Нейросетевые решения на ARM-микроконтроллерах

Дмитрий Швецов

В статье представлен обзор реализации нейронных сетей на недорогих микроконтроллерах с ядром ARM Cortex-M. Эти устройства подходят для граничных вычислений и могут быть использованы в различных приложениях, особенно в сфере видеоаналитики. Развитие аппаратных архитектур цифровых сигнальных процессоров (DSP – Digital Signal Processor) сделало внедрение таких систем реальным благодаря их преимуществам: низкой стоимости, энергопотреблению и малой задержке при выполнении задач в реальном времени. В статье рассматривается новое направление исследований – применение методов искусственного интеллекта в стандартных микроконтроллерах ARM Cortex-M.

В обзоре приведены данные о системах, которые показали высокую эффективность в приложениях с ограниченными ресурсами. Речь идёт о различных архитектурах глубоких нейронных сетей (DNN) и результатах их применения на базе микроконтроллеров ARM Cortex-M и недорогих аппаратных устройств и программных решений для разработки. В последние годы приложения, такие как распознавание лиц, речи, изображений или рукописного ввода, обработка естественного языка и автоматическая медицинская диагностика, продемонстрировали высокую эффективность благодаря глубокому обучению (DL). Если рассматривать структуру сложных глубоких нейронных сетей, то для них требуется ещё больше повысить производительность приложений и добавить дополнительные функции. Но это приводит ко всё более жёстким требованиям к вычислительным мощностям этих платформ. Чтобы их удовлетворить, производители интегральных схем увеличивают количество доступных ядер, рабочие частоты вычислительных ядер и систем памяти, а также разрабатывают специализированные аппаратные решения. В последнее время для выполнения алгоритмов с высокими вычислительными требованиями в условиях ограничений потребления энергии используют специализированные аппаратные ускорители вместе с системами памяти, которые обеспечивают высокую пропускную способность. Новый класс систем использует алгоритмы DNN для решения задач в области интеллектуальных встраиваемых устройств. Такие приложения проще для реализации, но должны иметь очень низкое энергопотребление, так как большинство этих систем работает продолжительное время от батарей. Некоторые из таких решений можно реализовать даже на недорогих микроконтроллерах с низким энергопотреблением, например, на базе ядра ARM Cortex-M.

Архитектура вычислителей, реализованных на базе ядра ARM Cortex-M, давно привлекает внимание разработчиков благодаря инструментам и поддержке встроенного программного обеспечения. Это позволяет сократить трудоёмкость и временны́е затраты на разработку. Недавний систематический обзор показал, что микроконтроллеры STM32 и серии ARM Cortex-M – лучшие аппаратные устройства для машинного обучения подобных систем.

При использовании микроконтроллеров на базе ядра ARM Cortex-M для граничных вычислений сравнивались следующие параметры:

- типичные встроенные архитектуры DNN;
- методы оптимизации (разметка, квантование и т.п.);
- ядро ARM Cortex-M (M3, M4, M7 и т.д.);
- результаты экспериментов с акцентом на заявленную точность, время вывода и энергопотребление.

Модели глубокого обучения обычно требуют высокой вычислительной мощности и значительного объёма доступной памяти, особенно современные модели (SOTA – State-Of-The-Art). Поэтому некоторые приложения на основе DL используют облачные сервисы крупных ИТ-компаний, популярные сервисы: для ускоренного обучения, развёртывания и управления проектами глубокого обучения. Однако у подхода с облачными вычислениями есть ряд недостатков. В классической парадигме облачных вычислений большое количество вычислительных задач выполняется в облаке. Перегрузка сети трафиком может вызвать задержки в некоторых сценариях реального времени. Вывод результатов работы нейросетей в реальном времени важен для чувствительных к задержкам приложений. В этом случае подход с облачными вычислениями может привести к значительным задержкам. Кроме того, есть опасения по поводу безопасности передачи данных в облако. По этим причинам новая тенденция состоит в том, чтобы выполнять вычисления моделей глубокого обучения непосредственно на устройстве, а не передавать данные на удалённые устройства с высокими вычислительными возможно-

стями. Этот подход очень часто называют пограничными вычислениями.

Для запуска модели DL на встраиваемых устройствах необходимо применить один или несколько алгоритмов сжатия, таких как квантование параметров модели, обрезка нейронной сети, дистилляция сети и бинаризация. Также существует подход, основанный на получении оптимизированной архитектуры, которая после обучения не требует использования методов сжатия. В этом направлении значительный вклад внесла архитектура SqueezeNet, цель которой получить небольшое количество параметров с минимальной потерей точности. Встраиваемые устройства подходят только для задач вывода, которые дешевле с точки зрения вычислительных ресурсов по сравнению с процессом машинного обучения. Среди этих устройств есть микроконтроллеры, эффективно используемые в различных областях, например, в приложениях Интернета вещей. За последнее десятилетие справляться с вычислительными ограничениями стало легче благодаря появлению на рынке специализированных аппаратных устройств. Эти устройства, используемые в контексте глубокого обучения, называются аппаратными ускорителями. С их помощью оптимизируют и специализируют аппаратные архитектуры, что позволяет снизить стоимость системы и энергопотребление, одновременно повышая производительность. Для реализации приложений глубокого обучения встроенная система должна обладать высокой вычислительной мощностью и способностью получать и обрабатывать данные в режиме реального времени. Процессор должен иметь достаточно памяти для хранения данных модели и параметров.

Устройства типа «система на кристалле» (SoC) могут быть привлекательным решением, включающим в себя множество периферийных устройств и высокие вычислительные возможности. Это делает SoC подходящим выбором для сложных требований приложений глубокого обучения.

