

Разработка голосового ассистента проводилась с использованием языка программирования Python, который обладает огромным количеством встроенного функционала для решения разнообразных задач. Для удобства работы пользователя был реализован графический интерфейс. После запуска приложения пользователю рекомендуется ознакомиться с основными функциями, которыми обладает приложение.

Активация голосового помощника происходит с помощью меню или по нажатию заданных горячих клавиш. Далее пользователь может осуществлять голосовой ввод команд, определенных приложением. Все действия дублируются в окне приложения.

Распознавание речи и преобразование ее в текстовый формат происходит посредством модуля SpeechRecognizer, который на данный момент работает только с английским языком. Для описания доступных функций использовался конфигурационный файл формата YAML.

Для определения функционала приложения была проанализирована компьютерная активность друзей, знакомых и коллег, и отображены основные действия, которые чаще всего ими выполняются.

Голосовой помощник «Severity» имеет ряд возможностей для работы с файлами в программах PowerPoint, Word, Excel, Paint, Notepad (например, создание и сохранение документа, открытие существующего документа). Вся работа с операционной системой и ее элементами осуществляется с помощью модуля os языка программирования Python. Так же организована работа с браузером Google Chrome. Имеется возможность открытия заданных конфигурационным файлом сайтов. Для этих целей были выбраны самые распространенные ресурсы: youtube, google, yandex, netflix, yandex music. Имеется возможность открытия произвольного сайта.

Р. И. Протченко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Андреев**, д-р физ.-мат. наук, профессор

НЕЙРОННЫЕ СЕТИ И ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON

Идея нейронных сетей была позаимствована у природы, а точнее в качестве аналога использовалась нервная система животных и чело-

века. Нервная система состоит из большого количества достаточно простых элементов – нейронов. Каждый нейрон получает сигналы от тысяч других нейронов и передает обработанную информацию тысячам других нейронов по одному единственному выводу.

Но с точки зрения программирования, нейронная сеть – способ решения проблемы эффективного параллелизма. В нейронных сетях, обычно, содержатся нейроны и связи между ними. Созданная при помощи компьютера нейронная сеть, имеет несколько слоев. Однако, количество нейронов в них бывает различным. Первый слой называется входной (принимает определенную информацию), последний – выходной (дает конечный результат), все промежуточные – скрытые слои (обрабатывают информацию, полученную с предыдущих слоев или входного).

В современном мире, многие сложные задачи требуют аналитические вычисления подобных тем, что делает человеческий мозг. Нейронные сети помогают нам в решении такого типа задач.

Основное применение нейронных сетей является:

Классификация – разбиение данных по параметрам. Например, на вход дается набор людей и нужно решить, кому из них давать кредит, а кому нет. Начальными данными будут люди и нужно решить, кому из них отказать в выдаче кредита, а кому его предоставить. Эту задачу может решить нейронная сеть, благодаря анализу определенных параметров: кредитная история, платежеспособность, кредитная история, возраст и т. д.

Предсказание – способность прогнозировать следующий шаг. Например, падение или рост цен на бензин, основываясь на конъюнктуры рынка нефтепродуктов в сопредельных государствах.

Распознавание – на данный момент, это самое распространенное применение нейронных сетей. Например: определение через камеру расположения лица, распознавание текста текста.

Наиболее распространенным языком программирования для искусственных нейросетей является Python [1, 2]. Из всех известных языков программирования, Python, для создания искусственных нейросетей, был выбран по ряду причин:

Исследовательские проекты. Исследовательскими проектами являются большое количество готовых примеров нейронных сетей на Python. Ученые, работающие над проектами, не владеют навыками программиста. Этот язык имеет низкий порог входа: на данный момент, никто не осуществляет написание нейронных сетей на Python с самого нуля, потому что это тратит много времени и сил. Имеются

библиотеки для нейронных сетей Python, которые уже составили профессионалы.

