УДК 004.896

ЗАПУСК МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ НА ARM-МИКРОКОНТРОЛЛЕРАХ В РАМКАХ ПАРАДИГМЫ TINYML

Акимов С.В. – студент,
Калужский филиал московского государственного
технического университета имени Н.Э. Баумана,
Россия, г. Калуга

В современную эпоху нейронные сети стали неотъемлемой частью повседневной жизни человека. С их помощью сегодня можно выполнять широкий спектр задач - от простого поиска информации в сети до создания полноценных рабочих продуктов. Наибольшую популярность приобрели крупные языковые модели (Large Language Models, LLM), такие как ChatGPT, DeepSeek и другие. Эти нейросети способны решать широкий круг задач, однако их функционирование требует значительных вычислительных ресурсов и мощных серверов.

Тем не менее, во многих случаях перед исследователями и разработчиками стоит узкоспециализированная задача, для решения которой нет необходимости использовать универсальные модели. В таких ситуациях возможно обучение специализированной нейронной сети, предназначенной для выполнения одной конкретной функции. Это позволяет существенно снизить требования к производительности и объёму вычислительных ресурсов.

Данный подход особенно актуален в условиях широкого распространения маломощных вычислительных устройств, которые окружают нас повсюду. К ним относятся многочисленные устройства Интернета вещей (Internet of Things, IoT). Подобные системы часто основаны на микроконтроллерах архитектуры ARM. Несмотря на их ограниченные вычислительные возможности, использование технологий машинного обучения на таких платформах открывает перспективные направления развития и расширяет возможности внедрения искусственного интеллекта в повседневную среду.

Микропроцессоры и микроконтроллеры архитектуры ARM составляют основу современных встраиваемых систем и устройств интернета вещей (IoT). С точки зрения запуска моделей машинного обучения ключевую роль играют два основных семейства процессорных ядер - Cortex-M и Cortex-A.

Cortex-M: Это семейство ориентировано на микроконтроллеры (MCU). Их главная цель - максимальное снижение энергопотребления и стоимости, что критически важно для автономных IoT-устройств и embedded-систем. Такие процессоры, как правило, не оснащены блоком управления памятью (MMU), что не позволяет запускать полноценные операционные системы, такие как Linux. Вместо них используются операционные системы реального времени

(RTOS), например, Zephyr RTOS или FreeRTOS. Примеры: популярные серии микроконтроллеров STM32[1].

Cortex-A: Это семейство представляет собой микропроцессоры (MPU), которые можно рассматривать как "мини-ПК". Они обладают высокой производительностью, оснащены MMU и способны запускать "тяжелые" ОС, включая различные дистрибутивы Linux. Благодаря этому они подходят для устройств с сложным пользовательским интерфейсом или требующих подключения к облачным сервисам. На таких системах могут быть запущены полноценные десктопные приложения. Пример: одноплатные компьютеры Raspberry Pi [2].

Для начала разберем основные направления использования AI в области встраиваемых систем [3]:

1. Компьютерное зрение - включает задачи классификации изображений и детекции объектов. Находит применение в системах видеонаблюдения, автоматизированного контроля качества промышленного оборудования и биометрических системах контроля доступа.

2. Обработка аудиосигналов - охватывает распознавание речевых команд, идентификацию ключевых слов (keyword spotting) и классификацию акустических событий, таких как звук выстрела или разрушения стекла.

3. Генеративное моделирование - применяется для синтеза текстового контента и других видов медиаданных непосредственно на устройстве.

4. Обработка естественного языка (Natural Language Processing) - используется для классификации намерений пользователя и анализа эмоциональной окраски речи (sentiment analysis).

А теперь рассмотрим программные решения, предоставляемые ARM, которые помогают перейти от написания нейросетей для стандартных мощностей ПК к возможностям TinyML. На платформе с готовыми моделями Hugging Face есть страница ARM, на которой представлены уже обученные и оптимизированные для запуска в условиях TinyML модели. У моделей указан рекомендованный способ их развертки: напрямую на системе или в облаке, только отправляю промты на модель. Модели решают частые задачи микроконтроллеров. Например, там есть нейросеть, распознающая человеческую речь на разных языках[4], с приведённой картой точности(рисунок 1)

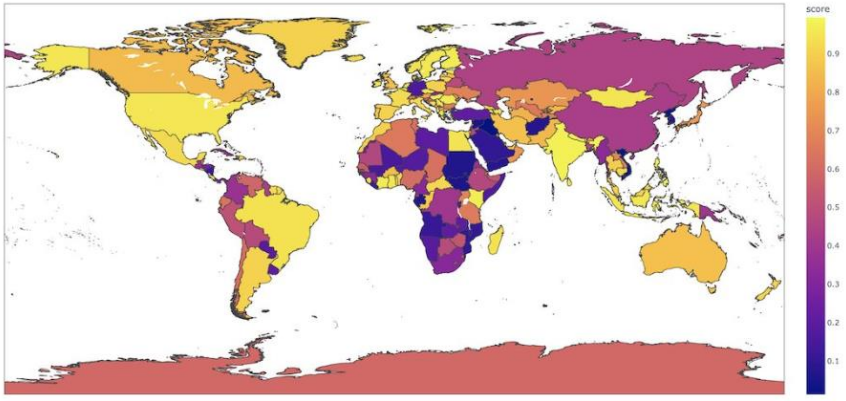

Рисунок 1 - Карта точности модели распознавания речи

Для самой развертки нейронной сети используется STM32Cube.AI - специализированное программное расширение, предназначенное для конвертациии уже обученных нейросетей в оптимизированный код для микроконтроллеров STM32. STM32Cube.AI является мостом между областью машинного обучения и микроконтроллером.

