Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет имени

Гагарина Ю.А.»

Институт машиностроения и материаловедения и транспорта (ИММТ)

Кафедра **«**Техническая механика и мехатроника»

**Курсовая работа**

по дисциплине: **«**Искусственный интеллект и нейрокомпьютеры»

на тему: «Разработка программного интерфейса для управления мобильными роботами с применением нейронных сетей»

Выполнил:   
студент группы

б-МХРТ-41  
Куприн Максим Сергеевич

Проверил: Старший преподаватель

Ревякин Владислав Анатольевич

Саратов 2024

Содержание

[1. Введение 3](#_Toc163722441)

[2. Теоретическая часть 3](#_Toc163722442)

[3. Практическая часть 6](#_Toc163722443)

[4. Заключение 11](#_Toc163722444)

[Список литературы 12](#_Toc163722445)

1. Введение

Использование мобильных роботов становится все более распространенным и востребованным. Они применяются в различных областях, начиная от производства и складского хозяйства, и заканчивая исследованиями в области искусственного интеллекта. Для эффективного управления мобильными роботами необходимо разработать программный интерфейс, который позволит управлять ими с применением нейронных сетей.

Целью данной работы является разработка программного интерфейса для управления мобильными роботами с использованием нейронных сетей и программное обеспечение, реализующее работу нейронной сети.

Задачи исследования включают в себя анализ существующих методов управления мобильными роботами, разработку архитектуры программного интерфейса, обучение нейронных сетей для управления роботами, тестирование и оптимизацию разработанного интерфейса. Полученные результаты могут быть применены в различных областях, где требуется эффективное управление мобильными роботами с использованием современных технологий и методов.

Тестировать разработанный интерфейс планируется в среде имитационного моделирования RSMA.

1. Теоретическая часть

Для разработки программного интерфейса для управления мобильными роботами с использованием нейронных сетей необходимо учитывать основные принципы работы нейронных сетей и способы их применения в управлении роботами.

Нейронные сети являются математическими моделями, которые могут обучаться на основе данных и использоваться для решения разнообразных задач, включая управление движением роботов. В данном случае, на вход нейронной сети подаются данные о текущих координатах робота и координатах точки, в которую он должен приехать. На выходе нейронная сеть выдает значения управляющих сигналов для моторов робота, позволяя ему двигаться в нужном направлении.

Для реализации программного интерфейса необходимо провести следующие шаги:

1. Подготовка данных: необходимо подготовить обучающую выборку, состоящую из пар координат робота и соответствующих управляющих сигналов для моторов. Также необходимо определить архитектуру нейронной сети и выбрать метод обучения.

2. Обучение нейронной сети: после подготовки данных необходимо обучить нейронную сеть на обучающей выборке. Обучение можно провести с использованием различных методов, таких как обратное распространение ошибки или генетические алгоритмы.

3. Разработка программного интерфейса: на основе обученной нейронной сети необходимо разработать программный интерфейс, который будет получать текущие координаты робота и точку, к которой он должен приехать, и отправлять управляющие сигналы на моторы.

4. Тестирование и оптимизация: после разработки программного интерфейса необходимо провести тестирование его работы на реальном роботе и провести оптимизацию параметров нейронной сети и алгоритмов управления для достижения лучших результатов.

В работе планируется использовать нейронные сети прямого распространения. Нейронные сети прямого распространения (Feedforward Neural Networks) – это один из наиболее распространенных типов нейронных сетей, которые представляют собой ациклический граф нейронов, где информация передается только в одном направлении – от входных нейронов к выходным.

Основными компонентами нейронной сети прямого распространения являются нейроны, слои и веса. Нейроны – это базовые элементы нейронной сети, которые объединены в слои. Существует три типа слоев нейронных сетей: входной слой, скрытые слои и выходной слой. Входной слой принимает входные данные, скрытые слои выполняют обработку информации, а выходной слой выдает результат обработки.

Каждый нейрон связан с нейронами предыдущего и следующего слоя с весами, которые определяют важность каждой связи. На вход нейрона поступают данные, которые затем умножаются на соответствующие веса и суммируются. Затем активационная функция применяется к полученной сумме для определения активации нейрона.

Обучение нейронной сети прямого распространения происходит путем настройки весов с использованием алгоритмов оптимизации, таких как обратное распространение ошибки (backpropagation). Этот процесс заключается в подаче обучающих данных на вход нейронной сети, расчете ошибки между выходом сети и ожидаемым результатом, и обновлении весов в соответствии с этой ошибкой с целью минимизации ее.

