Изменено	14/03/2023 10:38:00	Создано	10/03/2023 11:54:15	https://www.compel.ru/lib/140486

Нейронные сети на базе STM32G4. Теория и практика

19 октября 2020

Алексей Гребенников (г. Москва)

Наличие программного пакета X-CUBE-AI, расширяющего функционал STM32CubeMX, поможет разработчику построить искусственную нейросеть на базе микроконтроллера из линейки STM32G4 производства STMicroelectronics. Статья включает пошаговое описание реализации такой нейросети.

Идея построения искусственных нейронных сетей появилась в 50-х годах прошлого века. До недавних пор практическое применение таких сетей было крайне ограничено из-за недостаточной мощности вычислительной техники. Однако успехи современной микроэлектроники позволили существенно нарастить производительность электронных устройств, благодаря чему практическое использование нейронных сетей стало возможным даже на базе компактных микроконтроллеров. В данной статье рассматривается теория и практика применения искусственных нейронных сетей с использованием микроконтроллеров семейства STM32G4 производства компании STMicroelectronics.

Немного истории

С момента изобретения ЭВМ до настоящего времени огромное количество программного обеспечения пишется на алгоритмических языках, предполагающих жесткую фиксированную структуру алгоритма. То есть любое состояние программы получается после выполнения определенного набора условий и может быть заранее рассчитано. Такие алгоритмы хорошо себя зарекомендовали, так как обладают высокой степенью предсказуемости и при определенном объеме моделирования минимизируют риск внештатных ситуаций. Однако программы на алгоритмических языках плохо справляются с некоторыми классами задач, особенно с такими, где результат не выражается просто состояниями «1» или «0», а является некоторой вероятностью в диапазоне значений. Для решения таких задач были разработаны искусственные нейронные сети, работающие по принципу биологических нейронных сетей. Искусственная нейронная сеть (далее по тексту — просто нейронная сеть или нейросеть) представляет собой совокупность нейронов, соединенных между собой синапсами. Такая структура сети позволяет эффективно решать задачи классификации, предсказания, распознавания и так далее.

Процесс создания приложения на базе нейронной сети в микроконтроллерах STMicroelectronics состоит из пяти базовых этапов, изображенных на рисунке 1.

Рис. 1. Этапы создания нейросети

Как видно из рисунка 1, первые три этапа относятся к созданию нейросети, а четвертый и пятый – к эксплуатации.

На первом этапе происходит сбор данных для анализа. Обычно для этого используются датчики и сенсоры, которые располагаются рядом с объектом анализа и регистрируют изменения его состояния в пространстве и времени. Примерами физических величин для регистрации являются скорость, ускорение, температура, звуковые и видеохарактеристики объекта в зависимости от приложения нейросети. Компания STMicroelectronics предлагает

устройства, облегчающие сбор данных, такие как платформа SensorTile, которая работает в автономном режиме и поддерживает дистанционное управление через приложение для смартфона ST BLE Sensor. SensorTile содержит датчики движения и состояния окружающей среды, микроконтроллер, разъем для подключения SD-карты и модуль Bluetooth.

Данные, полученные от сенсоров, необходимо промаркировать. Для контролируемого обучения (обучения с учителем) необходима классификация полученных данных по определенным признакам — маркерам. Например, можно классифицировать изображения по наличию или отсутствию на них человека. Такие промаркированные данные являются эталоном для обучения и проверки нейросети. Разработчики должны определить оптимальную топологию нейросети для лучшего обучения и обеспечения корректных полезных данных конечному приложению. Обычно такие задачи решаются с помощью готовых интегрированных сред программирования для глубокого обучения. У компании ST есть ряд партнеров, которые предоставляют готовые инженерные решения в области нейросетей, а также оказывают поддержку со стороны архитекторов нейронных сетей и специалистов в данной области.

Тренировка нейросети представляет собой итерационный процесс обработки эталонных наборов данных с целью минимизации критерия ошибки. Эта задача обычно выполняется с помощью готовых интегрированных сред от сторонних разработчиков. Обучение, как правило, выполняется на мощных вычислительных машинах с практически неограниченными вычислительными ресурсами и объемами памяти, что позволяет выполнить большое число итераций за короткий промежуток времени. Результатом такого процесса является предварительно обученная нейросеть. Пакет STM32Cube.AI предоставляет простой и эффективный интерфейс с наиболее популярными средами для глубокого машинного обучения, такими как Keras, Caffe, TensorFlow Lite, ONNX, PyTorch, Matlab и другими. Эти среды представляют собой открытые программные библиотеки для машинного обучения, разработанные различными компаниями с целью решения задач построения и тренировки нейронных сетей для автоматического нахождения и классификации образов, сравнимых по качеству с человеческим восприятием. Выходные данные этих сред могут быть напрямую импортированы в пакет STM32Cube.AI.

Следующий, четвертый шаг построения нейросети – преобразование предварительно обученной нейросети, сгенерированной сторонней интегрированной средой, в программный код, оптимизированный для исполнения на микроконтроллере STM32. Оптимизация подразумевает минимизацию числа вычислений и объема используемой памяти. Этот шаг очень легко выполняется с помощью пакета STM32Cube.AI, интегрированного в экосистему разработки STM32 в качестве расширения известного инструмента STM32CubeMX. Пакет STM32Cube.AI позволяет выбрать необходимый микроконтроллер для конкретной задачи, предоставляет обратную связь по производительности нейросети на базе выбранного микроконтроллера, обеспечивает проверку сети как на компьютере, так и на целевом устройстве.

Финальный, пятый шаг – это внедрение созданной нейросети в пользовательское приложение. Для решения данной задачи компания STMicroelectronics предлагает широкий набор низкоуровневых драйверов, библиотек и пользовательских приложений, собранных в один пакет программного обеспечения. Для ускорения процесса проектирования разработчики могут использовать эти шаблоны, внося в них необходимые изменения.

Метод, в котором для обучения требуются эталонные данные, называется машинным обучением. Однако существует также метод, не требующий эталонных данных для обучения, он называется глубоким обучением. В случае глубокого обучения структура искусственных нейронных сетей состоит из нескольких входных, выходных и скрытых слоев. Каждый слой содержит единицы, преобразующие входные данные в сведения, которые следующий слой может использовать для определенной прогнозируемой задачи. Благодаря этой структуре вычислительное устройство может изучать собственную обработку данных. На рисунке 2 показаны семейства микроконтроллеров STM32, поддерживающие машинное и глубокое обучение.

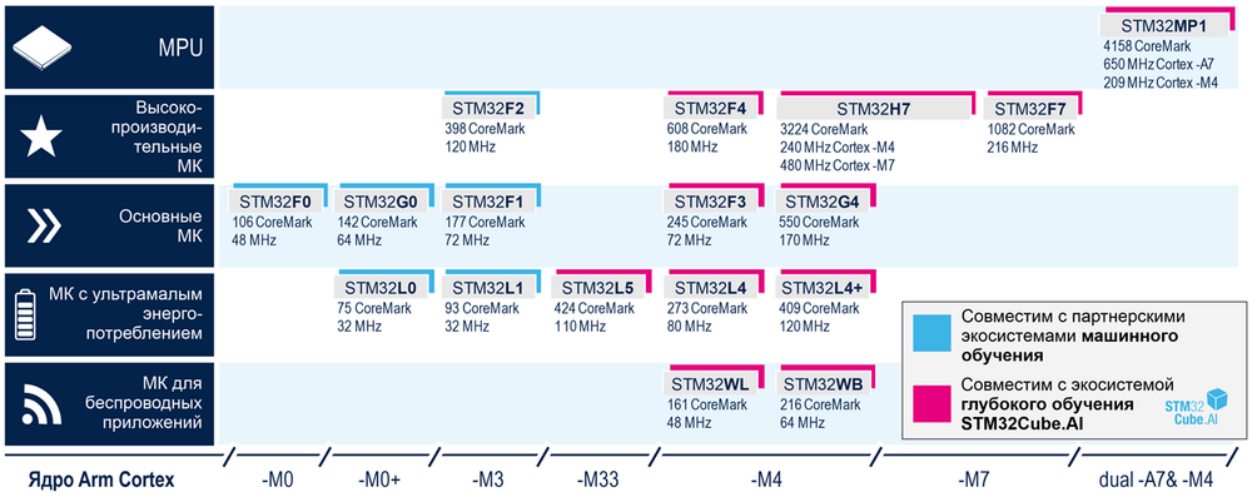

Рис. 2. Поддержка машинного и глубокого видов обучения

Пакет X-CUBE-AI

Пакет X-CUBE-AI, расширяющий функционал STM32CubeMX, используется на четвертом шаге построения нейронной сети (рисунок 1) и служит для преобразования предварительно обученной нейронной сети в формат, оптимизированный для выполнения на микроконтроллере.

X-CUBE-AI также позволяет на этапе создания проекта выбрать микроконтроллеры, подходящие для выбранной нейросети по определенным ресурсам, таким как размер памяти RAM или Flash. Этот пакет позволяет сгенерировать три типа проектов:

проект оценки системной производительности выполняется на микроконтроллере семейства STM32 и позволяет точно оценить объем используемой памяти и нагрузку вычислительного модуля;
проверочный проект позволяет оценить результаты, полученные от нейросети с использованием в качестве входных данных набора случайных чисел и пользовательского набора тестовых данных. Проект может исполняться как на компьютере, так и на микроконтроллере;
шаблон приложения позволяет быстро создавать приложения с элементами искусственного интеллекта (далее по тексту – ИИ).
На рисунке 3 показано ядро системы X-CUBE-AI.

