

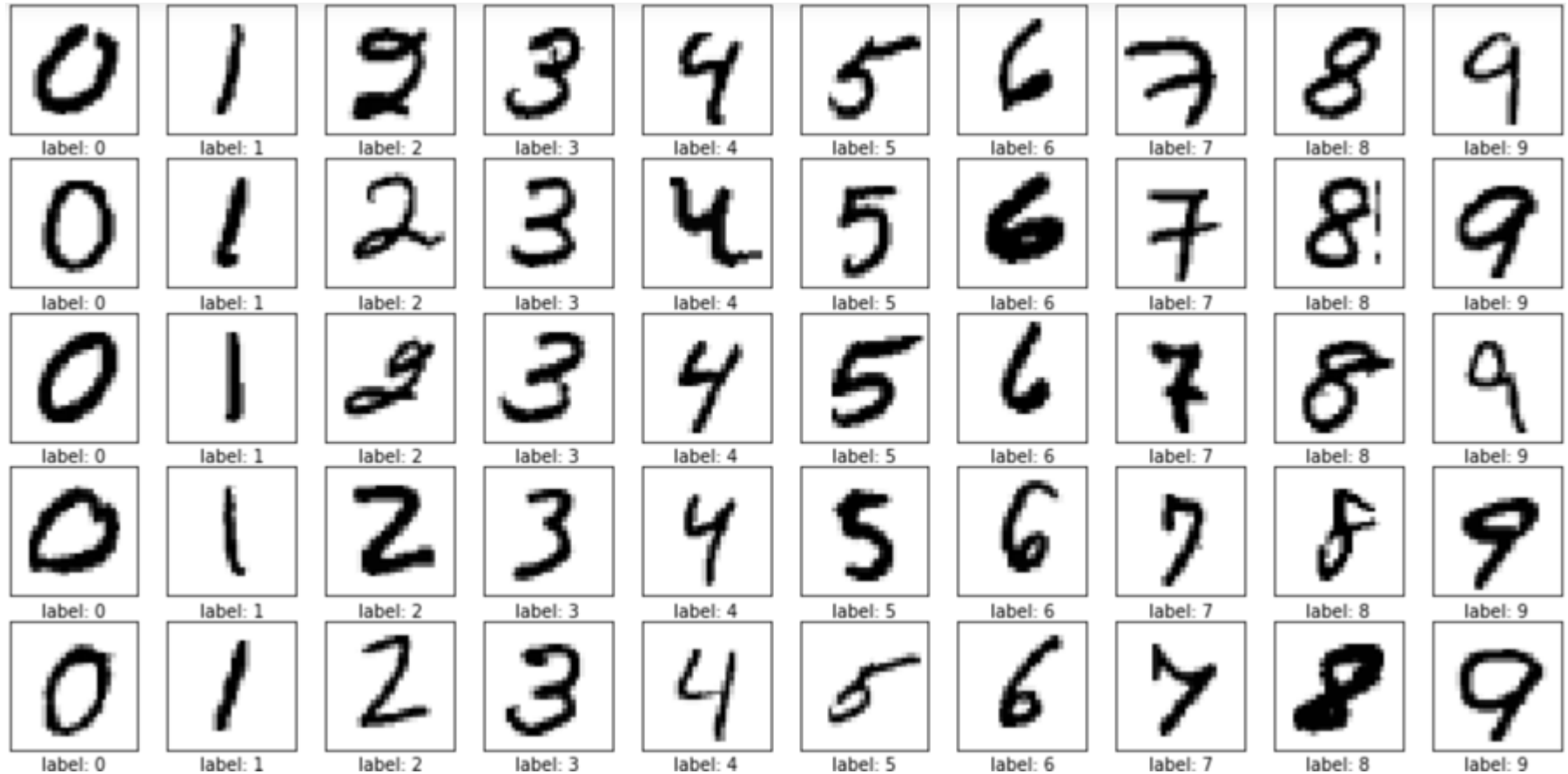
Cortex-M Lab 2

Embedded AI: MNIST Example

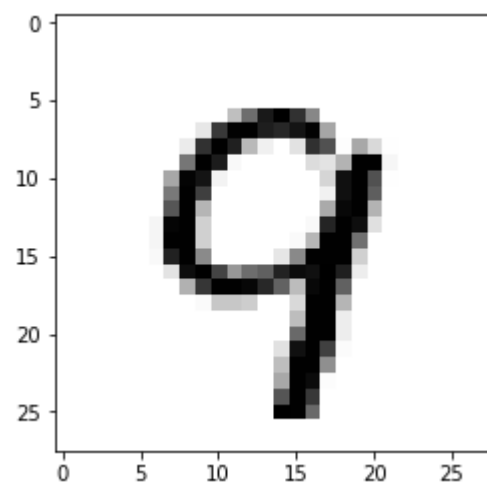


Embedded AI Example

- MNIST Data Set

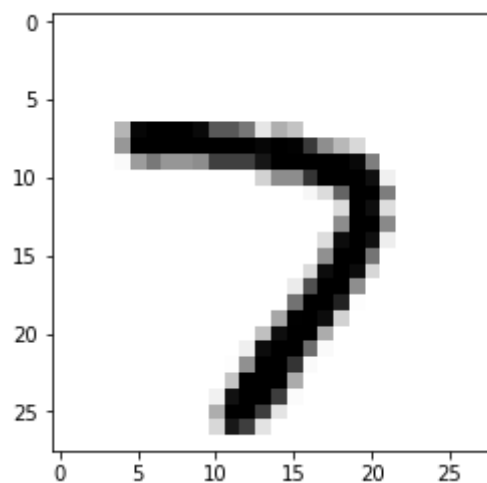


MNIST Data



| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 92 | e5 | ff | cd | 78 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 18 | c6 | fc | fd | e1 | d8 | eb | fc | 59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 14 | cd | fd | df | 46 | f | 0 | 1d | ce | ae | 2 | 57 | 26 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 89 | fd | e3 | 6 | 0 | 0 | 0 | 0 | 23 | 1c | 4c | fd | fd | 9 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 58 | fb | eb | c | 0 | 0 | 0 | 0 | 0 | 0 | 2a | ee | fd | ae | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 86 | fd | c0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | e | ee | fd | a1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | a9 | fd | 4a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 55 | f7 | fd | 4b | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | a | fa | fd | 2f | 0 | 0 | 0 | 0 | 0 | 0 | 6 | db | fd | f1 | 1f | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | a | fd | fd | 2f | 0 | 0 | 0 | 0 | 0 | 5 | 48 | fd | fd | 8f | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 6 | dd | fd | 75 | 0 | 0 | 0 | 0 | 19 | 76 | fd | fd | fd | 2f | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1c | f2 | fe | bb | 68 | 92 | 9f | dc | f4 | ef | fe | e0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 4e | c9 | fd | fd | f8 | d7 | 9c | 43 | f7 | fd | 9d | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 5 | 38 | 38 | 32 | 0 | 0 | 26 | fd | fd | 4a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 42 | fd | fd | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 95 | fd | fd | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1e | ee | fd | bf | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 42 | fd | fd | 70 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 57 | fd | f4 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | aa | fd | c6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | fe | fd | 95 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

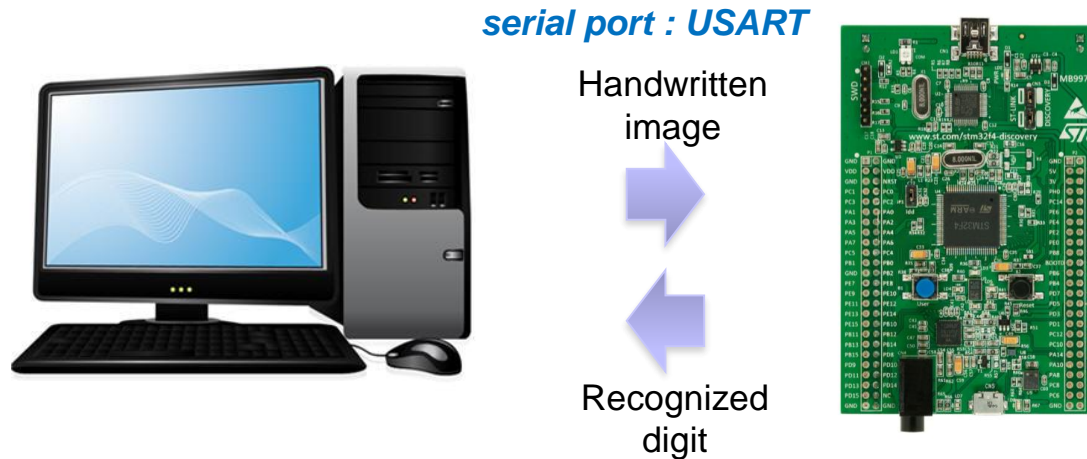
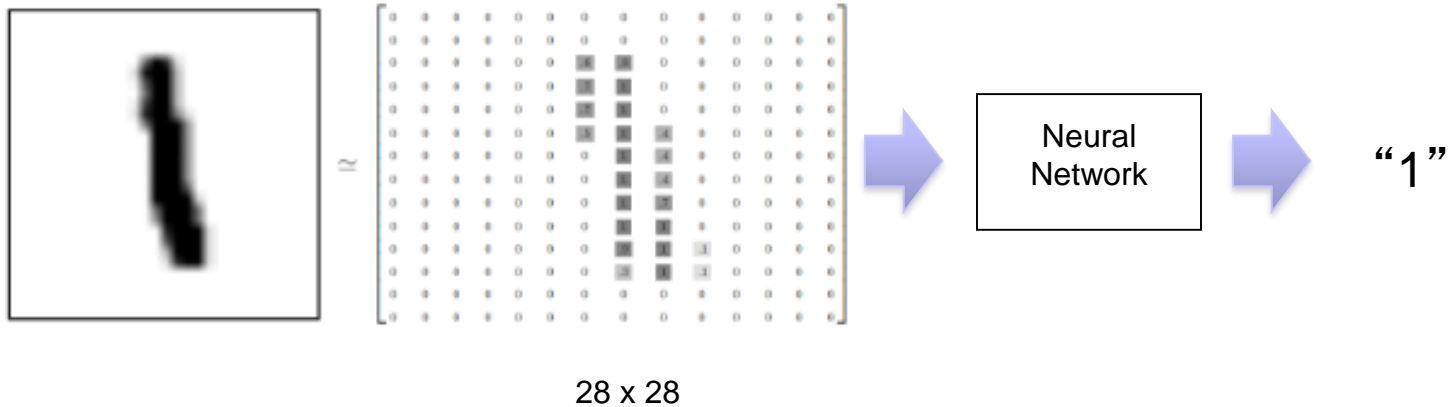
MNIST Data



| | | | | | | | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 4a | f9 | fe | fe | fe | f5 | a7 | a7 | 88 | 19 | 50 | 3c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 68 | fe | fe | fe | fe | fe | fe | fe | fe | f9 | fe | fc | c5 | 71 | 47 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 5 | 63 | 87 | 69 | 69 | 72 | c0 | c0 | c0 | e9 | fe | fe | fe | fe | fe | f6 | 81 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2d | 72 | 72 | cb | fe | fe | fe | f0 | f | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 23 | 9b | fe | fe | 82 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 | fe | f1 | 22 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 73 | fe | fe | 76 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | f3 | fe | f0 | 11 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6f | fe | fe | 8b | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | f3 | fe | f4 | 28 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | b0 | fe | fe | 71 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8c | fe | fe | dc | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 58 | fd | fe | f3 | 2d | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3f | f1 | fe | fe | 53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | f3 | fe | fe | 93 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 6f | fe | fe | cb | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3a | fe | fe | fe | 54 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | e | ed | fe | ff | c2 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 52 | fe | fe | c2 | 1b | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | e6 | c1 | 1c | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Embedded AI Example using MNIST Data Set

0–9 handwritten digit recognition:

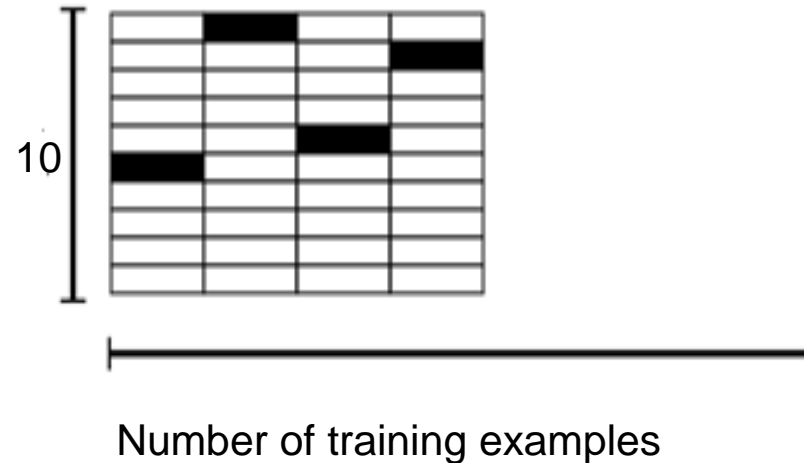
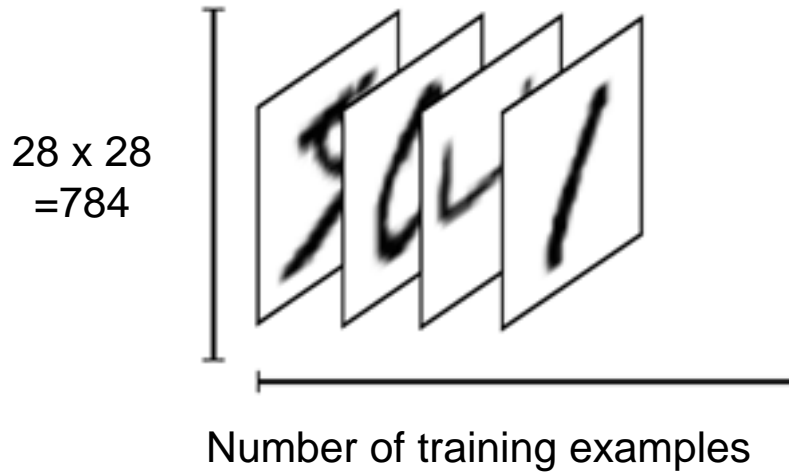