Один крупный производитель интегральных схем искусственного интеллекта для передовых вычислений сравнительно недавно выпустил встроенный микроконтроллер серии 78000 со сверхнизким энергопотреблением, специально разработанный для приложений искусственного интеллекта и объединяющий ускоритель свёрточных

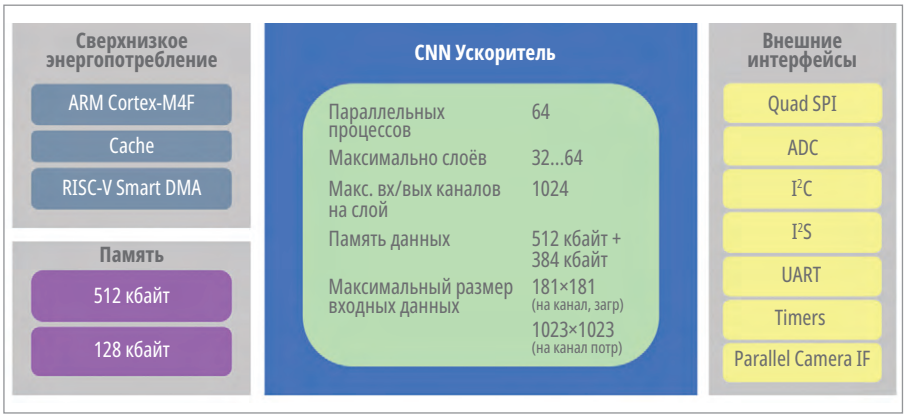

Рис. 1. Структура и основные характеристики ускорителя CNN ARM Cortex-M4 с FPU

нейронных сетей (CNN), маломощное ядро ARM Cortex-M4 и ядро RISC-V. Эта архитектура предоставляет множество вариантов конфигурации, что позволяет разрабатывать приложения искусственного интеллекта с низким энергопотреблением.

На рис. 1 кратко представлены основные характеристики ускорителя CNN и микроконтроллера: ядра, объём памяти и внешние интерфейсы. Микроконтроллер имеет двухъядерную архитектуру: процессор ARM Cortex-M4 с FPU (до 100 МГц) и 32-разрядный сопроцессор RISC-V (до 60 МГц).

Производительность устройства была продемонстрирована на примере двух приложений: поиск ключевых слов и распознавание лиц. Результаты по точности многообещающие: 99,6% для распознавания ключевых слов и 94,4% для распознавания лиц.

Ещё один крупный производитель электроники, который внёс важный вклад в развитие рынка передовых вычислений с искусственным интеллектом как новой парадигмы Интернета вещей, предлагает запускать нейронные сети на микроконтроллерах общего назначения STM32. Это значительно повлияет на продуктивность разработчиков пограничных систем искусственного интеллекта, сократив время развёртывания приложений. В данном случае основное внимание уделяется не аппаратным ускорителям, а обширному набору программных средств для переноса моделей DNN на стандартные микроконтроллеры STM32 с высокой эффективностью для процессорного ядра ARM Cortex-M4 и M7.

Аналогичное решение также разработано для автомобильных микроконтроллеров SPC5. Для этого используется плагин искусственного интеллекта под названием SPC5-STUDIO – AI среды разработки SPC5-STUDIO. Для приложений Интернета вещей нового поколения, которым требуется высокая вычислительная мощность (до гигабайт операций с памятью в секунду) и большой объём памяти (несколько мегабайт), была разработана параллельная архитектура SoC со сверхнизким энергопотреблением (PULP). Другие SoC также имеют решения для оптимизации и встраивания. Например, недавно был разработан 16-нм SoC со специальной оптимизацией для автоматического распознавания речи. Также есть платы серии TI TDAx. На рис. 2 представлено семейство процес-

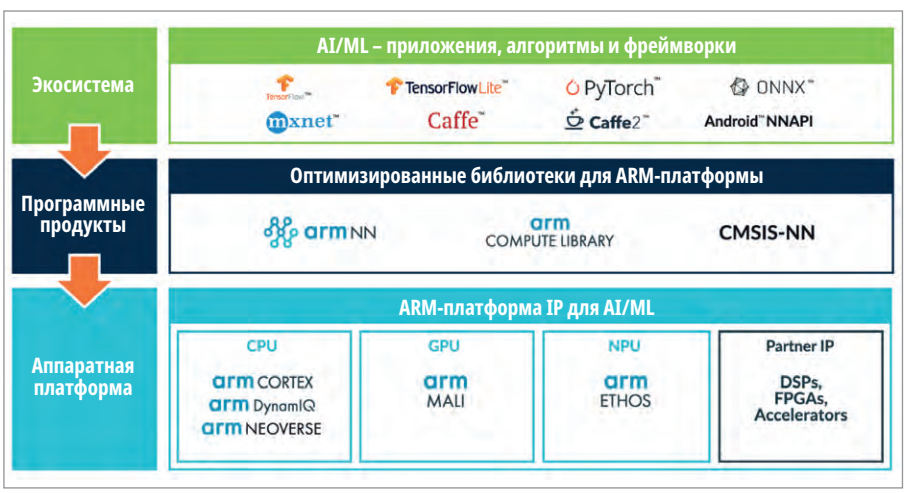

Рис. 2. Процессоры ARM Cortex, оптимизированные для задач с ИИ

соров ARM Cortex от простых моделей к более сложным с указанием возможностей решения функциональных задач.

В последние годы ускоренно развивается поддержка встроенного ПО для микроконтроллеров. Вот некоторые примеры встроенных программ и рамочных решений.

- CMSIS-NN от ARM – библиотека с открытым исходным кодом, которая состоит из эффективных ядер для максимизации производительности нейронных сетей на процессорах ARM Cortex-M.
- TensorFlow Lite Micro – фреймворк машинного обучения с открытым исходным кодом для создания моделей глубокого обучения во встраиваемых системах.
- X-CUBE-AI – пакет, расширяющий возможности STM32CubeMX.AI. Он позволяет преобразовывать предварительно обученные нейронные сети к формату библиотеки ANSI C, оптимизированной для микроконтроллеров STM32 на базе процессорных ядер ARM Cortex-M4 и M7.

