УДК 004.056, 004.3

ПОДХОД К АППАРАТНОЙ ПОДДЕРЖКЕ ДАННЫХ ТЕКСТА С ИСПОЛЬЗОВАНИЕМ ПЛИС

В.А. Вишняков
Доктор технических наук, профессор кафедры инфокоммуникационных технологий БГУИР
vish@bsuir.by

Ч. Юй
Аспирантка кафедры инфокоммуникационных технологий БГУИР
1259017720@qq.com

В.А. Вишняков
Область профессиональных интересов/исследований: управление и безопасность в инфокоммуникациях, Интернет вещей, блокчейн, электронный маркетинг, ИТ в образовании и медицине, интеллектуальные системы управления.

Ч. Юй
Аспирантка кафедры инфокоммуникационных технологий Белорусского государственного университета информатики и радиоэлектроники.

Аннотация. Цель данного исследования – разработка аппаратно-ориентированного метода ускорения классификации текста на основе метода случайного леса в условиях ограниченных вычислительных ресурсов. Оптимизирована модель, реализована облегченная архитектура, с использованием арифметики с фиксированной запятой и вычислительного модуля на ПЛИС. Создана сквозная конвейерная обработка от исходного текстового ввода до извлечения признаков методом TF-IDF и классификации с использованием случайного леса. Предложенный подход продемонстрировал числовую согласованность с эталонной реализацией на языке Python. Хотя обработка TF-IDF выполняется на центральном процессоре, её взаимодействие с модулем ПЛИС осуществляется через шину AXI-Lite, что закладывает основу для полной аппаратной интеграции в дальнейших исследованиях.

Ключевые слова: ПЛИС, случайный лес, анализ текста, TF-IDF, аппаратное ускорение.

Введение. Технологии анализа текста играют важную роль в различных областях, таких как медицина, мониторинг общественного мнения и обслуживание клиентов [1-2]. В этих сценариях применения зачастую требуется эффективная обработка больших объемов текстовых данных, извлечение ключевой информации и выполнение классификации с высокой точностью. Традиционные задачи обработки естественного языка (NLP) обычно выполняются с использованием вычислений на CPU или GPU, что часто оказывается вычислительно затратным и энергоемким процессом. В сценариях граничных вычислений, где требуется низкая задержка и высокая энергоэффективность, такие среды зачастую обладают ограниченными вычислительными ресурсами [3], что затрудняет выполнение задач в режиме реального времени или при ограниченном энергопотреблении.

Для решения этих проблем могут быть применены несколько стратегий: сжатие моделей, использование легковесных алгоритмов, совместное использование ресурсов на границе и в облаке, а также применение специализированных аппаратных ускорителей, таких как GPU и ПЛИС, для повышения эффективности обработки. Методы интеллектуального анализа текста на основе ПЛИС (программируемые логические

интегральные схемы) стали актуальной областью исследований, поскольку предоставляют высокоэффективные аппаратные ускорители. Однако при применении аппаратного ускорения на базе ПЛИС в задачах NLP необходимо учитывать ключевые проблемы, такие как эффективная обработка крупномасштабных словарей, предобработка текста и интеграция традиционных программных компонентов с аппаратными модулями.

В области машинного обучения сверточные нейронные сети (CNN) [4] и модели глубокого обучения, такие как BERT [5], уже были развернуты на ПЛИС и показали улучшение пропускной способности и энергоэффективности по сравнению с чисто CPU-ориентированными решениями. Хотя матричные операции SVM и BERT естественным образом подходят для параллельных вычислений на ПЛИС, структура случайного леса (Random Forest) характеризуется нерегулярными ветвлениями и разрывами в памяти, что требует индивидуальной оптимизации.

Кроме того, в типичном процессе обработки NLP извлечение признаков TF-IDF и связанные этапы (например, фильтрация стоп-слов и токенизация) по-прежнему выполняются на CPU. Это связано с тем, что задачи динамического выделения памяти и работы со строками затрудняют их непосредственное аппаратное ускорение с помощью инструментов высокоуровневого синтеза (HLS). Разделение между CPU-ориентированным извлечением признаков NLP и классификацией на ПЛИС не только влечет за собой накладные расходы на интерфейс (например, передачу векторных признаков), но и создает основу для гибридных решений на базе CPU-ПЛИС.