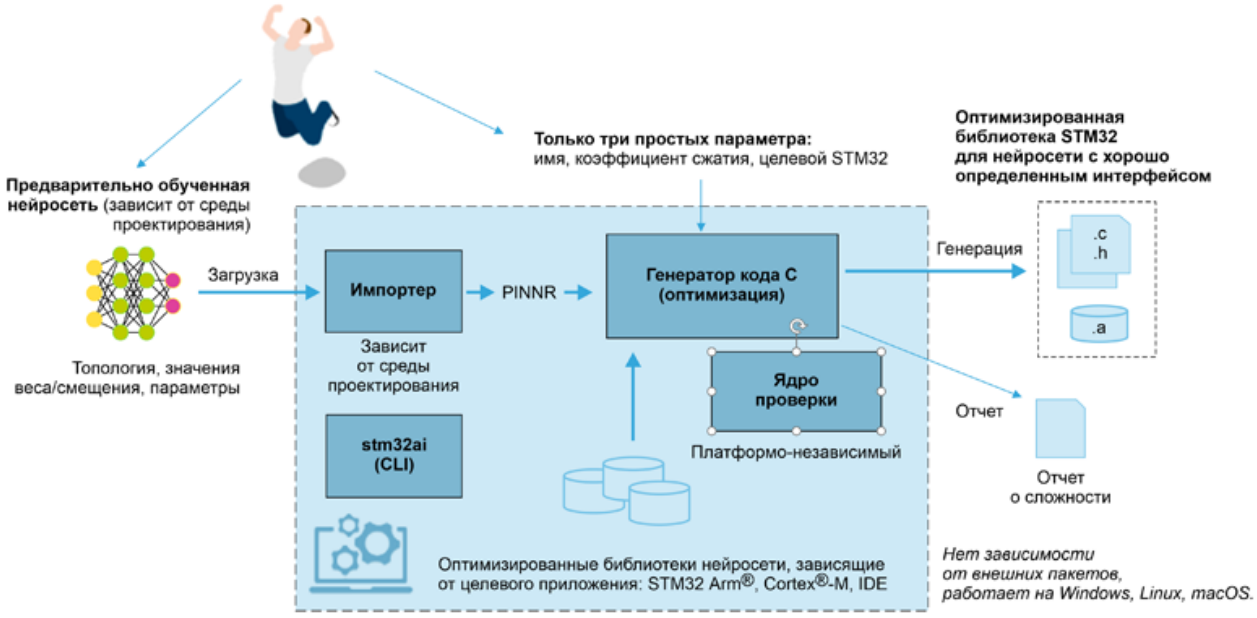

Рис. 3. Ядро системы X-CUBE-AI

Это ядро обеспечивает конвертацию нейросети в модель на языке C для использования во встроенных приложениях с ограниченными ресурсами аппаратного обеспечения. Сгенерированная библиотека нейросети (общая и специализированная части) может быть напрямую встроена в среду разработки. Для создания пользовательских приложений также экспортируется интерфейс API. Все функции ядра X-CUBE-AI доступны через интерфейс командной строки на уровне консоли. Упрощенный интерфейс конфигурации, доступный через графическую оболочку, содержит несколько параметров:

имя сгенерированной модели на C;
коэффициент сжатия для уменьшения размера модели;
возможность для семейства STM32 выбрать оптимальную библиотеку.
На рисунке 4 показан основной функционал, который поддерживается импортируемой моделью нейросети и целевой платформой.

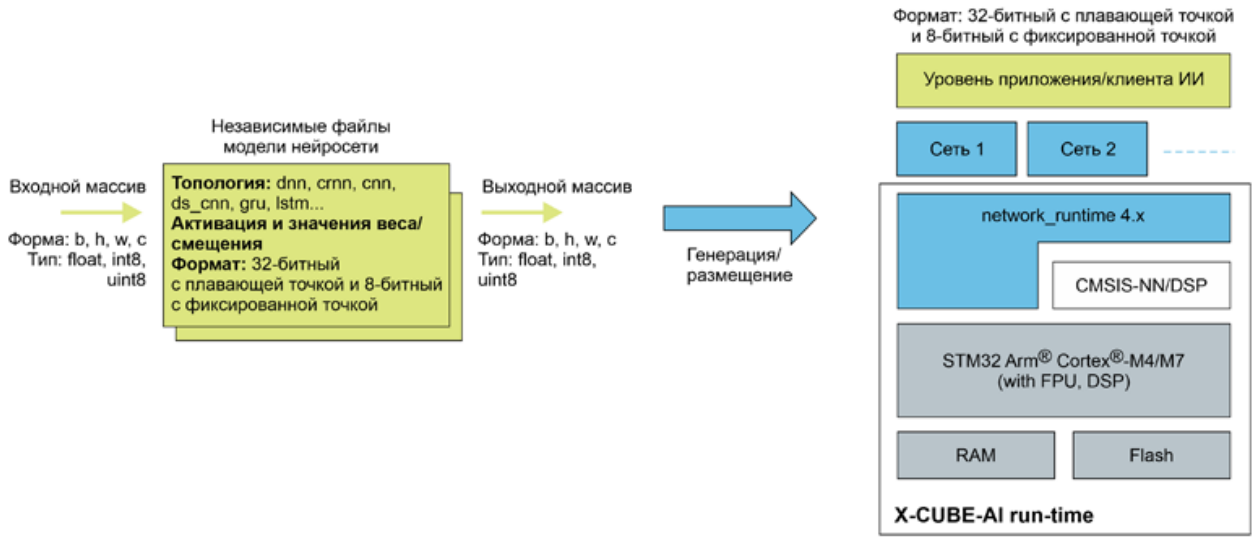

Рис. 4. Обзор X-CUBE-AI

Поддерживаются только простые массивы входных и выходных данных:

4-размерная форма: группа, высота, ширина, канал (формат “channel-last”);

32-битные данные с плавающей точкой и 8-ми битные данные с фиксированной точкой. Сгенерированные модели на C полностью оптимизированы для ядер STM32 Arm Cortex–M4/M7 с FPU и DSP.

Ниже – в практической части статьи – будет описан пример с использованием модели Keras. Keras – это библиотека глубокого обучения, представляющая собой высокоуровневый API, написанный на Python. Она позволяет легко и быстро создавать прототипы благодаря удобству в работе, модульности и масштабированию. Keras поддерживает сверточные, рекуррентные сети и их комбинации, может работать как на процессоре общего назначения (CPU), так и на графическом процессоре (GPU).

Генератор кода X-CUBE-AI может быть использован для моделей Keras с предварительным квантованием в формате 8-битных чисел с фиксированной точкой и для моделей TensorFlow Lite с квантованием. Для моделей Keras требуется файл переформатированной модели (h5*) и специализированный конфигурационный файл (json), как это показано на рисунке 5.

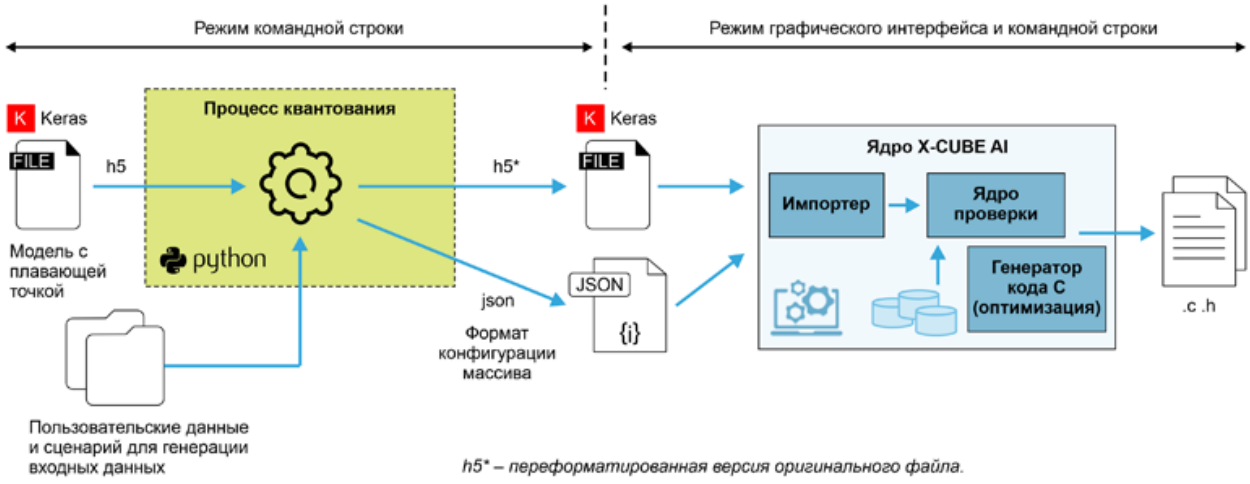

Рис. 5. Процесс квантования

Генератор кода преобразует значения веса и смещения, а также соответствующие активации из формата 32-битных чисел с плавающей точкой в формат 8-битных чисел с фиксированной точкой. Цель этой операции – уменьшить размер модели для увеличения скорости вычислений, обеспечивая при этом лишь незначительное снижение точности.

X-CUBE-AI является частью пакета STM32CubeMX, который используется для быстрой конфигурации микроконтроллеров STM32. STM32CubeMX позволяет с помощью графического пользовательского интерфейса создать полноценный проект для среды разработки, например, STM32CubeIDE, включая возможность генерации инициализационного кода на C для периферии, выводов, тактовых сигналов и других устройств. Структура STM32CubeMX показана на рисунке 6.