Пакет расширения STM X-CUBE-AI позволяет автоматически преобразовывать предварительно обученные нейронные сети для 32-разрядных микроконтроллеров. MicroTensor – облегчённый фреймворк машинного обучения, который используется для моделей TensorFlow и оптимизирован для ядер ARM. PyTorch Mobile позволяет выполнять ML-модели на периферийных устройствах с использованием экосистемы PyTorch. CMSIS-NN разработан для создания приложений Интернета вещей, которые запускают небольшие нейронные сети непосредственно в системах сбора данных. Этот подход предпочтительнее облачных вычислений, поскольку количество IoT-устройств растёт. Библиотека CMSIS-NN была полезна при использовании CNN для классификации изображений в наборе данных CIFAR-10. На платформе ARM Cortex-M7 удалось классифицировать 10,1 изображения в секунду с точностью 79,9%.

Преобразование и оптимизация модели вывода для запуска на устройстве – сложная задача из-за множества доступных встроенных платформ с разной аппаратной поддержкой. Сгенерированные исходные файлы ANSI C компилируются для логического вывода на микроконтроллере.

Процесс генерации с использованием этого фреймворка показан на рис. 3. На низком уровне используются ядра CMSIS-NN. Этот инструмент даёт разработчикам преимущества: графический пользовательский интерфейс, поддержка различных фреймворков глубокого обучения (Keras и TensorFlow Lite), 8-битное квантование и совместимость с различными сериями микроконтроллеров STM32.

Рассмотрим ряд реальных проектов с применением процессоров ARM Cortex, оптимизированных для задач с нейронными сетями, и сведём в таблицу для анализа результатов.

1. В проекте использования больших данных применили методы глубокого обучения для прогнозирования погоды с помощью глубокой крошечной нейронной сети (DTNN). Система автономна и не зависит от облачных сервисов, она основана на микроконтроллере STM32 и наборе инструментов X-CUBE-AI для автоматического преобразования модели в оптимизированную версию для микроконтроллера. В качестве входного параметра используется атмосферное давление. Авторы подробно описывают архитектуру системы на базе микроконтроллера STM32. Ядро системы – микроконтроллер ARM Cortex-M4 с 512 кбайт флеш-памяти и 96 кбайт памяти SRAM. Для управления визуализатором данных используется Raspberry PI.

Для управления потоками используется операционная система реального времени Miosix. Авторы исследовали несколько моделей, в том числе рекуррентные нейронные сети (RNN), такие как LSTM и GRU, которые часто используются для обработки данных временны́х рядов. Также была рассмотрена смешанная архитектура CNN-RNN из-за многообещающих результатов при обработке данных временны́х рядов. В итоге авторы выбрали четыре модели: LSTM, GRU, CNN-LSTM и CNN-GRU. Для каждого семейства была выбрана модель с наилучшей производительностью. Набор данных был получен с сертифицированной метеостанции и использовался в качестве входных данных на определённых этапах предварительной обработки. Для обучения были рассмотрены фреймворки Keras и TensorFlow. Результаты представлены с использованием показателей NRMSE (нормализованная среднеквадратичная ошибка) и NMAE (нормализованная

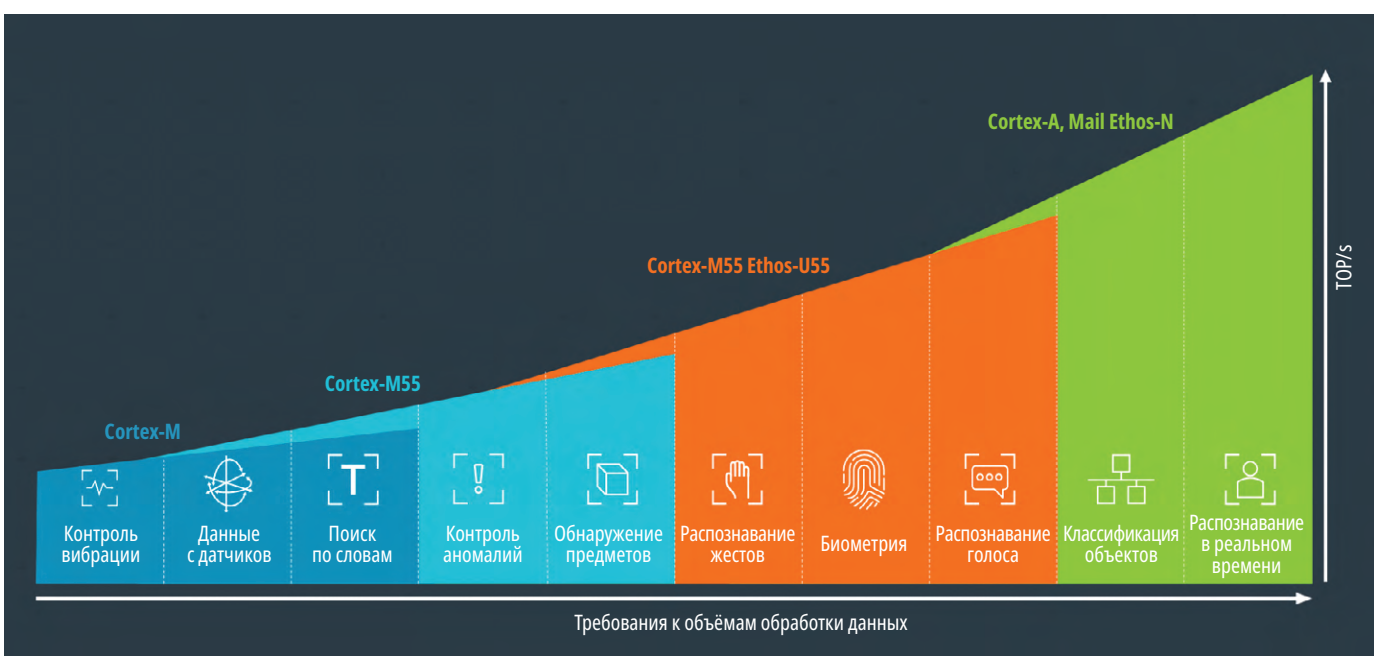

Рис. 3. Преобразование предварительно обученной модели с помощью STM32CubeMX.AI

средняя абсолютная ошибка). Описанные модели показали наилучшую производительность: количество слоёв, ячеек LSTM и фильтров. На основе результатов выбраны перспективные кандидаты – LSTM и GRU. После анализа объёма памяти с помощью инструмента X-CUBE-AI и учёта компромисса между сложностью и точностью для окончательной оценки выбрана модель LSTM. Производительность системы оценивалась в режиме реального времени в течение 30 дней. Результаты немного отличаются от полученных на этапе валидации.

