



پیاده سازی یک مدل Deep Learning بر روی میکرو های STM32

هدف از اجرای این پروژه قرار دادن یک مدل **fully connected** بر روی میکرو کنترلر STM32 است که بر روی دیتا ست **fashion mnist** پیاده شده باشد و سپس ارسال یک داده از میان داده های تست از روی PC بر روی میکرو کنترلر از روی پورت سریال و سپس بازگشت کلاس پیش بینی شده از سمت میکرو کنترلر بر روی PC از روی پورت سریال و نشان دادن جواب بر روی ترمینال سریال است که می توانید برنامه ای مثل Tera Term VT باشد و یا از پایتون و مائزول **pyserial** انجام شود. (برای ارتباط PC با میکرو کنترلر می توانید از یک مبدل **usb2serial** مثل **cp2102** استفاده نمایید)

می بایست ابتدا با استفاده از PC می بایست یک مدل با استفاده از **keras** از کتابخانه **TensorFlow** آموزش دهید ، سپس با استفاده از **X-CUBE-AI** که به صورت یک **extension** بر روی **cubemx** قابل نصب است آن را بر روی میکرو بگذارید ، در نهایت هم با استفاده از **AI validation application** بر روی مدل **inference** انجام دهید .

آشنایی با Deep Learning

شبکه های عصبی شامل تعداد زیادی نورون هستند که با تعدادی وزن به یکدیگر متصل هستند . اگر تعدادی شبکه ی عصبی بر روی هم قرار گیرند یک شبکه ی **Deep** تشکیل می دهند . یک شبکه **Deep** نیاز به پروسه ای به نام آموزش دارد که در نتیجه ی آن شبکه یاد میگیرد عمل خاصی را انجام دهد . به عنوان مثال یک تصویر را دریافت می کنند که آن تصویر داخل مجموعه ای از دیگر عکس هاست که به این مجموعه **dataset** می گویند و در خروج تشخیص می دهد این تصویر در چه کلاسی قرار می گیرد . امروزه از شبکه های عمیق برای انواع کار ها استفاده می شود ، از پیش بینی سهام بورس گرفته تا کنترل شبکه ی توزیع برق . انواع متفاوتی از این شبکه ها وجود دارد مانند شبکه های کانولوشنی ، **AUTO-ENCODER** ها ، **GAN** ها و

...

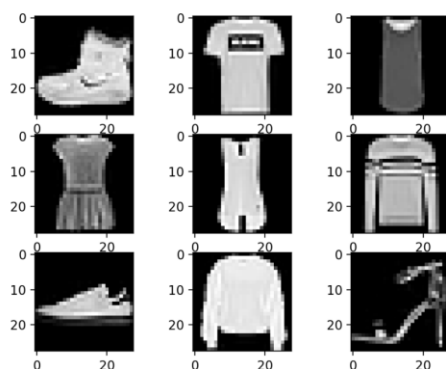
ساده ترین نوع این شبکه ها نوع Fully Connected است که همانطور که از نام آن پیداست تمامی نورون ها در یک طبقه تک به تک به لایه های بالایی و پایینی خود متصل هستند و با هم ارتباط دارند .

دیتاست fashion-mnist حاوی مجموعه عکس هایی سیاه و سفیدند که سایز آن ها ۲۸ در ۲۸ پیکسل است که مربوط به چند کلاس پوشش می باشد که به صورت زیر هستند .

T-shirt/top ,Trouser ,Pullover ,Dress ,Coat ,Sandal ,Shirt ,Sneaker ,Bag
,Ankle boot

که ترتیب شماره کلاس آن ها از چپ به راست است .

نمونه تصویر از این دیتاست به صورت زیر است :



در صورت انجام inference در خروجی شبکه عصبی یک بردار به صورت زیر دریافت می کنید که منظور از آن این است که شبکه حدس زده است کلاس تصویر وارد شده با چه احتمالی متعلق به هر کدام از کلاس ها است .

[.1, .1, .6, .05, .01, .02, .03, .03, .04, .02]

در نتیجه شبکه حدس زده است که تصویر متعلق به کلاس Pullover است .

یک دیتاست به دو قسمت تست و آموزش تقسیم می شود که از آموزش برای دادن آموزش به شبکه و از تست برای تعیین میزان صحیح بودن حدس شبکه در مورد کلاس تصویر استفاده می شود که به درصد بیان می شود .

یک نمونه کد برای آموزش این شبکه در پایتون به صورت زیر است :

https://github.com/aminzakizebarjad/fashion_mnist/blob/main/fashion_mnist.py

برای انجام نصب مازول تنسورفلو پایتون می توانید از لینک

<https://pypi.org/project/tensorflow/#files> استفاده کنید و فایل wheel را به صورت مستقیم

دانلود کنید که باید به سیستم عامل و ورژن پایتون هم دقت داشته باشید و با pip آن را نصب کنید و یا این که مستقیماً با pip دانلود و نصب کنید. قابل ذکر است که cuda و cudnn7 و cudnn8 هم باید نصب داشته باشید. دقت کنید نسخه ی cuda با نسخه ی tensorflow همخوانی داشته باشد.