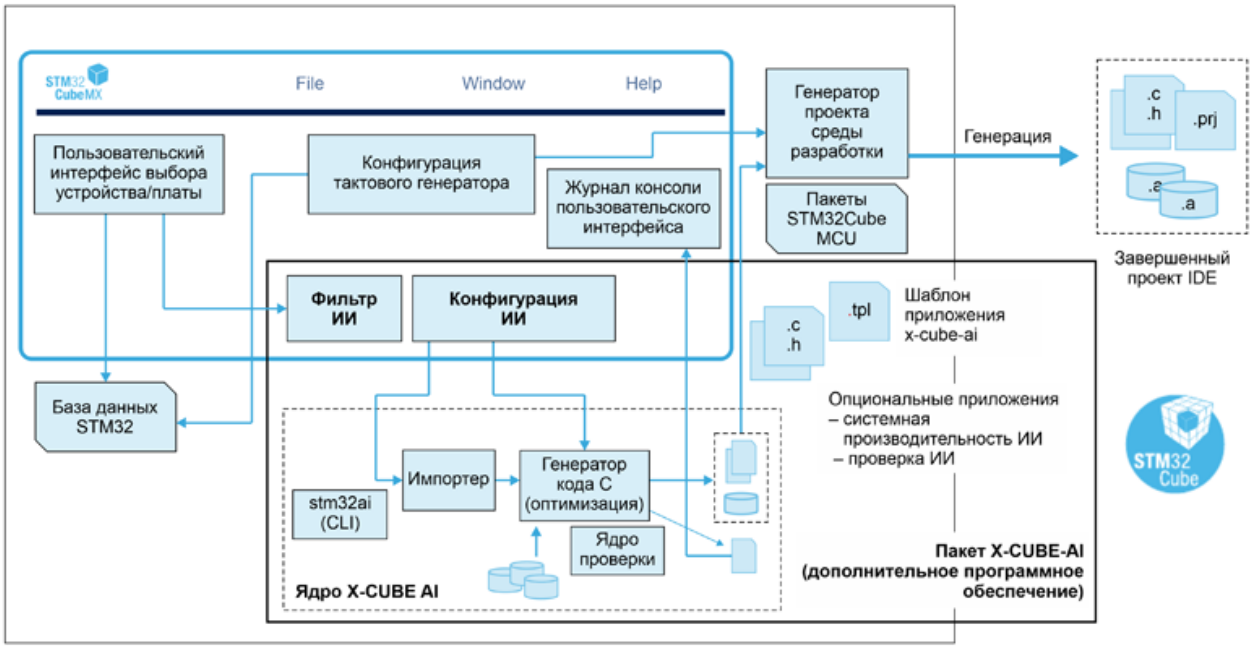

Рис. 6. Структура STM32CubeMX

Пример использования экосистемы X-CUBE-AI

В данном разделе рассмотрен практический пример проектирования нейронной сети на базе отладочной платы NUCLEO-G474RE.

Внешний вид платы показан на рисунке 7.

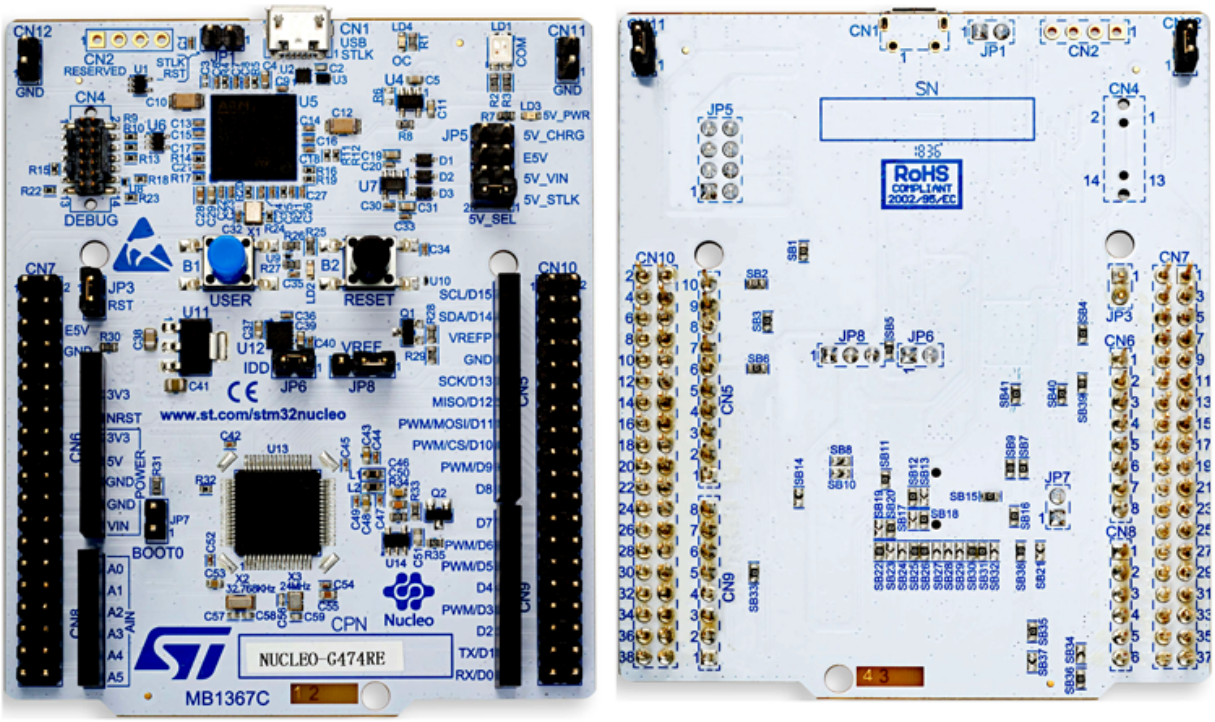

Рис. 7. Внешний вид платы NUCLEO-G474RE

Эта плата содержит микроконтроллер STM32G474RET6 в корпусе с 64 выводами. Максимальная частота ядра процессора – 170 МГц, объем Flash-памяти – 512 кбайт, объем статической памяти – 128 кбайт. Также плата содержит встроенный высокоскоростной отладчик/программатор ST-LINK/V3, что позволяет производить программирование и отладку платы без использования каких-либо внешних отладочных средств.

Микроконтроллеры семейства STM32G4 обладают высокой производительностью и богатым набором периферийных устройств. Также производитель предлагает широкий набор программных и отладочных средств для разработки, в том числе решений в области нейронных сетей, поэтому представитель именно этого семейства был выбран для практического рассмотрения.

Как было показано выше, для развертывания нейросети в микроконтроллере STM32 нужна готовая тренированная модель сети. Для данного примера будет использована нейросеть HAR-CNN-Keras, исходный код которой доступен для скачивания по ссылке: Это – сверточная нейросеть, написанная с использованием открытой нейросетевой библиотеки Keras. Сеть служит для распознавания нескольких элементов активности человека, таких как спуск по лестнице, пробежка, сидение, стояние, подъем по лестнице, пешая прогулка. Входными данными для сети являются показания нескольких акселерометров, расположенных в смартфоне и обеспечивающих 20 выборок данных в секунду.

Для загрузки модели нейросети переходим по ссылке, указанной выше, и в окне проекта кликаем мышкой на файл model.h5, как это показано на рисунке 8.

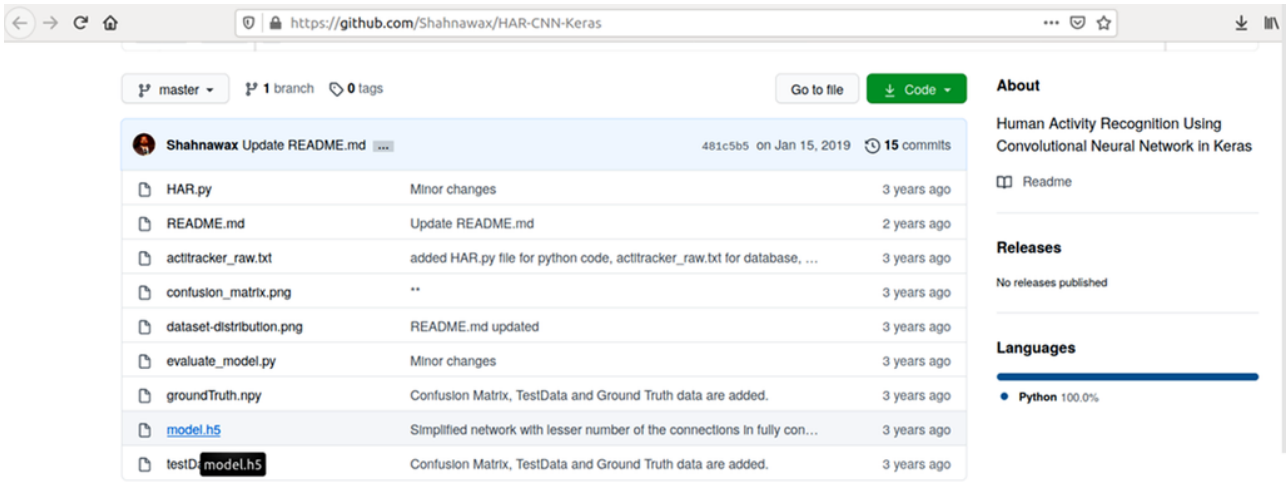

Рис. 8. Выбор модели нейросети

Далее нажимаем кнопку “Download” и сохраняем файл в удобном для работы месте (рисунок 9). Для целей этого примера файл модели переименован в «har_github.h5».