2. Интеллектуальная встраиваемая система для силовых агрегатов реализована на нейронной сети (DNN) на микроконтроллере STM32 для мониторинга приложенной нагрузки в системе трансмиссии. Цель – отслеживать состояние с помощью интеллектуальных датчиков, чтобы контролировать профилактическое обслуживание и снижать затраты. Если известны приложенные усилия, можно оценить условия эксплуатации и спрогнозировать возможные дефекты. Для определения приложенных усилий измеряют вибрации с помощью ёмкостного акселерометра. Использовался микроконтроллер STM32F469AI, ARM Cortex-M4 с 2 Мбайт флеш-памяти и 384 + 4 кбайт памяти SRAM, работающий на частоте 180 МГц. Быстрое преобразование Фурье (FFT) было рассчитано для сигнала вибрации и реализовано на том же микроконтроллере. Разработчики использовали фреймворк Keras для построения нейронной сети DNN. DNN состоит из трёх свёрточных слоёв, за которыми следуют три полностью связанных плотных слоя с примерно 45 000 параметров обучения. Основная цель этой топологии – извлечь полезные характеристики из сигнала вибрации в частотной области с помощью свёрточных слоёв и классифицировать их с использованием полностью связанных слоёв. Для классификации были выбраны семь факторов, что дало семь выходных классов. Предварительно обученная модель была автоматически преобразована в оптимизированную библиотеку на языке C с помощью STM32Cube.AI. Общая точность составляет 97,71%, что немного ниже точности исходной модели до преобразования. Но даже самая низкая точность превышает 90%. Этот пример реализации DNN на микроконтроллере ARM Cortex-M4 STM32 показывает хорошие результаты для промышленных приложений при использовании набора инструментов искусственного интеллекта STM32Cube. Микроконтроллер может обрабатывать алгоритмы DNN и FFT с помощью функций DSP STM32.

3. Проект реализации системы безопасности на транспорте на базе свёрточной нейронной сети (CNN) для обнаружения сонливости. Система была интегрирована в смарт-очки как носимое устройство. Метод сравнили с методом, который обычно используется в аналогичных приложениях и основан на механизмах определения порогов. Для получения входных данных использовали ИК-датчик. Обнаружение основано на событиях моргания. Это сложная задача, потому что могут быть разные обычные ситуации, которые можно интерпретировать как моргание. Авторы подробно описывают используемый набор данных и форму волны ИК-сигнала. Базовая архитектура CNN состоит из двух слоёв одномерной свёртки с 6 и 12 фильтрами, размер фильтра – 7 для обоих слоёв. За каждым уровнем свёртки следует уровень объединения средних значений. В конце для прогнозирования класса используется полностью связанный уровень. Модель была оптимизирована с использованием двоичной функции потери кросс-энтропии и эффективной версии градиентного спуска Adam. После обучения в течение 30 эпох с использованием пакета размером 10 средняя точность за 5 итераций составила 98,2% ± 0,8%. Для повышения точности были изменены количество свёрточных слоёв и количество фильтров на слой, а также размер фильтров и тип операции понижающей выборки. В результате было представлено семь моделей CNN с наилучшей точностью. Самая высокая средняя точность составила 99,5%. Микроконтроллер, встроенный в очки, принадлежит к семейству STM32L451xx. Он разработан на 32-разрядном ядре ARM Cortex-M4 и работал на частоте 40 МГц. Объём памяти ROM – 512 кбайт, оперативной памяти – 160 кбайт. Для преобразования предварительно обученных моделей использовался набор инструментов X-CUBE-AI. Помимо ограничения точности, было дополнительное требование приложения – ограничить объём используемой памяти ROM 90 кбайт. В результате X-CUBE-AI использует 47 кбайт.

Авторы предоставили подробное описание различных показателей производительности для всех оцениваемых моделей по сравнению с механизмом, основанным на пороге. Примеры рассмотренных показателей: время выполнения, среднее энергопотребление, чувствительность, специфичность и точность. Они пришли к выводу, что модели CNN обеспечивают лучшую точность, чем метод, основанный на пороге. Для каждой модели они обсудили поведение в отношении показателей производительности. Самая низкая средняя точность составила 87,4%, а самая высокая – 90,8%. Наконец, модель, показавшая наилучшую производительность, имела следующие особенности по сравнению с механизмом, основанном на пороге: чувствительность была улучшена на 10%, специфичность более чем на 4% и точность – почти на 6%.

4. Проект интеллектуальной распределённой системы на основе искусственного интеллекта для сельскохозяйственной отрасли. Система предназначена для обнаружения болезней кофейных растений с помощью нейронной сети. Она может работать на устройствах с ограниченными ресурсами, таких как маломощные микроконтроллеры. Из-за ограничений облачных вычислений (задержка и безопасность) более подходящим был признан подход граничных вычислений, поскольку обработка выполняется непосредственно на устройстве. Приложение называется Deep Leaf и использует квантованную свёрточную нейронную сеть (Q-CNN), работающую на микроконтроллере STM32.

С помощью инструмента X-CUBE-AI они реализовали пять различных моделей:

- 32-разрядную модель с плавающей запятой;
- сжатую модель;
- квантованную модель с использованием TensorFlow Lite converter;
- квантованную модель с использованием целочисленного представления;
- квантованную модель с использованием представления Q-формата с фиксированной запятой.