MNIST Data maintained by Yann LeCun: <http://yann.lecun.com/exdb/mnist/>
Keras provides data sets loading function at <http://keras.io/datasets>

Training

```
model.fit(x_train, y_train, batch_size=100, nb_epoch=20)
```

numpy array

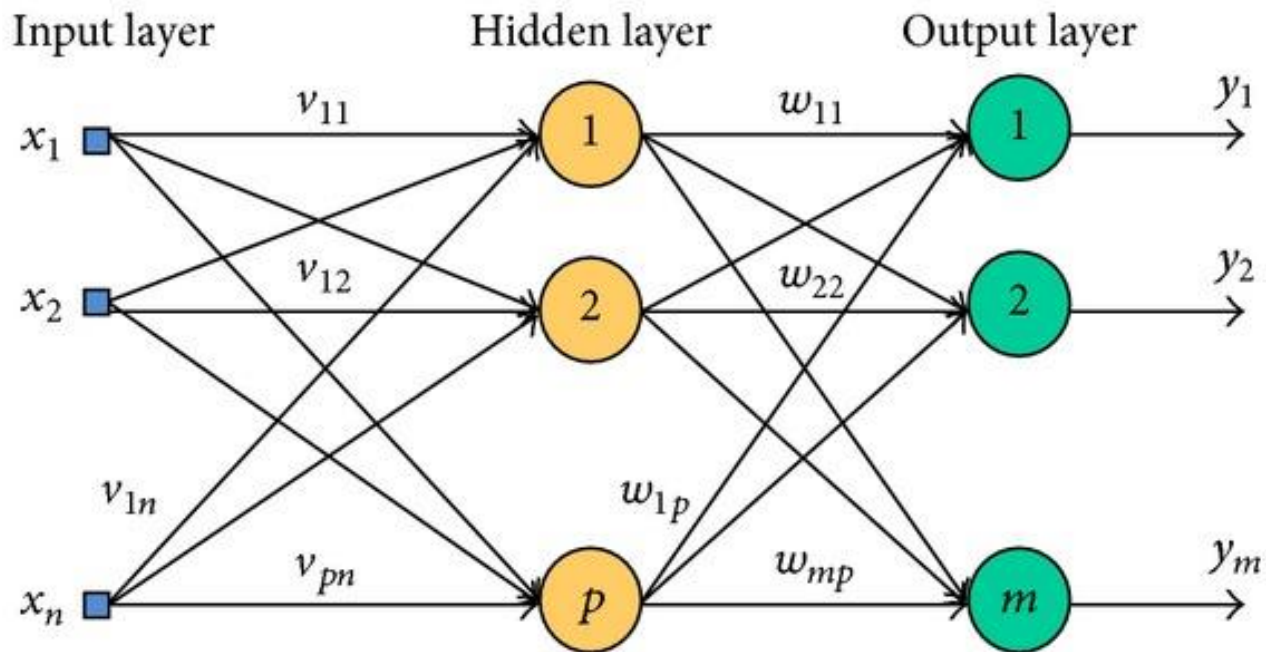


- Training on PC
- Save neural network model
- Convert model to C program
- Compile and download to target