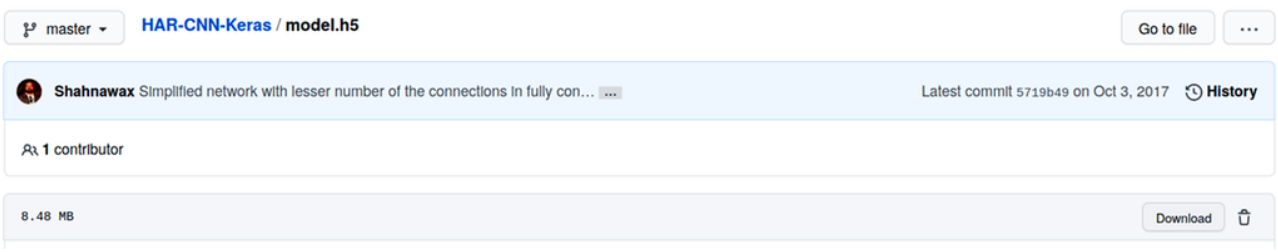

Рис. 9. Загрузка файла модели

Для программирования отладочной платы необходимо установить программное обеспечение. Для начала нужно установить STM32CubeMX версии 5.1.0 или новее. На момент написания статьи самой свежей является версия программы 6.0.0. STM32CubeMX доступен как для операционной системы Windows, так и для Linux. Заодно можно скачать пакет расширения X-CUBE-AI и среду STM32CubeIDE, которая будет необходима для компиляции сгенерированного кода.

После установки STM32CubeMX пакет X-CUBE-AI устанавливается следующим образом.

Запускается STM32CubeMX. В меню выбирается пункт «Help» → «Manage embedded software packages», можно также напрямую нажать кнопку «INSTALL/REMOVE», как это показано на рисунке 10.

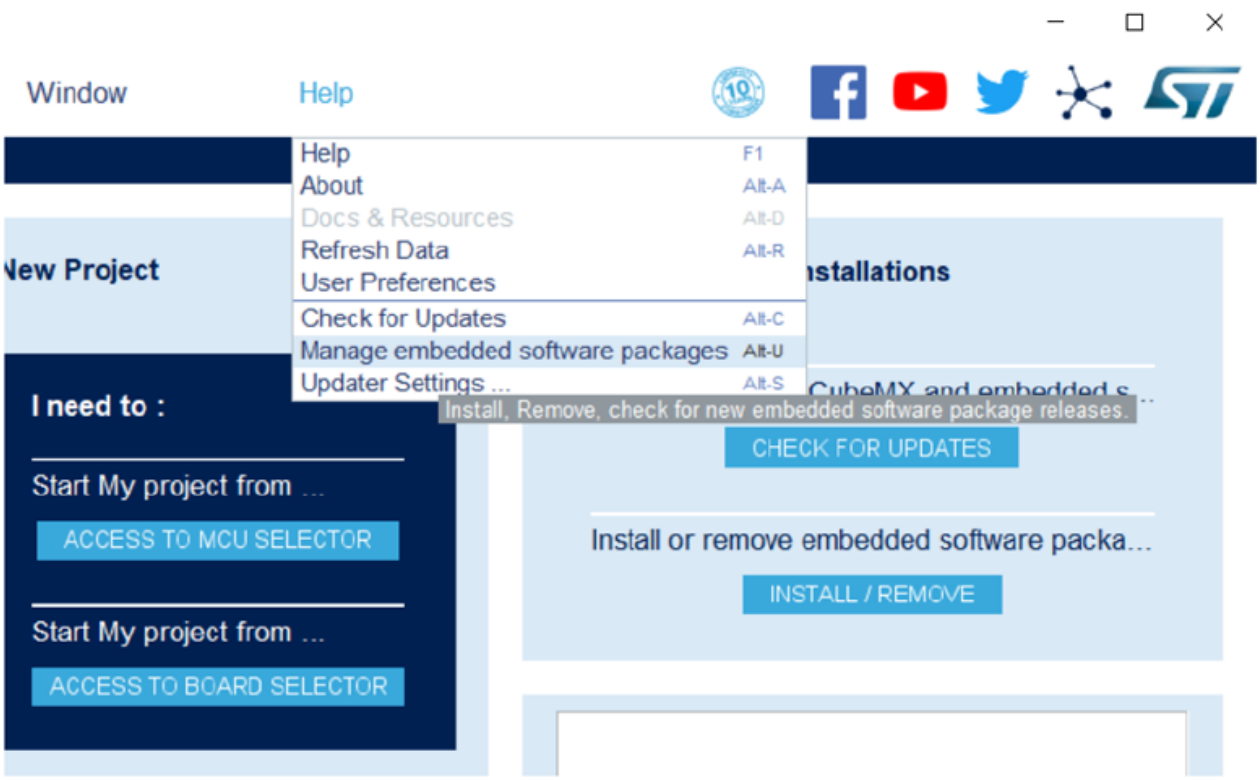

Рис. 10. Выбор встроенных пакетов в STM32CubeMX

В окне «Embedded Software Packages Manager» нажимаем кнопку «Refresh» для получения обновленного списка дополнительных модулей. Для поиска X-CUBE-AI выбираем вкладку «STMicroelectronics», как это показано на рисунке 11.

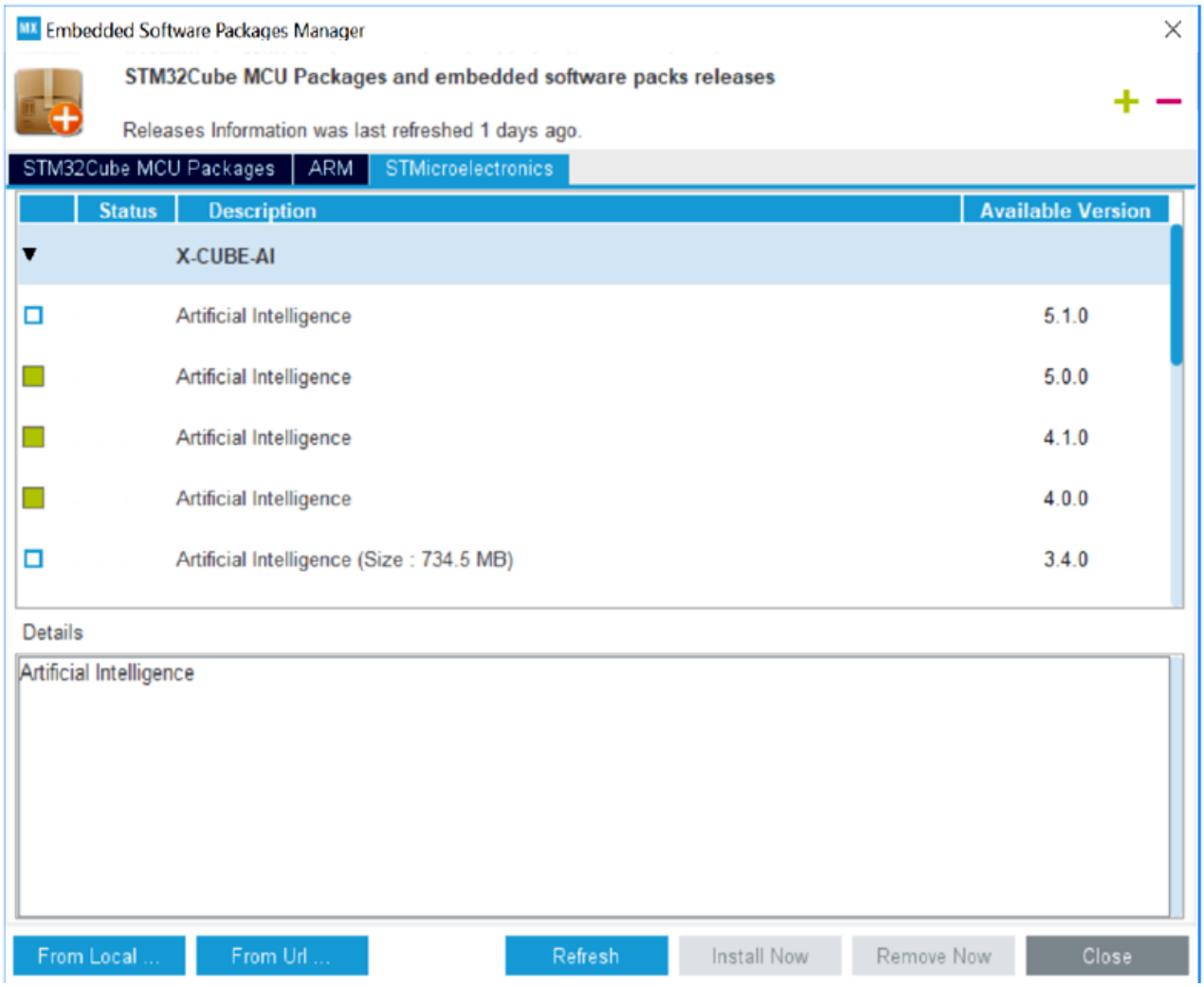

Рис. 11. Установка X-CUBE-AI в STM32CubeMX

Если X-CUBE-AI уже был установлен, желательно удалить его перед новой установкой.

Необходимая версия пакета выбирается и устанавливается путем нажатия кнопки «Install Now». После завершения установки квадрат рядом с необходимой версией пакета становится зеленым, после чего можно нажать кнопку «Close» (рисунок 12).