Обеспечивается анализ производительности с использованием следующих показателей: время вывода, использование памяти и энергопотребление. Использовалась платформа разработки STM32F746GDISCOVERY с микроконтроллером STM32F746NG на базе ядра ARM Cortex-M7. Он имеет 1 Мбайт флеш-памяти и 340 кбайт оперативной памяти. Функции инструмента X-CUBE-AI, такие как методы сжатия и квантова-

ния, были использованы для преобразования модели в соответствии с ограничениями микроконтроллера. Используемый набор данных состоит из изображений здоровых и больных кофейных листьев. Для увеличения количества изображений были применены методы увеличения объёма данных, а для повышения помехоустойчивости к изображениям был добавлен шум.

Система состоит из специального бокса, в который помещается лист для анализа. Для всех пяти моделей были измерены показатели точности и отзыва. Для моделей TensorFlow Lite и integer quantized была получена та же производительность, что и для 32-разрядной модели с плавающей запятой. Для квантованной модели Qm, n было получено небольшое снижение точности – на 1%. Это составило 95% от максимальной точности. Кроме того, модели были сравнены с точки зрения использования флеш-памяти и оперативной памяти, среднего времени вывода и среднего энергопотребления.

Авторы пришли к выводу, что квантованная модель, использующая представление Q-формата с фиксированной точкой, подходит для развёртывания на микроконтроллере. Для этой модели среднее энергопотребление составляет 134,12 МДж, что является самым низким показателем по сравнению с другими моделями. Важным выводом было то, что методы квантования превзошли метод сжатия по всем рассмотренным показателям производительности.

5. Проект контроля и предсказания действия лекарственных препаратов для анестезии на базе нейронной сети для управления моделями с предсказанием (DNN-MPC) на микроконтроллере ARM. Была исследована производительность MPC-модели на основе глубокого обучения для управления анестезией при внутривенной доставке лекарств. Линейные MPC-модели не подходят для реализации в реальном времени, так как требуют больших вычислительных затрат. Однако использование методов глубокого обучения позволяет обеспечить точную аппроксимацию линейного закона управления MPC и сократить сложность вычислений и объём памяти. Была выбрана рекуррентная нейронная сеть (RNN), поскольку такая модель обычно используется для приложений MPC. Данные для обучения были собраны в результате моделирования. Процесс обучения выполнялся в автономном режиме с использованием функции временны́х рядов нейронной сети на основе MATLAB и метода Левенберга–Марквардта. Обучение было остановлено, когда не стало видно дальнейшего улучшения обобщения, а среднеквадратичная ошибка (MSE) и регрессия (R) приблизились к нулю.

Микроконтроллер представлял собой ARM Cortex-M3 с 512 кбайт флеш-памяти и 96 кбайт памяти SRAM, работающий на частоте 84 МГц. Вычислительное время итерации сокращено с 11,354 мс до 2,99 мс. Эти результаты показаны как сравнение между линейным MPC и DNN MPC, включая использование памяти данных и программ. Объём памяти был аналогичным, но авторы упомянули, что в случае более крупных систем разница будет гораздо более заметной.

6. Решение для обнаружения присутствия человека на улице с помощью тепловизионной камеры низкого разрешения на базе свёрточной нейронной сети (CNN). Вывод выполняется на 32-разрядном микроконтроллере ARM Cortex-M4 с 1 Мбайт флеш-памяти и 128 кбайт памяти SRAM.

Для получения тепловизионного изображения размером 8×8 используется инфракрасный матричный датчик Grid-EYE, который делает снимки с частотой 10 Гц. Тепловизионные камеры больше подходят для этой задачи, но они намного дороже и потребляют больше энергии, чем PIR-датчики. Для обучения использовался пользовательский набор данных, расширенный за счёт добавления изображений, сделанных при различных температурных условиях. Перед использованием в качестве входных данных для CNN к изображению были применены некоторые этапы предварительной обработки: вычитание фона и выполнение усреднения фона. Архитектура сети проста и ориентирована на двоичную классификацию: «человек» или «без человека». Архитектура сети состоит из трёх свёрточных уровней и одного полностью связанного уровня. Гиперпараметры kernel size и stride были равны трём и одному соответственно. Для свёрточных слоёв использовалась функция активации выпрямленной линейной единицы (ReLU), а для последнего плотного слоя – сигмовидная, поскольку задача классификации двоичная.

Процесс обучения проходил с использованием фреймворка TensorFlow в течение 1000 эпох. Применялись функция кросс-энтропийных затрат и алгоритм оптимизации Adam. Использовались ядра, оптимизированные под CMSIS-NN для ARM Cortex-M. Было выполнено 8-битное квантование весов и активаций с фиксированной запятой. Для реализации применялась универсальная плата разработки STM NUCLEO-L476RG. Потребляемая мощность составила 16,5 мВт, время выполнения – 4,01 мс, а объём памяти – 25,08 кбайт (текст, BSS и данные). Производительность классификации была проанализирована с использованием обеих моделей: TensorFlow (32-разрядное представление с плавающей запятой) и CMSIS-NN (8-разрядная реализация с фиксированной запятой).

Для тестирования использовались все три раздела набора данных: train, validation и test. Производительность классификации снизилась на 0,2%, 1,0% и 0,2% соответственно. Производительность модели CMSIS-NN составила 80,9%, 76,4% и 76,7% соответственно.

В заключение было получено время вывода всего 4 мс при потребляемой мощности 2,3 мВт. Эксперименты показывают, что 8-битное представление с фиксированной точкой существенно не влияет на производительность, приводя к потере точности максимум на 1%.

7. Система предиктивного управления энергопотреблением интеллектуального здания с использованием нейронной сети с глубоким обучением для прогнозирующего управления смешанными целочисленными моделями. Набор данных, используемый для обучения, был получен из 500 различных запусков MPC. Они разделили набор данных между обучающим (90%) и оценочным набором (10%). Фреймворками, использованными для проектирования DNN, были TensorFlow и Keras с оптимизатором Adam. Были обучены три разные модели: две сети с мелким расположением и одна глубокая сеть. Авторы показывают, что глубокая сеть работает лучше, чем мелкие сети, за счёт меньшей ошибки обучения и меньшего объёма памяти. Поэтому для дальнейшего изучения была выбрана архитектура глубокой сети. Из-за простоты модели она была реализована на микроконтроллере. Для генерации C-кода использовался инструмент EdgeAI. Микроконтроллер использовался на ядре ARM Cortex-M3 с 96 кбайт оперативной памяти, 512 кбайт флеш-памяти и работал с частотой 89 МГц. В сети всего 5 скрытых слоёв по 10 нейронов

на слой. Время вычисления составило 2,9 мс, а объём памяти – всего 35 кбайт. Были проанализированы различия между использованием функций активации ReLU и tanh. В результате сделан вывод, что время вычислений значительно больше (7,3 мс), а код больше (37,3 кбайт) при использовании функции tanh из-за дополнительных математических библиотек, которые были необходимы.