Ключевые функциональные возможности [5]:

1. Конвертация моделей: Поддерживает импорт моделей из популярных фреймворков машинного обучения, таких как TensorFlow, Keras, PyTorch и Scikit-learn. Пользователь загружает обученную модель, STM32Cube.AI преобразует её в набор высокооптимизированных библиотек на языке C.

2. Статическое распределение памяти: В отличие от динамического выделения памяти в классических ML-фреймворках, STM32Cube.AI рассчитывает весь объем памяти, необходимый для работы сети (для входных, промежуточных и выходных тензоров), на этапе компиляции. Это полностью исключает использование кучи (heap), что критически важно для детерминированного поведения и надежности в системах реального времени.

3. Аппаратная оптимизация: Генерируемый код использует инструкции SIMD (Single Instruction, Multiple Data) и DSP-расширения ядер Cortex-M, а также аппаратное ускорение CRC для проверки целостности модели. Для поддержки будущих микроконтроллеров STM32 с NPU (например, серии STM32N6) заложена совместимость с ускорителями типа Arm Ethos-U55.

4. Валидация и бенчмаркинг: Инструмент предоставляет средства для проверки точности конвертированной модели на целевом устройстве, сравнивая её вывод с эталонной реализацией на ПК. Также он генерирует отчеты с ключевыми метриками производительности: объем потребляемой памяти (RAM и Flash) и количество процессорных циклов, необходимых для инференса.

На рисунке 2 представлено сравнение оптимизации модели посредствам STM32CubeAI и другого популярного оптимизационного фреймворка TensorFlow Lite for microcontrollers:

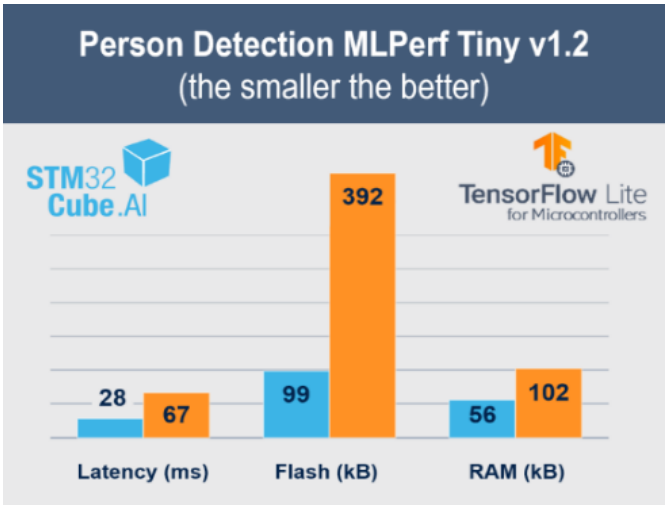

Рисунок 2 – диаграмма сравнения оптимизации

Для практической реализации проекта с использованием нейронной сети нам понадобиться отладочная плата с микроконтроллером или микропроцессором ARM. Из вариантов на Cortex M лучше всего подойдут Cortex M4 и M7. Примером таких плат могут послужить NUCLEO-F446RE на Cortex M4[5] и NUCLEO-F767ZI на Cortex M7. Из вариантов на Cortex A лучше всего рассмотреть Raspberry Pi 5. Все эти платы поддерживают вышеперечисленные программные инструменты и широко представлены в российских магазинах, что делает их лучшим выбором для создания проекта с AI[6].

В заключении хотелось бы добавить, что развертывание моделей машинного обучения на ARM-микроконтроллерах является технически осуществимой и стратегически важной задачей в контексте развития парадигмы TinyML. Успех в этой области основывается на комплексном подходе, который включает создание специализированных моделей, мощную инструментальную поддержку и грамотную оптимизацию. Современный программно-аппаратный стек, представленный такими инструментами, как STM32Cube.AI, и поддержкой со стороны популярных фреймворков, создает полноценный пайплайн развертывания, делая технологии искусственного интеллекта доступными для самых ограниченных устройств.

Ключевым фактором становится применение агрессивной оптимизации, прежде всего за счет квантования и использования аппаратных возможностей ядер Cortex-M, а в перспективе - специализированных NPU. Это позволяет достичь баланса между производительностью, потреблением памяти и энергопотреблением.

Таким образом, TinyML преодолевает барьер между облачным интеллектом и автономными embedded-системами, позволяя реализовывать алгоритмы ИИ непосредственно в точке сбора данных. Это не только решает задачи минимальной задержки, конфиденциальности и энергоэффективности, но и служит фундаментом для нового поколения умных и автономных устройств Интернета вещей, интегрируя искусственный интеллект в саму ткань окружающей нас физической среды.

Список литературы:

1. Cortex-M55 | Efficient DSP & ML with Enhanced AI Features [Электронный ресурс] // Офиц. сайт Arm. – URL: https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m55 (дата обращения: 31.10.2025).

2. Cortex-A75 | High-Performance CPU for AI and ML Tasks [Электронный ресурс] // Офиц. сайт Arm. – URL: https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a75 (дата обращения: 31.10.2025).

3. Warden P., Situnayake D. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers [Электронный ресурс] / P. Warden, D. Situnayake. – O'Reilly Media, 2019. – URL: https://www.oreilly.com/library/view/tinyml/9781492052036/.

4. DistilBERT base uncased finetuned SST-2 [Электронный ресурс] // Hugging Face. – URL: https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english (дата обращения: 31.10.2025).

5. STM32Cube.AI - STMicroelectronics [Электронный ресурс] // Офиц. сайт. – URL: https://stm32ai.st.com/stm32-cube-ai/ (дата обращения: 31.10.2025).

6. Cortex-M4 | High-Performance, Low Cost for Signal Control [Электронный ресурс] // Офиц. сайт Arm. – URL: https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m4 (дата обращения: 31.10.2025).