МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики   
Кафедра прикладной математики**

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА БОТОВ ДЛЯ DISCORD СЕРВЕРОВ**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_П.П. Лагун

Направление подготовки 09.03.03 Прикладная информатика

Направленность (Профиль) Прикладная информатика в экономике

Научный руководитель

канд. физ.-мат. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Письменский

Нормоконтролер

преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.С. Троценко

Краснодар

2023

**РЕФЕРАТ**

Курсовая работа 38 с., 7 ч., 27 рис., 1 табл., 13 источников.

БОТ, ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ, НЕЙРОН, ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Объектом исследования являются Discord боты с применением искусственных нейронных сетей, предназначенные для оптимизирования и добавления функций в повседневное общение пользователей приложения Discord.

Целью работы является ознакомление с литературой про ботов, искусственные нейронные сети, анализ данных и построение теоретической модели нейронной сети для её дальнейшей практической реализации.

В результате выполнения данной работы рассмотрены теоретические основы ботов, нейронных сетей, применяющиеся методы обучения, сбора и анализа данных, а также, проанализированы перспективы применения нейронных сетей в повседневной жизни.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc135579309)

[1 Понятие искусственных нейронных сетей 5](#_Toc135579310)

[2 Работа с данными и обучение нейронных сетей 9](#_Toc135579312)

[2.1 Сбор и анализ данных 9](#_Toc135579313)

[2.2 Методы обучения 11](#_Toc135579314)

[3 Обучение бота 14](#_Toc135579315)

[3.1 Обучение пониманию речи и текста 14](#_Toc135579316)

[3.2 Работа со смыслом: технология Natural Language Understanding 15](#_Toc135579317)

[3.3 Работа с контекстом 15](#_Toc135579318)

[4 Анализ предметной области и выбор метода решения задачи 20](#_Toc135579319)

[5 Нейронные сети на Python 21](#_Toc135579321)

[6 Примеры работы Discord ботов 23](#_Toc135579322)

[7 Практическая часть 27](#_Toc135579323)

[Заключение 36](#_Toc135579324)

[Список использованных источников 37](#_Toc135579325)

# **ВВЕДЕНИЕ**

Технологии с каждым годом просачиваются в жизни людей все больше и больше, позволяя упростить большинство повседневных задач. Одной из подобных технологий являются боты – специальные роботы или искусственный интеллект, которые выполняют автоматизированные задачи по определенному алгоритму, исходя из запроса от пользователя. Они позволяют сократить время пользователя и освободить человеческие ресурсы в виде отдельного рабочего места, которых и так может быть недостаточно или это попросту невыгодно, а также снизить нагрузку на реальных специалистов, уменьшив нецелевые обращения.

Области, в которых боты применяются, безграничны, такие как: техническая поддержка, распространение новостей и информационных сообщений, расчет стоимости товара, оформление заказа, нахождение информации в большом количестве данных, проигрывание музыки, подбор автомобилей, билетов на транспорт, напоминание о важных событиях и т.д.

В большинстве случаев, отличительной чертой работы бота является среда приложения, в которой он применяется, ведь именно в нее внедряется бот и использует интерфейс приложения. Самые известные приложения, в которых используются боты – это социальные сети, мессенджеры, всевозможные сайты, такие как: Telegram, VK, Discord и т.д.

Главной задачей бота является освободить человека от рутинной, однообразной работы и уменьшить время ожидания ответа на запрос клиента, поэтому многие компании внедряют технологии ботов в свои проекты для экономии рабочих мест и в тоже время увеличения клиентоориентированности. Поэтому важно для разработчика правильно настроить работу бота, чтобы он не отпугивал клиента бесконечными уточняющими вопросами, долгими или неправильными ответами. Если это не учитывать в процессе разработки, то при работе бота огромное количество пользователей попросту закроет приложение и обратится в другой сервис или к конкуренту.

В свою очередь, применение нейронных сетей совместно с ботом обеспечит больший функционал, позволит работать с большим количеством данных и поиском нужной информации, осуществит быстродействие работы приложения.

# **1 Понятие искусственных нейронных сетей**

Нейронные сети (НС) – математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма. После разработки алгоритмов обучения нейронных сетей, они используются в практических целях: в задачах прогнозирования, распознавания образов, управления и др.

Исследования в области нейронных сетей начались в 40-е годы XX века. Первое систематическое изучение искусственных нейронных сетей было предпринято Маккалокком и Питтсом в 1943 г. Позднее они исследовали сетевые модели для распознавания изображений, подвергаемых сдвигам и поворотам.

Искусственная нейронная сеть представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Процессоры подобного характера в большинстве случаев довольно просты, особенно если сравнивать с процессорами, которые изпользуются в персональных компьютерах. Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. Будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, простые процессоры по отдельности способны выполнять задачи, которые для человека бывают невыполнимы или трудозатратны, в купе с тратой огромного количества времени на выполнение вручную.

Нейронные сети представляют собой новую и очень перспективную вычислительную технологию, дающую новые подходы к исследованию динамических задач в финансовой области. Первоначально нейронные сети открыли новые возможности в области распознавания образов, затем к этому прибавились статистические и основанные на методах искусственного интеллекта средства поддержки принятия решений и решения задач в сфере финансов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна определять сложные зависимости между входными данными и выходными, а также осуществлять деление на классы по сходимости или объединять. Это означает, что в случае правильного и эффективного обучения искусственная нейронная сеть способна вернуть истинный результат на основании данных, которых не было при обучении, а также неполных и/или «зашумленных», частично искажённых данных.

Искусственную нейронную сеть можно рассмотреть, как направленный граф, который имеет связи между своими узлами, являющимися искусственными нейронами.

По архитектуре связей можно выделить три класса нейронных сетей:

1. однослойные сети прямого распространения (рисунок 1.a)
2. многослойные сети прямого распространения (рисунок 1.b)
3. рекуррентные сети (рисунок 1.c, 1.d)

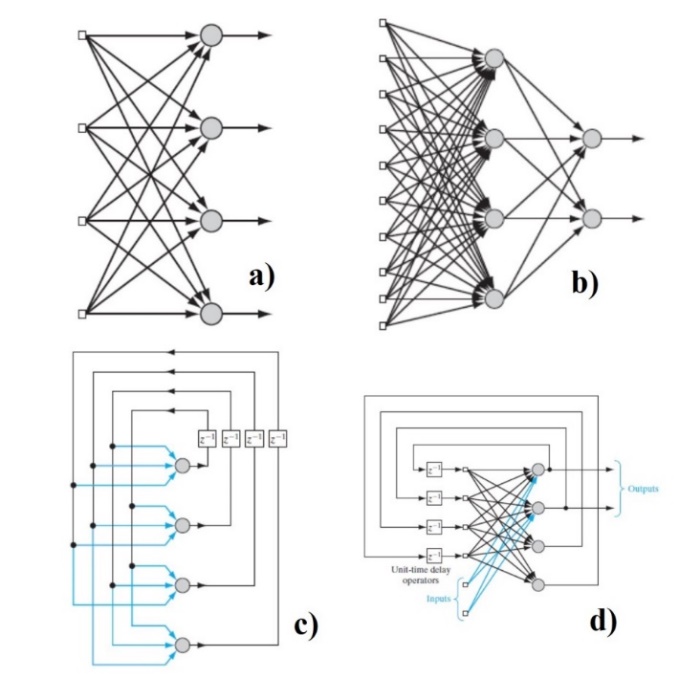


Рисунок 1 – Виды искусственных нейронных сетей

Нейронная сеть организована слоями. В простейшей однослойной форме такой сети имеются входные данные, которые проецируются соответственно на выходной слой нейронов. Такая конструкция имеет строго прямой тип подачи. К слою относятся вычислительные, то есть выходные узлы, но не учитывается входной слой, поскольку там не выполняется никаких вычислений.

Второй класс прямой нейронной сети отличается наличием одного или нескольких скрытых слоев. В данной форме слой входных данных также не учитывается.

В рекуррентных сетях нейронные связи к верхнему слою могут идти не только от нижнего слоя, но и от предыдущего значения самого этого нейрона или других нейронов того же слоя.