Данное исследование направлено на разработку архитектуры аппаратного ускорения классификации текста на основе случайного леса (Random Forest) с применением ПЛИС, что позволит обеспечить высокую производительность и низкую задержку интеллектуального анализа текста в условиях ограниченных вычислительных ресурсов.

Методологическая основа. Для обеспечения эффективного развертывания на ПЛИС рабочий процесс разделен на несколько этапов. Сначала алгоритм был оптимизирован и адаптирован для аппаратной реализации, чтобы гарантировать, что размер модели и структуры данных соответствуют ресурсам и характеристикам ПЛИС. Затем основные алгоритмические модули были реализованы на языке C++ с фиксированной запятой для достижения более высокой производительности и эффективности использования аппаратных ресурсов. Затем выполнена аппаратная симуляции с использованием инструментов HLS для верификации функциональности и предварительных показателей производительности. Алгоритмическая реализация извлечения текстовых признаков и использование случайного леса соответствует подходу, описанному в авторской разработке [6].

Архитектура системы гибридного конвейера CPU-ПЛИС представлена на рисунке 1. Модуль на стороне CPU включает в себя подмодули для предварительной обработки текста, извлечения признаков TF-IDF и снижения размерности методом SVD, а также преобразования и сохранения данных. Подмодуль предварительной обработки текста выполняет токенизацию входного тестового текста, очистку символов, приведение к нижнему регистру и фильтрацию стоп-слов. Модуль извлечения признаков TF-IDF и снижения размерности SVD преобразует обработанный текст в 108-мерный вектор признаков с фиксированной точкой. Модуль преобразования и сохранения данных одновременно сохраняет вектор признаков в двоичном и текстовом форматах для сравнения с реализацией на Python, а также передает извлеченный вектор признаков в модуль ПЛИС.

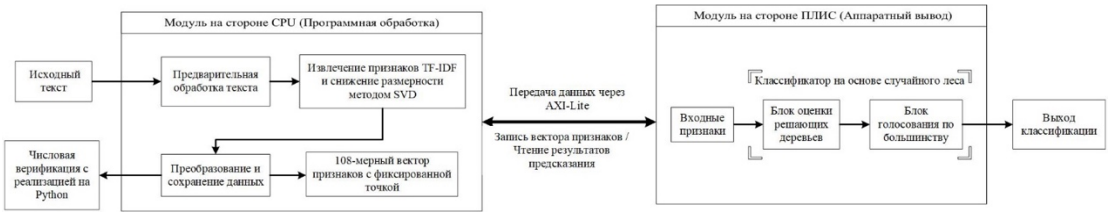

Рисунок 1. Схема архитектуры системы

Интерфейс AXI-Lite обеспечивает передачу 108-мерного вектора признаков, извлеченного на стороне CPU, в модуль классификатора случайного леса на ПЛИС, а вычисленное на аппаратной стороне значение «result» отображается в доступный для чтения и записи регистр AXI-Lite, позволяя пользователю считывать его со стороны CPU. Модуль на стороне ПЛИС включает классификатор на основе случайного леса, состоящий из блока оценки, содержащего решающие деревья, в которых каждый узел хранится с использованием структуры TreeNode. Выходные данные нескольких деревьев обрабатываются с помощью механизма голосования по большинству для получения финального предсказания. После завершения обработки на аппаратной стороне данные передаются из этого модуля на следующий этап системы или могут быть считаны извне.

Для проверки проведены функциональные и производительные тесты сквозного (Python-C++, ПЛИС) процесса классификации текста для обеспечения согласованности результатов на разных платформах. Предварительная обработка текста, извлечение признаков TF-IDF и снижение размерности SVD были выполнены с использованием Python для генерации векторов признаков, совместимых с модулем C++. Соответствующие параметры были экспортированы, а выходные данные реализации на C++ сравнивались с эталонной реализацией на Python. Результаты тестирования показали, что косинусное сходство между векторами признаков, сгенерированными из одного и того же тестового текста на разных платформах, составило 0,9990, что свидетельствует о высокой степени направленного совпадения выходных векторов C++ и Python, подтверждая корректность вычислений TF-IDF и SVD.