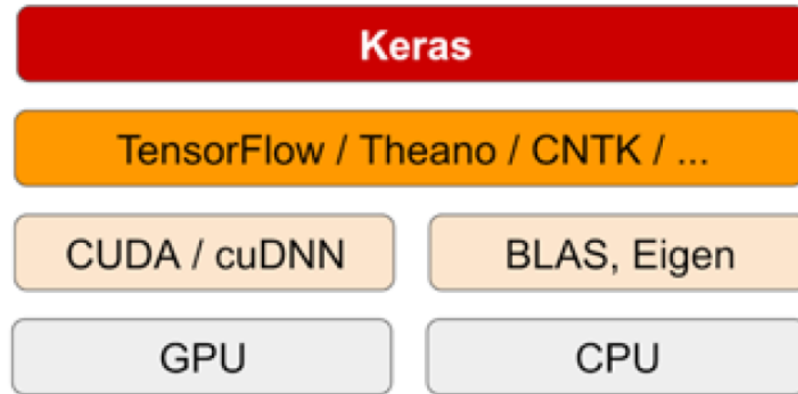


Neural Network Model

- $n=28 \times 28$
- $m=10$

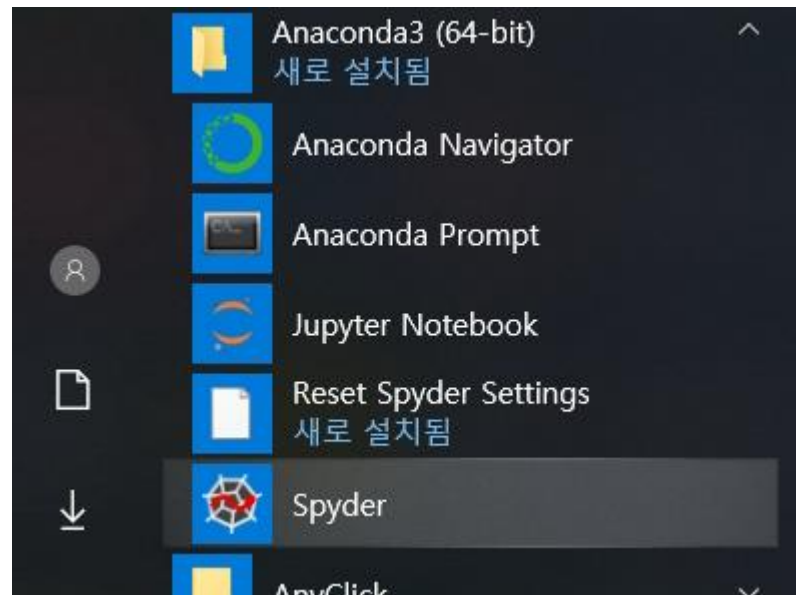


Deep-Learning Software and Hardware Stack

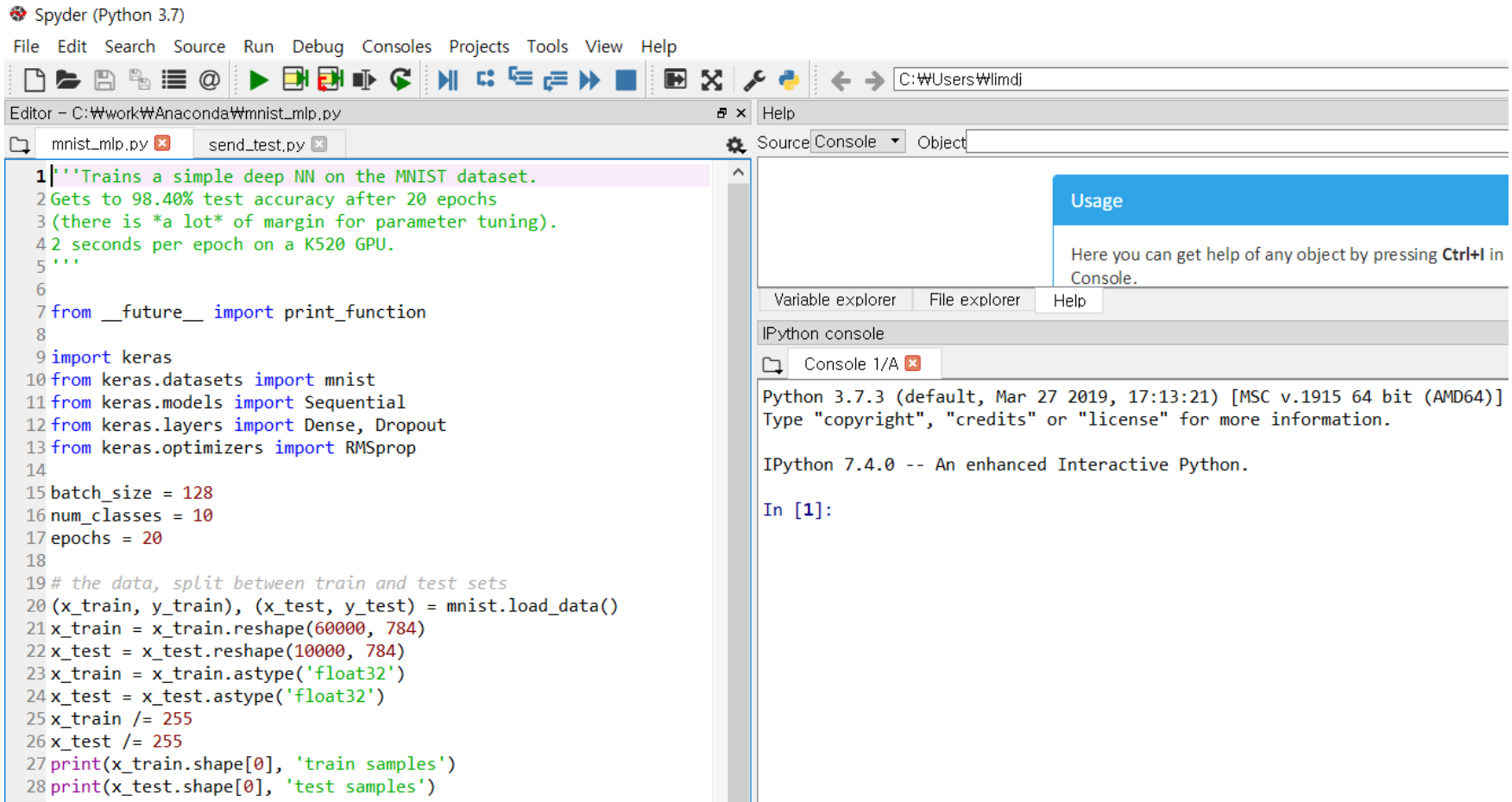


Anaconda : Python Data Science Platform

- Copy mnist_mlp.py, send_test.py to C:\work\Ananconda
- Run Spyder



■ Open mnist_mlp.py



The screenshot shows the Spyder Python IDE interface. The main editor window displays the file `mnist_mlp.py` with the following code:

```
1'''Trains a simple deep NN on the MNIST dataset.
2Gets to 98.40% test accuracy after 20 epochs
3(there is *a lot* of margin for parameter tuning).
42 seconds per epoch on a K520 GPU.
5'''
6
7from __future__ import print_function
8
9import keras
10from keras.datasets import mnist
11from keras.models import Sequential
12from keras.layers import Dense, Dropout
13from keras.optimizers import RMSprop