**1.2 Преимущества и недостатки искусственных нейронных сетей**

Применение нейронных сетей для решения различных задач с каждым годом становится популярнее. Создаются гораздо более эффективные программные реализации, построенные по принципу работы нейронных сетей живых организмов. Область применения этой технологии делается еще обширней. Но все же, использование этого подхода имеет свои преимущества и недостатки.

К преимуществам нейронных сетей можно отнести следующее:

1. устойчивость входных данных к шумам. Обученная нейронная сеть способна игнорировать сильно зашумленные данные, подобно тому, как человек может различать лица прохожих, не обращая внимание на посторонние объекты вроде асфальта, одежды, светофоров и пр.
2. адаптация к изменениям. Нейронная сеть способна адаптироваться к изменениям во входных данных, позволяя непрерывно продолжать обучение и работу.
3. отказоустойчивость. Функционирование нейронной сети может продолжаться даже при значительных повреждениях некоторого количества нейронов.
4. быстродействие. Решение задачи с помощью нейронных сетей находится быстрее, чем при использовании обычных алгоритмов.

Искусственные нейронные сети имеют также и ряд серьезных недостатков:

1. неточность решения. Нейронные сети не способны выдавать точный ответ, а дают приблизительное решение задачи, которое, возможно, отличается от правильного на некоторую достаточно малую величину.
2. многошаговость решения. В искусственных нейронных сетях каждый нейрон является независимым. Он получает сигнал и, преобразуя его, отправляет на вход следующему нейрону. В зависимости от соседнего нейрона идет изменение синапсов. Таким образом, нейронная сеть не способна решать задачу последовательно.
3. неспособность решать вычислительные задачи. Этот недостаток можно считать следствием двух предыдущих, то есть с помощью нейронных сетей невозможно получить решения задач, требующих последовательных действий и точного ответа. Такими являются, например, арифметические задачи.

Несмотря на некоторые недостатки, разработка искусственных нейронных сетей является перспективным направлением машинного обучения. Ведется активный поиск средств для минимизации этих недостатков и совершенствования работы алгоритмов обучения нейронных сетей.

# **2 Работа с данными и обучение нейронных сетей**

## **2.1 Сбор и анализ данных**

Для задач, решаемых с помощью нейронных сетей необходимо собрать данные, чтобы в дальнейшем начать обучение. Набор сигналов, подаваемых на вход для обучения, представляет собой совокупность значений входных переменных. Эти данные могут быть симулированными, файлами CSV, Excel или JSON, или информацией из базы данных. Это является отдельной задачей при практической реализации нейронной сети, поскольку от формата данных зависит дальнейшая структура проекта.

Достаточно сложным является вопрос о количестве наблюдений, необходимых для обучения нейронной сети. Большинству реальных задач достаточно нескольких сотен или тысяч таких данных, более сложным проектам может потребоваться большее количество.

Входные данные в обучающей выборке не должны быть противоречивыми, иначе это приведет к плохому качеству обучения сети. Также, необходимо, чтобы они содержали истинную информацию о предметной области.

Между данными, подающимися на вход нейронной сети может существовать функциональная или стохастическая связь. Эту зависимость можно выразить в числовой форме. Одним из способов оценки силы такой связи между параметрами является вычисление коэффициента корреляции. Но данная характеристика имеет смысл только для данных, имеющих стохастический тип связи, поскольку при функциональной зависимости параметров коэффициент равен единице.

Для работы с нейронной сетью, в основном, используются данные, которые являются случайными величинами и не имеют функциональной зависимости, следовательно, вычисление коэффициента корреляции можно считать оптимальным способом оценки возможности применения данных.

Пределы изменения коэффициента находятся на отрезке [-1;1]. Если вычисленный результат отрицательный, то это означает, что с увеличением значения одной переменной значение другой убывает. Если данные независимы, то коэффициент корреляции равен 0, но если результат не равен 0, то между переменными существует зависимость и чем ближе её значение к 1, тем она выше. Силу связи можно оценить, опираясь на классификацию, представленную шкалой Чеддока (таблица 1).

Таблица 1 – шкала Чеддока силы связей между переменными

|  |  |
| --- | --- |
| Значение | Интерпретация |
| от 0 до 0.3 | слабая |
| от 0.3 до 0.5 | умеренная |
| от 0.5 до 0.7 | заметная |
| от 0.7 до 0.9 | высокая |
| от 0.9 до 1 | очень высокая |

Обозначим через – фиксированные значения фактора, – функции отклика, – количество наблюдений, тогда для вычисления коэффициента парной корреляции используется формула:

|  |  |
| --- | --- |
|  | (1) |
| где |  |

, – выборочные средние;  
.

Для наиболее объективной оценки силы связи между данными, использующимися при разработке модели искусственной нейронной сети, имеет смысл применять коэффициент множественной корреляции, поскольку количество параметров, как правило, больше двух.

Тогда, пусть – число факторов, , где – значение фактора в исследовании и , , где – значение функции отклика в исследовании , .

Используя введенные обозначения и формулу (1), построим матрицу коэффициентов корреляции:

|  |  |
| --- | --- |
|  | (2) |

Используя данную матрицу, можно найти коэффициент множественной корреляции по формуле:

|  |  |
| --- | --- |
|  | (3) |

где

– определитель матрицы K;  
 – определитель без включения строки и столбца с номером 1.

Таким образом, проанализировав данные с помощью вычисления значений коэффициентов корреляции, можно сделать оценку о возможности дальнейшего применения этого набора параметров для практической реализации нейронной сети.

## **2.2 Методы обучения**

Процесс обучения нейронной сети можно организовать различными способами. Выбор оптимального алгоритма зависит от вида данных и задачи. Существует несколько методов обучения искусственных нейронных сетей.

1. Обучение с учителем предполагает, что набор данных для обучения является размеченным. Это означает, что каждому объекту в обучающем примере соответствует правильный ответ, который и должен получить алгоритм. Как правило, обучение с учителем применяется для решения задач двух типов: регрессии и классификации.

В задачах классификации алгоритм делает предсказание номера класса, к которому принадлежат объекты.

В задачах регрессии данные должны быть непрерывными. Например, при линейной регрессии, ожидаемое значение переменной вычисляется, учитывая определенные значения .

Таким образом, обучение с учителем больше всего подходит для решения задач, имеющих внушительную базу данных для обучения.

1. Нередко перед алгоритмом может быть поставлена задача найти заранее неизвестные решения. Для таких задач применяется обучение без учителя, когда у модели есть только набор данных и нет никакой информации по поводу дальнейших действий с ними. В этом случае нейронная сеть самостоятельно пытается проанализировать данные и извлечь из них полезные признаки. Наиболее популярными для данного алгоритма являются задачи кластеризации, обнаружения аномалий и ассоциаций. В задачах кластеризации алгоритм ищет общие признаки при анализе данных и группирует их вместе.

При поиске аномалий, выявляются данные, не соответствующие понятию нормального поведения.

В задачах ассоциаций, алгоритм, рассматривая некоторое количество ключевых признаков, может сделать предсказание о возможной связи одного объекта с другими.

Главным недостатком данного метода обучения является то, что вычислить точность алгоритма достаточно проблематично, поскольку отсутствуют правильные ответы. Но, как правило, размеченные данные затруднительно получить. В таких случаях используют алгоритмы обучения без учителя для поиска зависимостей и решения задачи.

1. В обучении с частичным привлечением учителя используют данные, в которых присутствуют как размеченные, так и неразмеченные объекты.

Этот метод обучения распространен при анализе медицинских изображений, где разметить все объекты является невыполнимой задачей, поскольку появляется вероятность ошибки при постановке правильного диагноза и данные не будут являться правдивыми. В этом случае, нейронная сеть извлекает информацию из размеченных данных и с ее помощью может улучшиться точность предсказаний, по сравнению с алгоритмом обучения, работающем исключительно на неразмеченных данных.

Метод обучения с частичным привлечением учителя полезен, когда проблематично извлечь из данных важные признаки, а разметить все объекты не представляется возможным.

1. Основная идея обучения с подкреплением состоит в принятии определенных действий агентом, взаимодействуя с окружающей средой, за которые он получает поощрение и продолжает совершать какую-либо деятельность, пытаясь увеличить свою награду. Такой подход очень полезен при обучении роботов, которые, например, управляют автономными транспортными средствами.