Лаконичность и интероперабельность. Язык Python обладает возможностью составлять сложные алгоритмы за короткий срок. Его особенности – это простота, лаконичность и выразительность. Помимо этого, он обладает мощным механизмом интероперабельности с C/C++, что дает возможность осуществлять быстрые вычисления. Можно заметить, что, на языке Python есть возможность создавать простые и сложные нейронные сети на Python.

Гибкость. Нейронные сети, по большей своей части, небольшие программы, но при этом есть потребность часто вносить изменения в них, подбирая наилучшую предобработку данных, архитектуру и другие параметры. Формирование и построение нейронных сетей на Python – вариант, удовлетворяющий этим требованиям лучше, чем использование C++ или Java.

В своей работе с помощью нейронных сетей, составлю программу по нахождению теста на картинке и распознаванию его. Для этого, проще говоря, нужно выполнить определённые действия: установка нужных библиотек (например: `opencv`, `easyocr`, `imutils`, `matplotlib`, `numpy`, `pandas`), оптимизировать изображение, нахождение контуров, вырезать нужный контур, чтение текста с вырезанного контура, вывод результата на экран.

Я выбрал эту тему, потому что у искусственных нейронных сетей есть перспективы в будущем, а также у них есть возможность самообучаться, а также способность отсеивать лишний шум. Однако нейросетевые технологии не должны рассматриваться как универсальное средство решения всех интеллектуальных задач. Их применение оправдано в тех областях, в которых существует значительное число однотипных примеров, отражающих скрытые взаимосвязи. Нейронные сетевые технологии предназначены для решения плохо формализованных задач. Такие технологии применяются для распознавания тех или иных предметов или событий. Благодаря им можно воспроизвести значительное количество связей между множеством объектов. Большое отличие искусственных нейронных сетей от обычных программных систем, заключается в том, что они не запрашивают программирования. Они сами настраиваются, т.е. обучаются тому, что требуется пользователю.

Литература

1. Нейронные сети на Python: как все устроено [Электронный ресурс] / Редакция сайта GeekBrains: электрон. версия газ. – URL: <https://gb.ru/blog/nejronnye-seti-python/>. – Дата доступа: 2.04.2022.

2. Нейронные сети для начинающих [Электронный ресурс] / Арнис@Arnis71. – URL: <https://habr.com/ru/post/312450/> – Дата доступа: 2.04.2022.

Д. Ю. Путьков

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

ПРИНЦИП РАБОТЫ ВЕБ-ПРИЛОЖЕНИЯ ПО ОТОБРАЖЕНИЮ WI-FI СЕТИ

Веб-приложение разработано для университета или других объектов, которым нужна безопасная Wi-Fi сеть. Приложение способно отображать Wi-Fi сеть на чертеже плана этажа, определять опасные зоны и защитить Wi-Fi сеть. Опасная зона – это зона Wi-Fi сети, которая выходит за пределы плана выбранного этажа. Безопасной сетью считается та сеть, которая не выходит за пределы этажа.

Веб-приложение состоит из трех частей: front-end, back-end и база данных. Все эти компоненты взаимодействуют между собой. Для взаимодействия между front-end и back-end используется протокол HTTP. Для взаимодействия между back-end и базой данных используется специальный фреймворк Gorm.

Расчет дальности Wi-Fi сигнала происходит на back-end уровне. Изначально сервер принимает запрос в виде JSON с front-end уровнем. В этот запрос входят координаты роутеров, которые указал пользователь, а также план этажа. После чего происходит расчет дальности Wi-Fi сигнала. Для расчета используются формулы:

1. $FSL = 33 + 20 (\lg F + \lg D)$.

2. $Y = Pt + Gt + Gr - Pmin - Lt - Lr$.

3. $FSL = Y - SOM$.

После расчета дальности Wi-Fi сигнала происходит отрисовка полученного результата на плане этажа пользователя. Для отрисовки используется восемь цветов, от зеленого, который обозначает, что в этой области наилучший Wi-Fi сигнал, до красного, где сигнал