ORES

Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

ИССЛЕДОВАНИЕ СПОСОБОВ ПРИМЕНЕНИЯ ЯЗЫКОВОЙ МОДЕЛИ НЕЙРОННОЙ СЕТИ ДЛЯ РАЗРАБОТКИ ИГРЫ В ИГРОВОМ ДВИЖКЕ UNITY

Сухоярский Денис Игоревич,

Маевский Илья Александрович,

Студенты, Донской Государственный Технический Университет,
кафедра «Медиатехнологии»,
344000, РФ, г. Ростов – на – Дону, пл. Гагарина 1
E-mail: densuh68@gmail.com, gagapool.55785@gmail.com
09.04.02 Информационные системы и технологии

Аннотация

Использование нейронных сетей в индустрии видеоигр началось относительно недавно. С каждым днём становится всё больше и больше видов нейронных сетей с различным функционалом, а также способов их использования при создании видеоигр.

В этой статье мы опишем некоторые существующие подходы интеграции и использования нейронных сетей в игровом движке Unity. Проанализируем каждую из них и выделим плюсы и минусы их использования.

Ключевые слова: ИИ, искусственный интеллект, разработка игр, квест, генерация сценариев, РПГ, НС, нейронные сети

RESEARCH ON HOW TO USE A NEURAL NETWORK LANGUAGE MODEL TO DEVELOP A GAME IN THE UNITY GAME ENGINE

Denis I. Sukhoyarsky,

Ilya A. Mayevsky

Students, Don State Technical University,
Department of «Media Technologies»
344000, Russian Federation, Rostov – on – Don, pl. Gagarina 1
E-mail: densuh68@gmail.com, gagapool.55685@gmail.com
09.04.02 Information systems and technologies

ABSTRACT

The use of neural networks in the video game industry is relatively new. Every day there are more and more types of neural networks with different functionality, as well as ways to use them when creating video games.

In this article we will describe some existing approaches for integrating and using neural networks in the Unity game engine. Let's analyze each of them and highlight the pros and cons of their use.

Keywords: AI, artificial intelligence, game development, quest, scenario generation, RPG, neural network, neural networks

Введение

Цель: исследовать существующие способы применения языковой модели нейронной сети при разработке игр с использованием игрового движка Unity для генерации текста.

Задачи:

1. Изучить подходы использования языковой модели нейронных сетей для разработки игр на Unity.
2. Выделить преимущества и недостатки каждого из изученных подходов.
3. Обзорно описать необходимые действия по внедрению каждого из подходов.

Искусственный интеллект (ИИ) на текущей стадии своего развития имеет ряд слабых сторон и ограничений. Однако уже сейчас существует большое количество видов и способов использования нейронных сетей и ИИ в игровой индустрии. Они используются для управления поведением неигровых персонажей (NPC), генерации различного уникального контента, создания игрового уровня или улучшения существующего и т.д. [3; 5; 4]

Нейронные сети пока не способны сами решать поставленные задачи, за ними нужен постоянный контроль. Но они могут стать хорошими помощниками для профессионалов и новичков. Новичкам эти технологии могут помочь с освоением базовых принципов в интересующей области. Профессионалам это поможет сэкономить время, поскольку ответы ИИ могут выполнять за них рутинную или специфическую работу.

Таким образом, нейронные сети и ИИ способны помогать в разработке: ускорять разработку и снижать затраты на нее. Также нейронные сети способны создавать необычные сценарии, которые могут быть новы для игрока, что, в свою очередь, разнообразит игровой опыт. Но помимо понимания целей, с которыми разработчики собираются использовать данные технологии, необходимо также знать подходящие способы их внедрения в разработку.

Основная часть (методология, результаты)

Языковая модель нейронной сети (Neural Language Model) – это модель, использующая нейронные сети для предсказания следующего слова или фразы в последовательности текста [1]. Она обучается на большом объеме текста и пытается предсказать наиболее вероятное следующее слово или фразу, учитывая предыдущую последовательность слов. Нейронные языковые модели могут использоваться для различных задач, таких как генерация текста, машинный перевод, обработка естественного языка и т.д.

Языковая модель нейронной сети может использоваться для генерации текста и диалогов в играх, созданных на движке Unity. Например, модель может использоваться для

создания персонажей, которые могут общаться с игроком, или для генерации случайных событий и квестов в игре. Также модель может использоваться для машинного перевода, если игра имеет многоязычный контент.

