эвтектического сплава Sn-8,8 мас. % Zn алюминием до 1 мас. % приводит к увеличению параметра ячейки c, что говорит об образовании пересыщенного твёрдого раствора замещения на основе цинка [4].

В заключение можно сделать вывод, что быстрозатвердевшие сплавы Sn-Zn без добавок демонстрируют двухфазную структуру, состоящую из твёрдых растворов олова и цинка. Введение *Al* и *Sb* ослабляет интенсивность дифракционных линий исходных компонентов (*Zn* и *Sn*), что свидетельствует о формировании пересыщенных твёрдых растворов и интерметаллических соединений. Результаты исследования открывают пути для разработки экологичных бессвинцовых припоев на основе системы *Sn-Zn*. Однако для практического применения необходимы дополнительные исследования механических свойств, коррозионной стойкости и смачиваемости модифицированных сплавов. Особое внимание следует уделить балансу между содержанием *Al* и *Sb* для достижения оптимальных эксплуатационных характеристик.

Подготовлено при финансовой поддержке БРФФИ (грант №Г24МП-054).

Список использованной литературы

1. Jae-Ean Lee1, Keun-Soo Kim, Katsuaki Suganuma, Junichi Takenaka, Koichi Hagio Interfacial Properties of Zn–Sn Alloys as High Temperature Lead-Free Solder on Cu Substrate // Materials Transactions, Vol. 46, No. 11 (2005) pp. 2413-2418.

2. Zernitsa, D. A. Study of the Structure and Properties of Rapidly Solidified Tin–Zinc Eutectic Alloys Doped with Antimony / D. A. Zernitsa, V. G. Shepelevich // Inorganic Materials : Applied Research. – 2023. – Vol. 14, № 1. – P. 86–95.

3. Шепелевич, В.Г. Быстрозатвердевшие легкоплавкие сплавы / В.Г. Шепелевич. – Минск : БГУ, 2015. – 192 с.

4. Шепелевич, В.Г. Структура быстрозатвердевшей фольги эвтектического сплава Sn – 8,8 мас. % Zn / В.Г. Шепелевич, Д.А. Зерница // Журнал Белорусского государственного университета. Физика, 1. С. 67–72.

РАЗРАБОТКА СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ УПРАВЛЕНИЯ ПРОЦЕССАМИ ПОИСКА И ОБРАБОТКИ ИНФОРМАЦИИ

Мухин Сергей (УО МГПУ им. И.П. Шамякина, г. Мозырь)
Научный руководитель – А.В. Макаревич, канд. физ.-мат. наук, доцент

Искусственный интеллект (ИИ) является одной из наиболее быстро развивающихся сфер науки и технологий, а создание систем на его основе играет важную роль в автоматизации процессов поиска, анализа и обработки информации. Такие решения широко применяются в различных областях, включая управление данными, обработку больших массивов информации, прогнозирование и поддержку принятия решений, что делает их незаменимыми в условиях современного цифрового общества [1].

Одним из простейших примеров ИИ-систем для анализа данных является алгоритм, использующий машинное обучение для классификации текстовой информации по определенным категориям. При этом одним из наиболее действенных инструментов в этой области считается рекуррентная нейронная сеть (RNN) – разновидность искусственных нейросетей, способных работать с последовательностями данных и учитывать контекст на разных временных шагах [2].

Рекуррентная нейронная сеть состоит из повторяющихся модулей, каждый из которых сохраняет информацию о предыдущих состояниях, что позволяет эффективно анализировать последовательности данных. В отличие от обычных полносвязных сетей, RNN способна учитывать временные зависимости, что делает её полезной для обработки естественного языка, анализа временных рядов и прогнозирования.

Рекуррентная нейронная сеть включает следующие компоненты:

1) входной слой (Input Layer), принимает входные данные (например, текстовые последовательности или числовые ряды);

2) скрытые слои (Hidden Layer), содержат рекуррентные нейроны, которые сохраняют состояние предыдущих шагов и обрабатывают последовательность данных;

3) выходной слой (Output Layer), формирует итоговый результат (например, следующий токен в тексте или предсказанное значение).

В выполненной автором настоящей статьи разработке для реализации и обучения RNN была использована библиотека PyTorch. Обучение сети включало несколько ключевых этапов: загрузку словаря токенов, инициализацию архитектуры модели, нормализацию данных, настройку функции потерь и оптимизатора, а также сохранение обученной модели. Эпохи обучения представляют собой циклы, в ходе которых модель последовательно обрабатывает все обучающие данные, обновляя параметры на основе градиентного спуска для минимизации ошибки эпохи обучения (рисунок 1).

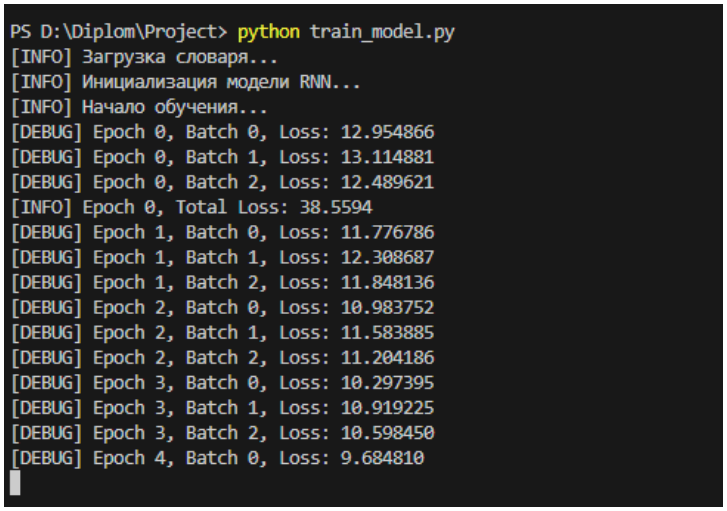

Рисунок 1 – Эпохи обучения RNN в разработанной модели ИИ

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