

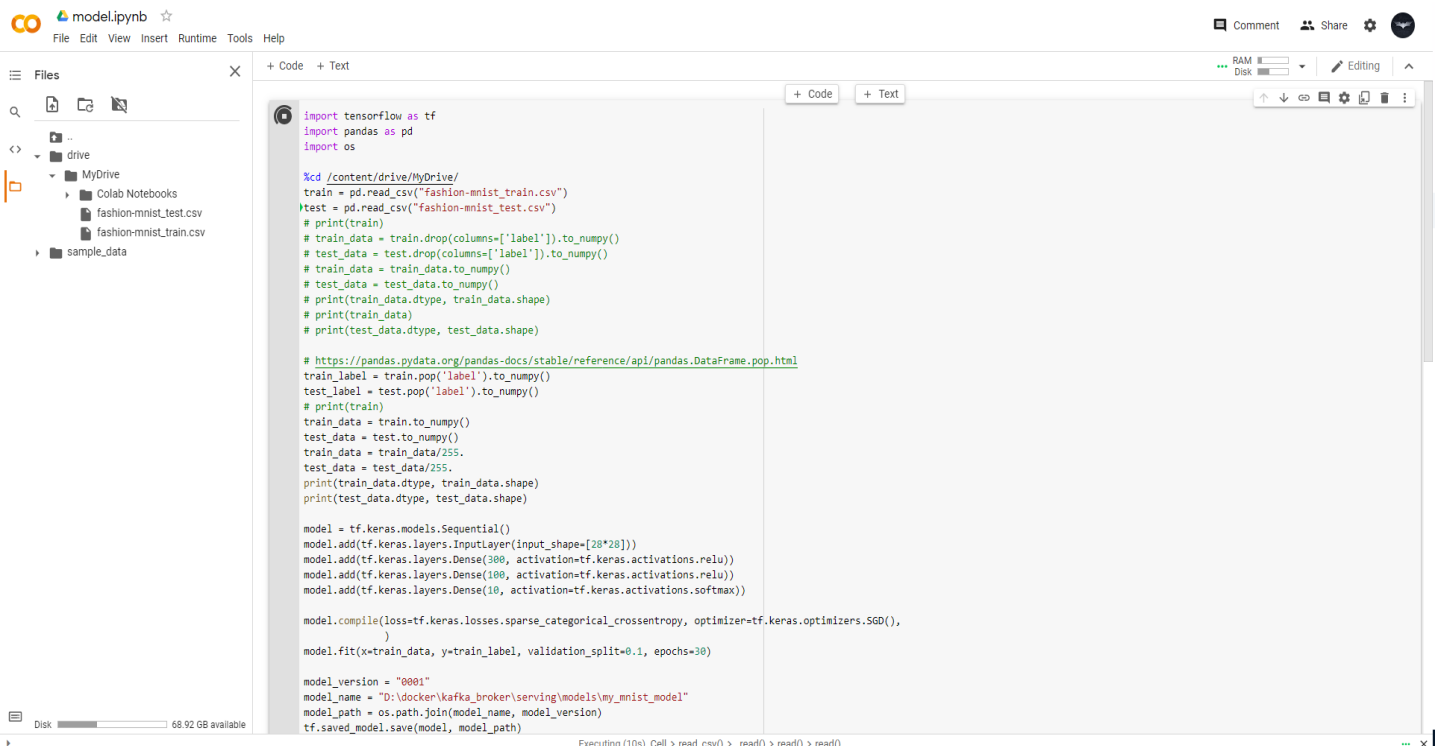
گزارش کار پروژه پایانی درس میکرو

سینا حیدری

9723116

شرح گزارش :