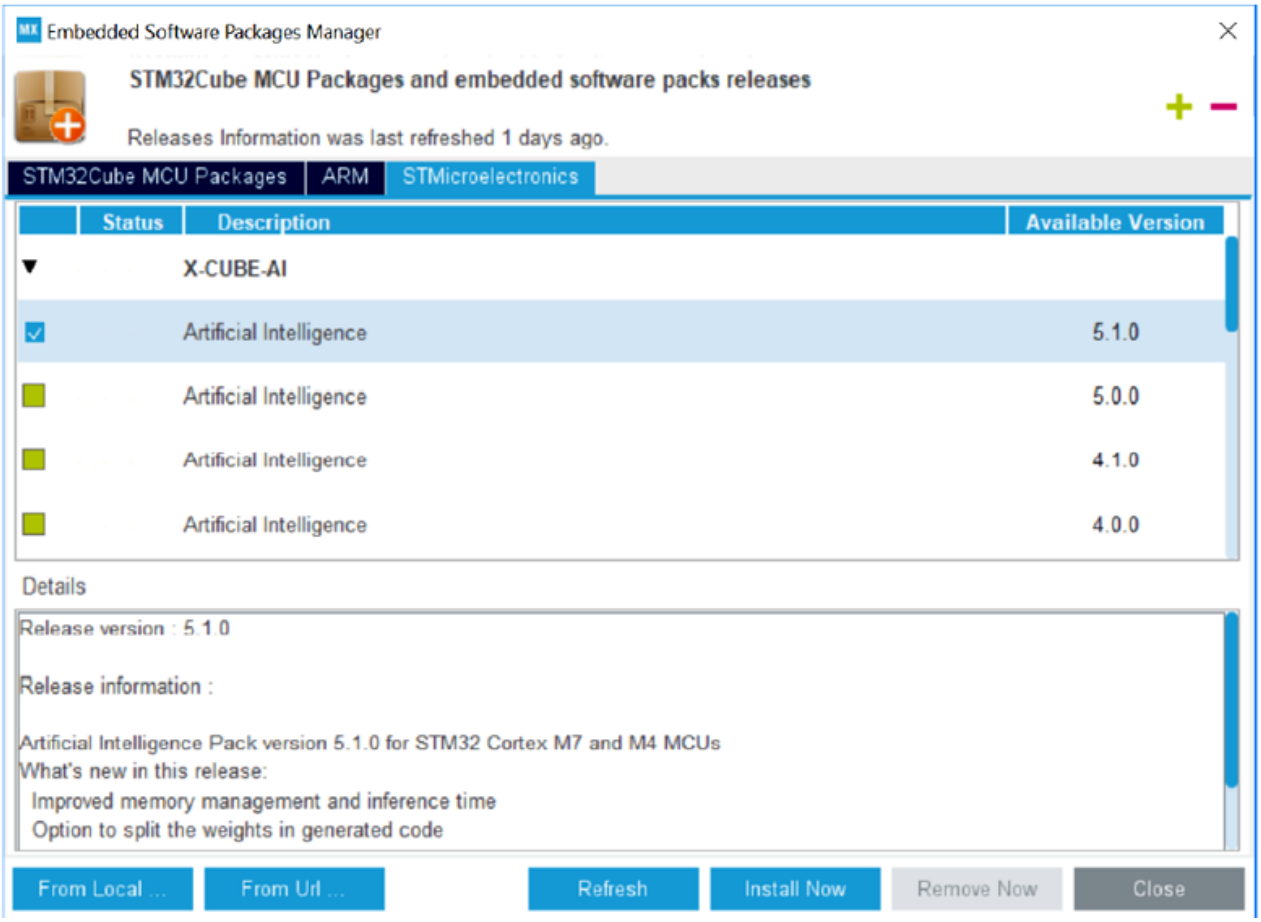

Рис. 12. X-CUBE-AI в STM32CubeMX

После установки и запуска приложения STM32CubeMX выбираем пункт «ACCESS TO BOARD SELECTOR» (рисунок 13).

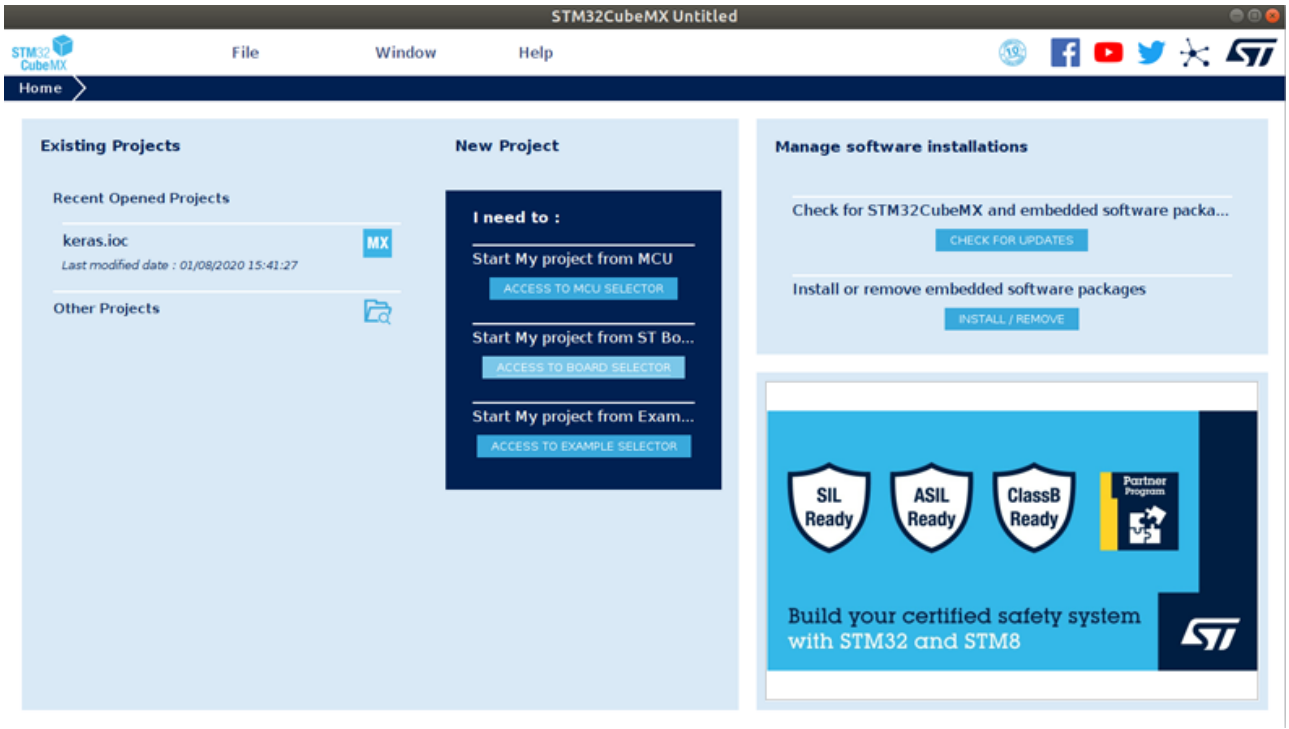

Рис. 13. Выбор отладочной платы NUCLEO-G474RE в STM32CubeMX

В открывшемся окне в поле «Commercial Part Number» набираем «NUCLEO-G474RE», в списке «Board List» выделяем необходимую отладочную плату (рисунок 14).

Рис. 14. Инициализация проекта

Двойным кликом запускаем проект (рисунок 15). При запуске проекта появляется сообщение “Initialize all peripherals with their Default Mode?”(«Инициализировать всю периферию в состояние по умолчанию?»). Отвечаем “Yes” («Да»).

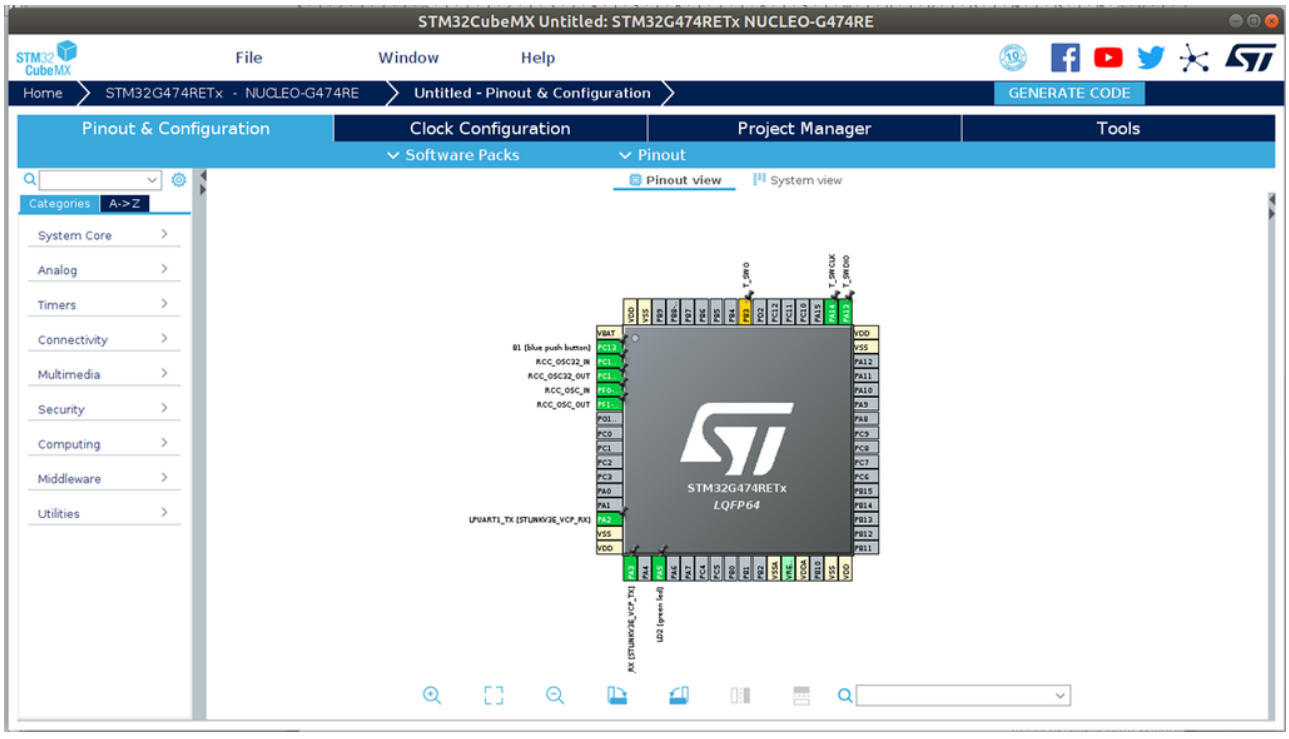

Рис. 15. Запуск проекта

Выводы контроллера также автоматически сконфигурированы согласно трассировке отладочной платы NUCLEO-G474RE.