Экспериментальная верификация. В модуле ПЛИС классификатор случайного леса был реализован с использованием инструментов HLS, при этом аппаратная реализация деревьев решений через распределённое хранение узлов в LUTRAM с конвейерной обработкой на фиксированной точке. Согласно отчету о синтезе, общая задержка функции инференса случайного леса составила примерно 11 193 такта, что соответствует времени выполнения предсказания 0,112 миллисекунды при тактовой частоте 100 МГц. В отношении использования ресурсов ПЛИС-реализация составила 5 % – 30 % от общего объема аппаратных ресурсов, при этом использование LUT, FF и DSP соответствовало заданным проектным целям и не превышало ограничений ресурсоемкости целевого устройства Xilinx XCVU11P. Дальнейшая оптимизация может дополнительно повысить вычислительный параллелизм и увеличить пропускную способность.

Заключение. Авторы, в рамках гибридной архитектуры CPU + ПЛИС, выполнили извлечение признаков TF-IDF + SVD на стороне C++ (или предварительную обработку на стороне Python), затем передали полученный 108-мерный вектор признаков с фиксированной точкой в IP-ядро случайного леса на ПЛИС и подтвердили числовую согласованность с эталонной реализацией на Python, достигнув косинусного сходства 0,9990, а также согласованности предсказанных результатов. Задержка аппаратного инференса и использование ресурсов указывают на то, что данный метод обладает потенциалом масштабируемости на крупномасштабных ПЛИС-устройствах.

Список литературы

[1] Janowski A. Natural Language Processing Techniques for Clinical Text Analysis in Healthcare. Journal of Advanced Analytics in Healthcare Management. 2023; 7(1):51–76.

[2] Hu Y. Text Mining and Data Information Analysis for Network Public Opinion. Data Science Journal. 2019; 18(1):7.

[3] Wisniewski M., Bec J.-M., Boguszewski G., Gamatié A. Hardware Solutions for Low-Power Smart Edge Computing. Journal of Low Power Electronics and Applications. 2022; 12(4):61.

[4] Mouri Zadeh Khaki A., Choi A. Optimizing Deep Learning Acceleration on FPGA for Real-Time and Resource-Efficient Image Classification. *Applied Sciences.* 2025; 15(1):422.

[5] Khan H., Khan A., Khan Z., Huang L.B., Wang K., He L. NPE: An FPGA-based Overlay Processor for Natural Language Processing. *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '21).* Association for Computing Machinery, New York, NY, USA, 2021; 227.

[6] Вишняков В.А., Юй Ч. Использование машинного обучения для распознавания болезни Альцгеймера на основе транскрипционной информации. *Доклады БГУИР.* 2023; 21(6):106–112.

Авторский вклад

Вишняков Владимир Анатольевич – предложение концепции метода.

Юй Чуюэ – реализация предложенного метода.

APPROACH TO HARDWARE-ASSISTED TEXT ANALYSIS USING FPGA

V.A. Vishnyakov
Doctor of Technical Sciences, Professor of the Department of Infocommunication Technologies, BSUIR

C. Yu
PhD student in the Department of infocommunication technologies at BSUIR

Abstract. The purpose of this study is to develop a hardware-oriented method for accelerating text classification based on a random forest in conditions of limited computing resources. The model has been optimized, a lightweight architecture has been implemented, using fixed-point arithmetic and FPGA computing modules. End-to-end pipelining has been created from the source text input to feature extraction using the TF-IDF method and classification using a random forest. The proposed approach demonstrated numerical consistency with the reference implementation in Python. Although TF-IDF processing is performed on the central processor, its interaction with the FPGA module is carried out via the AXI-Lite bus, which lays the foundation for full hardware integration in further research.

Keywords: FPGA, Random Forest, Text Analytics, TF-IDF, Hardware Acceleration.