Существует несколько вариантов внедрения языковой модели нейронной сети в игровой движок Unity.

Первый вариант - создать свою языковую модель и обучить ее на большом объеме текста, который мы хотим генерировать. После того, как модель будет обучена, мы сможем использовать ее для генерации нового текста. Данный подход может быть сложной задачей, так как это потребует много времени, значительных вычислительных ресурсов и специализированных знаний в области машинного обучения и искусственного интеллекта [2]. Поэтому данный способ не применим для обычных разработчиков.

Второй вариант – использовать интеграцию существующей модели нейронной сети по API. Для этого необходимо изучить существующие на рынке модели нейронных сетей и выбрать подходящую. При выборе модели необходимо сразу обратить внимание, есть ли у нее интеграция по API. После чего необходимо протестировать ответы нейросети на интересующую тему. Когда модель нейронной сети будет выбрана, необходимо ее интегрировать в игровой движок. Для генерации сценариев игрового квеста хорошо подходит ChatGPT [6]. На его примере будет описана интеграция в Unity.

API (Application Programming Interface) – это набор правил и процедур, которые позволяют различным программным компонентам взаимодействовать друг с другом. Он определяет, какие функции и методы доступны для использования, а также как они могут быть вызваны и какие параметры они принимают. API может быть как открытым и общедоступным, так и проприетарным и требовать авторизации для доступа.

Шаги для интеграции ChatGPT в Unity:

1. Необходимо зарегистрироваться на платформе OpenAI.
2. Оплатить подписку.
3. Получить API-ключ;
4. В Unity можно написать скрипт, который отправляет текстовый запрос к OpenAI API и получает ответ. Можно использовать HTTP запросы или любую подходящую библиотеку для работы с сетью;
5. Создать интерфейс, позволяющий игрокам вводить текст, который будет отправлен на сервер OpenAI для обработки;
6. Получить ответ от OpenAI API и отобразить его в игре. Можно сделать это, например, с помощью текстовых полей или областей текста;
7. Обязательно обработать случаи ошибок, например, отсутствия интернет-соединения или неправильного ответа от API.
8. Важно убедиться, что соблюдаются правила безопасности и конфиденциальности при работе с API.

Третий вариант - можно заранее сгенерировать большое количество пресетов, диалогов и другого и сохранить их в базе данных или текстовом файле. Затем во время игры в случайном порядке загружать сохраненные заготовки из этого списка. В большинстве случаев использование баз данных не требуется, можно просто использовать любые текстовые файлы поддерживаемые Unity.

Шаги для интеграции заранее сгенерированного текста в Unity:

1. Зарегистрироваться на сайте нейронной сети.
2. Оплатить подписку.
3. Хорошим подходом будет разделить NPC по категориям (крестьянин, рыцарь, король). Это поможет отсортировать задания по сложности и по награде и избавиться от нереалистичных ситуаций.

4. Сгенерировать необходимый объем текста для каждой категории NPC.
5. Перенести сгенерированный текст каждой категории NPC в отдельный файл.
6. В Unity создать массив элементов для каждой категории NPC, который будет загружать соответствующий файл и получать из него данные.
7. Создать скрипт, который будет случайным образом собирать задание по частям (описание задания, награда). Таким образом, один и тот же NPC сможет создать для каждого игрока уникальный игровой опыт.
8. Создать интерфейс для вывода получившихся заданий на экран.

Четвертый вариант - использовать языковую модель нейронной сети для генерации идей, которые в дальнейшем будут отредактированы сценаристом. Такой подход, в первую очередь, сокращает время на создание задания. А также может служить источником необычных идей.

Каждый из этих подходов имеет свои преимущества и недостатки, и выбор между ними будет зависеть от конкретных требований и ограничений.

Можем выделить следующие плюсы и минусы подходов, представленных в статье.

Разработка собственной языковой модели:

Плюсы:

1. Не требуется платить деньги за использование.
2. Модель можно адаптировать под конкретные задачи и требования.
3. Разработчик может контролировать работу модели и улучшать ее.

Минусы:

1. Требует много времени на создание и обучение
2. Требует значительных вычислительных ресурсов
3. Требует специализированных знаний в области машинного обучения и искусственного интеллекта.