Нейронные сети прямого распространения широко применяются в различных областях, таких как обработка изображений, распознавание речи, прогнозирование временных рядов и управление роботами. Их преимущества включают возможность обучения на больших объемах данных, способность к обобщению и высокую гибкость в моделировании сложных взаимосвязей в данных.

1. Практическая часть

С применением языка высокого уровня Python разработан класс Neuron, реализующий поведения и свойства модели нейрона МакКаллока-Питтса., дополненные возможностью смены функции активации.

Класс Neuron содержит следующие поля:

* inputs – список входных значений нейрона;
* weights – список весов нейрона;
* activation – Enum, описывающий тип функции активации;
* normalize – опция отвечающая за нормализацию входных значений нейрона при его расчете;
* offset – смещение значения взвешенной суммы нейрона;
* output – выходное значение нейрона;
* error – величина ошибки, используется при обучении;
* gradient – градиент ошибки, используется при обучении;
* initial\_weights – список изначальных весов, используется при обучении.

Класс Neuron содержит один метод calculate, выполняющий расчет выходного значения нейрона в соответствии с моделью МакКаллока-Питтса. Блок-схема метода представлена на рисунке 1.

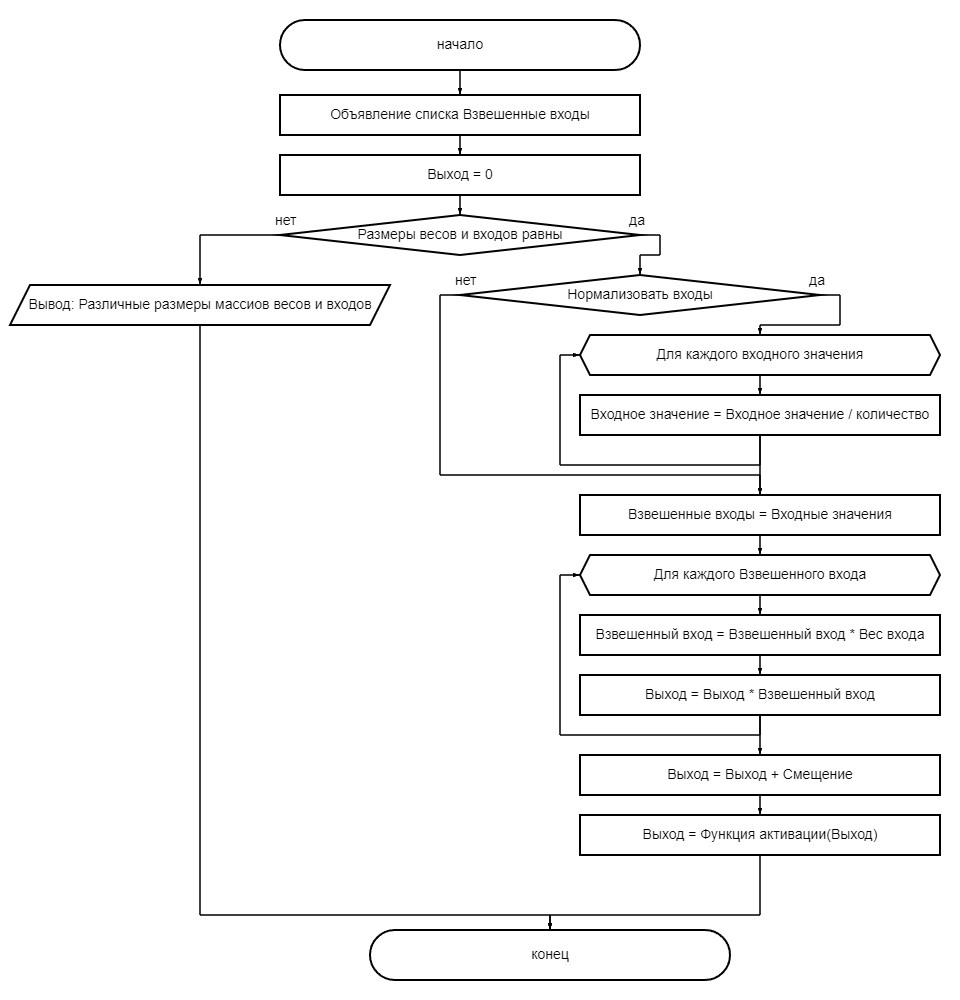


Рис. 1. Блок-схема метода calculate

С применением языка высокого уровня Python разработан класс Layer, хранящий список нейронов и список их выходных значений.