# **3 Обучение бота**

## **3.1 Обучение пониманию речи и текста**

На этом этапе бот только распознает звук или его визуальные характеристики. Требуется понимать и анализировать эти характеристики так, чтобы в итоге получилось так, будто это сказал человек.

С этим большое количество проблем: запрос от пользователя нужно проанализировать, тем самым убрать шумы или не воспринимать их в учет, отличить друг от друга слова омонимы или похожие по звучанию слова («мыла» и «мыло»), в итоге, исходя из полученных результатов, выбрать наилучший вариант.

Чат-бот использует языковую и акустическую модели. Они предварительно обучаются на огромном объёме данных, накапливают опыт.

Акустическая модель в реальном времени переводит звук в цифровой формат, нарезает на множество микро-отрезков и относит каждый отрезок к определённой части слова. Таких соотношений большое количество: языковая модель выстраивает последовательность, не путая слова омонимы или же похожие по звучанию слова. Она учитывает, с какой частотой входят или стоят рядом друг с другом звуки.

Если искусственный интеллект принял фразу «выведи значения, полученные в икре», языковая модель поймет, что большая вероятность того, что имеется в виду «выведи значения, полученные в игре». Важно, что обе модели зависят от контекста беседы. Если пользователь общается с чат-ботом на сайте магазина автозапчастей, искусственный интеллект поймёт, что в запросе подразумевается какое-то конкретное наименование детали, например, «Поперечный рычаг верхней подвески».

## **3.2 Работа со смыслом: технология Natural Language Understanding**

Запрос клиента переведен в текст, но его желание и смысл пользователя не определены. Искусственный интеллект начинает мыслить, как человек - соотносит запрос с примерами, на которых учился, и находит наиболее подходящие по смыслу. Бот действует классами, сравнивая каждый запрос с одним из них. А классы, в свою очередь, формируются, исходя из сферы применения бота: покупка авиабилетов, подбор автозапчастей, техническая поддержка.

Например, в класс «подобрать автозапчасть» попадают запросы «автозапчасть по VIN коду», «оформление заказа», «гарантия возврата» и нестандартные, вида «подобрать как». С каждым таким этапом происходит обучение бота, в дальнейшем самостоятельно определяя, какие запросы подходят больше всего по смыслу в определенный класс.

## **3.3 Работа с контекстом**

Чат-бот должен самостоятельно определять и учитывать контекст беседы. К примеру, вопрос «как будет “подобрать автозапчасть” по-английски?» можно определить и как подбор детали, и как запрос к переводчику.

Основная технология современных ботов – понимание естественного языка (NLU). Она позволяет искусственному интеллекту понимать пользователей и запускать нужные параметры для обработки запросов. Для этого нужны все технологические решения, связанные с обработкой естественного языка. К ним относится подход к обработке (гибридный, rule-based, статистический), технологии применения чат-ботов в бизнес-процессы компании (локальные или облачные), технологии распознавания и синтеза речи.

Обучения чат-ботов отличаются по технологии. Самыми конкурентоспособными на рынке стали технологии обучения искусственного интеллекта по принципу нейросетей, когда чат-бота обучают на выборках ответов. Ему показывают конкретные запросы от пользователей, а он обучается сортировать такие запросы и схожие с ними по смыслу в определенный класс. Высокая эффективность алгоритма напрямую зависит на скорость обучения, ведь для обучения нужно намного меньше примеров для более качественного результата вывода ответа.

Зачем чат-боту нужны лингвисты? Формирование правил, по которым искусственный интеллект в чат-боте обучается определять смысл запроса, а не просто ищет ключевые слова в тексте, лежит на плечах лингвистов. Эти специалисты умеют программировать и пишут на специальном языке алгоритм поведения бота. В компании Just AI, которая занимается разработкой ботов, должность программиста, который занимается ботами, называется «лингвист-разработчик». Специалисты подобного характера обучают ботов определять числительные, написанные текстом, распознавать смысл запроса от пользователя с ошибками в произношении или орфографии, с произвольным порядком слов или сленгом.

Почему чат-боты несовершенны? В интернете существует огромное количество постов, скриншотов и форумов, которые в формате высмеивания анализируют чат-ботов по их ответам. Но это не исключает факт того, что в работе настоящих сотрудников число подобных публикаций не меньше. В обоих случаях проблемы одинаковы: нет времени на дополнительное обучение, поверхностное планирование задачи, недостаточная база ответов.

Чат-бот на 80% состоит из уже заготовленных ответов, которыми предварительно обучили. Без обучения на примерах ни один искусственный интеллект или нейросеть не способны в совершенстве определять запросы, а грамотно реализованный код программы не справится с продажей или возвратом товара в магазине. Это способен решить правильный контент. Чем лучше компания осведомлена о пожеланиях своих клиентов, чем больше информации способна предоставить об их запросах, тем эффективнее будут ответы бота.

Программист должен принимать в учет все мелочи в работе с контентом при разработке программного кода: каждая фраза должна обрабатываться верно, извлекая и предоставляя нужные данные. Подход к обучению чат-бота преимущественно должен быть осуществлен самым простым способом: каждый день запросы клиентов меняются, и бот должен следовать каждому обновлению, дабы не терять актуальности.

Преимущества чат-ботов перед мобильными приложениями. Компании часто спешат – поскорее осуществить запуск, опередить всех своих конкурентов, лучше и быстрее приступить к работе. Владельцам компаний кажется, что работать через сайт или почту лучше. Бот, в свою очередь, кажется бесполезным, а на самом деле часто это – новый элемент продаж, который повышает коэффициент клиентоориентированности. На практике клиенты часто не хотят пользоваться сайтами. А приложений на рынке огромное количество, что они быстро приедаются и от них избавляются, иногда даже ни разу попользовавшись. Памяти на устройствах у многих пользователей также не хватает, поэтому в большинстве случаев установка нового приложения не их выбор. По статистике GlobalWebIndex около 80 % пользователей активно пользуются всего тремя приложениями, среди которых есть хотя бы одна соцсеть или мессенджер.

Бот для компании создается куда дешевле, чем создание отдельного полноценного приложения. Настройка и откладка приложения, что является немаловажным фактором для правильности работы приложения, занимает длительный отрезок времени. Вложение денег в данный процесс не гарантирует, что после успешной реализации отдельного приложения клиентам оно будет интересно.

В боте легче прописать сценарий для коммуникации. Например, для покупки билетов в кино пользователю не захочется устанавливать приложение на свое устройство. Ему проще напишет боту в списке контактов.

Боты портативны. Они интегрируются в приложение, на сайт, в программу технической поддержки или на другую платформу, которую хочет клиент или же владелец компании. Яркими примерами можно назвать ботов на сайте «Сбербанка», бот Яндекс.Навигатора, в чатах в Telegramm, Discord боты, осуществляющие огромное количество функций.

Поэтому заинтересовать аудиторию через какие-то новые приложения намного сложнее, чем через уже установленные. Это подтверждают данные из Quantcast. Особенно это касается мессенджеров и соцсетей, которые установлены на устройства пользователей уже давно. Эти достоинства использования ботов не единственные.

Как используются боты для бизнеса, для работы с клиентами?

Во время работы с клиентом бот выполняет такие операции:

– оформление;

– ответы на вопросы;

– уточнение;

– заказ;

– техническая поддержка.

Для работы с сотрудниками:

Не менее популярны корпоративные чат-боты, которых используют для оказания услуг сотрудникам фирмы. Это определенные электронные действия, которые предприятие предлагает работникам. Через него можно:

– получать информацию о товаре на складе;

– выполнять заказы недостающего товара;

– уточнять время работы руководства или определенного сотрудника.

Крупные компании предлагают клиентам на выбор несколько разных чат-ботов из социальных сетей. Можно выбрать удобный способ связи.

На данный момент закладывают 3 базовые функции в ботов, которые в будущем возможно адаптировать под конкретную цель. Это три наиболее популярных среди них:

– бот-диалог. Все общение с пользователем будет происходить в режиме вопрос-ответ. Пользователь вводит запрос, а бот – воспринимает их и реагирует. Он может оформить заказ, найти нужную информацию и так далее. По этому принципу работает техническая поддержка или большинство Telegramm ботов. Данный вариант используется при потребности выявить желания покупателя.

