Процесс обучения разработанной модели ИИ включает несколько этапов. Файл tokenizer.py выполняет токенизацию базы данных, создавая словарь vocab.json, где каждому слову или знаку присваивается уникальный номер. В файле train_model.py загружается этот словарь, задаются параметры обучения (размер входного вектора, количество нейронов, скорость обучения, число эпох), после чего инициализируется рекуррентная нейронная сеть. Затем текстовые данные преобразуются в токены, конвертируются в тензоры, аналогично обрабатываются метки классов. После инициализации модели, функции потерь и оптимизатора запускается процесс обучения, а по его завершении модель сохраняется. Файл main.py отвечает за запуск обученной модели, содержащий функции для преобразования текста в токены и обратно, а также реализацию улучшенной версии RNN с её инициализацией [3].

В ходе исследования была разработана и обучена рекуррентная нейронная сеть для обработки текстовых данных. Были освоены ключевые аспекты машинного обучения, включая работу с токенизированными последовательностями, построение архитектуры модели, её обучение и сохранение. Полученная модель демонстрирует базовую способность к анализу текстовой информации. В дальнейшем её можно улучшить, используя более сложные архитектуры и увеличивая объем обучающих данных, что позволит повысить точность классификации.

Список использованной литературы

1. Харбанс, Р. Грокаем Алгоритмы искусственного интеллекта / Р. Харбанс. – СПб. : Питер, 2023. – 368 с.

2. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвиль. – М. : ДМК Пресс, 2018. – 652 с.

3. Chollet, F. Deep Learning with Python / F. Chollet. – Shelter Island, NY : Manning Publications, 2021. – 384 p.

РЕАЛИЗАЦИЯ ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ «ГОЛОВОЛОМКА СУДОКУ»

Пилипейко Александр (УО МГПУ им. И.П. Шамякина, г. Мозырь)
Научный руководитель – А.А. Голуб, канд. физ.-мат. наук, доцент

В современных условиях программные решения для настольных и мобильных платформ должны обладать удобным интерфейсом, высокой производительностью и кроссплатформенностью. Разработка интерактивных приложений требует применения современных методик программирования.

Судоку – это логическая головоломка, состоящая из сетки размером 9×9, разделенной на меньшие блоки 3×3. Цель игры – заполнить пустые клетки цифрами от 1 до 9 таким образом, чтобы в каждой строке, каждом столбце и каждом блоке 3×3 все числа были уникальными. Головоломка популярна благодаря своей простоте в освоении и глубине логических

решений, что делает ее востребованной среди любителей интеллектуальных игр.

Основной целью разработки являлось создание интерактивного приложения «Судоку», отвечающего следующим требованиям: отображение игрового поля (9×9) с возможностью ввода данных пользователем, генерацию случайных головоломок с корректными решениями, проверку правильности введенных пользователем значений и реализацию графического интерфейса с поддержкой событий ввода [1].

Приложение реализовано на C++ в среде Visual Studio для платформы Windows с использованием библиотеки SDL и объектно-ориентированного подхода. Основными элементами являются: графический интерфейс с кнопками для начала новой игры и проверки ответа, где введённые пользователем числа динамически отображаются на поле игры; игровое поле, отвечающее за «отрисовку» окна, обработку ввода и проверку решения; генератор игры, создающий корректное заполненное поле Судоку и формирующий головоломку путём удаления случайных чисел; а также решатель Судоку, использующий метод возврата для проверки корректности решения [2].

Основные методологические подходы, использованные при разработке, включают алгоритм генерации головоломки, при котором создается полностью заполненная сетка Судоку, после чего удаляются случайные числа с контролем на единственность решения; проверку ввода пользователя, где каждое введенное значение проверяется на соответствие правилам игры (отсутствие повторений в строках, столбцах и блоках 3×3); обработку пользовательского ввода с возможностью управления игровым процессом [3].

Пример готового рабочего окна приложения продемонстрируем на рисунке 1:

Рисунок 1 – Рабочее окно приложения

Таким образом, разработанное приложение представляет собой полноценную игровую реализацию Судоку с возможностью генерации новых задач и проверки решений. Дальнейшее развитие проекта может включать добавление уровней сложности с разным количеством заполненных ячеек, улучшение графического интерфейса с анимацией ввода, поддержку сохранения и загрузки игр, а также перенос на мобильные устройства. В результате получилось удобное и производительное приложение для игры в Судоку, в котором использованные методы генерации и проверки головоломок обеспечивают корректную работу программы, а графический интерфейс делает взаимодействие с пользователем интуитивным и удобным.

Список использованной литературы

1. Metanit [Electronic resource] // История создания Судоку – Mode of access: https://metanit.com/cpp/tutorial/1.1.php – Date of access: 26.03.2025.

2. Habr [Electronic resource] // Разработка логических игр – Mode of access: https://habr.com/ru/articles/454396/ – Date of access: 26.03.2025.

3. Geeksforgeeks [Electronic resource] // Алгоритмы решения Судоку – Mode of access: https://www.geeksforgeeks.org/sudoku-backtracking/ – Date of access: 26.03.2025.

ИСПОЛЬЗОВАНИЕ КЛЕТОЧНЫХ АВТОМАТОВ ДЛЯ СОЗДАНИЯ РАЗЛИЧНЫХ ФИЗИЧЕСКИХ МОДЕЛЕЙ НА ОСНОВЕ ИГРЫ «ЖИЗНЬ»

Пилипейко Александр (УО МГПУ им. И.П. Шамякина, г. Мозырь)
Научный руководитель – А.П. Сафронов, старший преподаватель

В данной работе будет показано конечное состояние «живых» клеток и будут разобраны некоторые нюансы, связанные с моделью, такие как правила игры и конечные состояния клеток.

Для начала нужно определить, что представляют собой клеточные автоматы, правила игры «Жизнь».

Клеточные автоматы представляют собой дискретные пространственно-временные системы, состоящие из ячеек, которые могут находиться в различных состояниях. Состояние каждой ячейки в следующий момент времени определяется ее собственным текущим состоянием и состояниями соседних ячеек в соответствии с заданными правилами перехода [1].

Правила игры «Жизнь» включают в себя условия жизни и смерти клеток в зависимости от количества соседей.

Игра моделирует жизнь простых организмов на двумерной решетке. Каждая ячейка решетки может находиться в одном из двух состояний: «живая» или «мертвая». Состояние ячейки в следующий момент времени определяется ее собственным текущим состоянием и состояниями восьми соседних ячеек в соответствии с простыми правилами: рождение – мертвая ячейка становится живой, если ровно три ее соседа живы; выживание – живая ячейка остается живой, если у нее ровно два или три живых соседа;