8. Мобильная система безопасности для обнаружения падений с использованием рекуррентных нейронных сетей (RNNS) LSTM, работающих на микроконтроллере. Такая система полезна для мониторинга пожилых людей на предмет непреднамеренных падений и для отправки предупреждающих уведомлений в систему удалённого мониторинга при положительном результате.

Авторы исходили из следующих трёх основных требований:

- должно быть обеспечено постоянное беспроводное соединение для оповещения;
- система должна быть как можно меньше и легче, чтобы избежать возможных неудобств;
- система должна представлять собой устройство с низким энергопотреблением, поскольку оно питается от аккумулятора.

Эти требования легли в основу необходимости внедрения системы обнаружения падений в реальном времени, которая выполняет вычисления непосредственно на встроенном устройстве. Миниатюрная плата SensorTile была выбрана в качестве подходящего устройства для этого приложения. Это стало возможным благодаря маломощному микроконтроллеру STM32L476JGY на базе ядра ARM Cortex-M4 и дополнительным доступным встроенным функциям, таким как трёхосевые акселерометры.

Микроконтроллер имеет 1 Мбайт флеш-памяти и 128 кбайт оперативной памяти. Используется арифметика с плавающей запятой одинарной точности, чтобы избежать значительных потерь в точности, которые могли быть вызваны методами обрезки или квантования. В качестве набора данных использовался SisFall, который является одним из распространённых наборов данных, доступных для этого приложения. Он был размечен вручную и разделён на обучающий и проверочный набор (80% обучающего набора и 20% тестового набора).

Соответствующими классами, которые были определены, были FALLS, ALLERTS и фоновый класс (BKG), который охватывает обычные действия, не связанные с падением.

Результаты сравнения решений на ARM Cortex

Использование микроконтроллеров на базе ядра ARM Cortex-M для приложений с глубоким обучением является перспективным решением. Это подтверждается тем, что недорогая архитектура ARM Cortex-M очень популярна и уже широко используется в различных встраиваемых приложениях. Поэтому добавление алгоритмов ИИ на эти платформы является следующим логичным шагом. В табл. 1 представлены решения с использованием алгоритмов периферийных вычислений на основе глубокого обучения для следующих областей применения: прогнозирование погоды, профилактическое обслуживание, распознавание сонливости, сельское хозяйство, классификация эпизодов приёма пищи, распознавание звуковых событий на открытом воздухе, прогнозное управление моделями, распознавание присутствия человека на открытом воздухе, мониторинг литий-ионных аккумуляторов, системы обнаружения падений и медицинское применение. В обзоре указаны инструменты, используемые для обучения моделей или преобразования их в совместимый формат для работы на микроконтроллере, архитектура модели, приложение, аппаратные ресурсы и результаты.

Для встраиваемых устройств наиболее распространённым подходом к построению модели является обучение на главной рабочей станции с последующей оптимизацией предварительно обученной модели. Однако в последнее время растёт интерес к получению оптимизированной модели непосредственно во время обучения, например, для обучения с учётом квантования.

Анализ архитектуры решений

Наиболее распространёнными архитектурами глубоких нейронных сетей, работающими во встроенных системах на базе ядра ARM Cortex-M, являются CNN, RNN, LSTM, GRU или их комбина-

Таблица 1. Сравнение решений по глубокому обучению с использованием ARM Cortex-M

№	Модель	Сфера применения	Инструментарий	Аппаратная платформа	Итоги
1	LSTM, GRU, CNN-LSTM, CNN-GRU	Прогноз погоды	X-CUBE-AI toolchain, Keras, TensorFlow	STM32F401RET6 ARM Cortex-M4 512 kB of Flash 84 MHz	NRMSE: 0.0328 NMAE: 0.0251
2	CNN	Мониторинг нагрузки в системе трансмиссии	STM32Cube.AI, Keras	STM32F469AI ARM Cortex-M4 2 MB of Flash 180 MHz FPU	Точность: 97,71%
3	CNNs	Обнаружение сонливости на основе моргания глаз	X-CUBE-AI toolchain	STM32L451xx ARM Cortex-M4 512 kB of Flash 80 MHz FPU	Точность в диапазоне: 87,4–90,8%
4	Q-CNN	Выявление болезней кофейных растений	X-CUBE-AI toolchain, TensorFlow Lite	STM32F746NG ARM Cortex-M7 1 MB of Flash 216 MHz FPU	Точность: 96%
5	RNN	Модель предиктивного контроля для анестезии	Matlab	RM Cortex-M3 512 kB of Flash 96 kB of SRAM 84 MHz	Время ответа: 2,99 мс
6	CNN	Обнаружение присутствия человека на открытом пространстве	TensorFlow, CMSIS-NN	STM32L476RG ARM Cortex-M4 1 MB of Flash 80 MHz FPU	Время ответа: 4,01 мс
7	DNN	Предиктивное управление энергопотреблением интеллектуального здания	TensorFlow, Keras, EdgeAI	ARM Cortex-M3 512 kB of Flash 96 kB of SRAM 89 MHz	Время ответа: 2,9 мс
8	LSTM, RNNs	Мобильная система обнаружения падений	TensorFlow, CMSIS	STM32L476JGY ARM Cortex-M4 1 MB of Flash 80 MHz FPU	Точность: 98%