Класс Layer содержит следующие поля:

* neurons – список нейронов, входящий в слой;
* outputs – список выходных значений нейронов слоя;

Класс Layer, аналогично классу Neuron, содержит метод calculate, выполняющий расчет выходных значений каждого нейрона в слое, последовательно вызывая метод calculate каждого нейрона. Блок-схема метода представлена на рисунке 2.

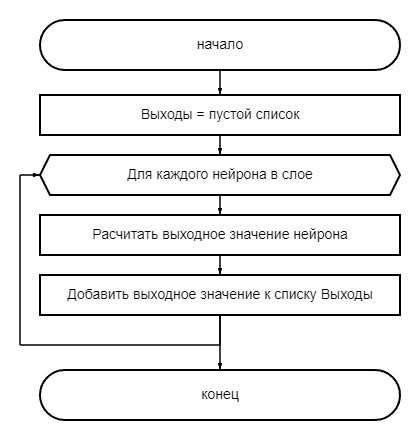


Рис. 2. Блок-схема метода calculate класса Layer

С целью расширения функционала программного обеспечения разработан класс Network, реализующий функционал нейронной сети, состоящей из одного и более слоев.

Класс Network содержит следующие поля:

* inputs – список входных значений сети;
* layers – список слоев, входящих в нейронную сеть;
* outputs – список выходных значений сети;

Класс Network, содержит метод calculate, выполняющий последовательный расчет выходных значений каждого слоя сети. Блок-схема метода calculate представлена на рисунке 3.

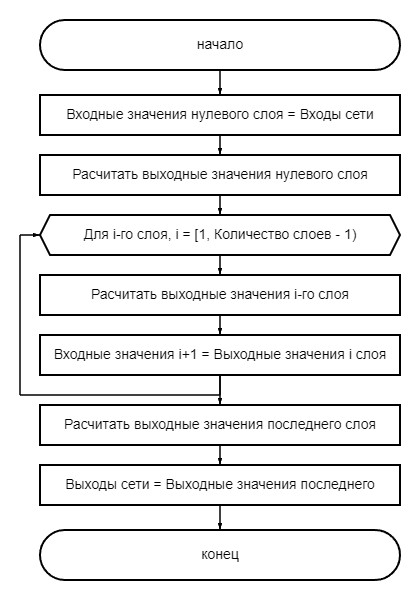


Рис. 3. Блок-схема метода calculate класса Network

Для управления мобильным роботом применялся программный интерфейс RSMA-API и язык программирования Python. Архитектура сети – трехслойная сеть прямого распространения. Входной слой содержит 2 нейрона с шестью входами для чтения двух векторов координат, второй слой – 4 нейрона, третий слой – 2 нейрона. Выходы сети подключены к электродвигателям мобильного робота, представленного на рисунке 4.

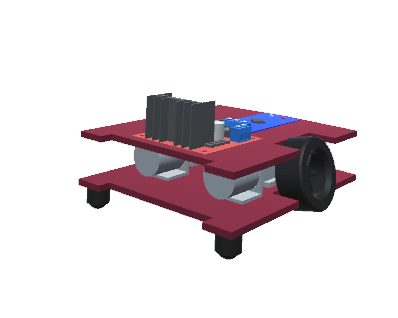


Рис. 4. Мобильный робот DarkieBot

Для настройки весов применялся алгоритм искусственного отбора нейронных сетей. При создании робота, сеть, выполняющая управление, получала случайные веса. Затем в цикле выполнялся расчет сети и управление электродвигателями. В течении 10 секунд робот может перемещаться по сцене. По истечении срока результат робота сохраняется, как и веса сети.

По истечении 10 циклов обучения, из сохраненных данных выбирается одна сеть, которая наиболее близко приблизилась к целевой точке.

Веса выбранной сети случайным образом изменяются (мутируют) и запускается новый цикл обучения.

В ходе выполнения работы собирались данные о положении робота, представленные на рис. 5. За 5 часов не удалось достичь заданной точки. По графику отклонения видно, что ошибка со временем уменьшается, соответственно обучение происходит в верном направлении.

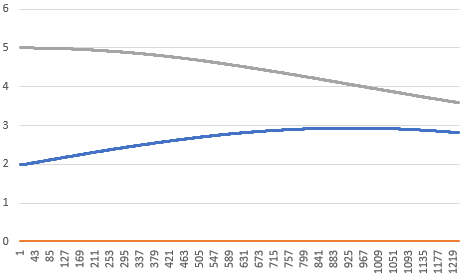


Рис. 5. Отклонение положения от заданной точки

1. Заключение

В ходе данной работы был разработан программный интерфейс для управления роботами с использованием нейронных сетей, что позволяет эффективно управлять и обучать роботов для выполнения различных задач. Благодаря применению принципов искусственного отбора нейронных сетей, удалось успешно обучить нейронную сеть, оптимизировав её структуру и параметры для достижения оптимальной производительности и результатов.