14
15batch_size = 128
16num_classes = 10
17epochs = 20
18
19# the data, split between train and test sets
20(x_train, y_train), (x_test, y_test) = mnist.load_data()
21x_train = x_train.reshape(60000, 784)
22x_test = x_test.reshape(10000, 784)
23x_train = x_train.astype('float32')
24x_test = x_test.astype('float32')
25x_train /= 255
26x_test /= 255
27print(x_train.shape[0], 'train samples')
28print(x_test.shape[0], 'test samples')
--
```

The right-hand side of the IDE contains the IPython console and a help panel. The help panel shows the "Usage" section, which states: "Here you can get help of any object by pressing **Ctrl+I** in Console." Below this, the "Variable explorer", "File explorer", and "Help" tabs are visible. The IPython console shows the following output:

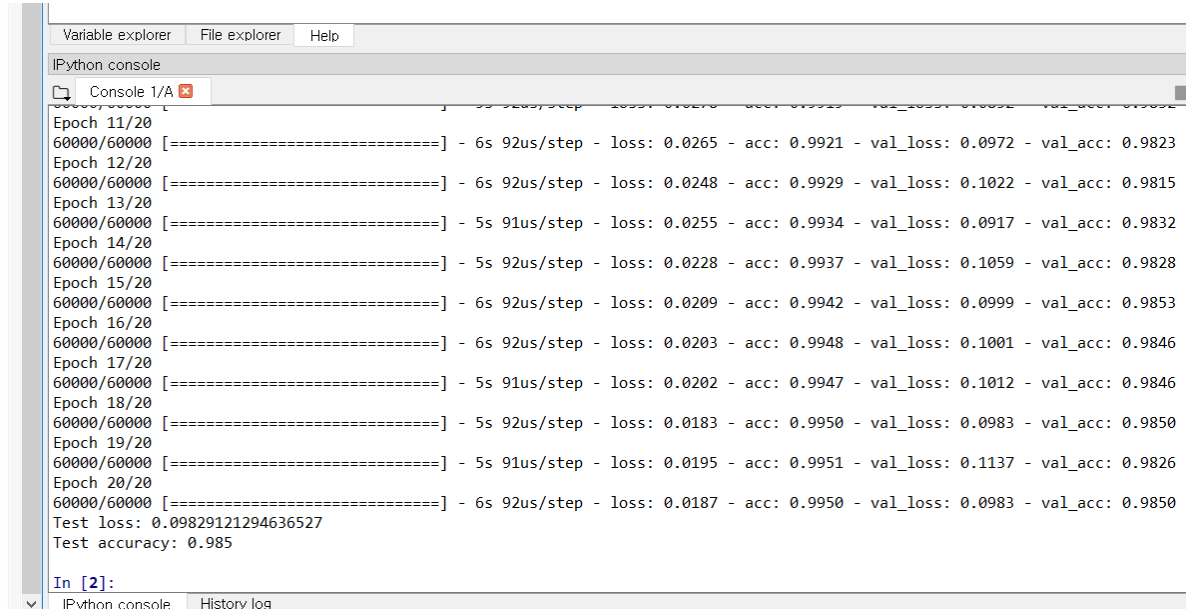
```
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]:
```