در ابتدای کار ما باید بریم به سراغ اینکه COLAB که بتواند فایل ما را پردازش کند در برنامه ی COLAB ما ابتدا در بخش Google drive دو فایل fashion-mnist_train.csv و fashion-mnist_test.csv را میریزیم تا بتوان فایل را برای ما تحلیل کند. و کد هم همان کد پایتونی است که استاد برای ما قرار داده است مانند شکل زیر :



```
import tensorflow as tf
import pandas as pd
import os

%cd /content/drive/MyDrive/
train = pd.read_csv("fashion-mnist_train.csv")
test = pd.read_csv("fashion-mnist_test.csv")
# print(train)
# train_data = train.drop(columns=['label']).to_numpy()
# test_data = test.drop(columns=['label']).to_numpy()
# train_data = train_data.to_numpy()
# test_data = test_data.to_numpy()
# print(train_data.dtype, train_data.shape)
# print(train_data)
# print(test_data.dtype, test_data.shape)

# https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.pop.html
train_label = train.pop('label').to_numpy()
test_label = test.pop('label').to_numpy()
# print(train)
train_data = train.to_numpy()
test_data = test.to_numpy()
train_data = train_data/255.
test_data = test_data/255.
print(train_data.dtype, train_data.shape)
print(test_data.dtype, test_data.shape)

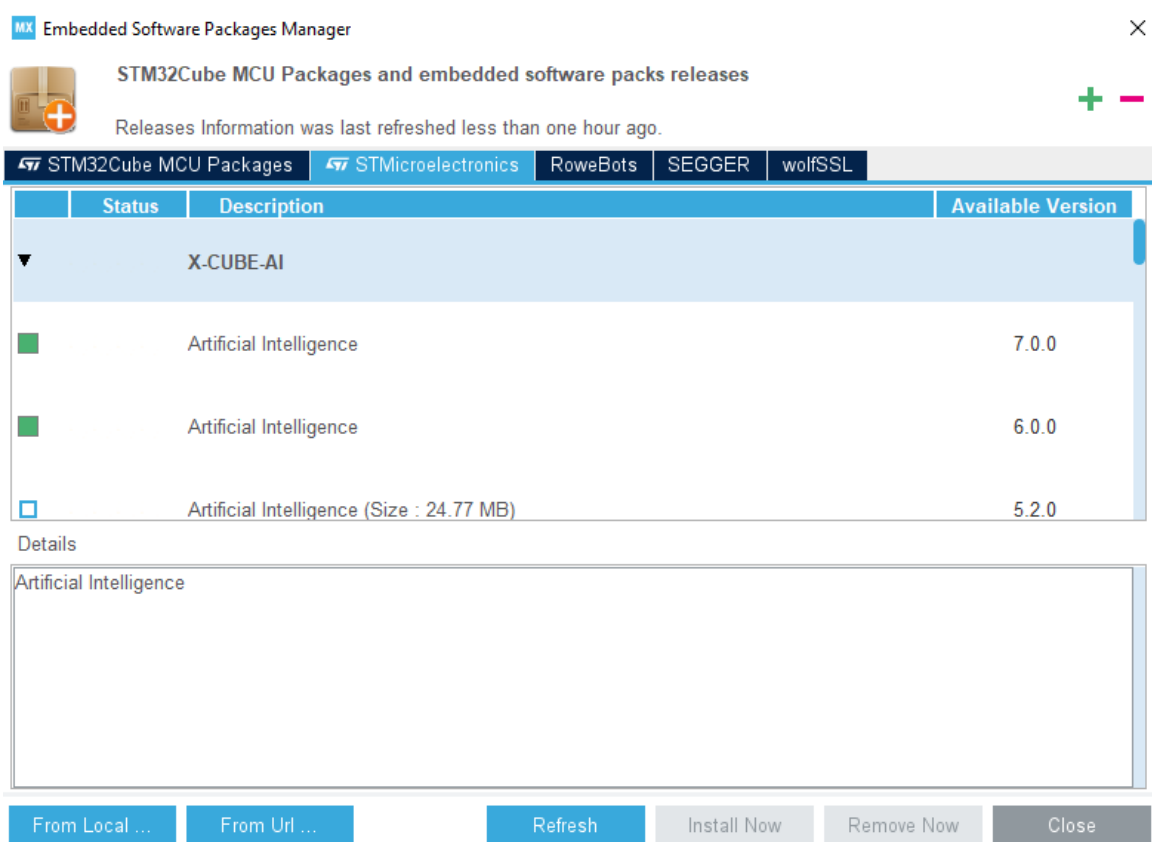
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.InputLayer(input_shape=[28*28]))
model.add(tf.keras.layers.Dense(300, activation=tf.keras.activations.relu))
model.add(tf.keras.layers.Dense(100, activation=tf.keras.activations.relu))
model.add(tf.keras.layers.Dense(10, activation=tf.keras.activations.softmax))

model.compile(loss=tf.keras.losses.sparse_categorical_crossentropy, optimizer=tf.keras.optimizers.SGD(),
              )
model.fit(x=train_data, y=train_label, validation_split=0.1, epochs=30)

model_version = "0001"
model_name = "D:\docker\kafka_broker\serving\models\my_mnist_model"
model_path = os.path.join(model_name, model_version)
tf.saved_model.save(model, model_path)
```