Тактовую частоту проекта можно проверить и, при необходимости, изменить во вкладке “Clock configuration” («Настройка тактовой частоты»), как это показано на рисунке 16.

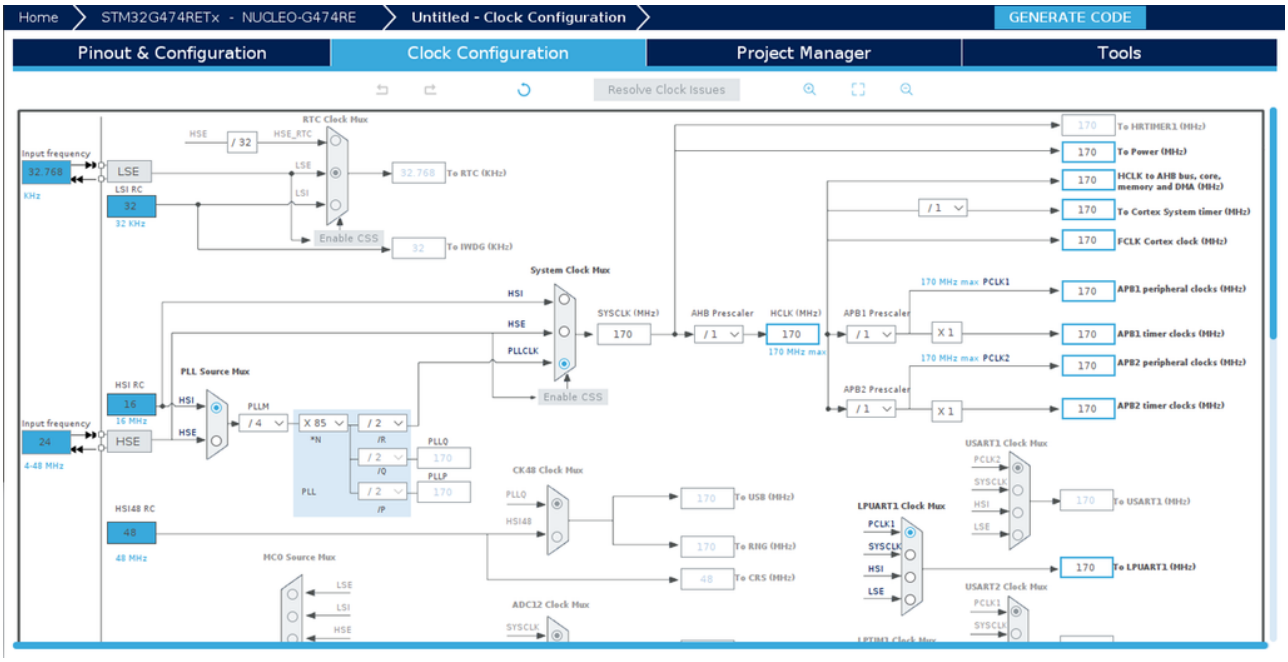

Рис. 16. Настройка тактовой частоты

Для отладочной платы NUCLEO-G474RE максимальная частота ядра – 170 МГц. Именно это значение (параметр «HCLK (MHz)») и было выбрано (рисунок 16).

Для обмена данными с компьютером используется порт LPUART1, основные настройки которого можно посмотреть, выбрав «Categories» à «Connectivity» à «LPUART1», как это показано на рисунке 17.

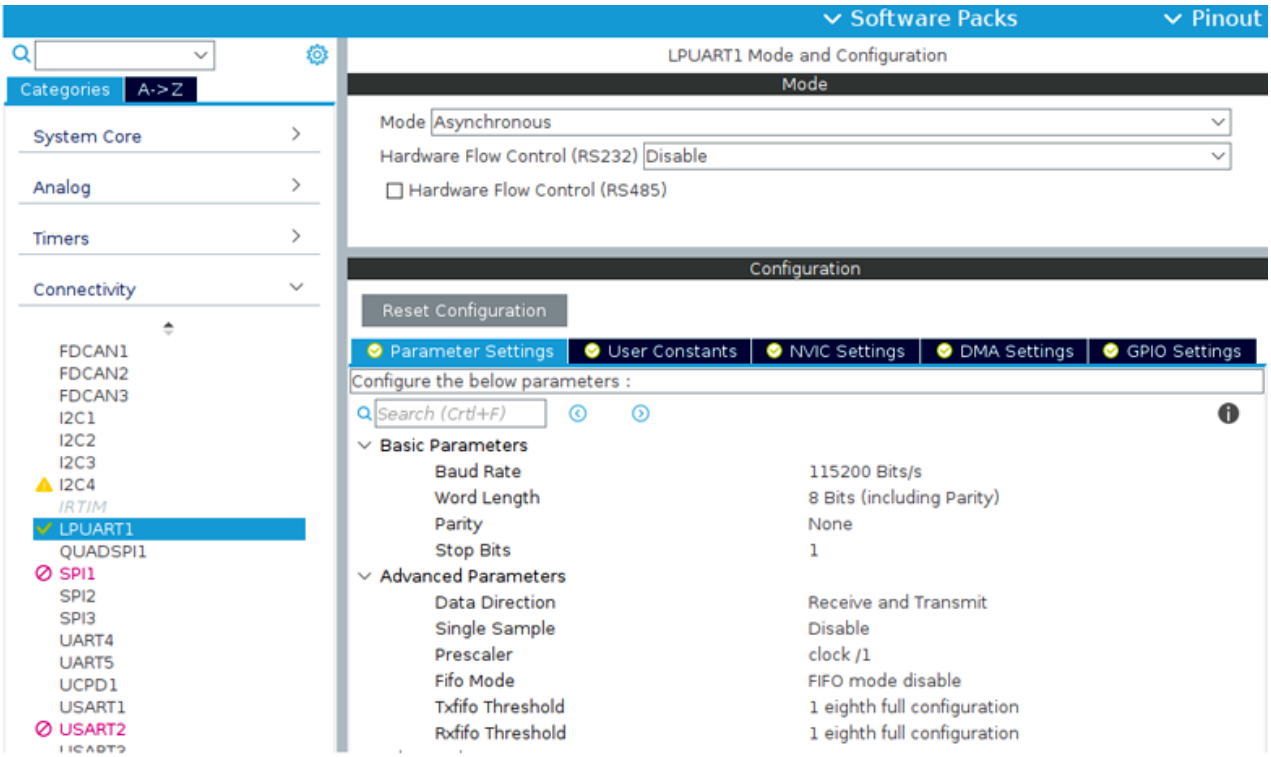

Рис. 17. Настройки порта UART

Далее необходимо подключить к проекту модуль X-CUBE-AI. Для этого выбираем в меню «Software Packs» → «Select components.» Затем в открывшемся окне выбираем «STMicroelectronics.X-CUBE-AI» → «Artificial Intelligence X-CUBE-AI» → «Core», и «Application» → «Application template», как это показано на рисунке 18.