ции. Архитектура ограничена вычислительными ресурсами из-за аппаратных ограничений и низкой частоты работы микроконтроллеров для снижения энергопотребления. Современные архитектуры обычно поставляются с новыми функциями, которые не поддерживаются существующими библиотеками SW. Поэтому архитектура NN должна определяться с учётом ограничений, связанных с реализацией SW, вычислительной мощностью и памятью, в зависимости от используемого микроконтроллера. Когда предварительно обученные модели преобразуются для развёртывания на встроенных устройствах, это нормально. При изменении исходной модели ожидается незначительное снижение производительности. Например, в приложении для обнаружения сонливости одной из основных целей автора является сравнение с традиционным методом – механизмом на основе порога. Также можно наблюдать среднюю точность до и после применения методов преобразования. Снижение средней точности для наиболее эффективной модели CNN составляет 8,7%. Заметно снижение точности и для приложения predictive maintenance – на 5,8%. Это указывает на необходимость дальнейшей оптимизации методов преобразования. Несмотря на снижение производительности при преобразовании, всё ещё можно получить высокопроизводительные модели. Например, наилучшие средние результаты для архитектуры CNN были получены в таких приложениях, как профилактическое обслуживание и обнаружение болезней на кофейных листьях. CNN, используемая для определения присутствия человека на улице, имеет невысокую точность (менее 80%), однако это связано с низким разрешением входного изображения (всего 8×8). Использование изображения с более высоким разрешением может повысить точность обнаружения. RNN широко используются в периферийных вычислениях, так как они эффективны при работе с данными временны́х рядов. Наиболее распространёнными архитектурами являются LSTM и GRU. Для некоторых приложений полезны смешанные архитектуры с классом CNN. Наилучшие результаты точности были получены для таких приложений, как носимые системы для обнаружения падений (94,41%) или классификации эпизодов приёма пищи (98%). В целом результаты аналогичны тем, что были получены для архитектур CNN, с небольшим снижением производительности при использовании методов оптимизации. В некоторых работах оценивается энергопотребление, которое указано в табл. 1. Например, при обнаружении заболеваний на кофейных листьях максимальное энергопотребление составляет 5 мВт. Энергопотребление анализируемых систем составляет менее 10 мВт, поэтому такие системы легко могут быть спроектированы как устройства с батарейным питанием. Эффективность можно повысить с помощью методов оптимизации модели, таких как квантование. После квантования получается разница в энергопотреблении в 380,4 нДж.

Аппаратная платформа

Большинство анализируемых решений используют ядра ARM Cortex-M4 или M7, поскольку они демонстрируют высокую производительность в категории недорогих систем. Ядра ARM Cortex-M оптимизированы для приложений с преобладанием управления потоками. Однако вывод DNN состоит из параллельной обработки данных и будет хуже работать только на центральном процессоре. Для маломощных и недорогих датчиков не подходит добавление второго DSP или ускорителя. Чтобы преодолеть разрыв между управлением потоком и параллельными вычислениями, ARM предлагает семейства ядер M4 и M7 с инструкциями DSP непосредственно в ядре без сопроцессора.

Энергопотребление, требуемый объём памяти и время вывода сильно зависят от используемой аппаратной платформы, сложности модели и частоты работы процессора. Эти требования устанавливаются в зависимости от приложения, чтобы получить максимально эффективную систему. Важным шагом при разработке приложений на встраиваемых устройствах является оптимизация модели. Например, квантование может значительно сократить требуемый объём памяти. Однако этот метод может привести к снижению точности, поэтому правильный метод должен быть тщательно выбран. Негативное влияние зависит от используемого метода квантования. Сейчас искусственный интеллект часто используется с STM32Cube IDE. Пакет расширения X-CUBE-AI предоставляет комплексные решения для автоматического преобразования модели нейронной сети, проверки достоверности и измерения производительности системы. Поэтому 32-разрядные микроконтроллеры ARM Cortex-M – наиболее распространённая платформа. Популярность решений на базе ARM объясняется доступностью этих наборов инструментов.

Большинство проектов показали многообещающие результаты с точки зрения точности, времени выполнения, энергопотребления и объёма памяти. Однако парадигма пограничных вычислений – новая тема исследований со множеством задач. Эти проблемы касаются как аппаратных, так и программных решений. В этом разделе мы обсудим высокоуровневые проблемы и возможности для будущих исследований по внедрению DL на недорогих микроконтроллерах. Они связаны с аппаратными устройствами, программной реализацией и сжатием глубоких нейронных сетей.

Использование специализированных аппаратных ускорителей эффективно, но разработка таких ускорителей для конкретных приложений слишком дорога. При использовании микроконтроллеров общего назначения определённые возможности могут повысить производительность и эффективность вычислений (например, SIMD или векторные расширения, аппаратно реализованные вычисления с плавающей запятой, иерархия кэша или энергонезависимая память большего размера для хранения большого количества параметров). Также предлагается реализовать математические алгоритмы, такие как разложение по сингулярным значениям, для систем с «голым металлом» серии ARM Cortex-M. Это может быть полезно для реализаций глубокого обучения. Доступ к памяти важен в этих приложениях из-за большого количества перемещений данных, что сильно влияет на потребление энергии и задержку. Чтобы компенсировать это, появились передовые методы, например, вычисления в памяти. В отличие от архитектуры фон Неймана, где память и процессоры физически разделены, с помощью этого метода определённые вычислительные задачи могут выполняться в самой памяти на основе физических атрибутов устройств памяти. Используются энергонезависимые аналоговые мемристорные перемычки, которые физически представляют веса в виде проводимостей в каждой точке пересечения. При подаче напряжения на строки векторно-матричное умножение генерируется как ток в строках столбцов по законам Кирхгофа и Ома. Пока эта пе-

Marionix

ПРОСТО. НАДЕЖНО. ДОСТУПНО

IES6200-PN

IES618 - управляемые промышленные коммутаторы с поддержкой PROFINET

- 8 x 10/100BASE-T(X) (RJ45)
- 6 x 10/100BASE-T(X) (RJ45) + 2 x 100BASE-FX/LX (SC/ST/FC)
- 4 x 10/100BASE-T(X) (RJ45) + 4 x 100BASE-FX/LX (SC/ST/FC)
- Поддержка протоколов резервирования ERPS V2, SW-RING, RSTP, LACP
- Резервированный вход по питанию 12..60 В (DC)
- Диапазон рабочих температур: -40..75°C