سپس خروجی به ما یک فایل model.h5 میدهد که آن را باید به Cube بدهیم.

سپس در خود برنامه ی Cube و در بخش install/remove ما Ai را نصب میکنیم و به رنگ سبز در میآید



Artificial Intelligence

☒ Enable

Model

Type

Model

Compression

و سپس در بخش new project پروژه ای جدید درست میکنیم و برای تنظیمات آن در بخش Ai آن را با شکل زیر درست میکنیم

و بعد از آنالیز به ما میکرو هایی را پیشنهاد میکند که ما میکرو STM32F407ZG استفاده میکنیم. و بعد شروع پروژه را میزنیم تا برای ما میکرو را بالا بیاورد

در بخش تنظیمات software Cube ما این تنظیمات را اوکی میکنیم

Package	Status	Version	Selection
> RoweBots.I-CUBE-UNISONRTOS	<input type="checkbox"/>	5.5.0-4	<input type="button" value="Install"/>
> SEGGER.I-CUBE-embOS	<input type="checkbox"/>	1.2.0	<input type="button" value="Install"/>
▼ STMicroelectronics.X-CUBE-AI	<input checked="" type="checkbox"/>	7.0.0	
▼ Artificial Intelligence X-CUBE-AI	<input checked="" type="checkbox"/>	7.0.0	
Core	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
▼ Device Application	<input checked="" type="checkbox"/>	7.0.0	
Application	<input checked="" type="checkbox"/>		<input type="text" value="Applicatio..."/>
> STMicroelectronics.X-CUBE-ALGOBUILD	<input type="checkbox"/>	1.2.0	

و بعد در بخش Software packs میریم داخل بخش STM و در تب نتوورک اونجا مدل ما داده شده و آنالیز میکنیم و Validate on Desktop را نیز میزنیم و مانند شکل زیر است :

Configuration

Reset Configuration Add network Delete network

Main network_1626603565830 +

Model: C:\Users\Sina\Desktop\Uni\Micro\model.h5 Browse... Browse...