■ Saved model: mnist_mlp_model.h5

```
28 print(x_train.shape[0], 'train samples')
29 print(x_test.shape[0], 'test samples')
30
31 # convert class vectors to binary class matrices
32 y_train = keras.utils.to_categorical(y_train, num_classes)
33 y_test = keras.utils.to_categorical(y_test, num_classes)
34
35 model = Sequential()
36 model.add(Dense(512, activation='relu', input_shape=(784,)))
37 model.add(Dropout(0.2))
38 model.add(Dense(512, activation='relu'))
39 model.add(Dropout(0.2))
40 model.add(Dense(num_classes, activation='softmax'))
41
42 model.summary()
43
44 model.compile(loss='categorical_crossentropy',
45               optimizer=RMSprop(),
46               metrics=['accuracy'])
47
48 history = model.fit(x_train, y_train,
49                    batch_size=batch_size,
50                    epochs=epochs,
51                    verbose=1,
52                    validation_data=(x_test, y_test))
53 score = model.evaluate(x_test, y_test, verbose=0)
54 print('Test loss:', score[0])
55 print('Test accuracy:', score[1])
56
57 model.save('mnist_mlp_model.h5')
```



Variable explorer | File explorer | Help

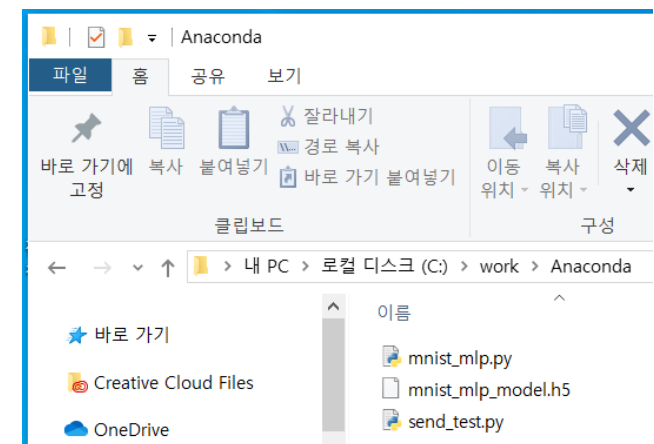
IPython console

Console 1/A

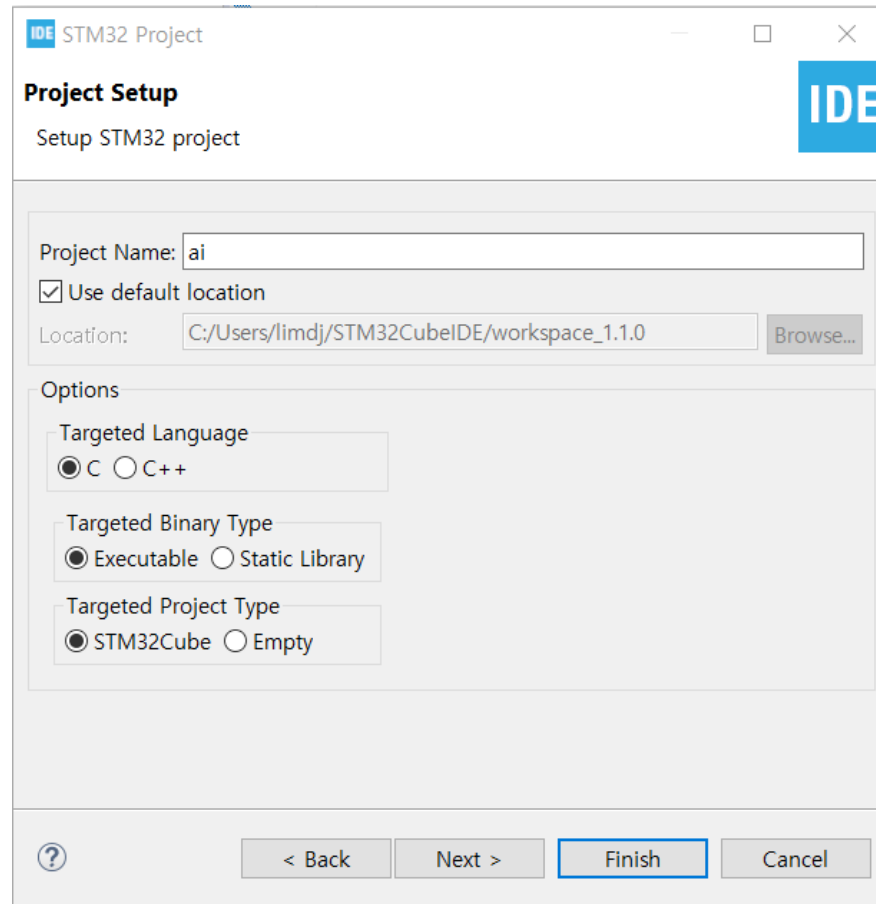
Epoch 11/20
60000/60000 [=====] - 6s 92us/step - loss: 0.0265 - acc: 0.9921 - val_loss: 0.0972 - val_acc: 0.9823
Epoch 12/20
60000/60000 [=====] - 6s 92us/step - loss: 0.0248 - acc: 0.9929 - val_loss: 0.1022 - val_acc: 0.9815
Epoch 13/20
60000/60000 [=====] - 5s 91us/step - loss: 0.0255 - acc: 0.9934 - val_loss: 0.0917 - val_acc: 0.9832
Epoch 14/20
60000/60000 [=====] - 5s 92us/step - loss: 0.0228 - acc: 0.9937 - val_loss: 0.1059 - val_acc: 0.9828
Epoch 15/20
60000/60000 [=====] - 6s 92us/step - loss: 0.0209 - acc: 0.9942 - val_loss: 0.0999 - val_acc: 0.9853
Epoch 16/20
60000/60000 [=====] - 6s 92us/step - loss: 0.0203 - acc: 0.9948 - val_loss: 0.1001 - val_acc: 0.9846
Epoch 17/20
60000/60000 [=====] - 5s 91us/step - loss: 0.0202 - acc: 0.9947 - val_loss: 0.1012 - val_acc: 0.9846
Epoch 18/20
60000/60000 [=====] - 5s 92us/step - loss: 0.0183 - acc: 0.9950 - val_loss: 0.0983 - val_acc: 0.9850
Epoch 19/20
60000/60000 [=====] - 5s 91us/step - loss: 0.0195 - acc: 0.9951 - val_loss: 0.1137 - val_acc: 0.9826
Epoch 20/20
60000/60000 [=====] - 6s 92us/step - loss: 0.0187 - acc: 0.9950 - val_loss: 0.0983 - val_acc: 0.9850
Test loss: 0.09829121294636527
Test accuracy: 0.985