– бот-автоматическая серия ответов. Больше всего такие боты похожи на автоматическую рассылку, которая необходима для прогрева и подготовки клиентов. С его помощью можно проинформировать клиентов об услугах, нововведениях и изменениях в сервисе.

– бот-меню. Клиент должен нажимать на кнопки, чтобы получить нужную для него информацию. Обычно так себя ведут боты технической поддержки или навигации по приложению или сайту. Можно осуществить список или справочник, в которых вопросы и ответы будут храниться.

# **4 Анализ предметной области и выбор метода решения задачи**

Общение между людьми в мессенджерах является неотъемлемой частью жизни почти каждого человека, живущего в мире технологий, развития и прогресса. И как раз внедрение искусственного интеллекта в повседневное общение людей является перспективным направлением применения нейронных сетей, поскольку задача упрощения общения, экономии времени, удобство использования приложений является немаловажным аспектом пользователей. Это техническая поддержка, поиск и проигрывание музыки, создание чатов для общения с адаптированными ролями участников и т.д. Нейронные сети очень хорошо справляются с задачами подобного рода данных, поскольку они способны решать их, основываясь на выявленных скрытых закономерностях.

В данной работе будет рассмотрена теоретическая модель искусственной нейронной сети, способной получать информацию о пользователе и сервере, введение отчетов или любого другого вида данных, исходя из заполненных Google форм, поиска музыки и предоставления всевозможного развлекательного контента. В дальнейшем, будет рассмотрена практическая реализация решения данной задачи.

# **5 Нейронные сети на Python**

В последние десятилетие наблюдается высокий интерес к нейронным сетям. Глубокие нейронные сети сейчас являются одним из самых популярных методов интеллектуального анализа данных. Почти во всех применяемых областях они показывают более качественные результаты, чем другие методы машинного обучения. Интенсивное развитие технологий машинного обучения привело к появлению готовых к использованию библиотек, в которых уже реализована аппаратная часть, создание архитектур сетей, отвечающая за вычисление функций ошибок, представление нейронов и их связей между собой, их типов и т.п. Более того, стали появляться программные средства, позволяющие создавать сложные нейронные сети без необходимости писать код.

Одним из лидирующих языков, используемых в создании искусственных нейронных сетей, является язык Python, обладающий простотой в использовании и удобством, позволяющий с минимальными затратами времени и сил разрабатывать сложные алгоритмы. Python имеет большое количество библиотек для работы с нейронными сетями. Наиболее популярными считаются TensorFlow, разработанная компанией Google, Theano, являющаяся библиотекой численного вычисления, и Keras, являющаяся надстройкой для вышеперечисленных библиотек.

Перед началом применения на практике необходимо разобраться с проблемами, которые могут возникнуть. Так как заранее размер сети (её сложность), необходимый для точного отображения, неизвестен, то построенная сеть может оказаться чрезмерно большой. Заранее должны быть определены число слоев и число элементов в них, т.е задана архитектура сети. Эта архитектура нейронной сети порождает сложные нелинейные разделяющие поверхности в пространстве входов. В нейронной сети с одним скрытым слоем, векторы образцов первоначально преобразуются в новое пространство представлений (пространство скрытого слоя), а затем пространство гиперплоскости, соответствующие выходным узлам, размещаются так, чтобы разделить классы уже в новом пространстве. А значит, сеть распознает уже иные характеристики – «характеристики характеристик», полученные в скрытом слое. Все это указывает на необходимость этапа предварительной обработки входных данных.

При построении классификатора на основе нейронной сети можно выделить следующие этапы:

1. данные:
2. разработать базу данных из примеров, характеристик для поставленной задачи.
3. разделить весь объем данных на два множества: обучающее и тестовое.
4. предварительная обработка:
5. выбрать систему признаков, характерных для поставленной задачи, и преобразовать данные соответствующим для подачи на вход сети. В результате желательно получить линейно отделимое представление множества образцов.
6. выбрать систему кодирования выходного значения или значений.
7. конструирование, обучение, тестирование и оценка качества сети:
8. выбрать топологию сети: число элементов и структуру связей (входы, слой, выходы).
9. определить функцию активации, которая будет использоваться.
10. определить нужный алгоритм обучения.

Оценить работу нейронной сети в зависимости от её сложности с целью оптимизировать архитектуру.

# **6 Примеры работы Discord ботов**

* Настройка приветствия и прощания с участником с выдачей роли:

Данная функция позволяет автоматизировать приглашение и выдачу роли новым участникам сервера, что намного облегчает этот процесс и экономит время для создателя сервера.

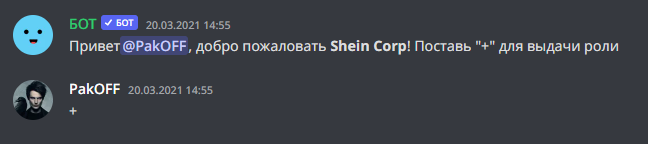


Рисунок 2 – пример приветствия и выдачи роли

* ведение отчетности с тегом роли ответственного участника, исходя из заполненной Google формы:

Данная функция позволяет вести отчетность по разному виду информации, предварительно полученной из заполненной пользователем Google формы. Здесь реализован тег ответственной за ведение отчетности роли, что, в свою очередь, осуществляет заметность данного сообщения в виде отдельного уведомления с большим приоритетом.

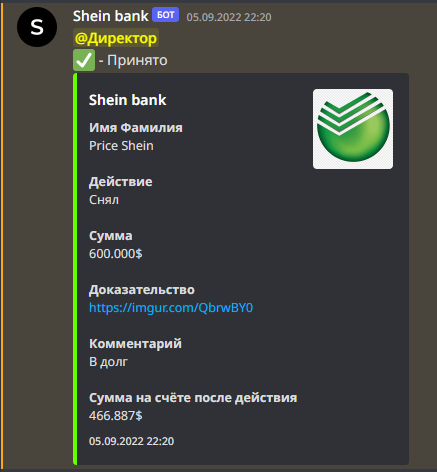


Рисунок 3 – пример введения отчетности с тегом роли ответственного участника

* поиск и проигрывание музыки:

Данная функция позволяет найти нужный пользователю трек только по названию композиции, что облегчает и ускоряет процесс поиска. После успешного нахождения трека, бот автоматически добавляется в голосовой канал, где уже находится пользователь и начинает воспроизведение найденной композиции.

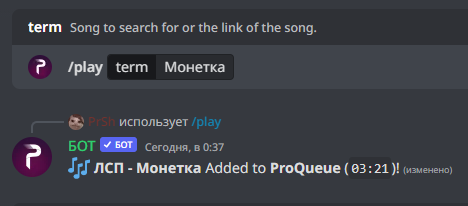


Рисунок 4 – пример поиска трека по названию и последующего проигрывания

* генерирование случайного числа в заданном диапазоне:

Данная функция позволяет сгенировать случайное число в заданном пользователем диапазоне от 1 до n, где n – число, введенное пользователем.

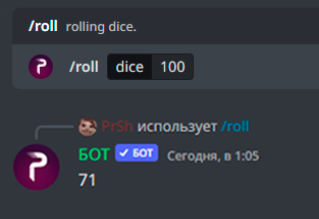


Рисунок 5 – пример генерации случайного числа, исходя из заданного пользователем диапазона

* получение информации об участнике сервера и о самом сервере:

Данная функция позволяет получить информацию о любом участнике сервера. На рисунке 6 пример информации о профиле конкретного участника сервера, а именно его уровень, репутацию и кредиты, ранг и количество опыта на данный момент. Настройка вывода определенной информации об участнике прерогатива разработчика.

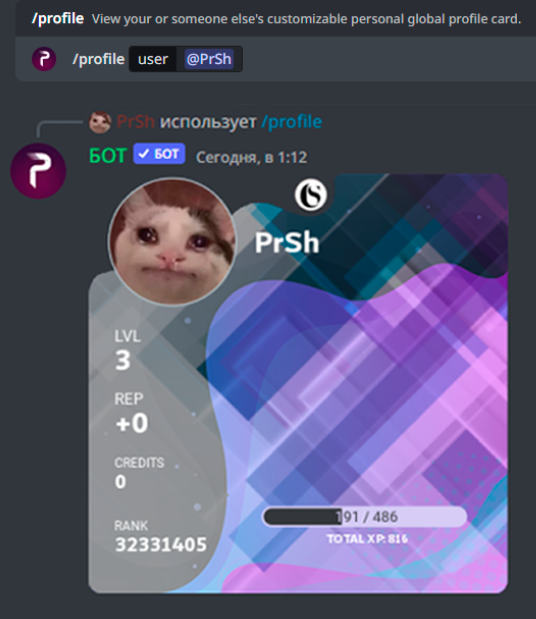


Рисунок 6 – пример информации об участнике сервера

На рисунке 7 уже вывод информации в целом о пользователе Discord, а именно идентификатор, дата создания учетной записи Discord и присоединения на сервер.