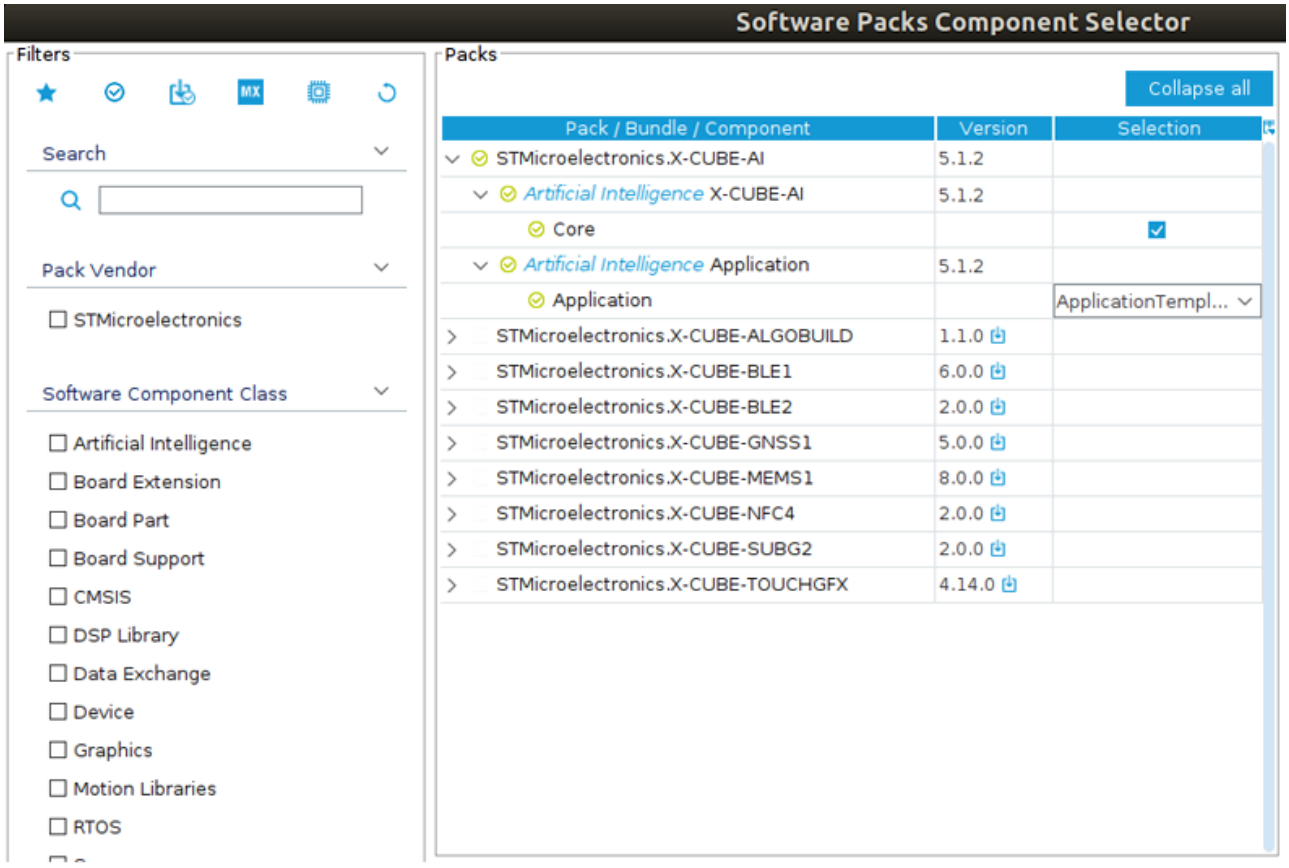

Рис. 18. Выбор модуля X-CUBE-AI

Существует несколько типов приложений, которые можно выбрать:

System performance – для проверки производительности реализаций различных нейросетей на целевой платформе;
Validation – для проверки производительности нейросетей и сравнения результатов вычислений;
Application template – шаблон, позволяющий создавать пользовательские приложения. Этот тип приложения используется в данном примере и также позволяет оценить производительность и сравнить результаты вычислений.
Далее выбираем «Categories» → «STMicroelectronics.X-CUBE-AI». В окне «Configuration» → Model inputs» выбираем название для модели нейросети – в данном примере это network (рисунок 19). Тип сети – «Keras», выбираем «Saved model», то есть используется заранее сохраненная сеть. Указываем путь до файла нейросети *.h5, который был ранее загружен. Выбираем степень сжатия модели – в данном случае 8, и нажимаем кнопку «Analyze». После окончания анализа проверяем вычислительные ресурсы, необходимые для реализации

данной сети с заданной степенью сжатия. Согласно рисунку 19, необходимо 367 кбайт Flash-памяти и 25 кбайт памяти RAM. В скобках рядом указаны размеры памяти выбранного микроконтроллера.

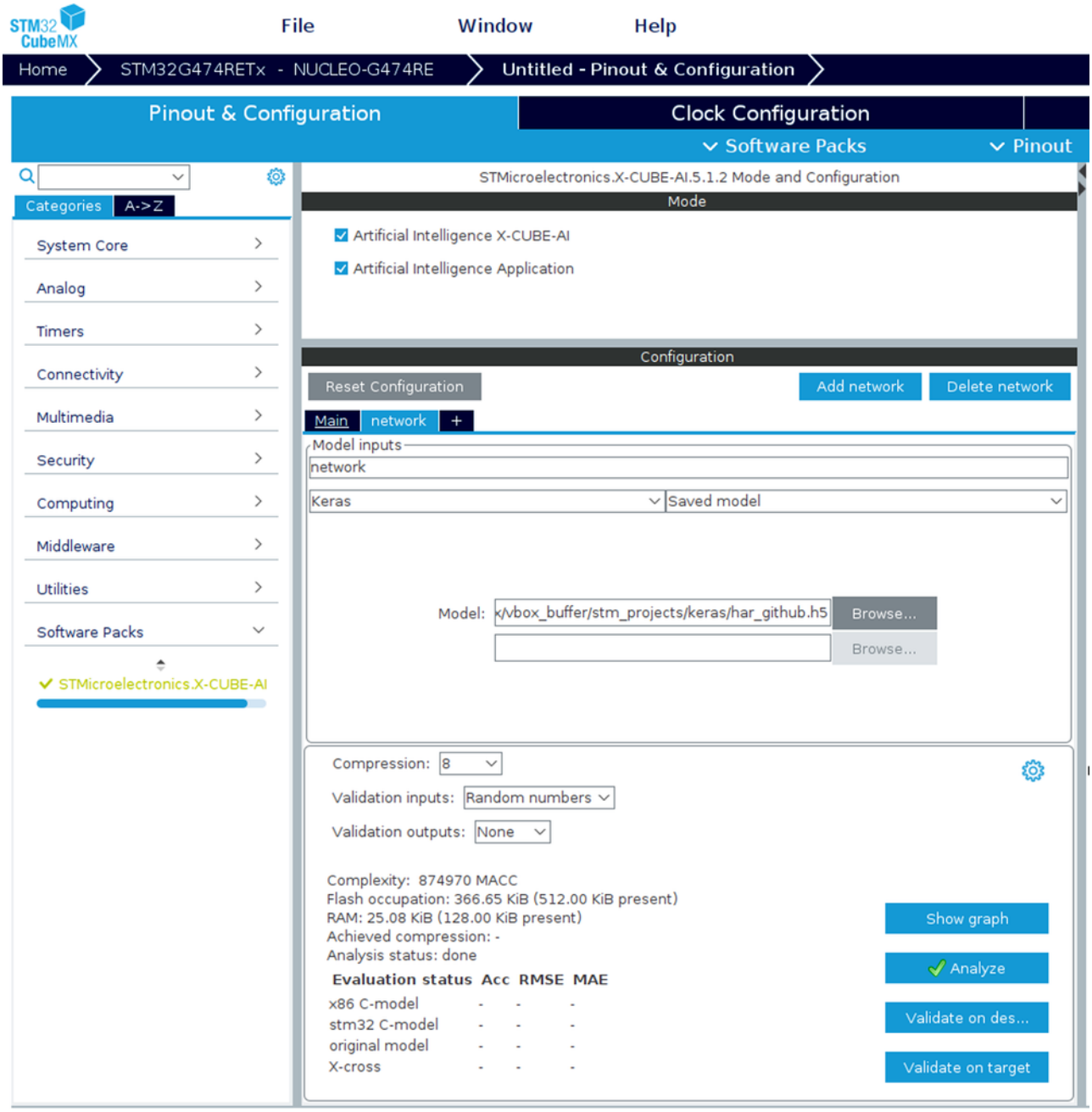

Рис. 19. Расчет вычислительных ресурсов

Коэффициент сжатия 8 был выбран из-за ограниченного объема памяти, так как модель с коэффициентом 4 уже не вместилась бы во Flash-память выбранного микроконтроллера.

Далее необходимо проверить, насколько точно код, сгенерированный для микроконтроллера, соответствует оригинальной модели нейросети. Для этого существует два вида тестов: на компьютере и на устройстве STM32 (рисунок 20).

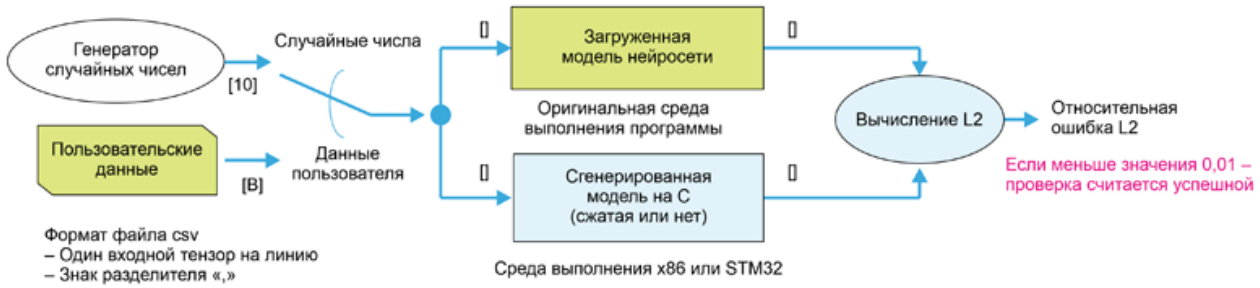

Рис. 20. Виды проверки нейросети

Модель нейросети, сгенерированная на языке C, запускается на выполнение на процессоре x86 или STM32, и результат выполнения сравнивается с оригинальной моделью. При ошибке L2 меньше 0,01 результат построения C-модели считается успешным. Большое влияние на ошибку оказывает степень сжатия.

Сначала проверяется модель на компьютере в среде x86. После нажатия кнопки «Validate on desktop» открывается окно, отображающее процесс проверки. В данном случае в качестве входных данных используются случайные числа. Результат выполнения проверки на компьютере показан на рисунке 21.