редовая технология не может быть использована для реальных приложений. Более эффективным решением для увеличения параллельных вычислений при сохранении низкого энергопотребления является многоядерная архитектура. Однако управлять такой системой сложнее. Более эффективной архитектурой может стать объединение специально разработанных аппаратных ускорителей для глубокого обучения с процессором общего назначения и периферийными устройствами ввода-вывода, такими как представленное устройство в разделе II-A контроллера серии 78000. Производители новейших аппаратных устройств не предоставляют базовых программных библиотек для разработки приложений. Поэтому для использования современных аппаратных устройств часто требуется разработка с нуля, что отнимает много времени и является сложной задачей.

Микроконтроллеры STM32 широко используются в качестве микроконтроллеров общего назначения, поскольку поставщик предоставляет пакет X-CUBE-AI. Однако свобода ручной оптимизации ограничена, так как это предварительно скомпилированная среда с высокоуровневой конфигурацией. С другой стороны, это преимущество для разработчиков без большого опыта в данной области. Популярные фреймворки ML, такие как TensorFlow, Keras и Caffe, поддерживаются их наборами программных инструментов. Недавно появились новые фреймворки, которые могут улучшить результаты: MicroAI и MCUNet. MicroAI – это платформа для развёртывания глубоких нейронных сетей на микроконтроллерах, включая квантование. Основные особенности фреймворка MicroAI:

- реализация CNN с непоследовательными топологиями;
- поддержка 16-битного квантования;
- не предназначен для ограниченного семейства аппаратных целей.

MCUNet – это фреймворк, который обеспечивает эффективное проектирование нейронной архитектуры с использованием двухэтапного подхода к поиску нейронной архитектуры наряду с библиотекой вывода. Они достигли рекордно высокой точности в 70,7% на микроконтроллерах, используя крупномасштабный набор данных ImageNet.

Оба фреймворка превосходят существующие решения, такие как TensorFlow Lite Micro и CMSIS-NN, что относит их к самым перспективным фреймворкам. Методы сжатия сетей, такие как квантование и обрезка, постоянно совершенствуются. Однако разработка с нуля остаётся сложной задачей. Чтобы её решить, разрабатываются различные решения с открытым исходным кодом, например:

- сжатие моделей с помощью Neural Network Intelligence (NNI);
- инструментарий эффективности моделей искусственного интеллекта (AIMET);
- SparseML.

Заключение

Глубокое обучение и глубокие нейронные сети – многообещающие решения для сложных задач. Традиционно такие задачи решаются с помощью больших компьютерных систем со специализированным оборудованием, так как требуют высоких вычислительных мощностей и ресурсов памяти. Однако недавние исследования показывают, что парадигма глубокого обучения и реализация периферийных вычислений могут быть полезны и для простых приложений. Периферийные вычисления помогают решать многие реальные проблемы, которые необходимо решить в ближайшее время в контексте недорогих/маломощных приложений. В этом случае процессор ARM Cortex-M является одним из лучших возможных кандидатов. В статье описана реализация глубоких нейронных сетей с использованием микроконтроллеров на базе ядра ARM Cortex-M зарубежных производителей.

Внедрение глубоких нейронных сетей на встраиваемых устройствах, таких как микроконтроллеры, – сложная задача. Это связано с ограничениями на вычисления и объём памяти. Поэтому разработчики вынуждены настраивать существующие архитектуры или даже разрабатывать инновационные модели, которые лучше подходят для встраиваемых процессоров.

Все описанные примеры и нейросетевые решения применимы на отечественных процессорах ARM-архитектуры. В итоге использование оптимизированного оборудования в сочетании с оригинальными архитектурами глубоких нейронных сетей приводит к созданию интеллектуальных и энергоэффективных систем. ●

Автор – сотрудник фирмы ПРОСОФТ
Телефон: (495) 234-0636
E-mail: info@prosoft.ru

НОВОСТИ реклама

Российские ИБП Эксперт и Легион от «Сайбер Электро» получили сертификат сейсмостойкости в 9 баллов по шкале MSK-64

Отраслевая защищённость, прочность и полное соответствие российским нормативам качества продукции торговой марки «Сайбер Электро» подтверждена новым сертификатом на сейсмостойкость.

Источники бесперебойного питания серий ЭКСПЕРТ и ЛЕГИОН от «Сайбер Электро» успешно прошли тестовые испытания на сейсмоустойчивость и получили действующий сертификат MSK-64.

Экспериментальные данные были получены расчётным методом с использованием математической модели и показали, что ИБП ЭКСПЕРТ и ЛЕГИОН, мощностью от 1/10 до 10/500 кВА, сохраняют свои рабочие характеристики при сильных разрушительных землетрясениях с амплитудой 9 баллов.

Сейсмоустойчивые и надёжные бесперебойники ЭКСПЕРТ и ЛЕГИОН подходят для эксплуатации на промышленном производстве и ЦОД, в составе телекоммуникационных систем, АСУ ТП и серверных на объектах, расположенных в сейсмоопасных климатических зонах.

Производителю выдан соответствующий сертификат ГОСТ Р, подтверждающий исполнение его продуктов сейсмостойкости в 9 баллов по шкале MSK-64.

Об уровне 9 сейсмозащиты ИБП по шкале MSK-64

Шкала сейсмостойкости MSK-64 применяется в России для определения устойчивости зданий, сооружений, оборудования и различных конструкций к разрушительной силе толчков при землетрясениях. Уровень сейсмостойкости в 9 баллов, максимальный для зданий и сооружений, при котором они ещё могут устоять без обрушений, означает, что в случае сильных толчков ИБП «Сайбер Электро» не будут повреждены на объектах с таким же высоким уровнем сейсмозащиты. ●