Список литературы

1. Васильев, А.Н. Тархов Д.А. Принципы и техника нейросетевого моделирования / А.Н. Васильев. - Москва: Наука, 2017. - 999 c.
2. Галушкин, А. И. Нейрокомпьютеры. Учебное пособие / А.И. Галушкин. - М.: Альянс, 2018. - 528 c.
3. Гелиг, А. Х. Введение в математическую теорию обучаемых распознающих систем и нейронных сетей. Учебное пособие / А.Х. Гелиг, А.С. Матвеев. - М.: Издательство СПбГУ, 2018. - 224 c.
4. Unity - Manual: Introduction to components [Электронный ресурс] // Unity User Manual 2022.3 (LTS) URL: https://docs.unity3d.com/Manual/Components.html
5. Unity - Manual: Hinge Joint component reference [Электронный ресурс] // Unity User Manual 2021.3 (LTS) URL: https://docs.unity3d.com/2021.3/Documentation/Manual/class-HingeJoint.html
6. Unity - Manual: Fixed Joint component reference [Электронный ресурс] // Unity User Manual 2021.3 (LTS) URL: https://docs.unity3d.com/Manual/class-FixedJoint.html
7. Unity - Руководство: Префабы (Prefabs) [Электронный ресурс] // Unity User Manual 2021.3 (LTS). URL: https://docs.unity3d.com/ru/530/Manual/Prefabs.html
8. Unity - Manual: Rigidbody component reference [Электронный ресурс] // Unity User Manual 2022.3 (LTS) URL: https://docs.unity3d.com/Manual/class-Rigidbody.html
9. Unity - Руководство: Создание и Использование Скриптов [Электронный ресурс] // Unity User Manual 2022.3 (LTS) URL: https://docs.unity3d.com/ru/530/Manual/CreatingAndUsingScripts.html
10. Unity - Manual: The Inspector window [Электронный ресурс] // Unity User Manual 2022.3 (LTS). URL: https://docs.unity3d.com/Manual/UsingTheInspector.html
11. Справочник по Markdown для Microsoft Learn - Contributor guide | Microsoft Learn [Электронный ресурс] // Microsoft Learn URL: https://learn.microsoft.com/ru-ru/contribute/content/markdown-reference
12. GitHub [Электронный ресурс] // GitHub. URL: https://github.com/
13. Репозиторий RSMADocs [Электронный ресурс] // GrimDarkTech/RSMADocs. URL: https://github.com/GrimDarkTech/RSMADocs
14. Функциональный возможности RSMA · GrimDarkTech/RSMADocs [Электронный ресурс] // GrimDarkTech/RSMADocs. URL: https://github.com/GrimDarkTech/RSMADocs/blob/main/Features.md
15. Установка RSMA [Электронный ресурс] // GrimDarkTech/RSMADocs. URL: https://github.com/GrimDarkTech/RSMADocs/blob/main/Installation/InstallationRU.md
16. RSMA Manual En[Электронный ресурс] // GrimDarkTech/RSMADocs. URL: https://github.com/GrimDarkTech/RSMADocs/tree/main/Manual/en
17. Unity - Manual: UI Toolkit [Электронный ресурс] // Руководство Unity. URL: https://docs.unity3d.com/Manual/UIElements.html
18. Unity - Scripting API: HelpURLAttribute [Электронный ресурс] // Unity Scripting API 2022.3 (LTS)URL: https://docs.unity3d.com/ScriptReference/HelpURLAttribute.html
19. Unity - Scripting API: Transform [Электронный ресурс] // Unity Scripting API 2022.3 (LTS). URL: https://docs.unity3d.com/ScriptReference/Transform.html
20. Interface. Справочник по C# - C# | Microsoft Learn [Электронный ресурс] // Microsoft Learn URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/interface
21. Наследование в C# - C# | Microsoft Learn [Электронный ресурс] // Microsoft Learn URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/fundamentals/tutorials/inheritance
22. ФИПС - Федеральное государственное бюджетное учреждение Федеральный институт промышленной собственности [Электронный ресурс] // Официальный сайт ФИПС. URL: https://new.fips.ru/
23. Grim Dark // Официальный сайт ООО "ГРИМДАРК ТЕХНОЛОГИИ" URL: https://grimdark.ru/