پیاده سازی بر روی STM32

همان طور که در ابتدا اشاره شد، CUBEMX دارای یک افزونه به نام X-CUBE-AI است که برای راهنمایی نحوه ی کار با آن می توانید از لینک

https://my.st.com/resource/en/user_manual/dm00570145-getting-started-with-xcubeai-expansion-package-for-artificial-intelligence-ai-stmicroelectronics.pdf

پرونده را دانلود کنید و خود آن را از داخل cubeMX نصب کنید و یا مستقیماً از لینک

https://my.st.com/content/my_st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-ai.license=1624004653224.product=X-CUBE-AI-Linux.version=5.2.0.html

را استفاده کنید. این افزونه به صورت داخلی می تواند یک پورت UART را

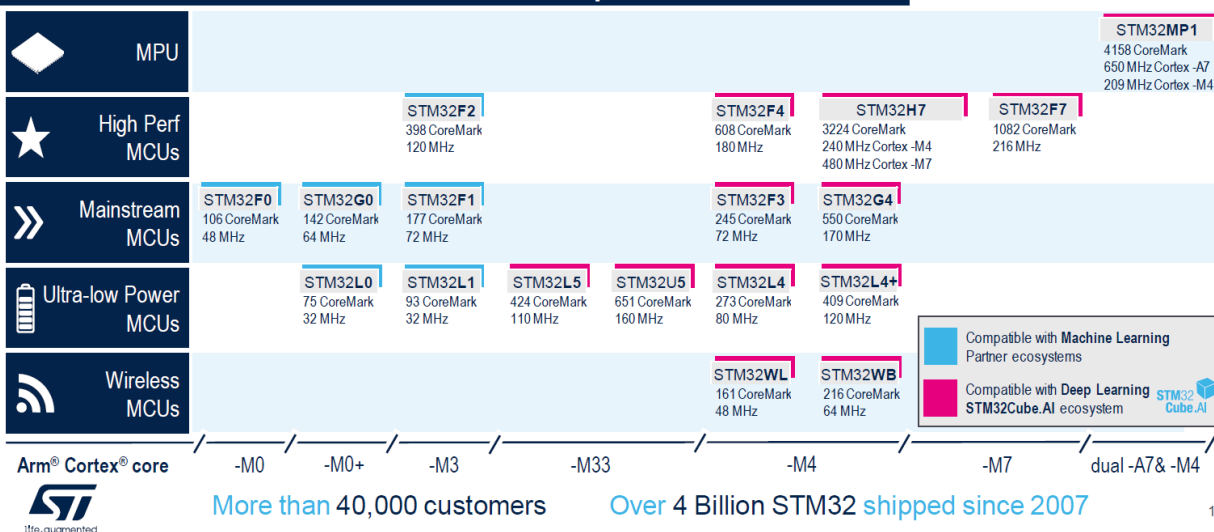
برای انجام inference اختصاص دهد که باید پیکسل های تصویر را بر روی این پورت می توانید انتقال دهید. این کار را می توانید با استفاده از pyserial انجام دهید.

دقت کنید هر میکرویی از STM32 برای پیاده سازی مدل شبکه ی deep مناسب نیست، در انتخاب آن دقت کنید.

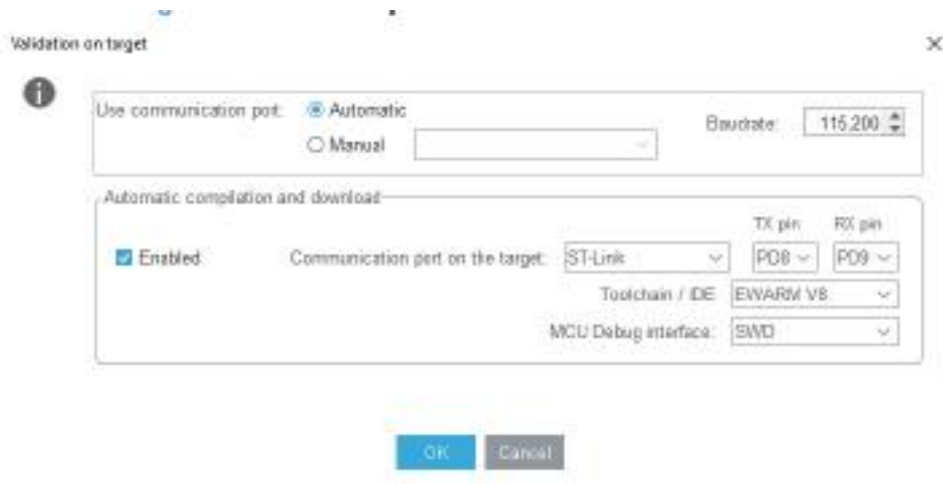


Making AI Accessible Now

Leader in Arm® Cortex®-M 32-bit General Purpose MCU



برای انجام inference از AI validation application استفاده کنید که داخل cubemx قابل دسترسی و تنظیم است.



گزارش

در نهایت شما بایست ۱۰ عدد inference انجام دهید و جواب آن را بر روی tera term یا در صورت استفاده از ترمینال pyserial برای ارسال پیکسل ها، با استفاده از پایتون بردار جواب inference که یک بردار تک بعدی دارای ۱۰ مولفه است دریافت کنید.

فایل فشرده ی تحویلی بایست حاوی کد های پایتون آموزش و تست شبکه به صورت فایل **py** فایل انجام inference با استفاده از **pyserial** به صورت **py** ، فایل پروژه ی **CUBEMX** به صورت پسوند **ioc** و فایل پروژه ی **keil** و یک فایل **pdf** که گزارش قدم به قدم از آموزش شبکه ، تست آن و نحوه ی قرار دادن شبکه بر روی میکرو و انجام **inference** باشد.

گروه ها می توانند به صورت دو نفره باشند .

موفق باشید.