Compression: 4

Validation inputs: Random numbers

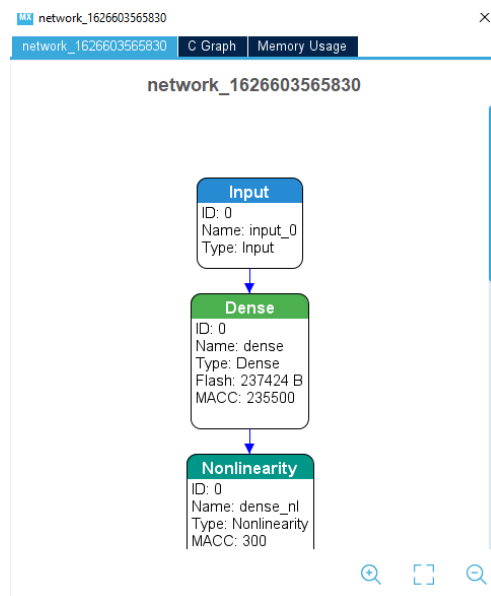
Validation outputs: None

Complexity: 267160 MACC
Flash occupation: 266.49 KiB (1024.00 KiB present)
RAM: 4.66 KiB (320.00 KiB present)
Achieved compression: 3.94
Analysis status: done

Show graph
Analyze
Validate on desktop
Validate on target

Validation metrics
(No data to display, please perform a validation first)

و بعد روس show graph میزنیم و گراف زیر بالا می آید :



بعد پورت Usart3 را تنظیم میکنم و خود میکرو پورت CRC را چون بخش هوش مصنوعی بهش نیازمند است فعال کرده است :

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

☐ Hardware Flow Control (RS485)

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Search Signals

Search (Ctrl+F)

☐ Show only Modified

Pin Name	Signal on Pin	GPIO output...	GPIO mode	GPIO Pull-u...	Maximum o...	User Label	Mod
PB10	USART3_TX	n/a	Alternate Fu...	No pull-up a...	Very High		<input type="checkbox"/>
PB11	USART3_RX	n/a	Alternate Fu...	No pull-up a...	Very High		<input type="checkbox"/>

و سپس بعد از درست کردن تنظیمات در برنامه CubeMX مد را برای ARM و برنامه ی Keil میسازیم. و پس از config کردن بخش Cube به سراغ بخش Python میرویم که با پورت سریال داده میخواهد بفرستد

قبل از پایتون چون برنامه Cube با کراس کار نمیکند یعنی کدی generate نمیشود پس باید فایل را از h5 به tflite تبدیل کنیم که هر دو فایل در فولدر پروژه وجود دارد و برای تبدیل h5 به tflite از کد زیر استفاده میکنیم

```
import tensorflow as tf

model = tf.keras.models.load_model('model.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
open("converted_model.tflite", "wb").write(tflite_model)
```

و چون فایل هوش مصنوعی ما عوض شده است پس باید مراحل بالا را دوباره ببریم و با همان میکرو و دقیقا همه چی را دوباره استفاده میکنیم و اگر دوباره آنالیز کنیم دوباره دقیقا همان گراف را میدهد به ما و اگر این دفعه کد را generate کنیم فایل به درستی generate میشود و بخش Cube آن به پایان میرسد و چون دقیقا همان مراحل بالا را برای h5 انجام داده ایم دیگر برای tflite تکرارش نکرده ام چون دقیقا تمامی مراحل آن یکی است و فقط به جای KERAS از TFLite استفاده میکنیم و به جای فایل H5 از فایل tflite استفاده میکنیم

حال به سراغ کد پایتون میرویم :

```
import serial

data[10]

ser = serial.Serial(    port="COM4", baudrate=9600, bytesize=8, timeout=2, stopbits=serial.STOPBITS_ONE)
ser.open()
for i in range (10)
    ser.write(data[i])
    sleep(1)
ser.close()
```

در اینجا اول سریالو ایمپورت میکنیم و سپس با فرض اینکه دیتا ها در آرایه دیتا هستند و چون میکرو نداریم که پردازش کند دیتا ها را و سپس init میکنیم سریال را و پس از init کردن باید سریال را باز کنیم و چون آرایه ما دارای 10 دیتا است نیازمند یک for است تا بتوان 10 تا دیتا را بتوان بفرستیم و سپس با استفاده از ser.write آن را در پورت سریال مینویسیم و بعد برای قاطی نشدن دیتا ها sleep استفاده میکنیم و بعد از فرستادن دیتا ها پورت سریال را میبندیم