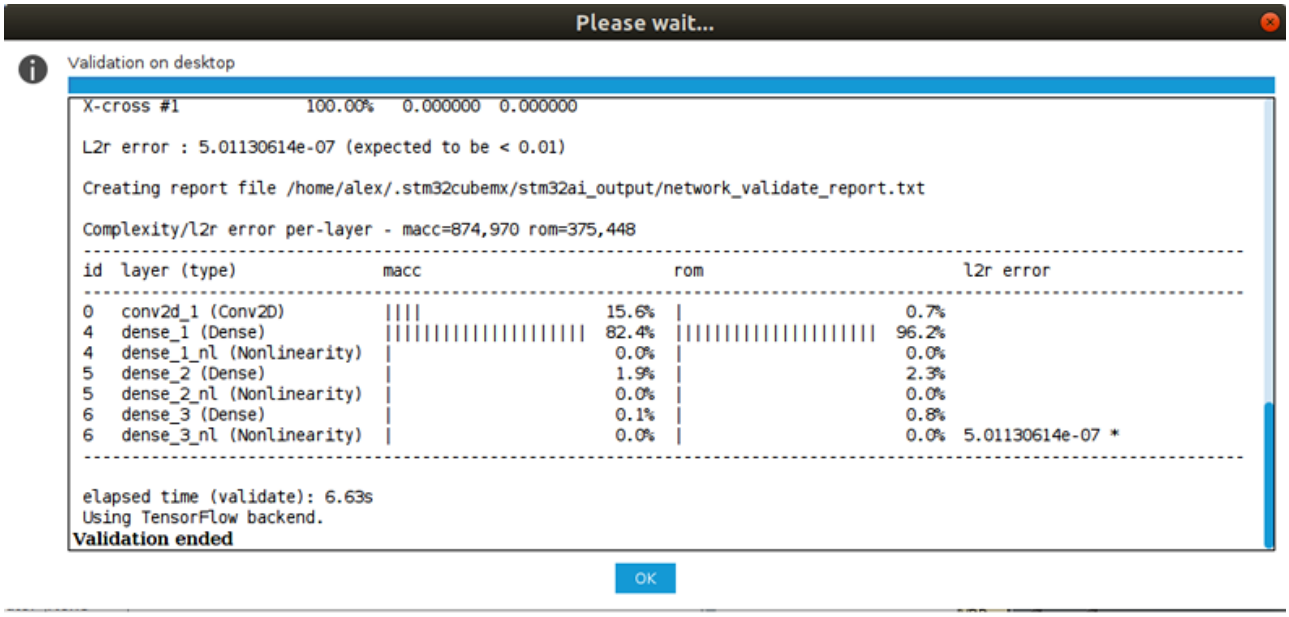

Рис. 21. Результат проверки модели на компьютере

Ошибка L2 = 5,01e-07, что значительно меньше допустимой погрешности 0,01. Также виден процент использования вычислительных ресурсов разными слоями нейросети.

Для проверки модели на STM32 нажимаем кнопку «Validate on target». При этом открывается окно конфигурации проверки (рисунок 22).

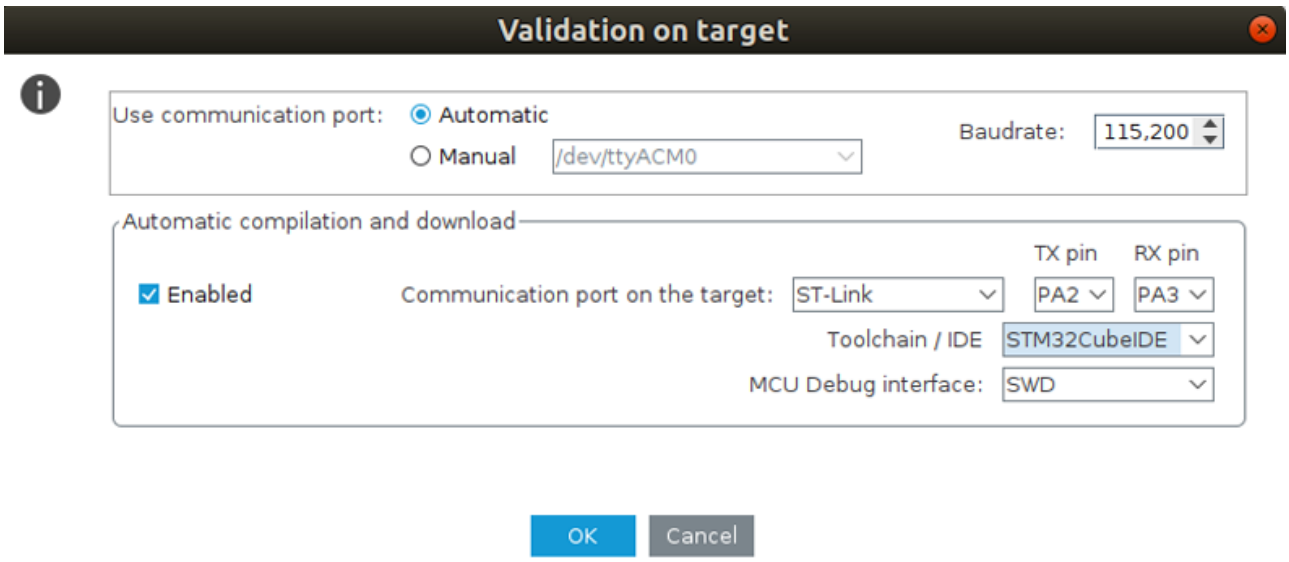

Рис. 22. Настройка проверки модели в STM32

Имя порта зависит от операционной системы. В данном примере использовалась система Ubuntu18.04, где порт для связи с отладочной платой получил автоматическое название /dev/ttyACM0. В операционной системе Windows это будут названия типа COM1, COM2 и так далее. Скорость 115200 соответствует настройкам проекта. Для проверки модели в микроконтроллере необходима среда (Toolchain/IDE) для компиляции исходных кодов, сгенерированных с помощью STM32CubeMX. В данном случае для компиляции используется среда STM32CubeIDE, которая была предварительно установлена на компьютер.

После выполнения необходимых настроек нажимаем «OK» и ожидаем выполнения теста, который включает в себя компиляцию проекта, прошивку платы, выполнение теста, анализ полученных результатов. На рисунке 23 показан результат выполнения теста.

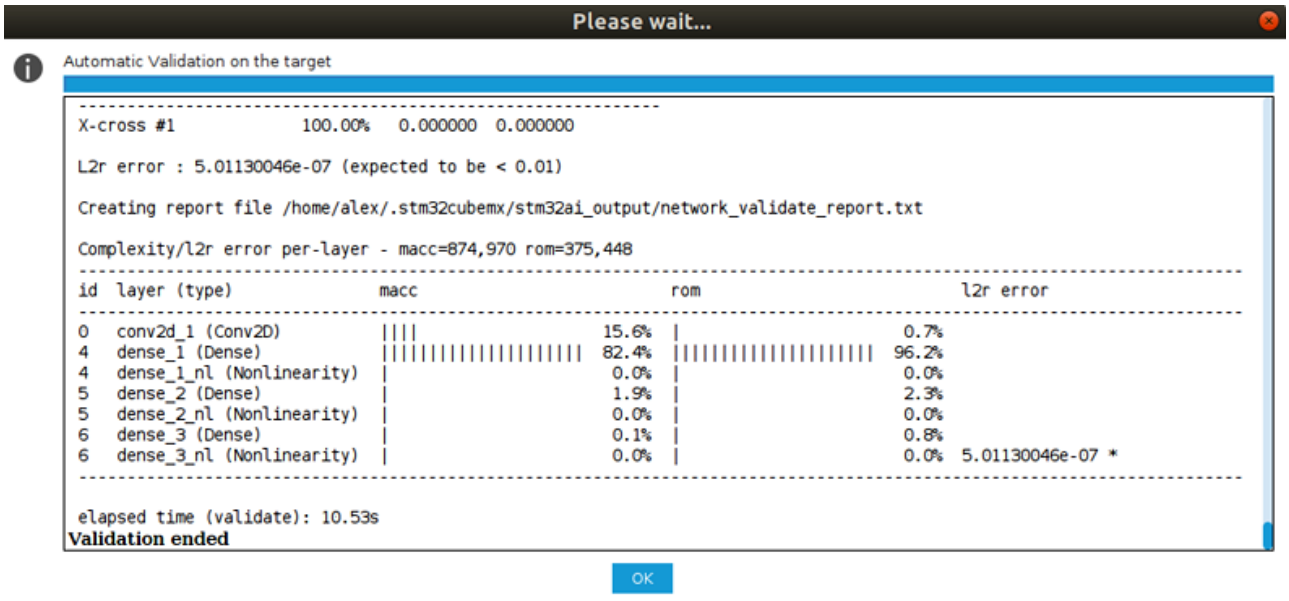

Рис. 23. Результат выполнения теста в STM32

Ошибка L2 в этом случае также значительно меньше 0,01, а значит модель, сгенерированная для микроконтроллера STM32, соответствует оригинальной.

Подведем итог

Искусственные нейронные сети достаточно сложны в реализации. Для их практического применения очень важны инструменты, облегчающие и ускоряющие процесс создания прикладных приложений с их использованием. Компания STMicroelectronics предлагает широкий набор программных инструментов, значительно облегчающих процесс создания приложений на базе искусственных нейросетей. В совокупности с аппаратными отладочными средствами это позволяет значительно сократить цикл разработки устройств и в кратчайшие сроки получить конкурентоспособный продукт. Специалисты компании КОМПЭЛ всегда рады помочь в выборе правильных ресурсов и инструментов для решения задач построения нейросетей на базе микроконтроллеров семейства STM32G4.

•••