API для генерации текста:

Плюсы:

1. Большая часть вычислений выполняется на сервере API, а не на устройстве пользователя, что снижает требовательность игры к ресурсам компьютера.
2. Поставщики API обычно регулярно обновляют и улучшают свои модели, поэтому пользователи автоматически получают все эти улучшения.
3. Не требуются знания о машинном обучении или искусственном интеллекте, чтобы использовать API.

Минусы:

1. Использование API обычно стоит денег, особенно если делать много запросов.
2. Для использования API требуется подключение к интернету.
3. Отправка данных на сервер API может вызвать проблемы с конфиденциальностью.

Предварительно сгенерированные задания:

Плюсы:

1. Простота реализации
2. Контроль над содержанием. Возможность контролировать, какие пресеты будут в игре.

Минусы:

1. Ограниченное количество заданий. Количество пресетов ограничено тем, сколько их было сгенерировано заранее. Это может негативно повлиять на разнообразие заданий.

Использование нейронных сетей для генерации идей:

Плюсы:

1. Качественная проработка каждого задания.
2. Возможность получить нестандартную, интересную идею.

Минусы:

1. Затраты времени на редактирование и доработку.

Выводы

После выделения четырех способов применения языковой модели нейронной сети в разработке видеоигр и их анализа можно сделать несколько выводов. Каждый из подходов применим на практике, все зависит от требований и ресурсов команды разработчиков.

Первый подход с созданием своей языковой модели нейронной сети больше подходит для игровых студий, располагающих финансами, временем и специализированными знаниями. Направлен, в первую очередь, на разработку ААА-игр. В этом случае возможно более узкое обучение модели для выполнения конкретных задач, а также доступен полный контроль над моделью и обучающим материалом.

Второй подход с интеграцией по API хорошо подойдет игровым студиям и инди-разработчикам, но из-за своих особенностей его лучше использовать для небольших проектов.

Третий подход с заранее сгенерированным текстом самый простой подход из представленных в статье, а также самый быстрый в реализации. В первую очередь, применим инди-разработчиками.

Четвертый подход, заключающийся в использовании нейронных сетей для генерации идей, может использоваться как игровыми студиями, так и инди-разработчиками. Он позволяет создать качественно проработанный текст.

Список литературы:

1. Прошина М. В. Современные методы обработки естественного языка: нейронные сети //Экономика строительства. – 2022. – №. 5. – С. 27-42.
2. Звайгзне А. Ю. Развертывание и настройка собственной языковой модели нейронной сети на локальной машине //Постулат. – 2024. – №. 1 январь.
3. Гузненков А. В., Манукян Э. С. Использование нейронных сетей в разработке игрового искусственного интеллекта //ББК 1 Н 34. – С. 4951.
4. Новак Д. О., Попова Е. С., Василькова А. Н. Искусственный интеллект как инструмент создания компьютерных игр. – 2023.
5. Фролов И. И., Богдан Е. В. Игровой искусственный интеллект в контексте больших данных. – 2024.
6. Сухоярский Д. И., Маевский И. А., Савельев И. М. ИССЛЕДОВАНИЕ ЯЗЫКОВЫХ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ГЕНЕРАЦИИ СЦЕНАРИЕВ ИГРОВОГО КВЕСТА //Оригинальные исследования. – 2023. – №12.

References:

1. Proshina M.V. Modern methods of natural language processing: neural networks // Construction Economics. – 2022. – No. 5. – pp. 27-42.
2. Zvaigzne A. Yu. Deployment and configuration of your own language model of a neural network on a local machine // Postulate. – 2024. – No. January 1st.

3. Guznenkov A.V., Manukyan E.S. The use of neural networks in the development of gaming artificial intelligence //BBK 1 N 34. – P. 4951.
4. Novak D. O., Popova E. S., Vasilkova A. N. Artificial intelligence as a tool for creating computer games. – 2023.
5. Frolov I. I., Bogdan E. V. Game artificial intelligence in the context of big data. – 2024.
6. Sukhoyarsky D. I., Mayevsky I. A., Savelyev I. M. RESEARCH OF LANGUAGE MODELS OF NEURAL NETWORKS FOR GENERATING GAME QUEST SCENARIOS //Original Research. – 2023. – No. 12.