллельную обработку данных. То есть, когда одну и ту же последовательность математических операций применяют к большому объёму данных. При этом лучшие результаты достигаются, если отношение числа арифметических инструкций к числу обращений к памяти достаточно велико. Это предъявляет меньшие требования к управлению исполнением (flow control), а высокая плотность математики и большой объём данных отменяет необходимость в больших кэшах, как на CPU.

Во **втором** разделе описывается процесс разработки нейронной сети и обработки для распознавания текста.

Разработка приложения по распознаванию математических текстов и состоит из двух этапов: создание обучающей выборки и обучение нейронной сети, выделение элементов текстового изображения и их анализ нейронной сетью.

Тренировочный датасет для нейронной сети состоит из более 100 тыс. изображений. За их основу были взяты около 200 символов из математики, греческого и латинского алфавитов. Используемый датасет был создан путем деформации изображения каждого из символов. За счет этого небрежно сканированный текст может быть подан для анализа обученной нейронной сети. У каждого символа были определены координаты его прямоугольного контура, а затем изображения деформированы растяжением. В результате была получена выборка для каждого символа, ограниченная несколькими тысячами изображениями. Каждое квадратное изображение сжато до размера  $32 \times 32$  пикселей.

Архитектура нейронной сети создавалась на языке Python с использованием открытых программных библиотек для машинного обучения Keras и Tensorflow. Построенная нейронная сеть состоит из трех слоёв:

— входной слой – 1024 нейронов,

- скрытый слой – 512 нейронов,
- выходной слой – соответствует количеству заданных классов.

Выборка данных прошла десятки эпох (epoch) — весь датасет прошел через нейронную сеть в прямом и обратном направлении. Чем больше эпоха, тем лучше натренирована сеть и, соответственно, ее результат. Эпоха увеличивается каждый раз, когда мы проходим весь набор тренировочных сетов.

Так как одна эпоха слишком велика, она была разбита на небольшие партии (batches) по 6000 штук. Данные расположены в папках по категориям, откуда изображения импортируются и формируются в датасет состоящий из пар: изображение символа, класс символа. Датасет случайно перемешивается, а структура изображений изменятся под требуемую для обработки.

Полученные изображения и список их классов подаются для обучения в саму нейронную сеть, а обученная нейронная сеть сохраняется в отдельный файл, с помощью которого будут распознаваться символы.

Алгоритм распознавания символов в текстовом изображении позволяет находить отдельные символы в строках, анализировать их нейронной сетью. Далее все символы группируются в отдельные строки, где проходят через фильтр, предназначенный для выявления и объединения составных символов, таких как, например, 'Й', 'ы', 'ё', 'i', 'j', 'Ξ', '=', ':', ';' и т.п. После символы группируются в самих строках и результат предоставляется в формате LaTeX, как представлено на рисунке.

<b>a</b>	$f(x, y) = \sum \alpha \cdot x + \sum y/\beta, \quad 0 < \alpha \leq 2, \quad \beta > 2$
<b>б</b>	$f(x, y) = \sum \alpha \cdot x + \sum y/\beta, \quad 0 < \alpha \leq 2, \quad \beta \geq 2$
<b>в</b>	<pre>f ( x , y ) = \sum \alpha \cdot x + \sum y / \beta , \ \ 0 \ \ &lt; \ \alpha \ \leqslant \ 2 ,\ \ \ \beta \ \geqslant \ 2 &gt;&gt;&gt;</pre>

Рисунок 1 – Анализ текстового изображения функции: *a* – начальное изображение; *б* – области вокруг символов; *в* – результат обработки символов нейронной сетью в соответствии с классами

Программа способна достаточно точно распознавать символы, а, добавляя новые символы и увеличивая набор данных, можно заново обучить нейронную сеть, что значительно расширит возможности распознавания.

## ЗАКЛЮЧЕНИЕ

Основные результаты работы:

1. Определены методы и технологии разработки программы для распознавания текстов.
2. Изучены и применены библиотеки и язык программирования необходимые для разработки:
  - Python 3,
  - TensorFlow и Keras,
  - OpenCV.
3. Разработана и реализована программа распознавания математических текстов.