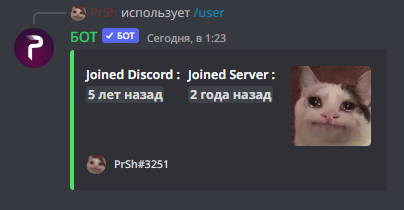


Рисунок 7 – пример информации о пользователе

На рисунке 8 виден вывод информации о созданном сервере, а именно ID сервера, дата создания, кем был создан, количество участников, количество текстовых и голосовых каналов, количество ролей.

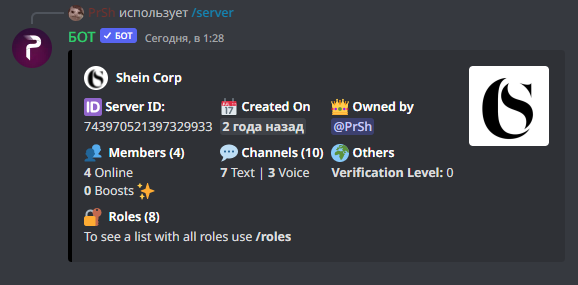


Рисунок 8 – пример информации о сервере

# **7 Практическая часть**

Перед написанием кода для Discord бота сначала нужно его создать. В этом нам помогает официальный сервис Discord Developer Portal, в котором мы можем настроить имя бота и изображение профиля с настройкой описаний, получить так называемый Token для дальнейшей идентификации бота в нашем коде и подключении его к серверам, настроить его права, функционал и возможности.

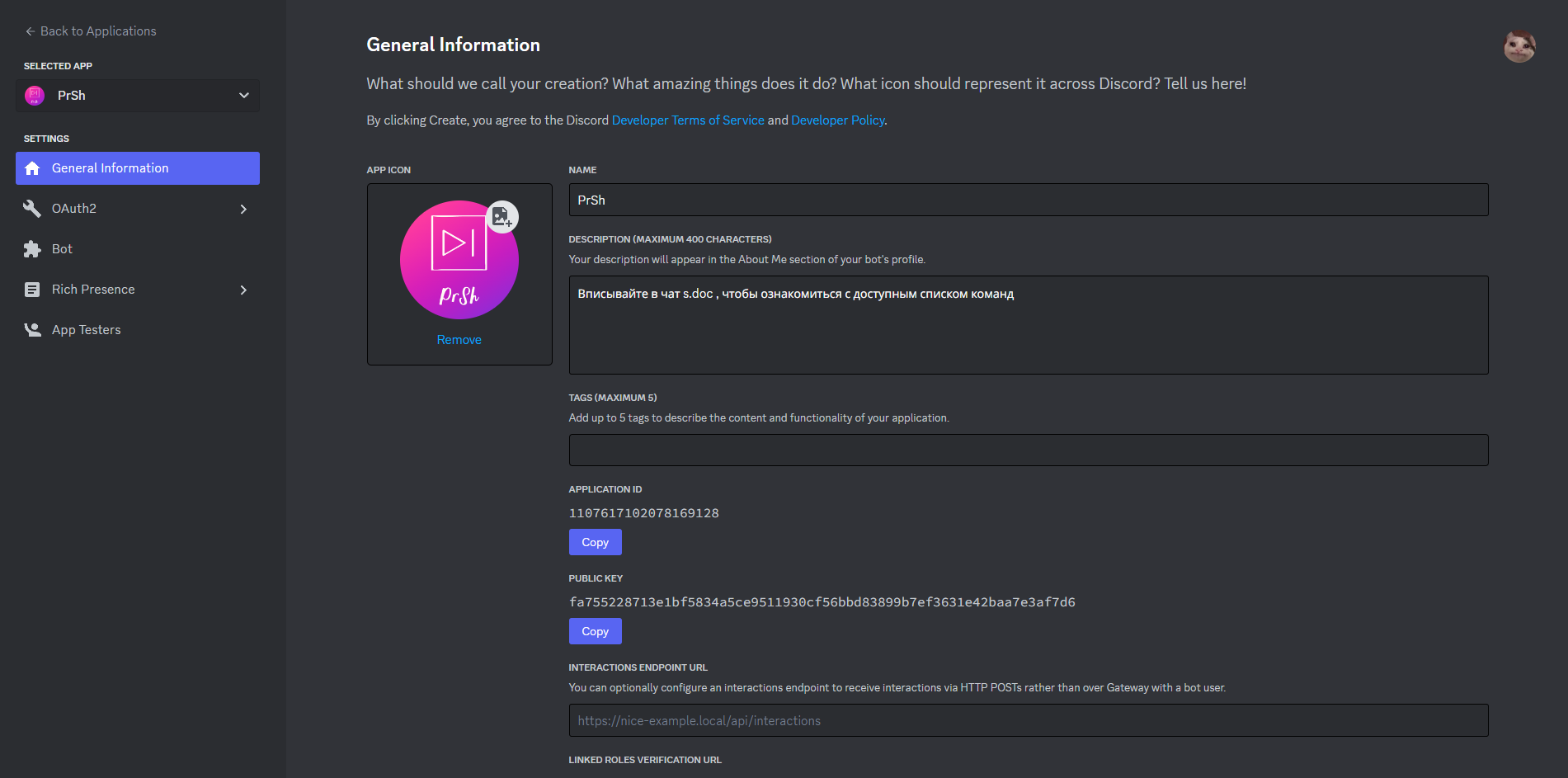


Рисунок 9 – пример созданного аккаунта бота

Рассмотрим получение Token’а для дальнейшего взаимодействия с ботом и подключения его к серверам на рисунке 10:

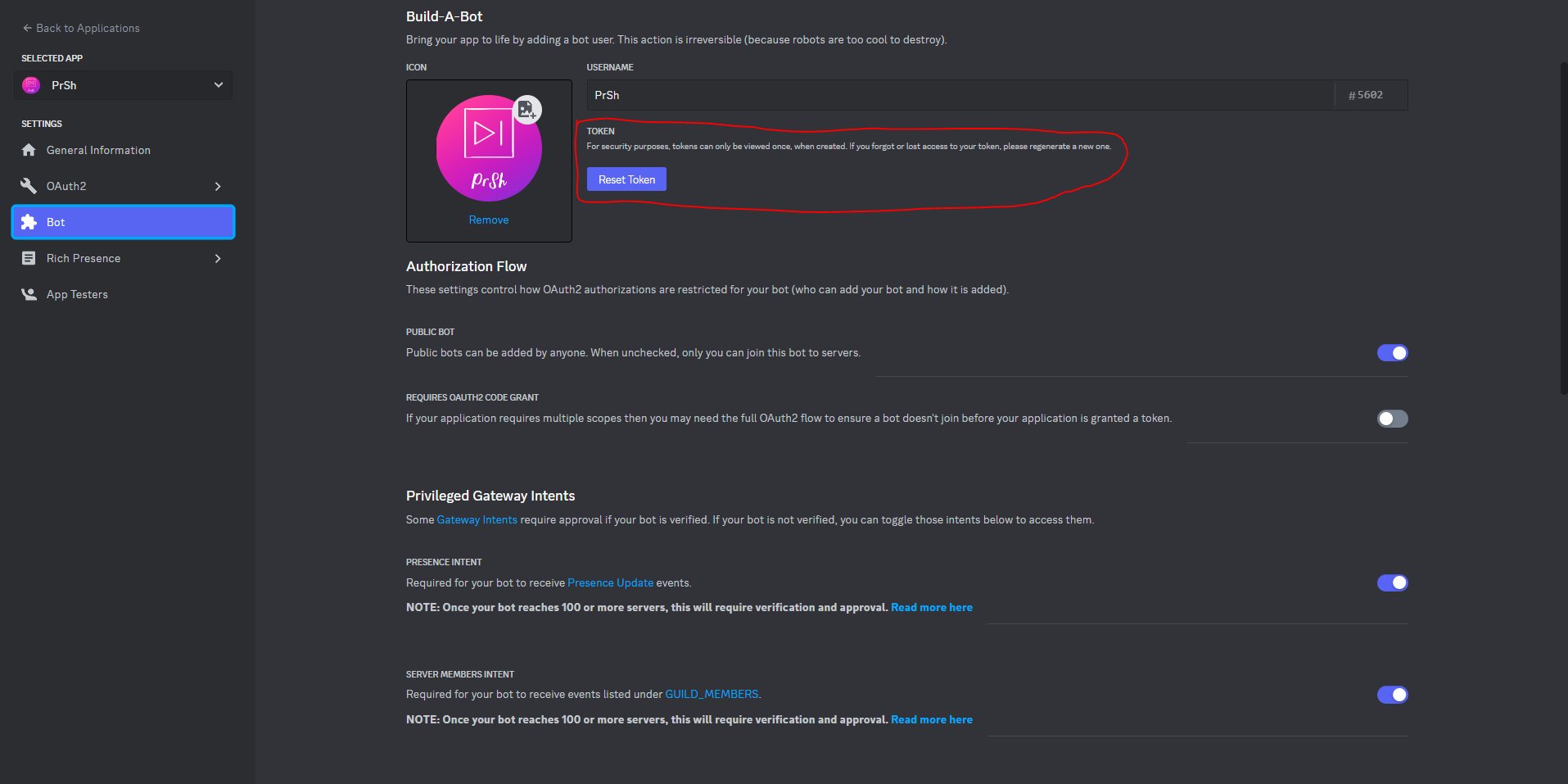


Рисунок 10 – пример получения Token’а

А также предоставление прав, функционала и возможностей боту на рисунке 11:

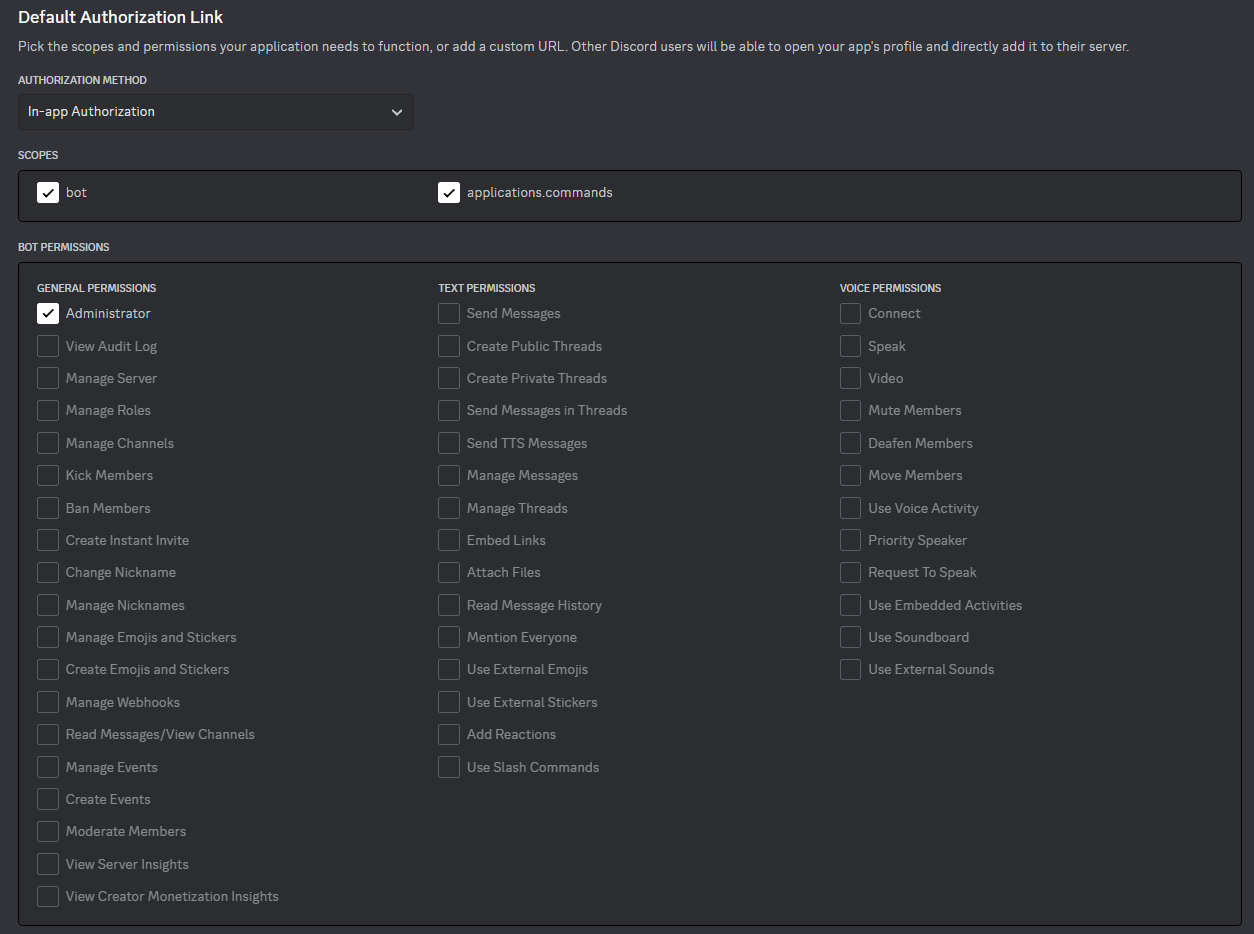


Рисунок 11 – пример предоставления прав и функционала боту

При выборе роли Administrator бот получает все права и обладает возможностями подобно создателю сервера. При разработке бота я выбрал именно роль Administrator, т.к. бот в первую очередь был создан для применения на личных серверах. В дальнейшем уже при продвижении своего проекта рекомендуется выбирать конкретные права и возможности бота для гарантии безопасности, ведь если Token бота попадет в плохие руки, то взломщик получит все права от сервера, на котором находится бот с ролью Administrator.

После выполнения первых шагов получим:

1. аккаунт бота;
2. Token для подключения бота.

Собственно, вот и всё. На данном этапе наш бот полностью пассивен.

Затем мы переходим уже непосредственно к написанию кода программы, создав предварительно файл main.py, первым делом нам стоит подключить огромное количество библиотек с последующей загрузкой их через командую строку и добавлением переменных в PATH системы, для дальнейшей их успешной работы. В нашем случае при выборе языка программирования Python основной библиотекой является discord.py. Все остальные библиотеки подключаются по мере надобности и добавления функционала боту. Пример подключенных библиотек можно увидеть на рисунке 12:

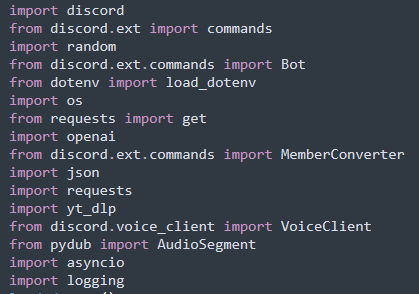


Рисунок 12 – пример подключения библиотек

Также для взаимодействия с ботом и подключения его на сервера, нам нужно вставить в программу Token и префикс для взаимодействия, в моем случае префиксы поделились на две категории, a. для музыки и s. для всех остальных команд. При реализации своего бота я добавил взаимодействие с популярным сервисом искусственного интеллекта OpenAI, который в дальнейшем поможет получать информацию для некоторых команд. Пример вставки в код видно на рисунке 13:

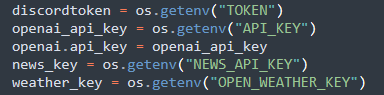


Рисунок 13 – пример вставки Token’а и подключение сервиса OpenAI

После всех вышеописанных шагов приступаем к запуску бота и подключения его к серверу с помощью функций @bot.event и on\_ready(), с возможностью записи всех ошибок в отдельный файл discordBot.errors, продемонстрировано на рисунках 14 и 15:

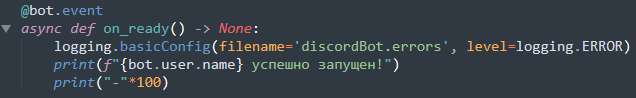


Рисунок 14 – пример запуска бота

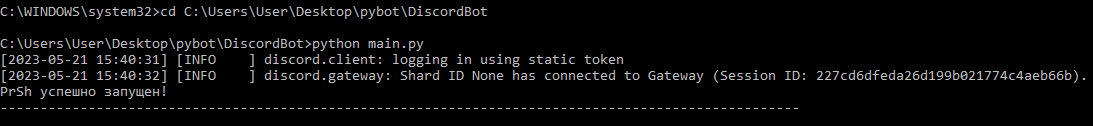


Рисунок 15 – успешный запуск бота

Также одна из функций бота является уникальное приветствие к нескольким личным серверам. Бот проверяет, к какому именно серверу он подключен и от этого выбирает уникальное приветствие для новоприбывшего пользователя. Видно это на рисунках 16 и 17:

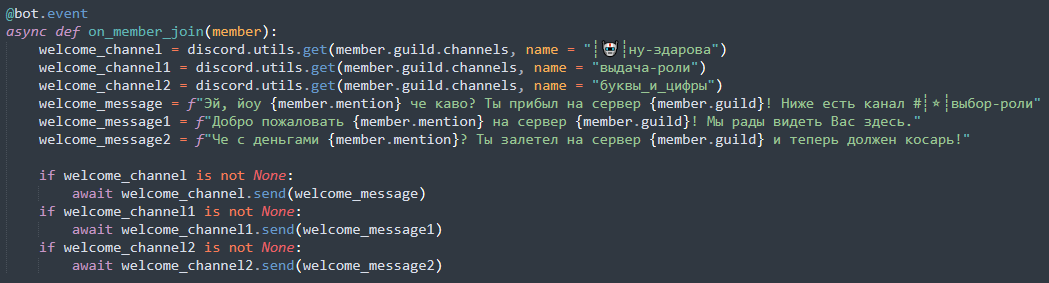


Рисунок 16 – код уникальных приветствий



Рисунок 17 – результат работы кода уникальных приветствий

Переходим уже непосредственно к написанию функциональных команд, с которыми каждый пользователь сервера, на который подключен бот, сможет взаимодействовать. Одна из основных команд является ознакомление со всем списком доступных команд пользователя, ведь без них невозможно взаимодействие с ботом. Реализация показана на рисунках 18 и 19:

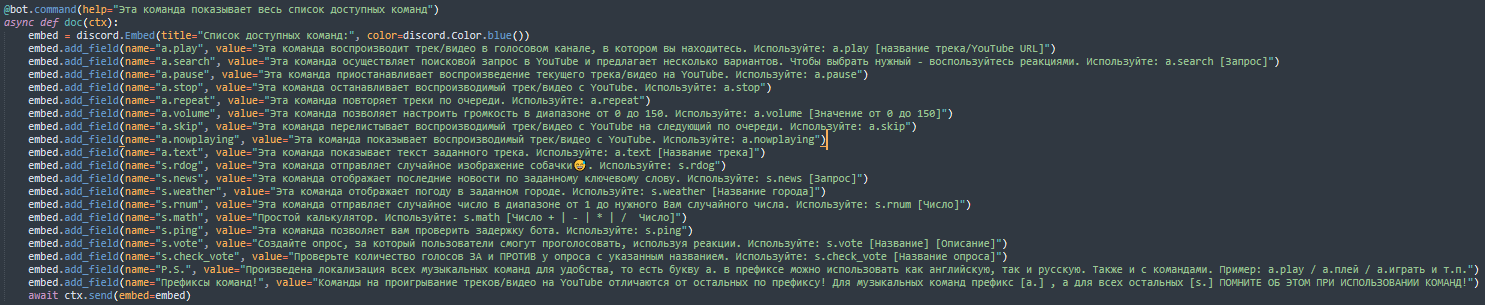


Рисунок 18 – код списка доступных команд

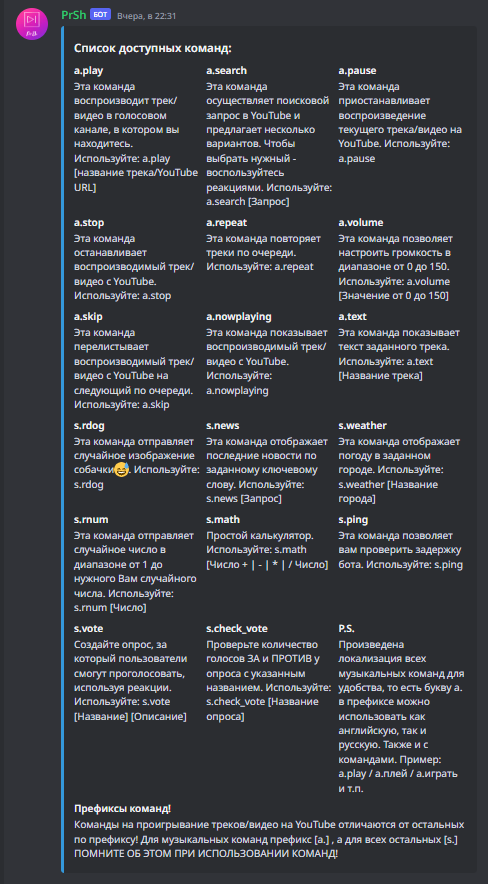


Рисунок 19 – результат работы кода доступных команд

Одним из основных направлений при разработке Discord бота – проигрывание музыки, ведь по личному опыту именно за этим функционалом я раньше добавлял на свои сервера Discord ботов. Реализация происходила с помощью удобного и популярного сервиса видео хостинга YouTube, что видно на рисунке 20:

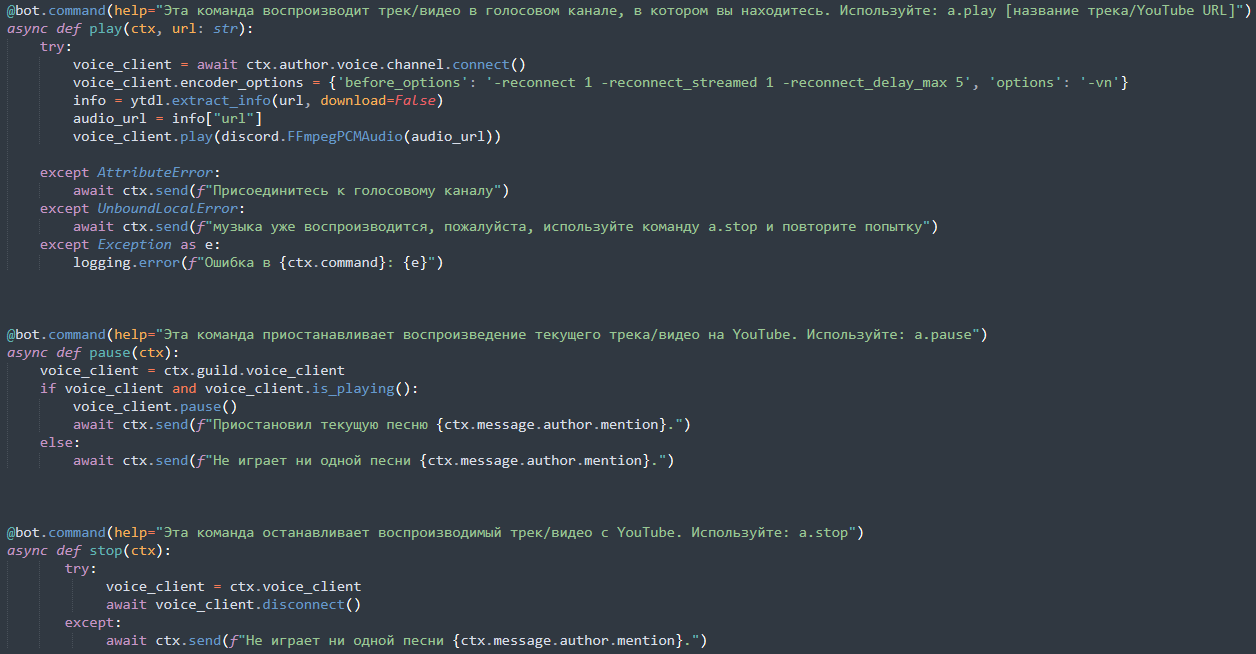


Рисунок 20 – фрагмент кода реализации проигрывания музыки

Ну и конечно добавлена возможность взаимодействия с сервисом искусственного интеллекта OpenAI и сторонним сервисом, который выдает случайное фото собаки, для получения которой пользователю всего лишь нужно вписать команду s.rdog. А уже для взаимодействия с сервисом искусственного интеллекта OpenAI предоставлено две команды: s.news – выдает 5 последних найденных новостей по заданному запросу от пользователя и s.weather – отображает погоду в заданном городе. Реализация отображена на рисунке 21, а результат работы на рисунках 22, 23, 24:



Рисунок 21 – код команд rdog, news и weather

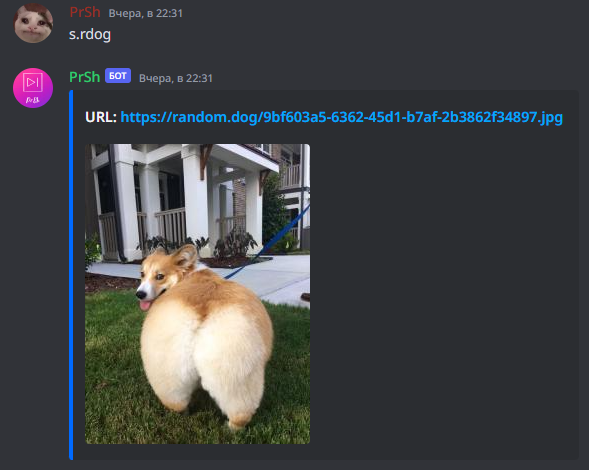


Рисунок 22 – результат команды rdog



Рисунок 23 – результат команды news

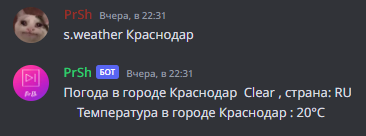


Рисунок 24 – результат команды weather

Добавлены команды, которые работают исходя из внутренних возможностей языка программирования Python, таких как randint() и eval(expression). Команда rnum отправляет случайное число в диапазоне от 1 до заданного числа, а команда math – это простой калькулятор. Реализация отображена на рисунке 25, а результат работы на рисунках 26, 27:

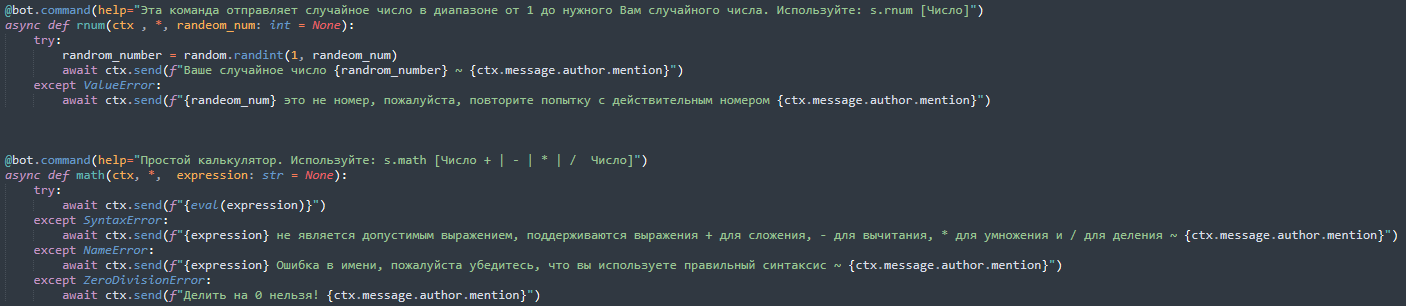


Рисунок 25 – код команд rnum и math

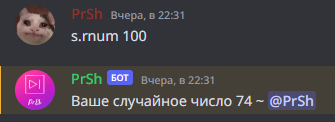


Рисунок 26 – результат команды rnum

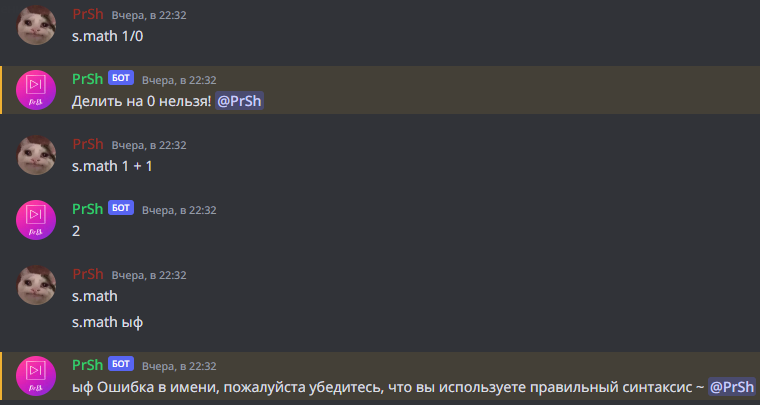


Рисунок 27 – результат команды math

# **ЗАКЛЮЧЕНИЕ**

Системы искусственного интеллекта, работающие на основе нейронных сетей, показывают большие успехи в анализе и классификации огромных наборов данных. Они позволяют автоматизировать различные процессы деятельности человека. В результате, находятся новые области для применения таких систем и методы оптимизации алгоритмов, разработанных для решения различных прикладных задач.

В данной курсовой работе рассмотрены основные принципы работы искусственных нейронных сетей: строение, виды, методы обучения, сбора и анализа данных. На основе изученной информации создана теоретическая модель нейронной сети для работы бота, который решает задачи модерации текстовых каналов, получения информации о пользователе или сервере, настройка приветствия и прощания с участниками, управление музыкой.

В данной курсовой работе осуществлена реализация Discord бота на языке программирования Python, произведена работа по подключению нейронных сетей и внутреннего функционала языка программирования Python в доступный список команд, произведена оценка возможности будущего продвижения проекта на рынок.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Обучение нейросети с учителем, без учителя, с подкреплением – в чем отличие? Какой алгоритм лучше? URL:<https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/> (дата обращения: 10.12.2022).

2. От чат-ботов к разговорному ИИ: разработка умных ассистентов для бизнеса. <URL:https://vc.ru/promo/42838-aimylogic> (дата обращения: 10.12.2022).

3. От чат-ботов к разговорному искусственному интеллекту. <URL:https://konveier.com/blog/2018/08/02/ot-chat-botov-k-razgovornomu-iskusstvennomu-intellektu/> (дата обращения: 11.12.2022).

4. Что такое нейронная сеть? URL:<https://ru.wikipedia.org/wiki/Нейронная_сеть> (дата обращения: 11.12.2022).

5. Зенин А. В. Исследование возможностей использования нейронных сетей / А.В.Зенин // Журнал молодой ученый. – 2017. – №16(150). – С. 130–140.

6. Рашид Т. Создаем нейронную сеть / Т.Рашид. – Пер. с англ. – Спб.: ООО «Альфа-книга», 2017. – 272 с.

7. Николенко, С. Глубокое обучение / С. Николенко, А. Кадурин, Е. Архангельская. – СПб.: Питер, 2019. – 480 с.

8. Haykin S. Neural networks and learning machines / S. Haykin. – by Pearson, 2009. – 938 p.

9. Элбон К. Машинное обучение с использованием Python. Сборник рецептов / К. Элбон. – Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 384 с.

10. Горбань А. Н. Обобщённая аппроксимационная теорема и вычислительные возможности нейронных сетей. Архивная копия от 27 января 2012 на Wayback Machine // Сибирский журнал вычислительной математики, 1998, т. 1, № 1. – С. 12–24.

11. Открытые системы – введение в нейросети. Архивная копия от 31 октября 2005 на Wayback Machine.

12. Руководство по использованию библиотеки discord.py. URL:<https://github.com/denisnumb/discord-py-guide/blob/main/discord-py.md> (дата обращения: 13.05.2023).

13. Создаем Discord-бота на Python. <URL:https://tproger.ru/articles/sozdajom-discord-bota-na-python/> (дата обращения: 14.05.2023).