In [2]:

IPython console | History log



New STM32 Project: ai



The image shows a 'Project Setup' dialog box for creating a new STM32 project. The window title is 'STM32 Project'. The main heading is 'Project Setup' with the subtitle 'Setup STM32 project'. The 'Project Name' field contains 'ai'. The 'Use default location' checkbox is checked. The 'Location' field shows 'C:/Users/limdj/STM32CubeIDE/workspace_1.1.0' with a 'Browse...' button. The 'Options' section has three groups: 'Targeted Language' with 'C' selected, 'Targeted Binary Type' with 'Executable' selected, and 'Targeted Project Type' with 'STM32Cube' selected. At the bottom are buttons for '?', '< Back', 'Next >', 'Finish' (highlighted), and 'Cancel'.

STM32 Project

Project Setup
Setup STM32 project

Project Name: ai

☒ Use default location

Location: C:/Users/limdj/STM32CubeIDE/workspace_1.1.0 Browse...

Options

Targeted Language
☒ C ☐ C++

Targeted Binary Type
☒ Executable ☐ Static Library

Targeted Project Type
☒ STM32Cube ☐ Empty

? < Back Next > **Finish** Cancel

■ Minimum Heap Size: 0x2000

The screenshot shows the STM32CubeIDE Project Manager interface. The top bar displays the project name 'ai.ioc' and the files 'aiSystemPerformance.c', 'network_data.c', and 'main.c'. The left sidebar has four tabs: 'Project', 'Code Generator', 'Advanced Settings', and 'Pinout & Configuration'. The 'Advanced Settings' tab is selected, showing the following configuration:

- Project Settings**
 - Project Name: ai
 - Project Location: C:\Users\lindj\STM32CubeIDE\workspace_1.1.0
 - Application Structure: Advanced (dropdown) ☐ Do not generate the main()
 - Toolchain Folder Location: C:\Users\lindj\STM32CubeIDE\workspace_1.1.0\ai\
 - Toolchain / IDE: STM32CubeIDE (dropdown) ☒ Generate Under Root
- Linker Settings**
 - Minimum Heap Size: 0x2000 (highlighted with an orange box)
 - Minimum Stack Size: 0x400
- Mcu and Firmware Package**
 - Mcu Reference: STM32F407VGTx
 - Firmware Package Name and Version: STM32Cube FW_F4 V1.24.1 (dropdown) ☒ Use latest available version

■ Enable USART2

The screenshot displays the STM32CubeMX Pinout & Configuration window. The left sidebar shows a tree view of components, with **USART2** selected under the **Connectivity** category. The main panel is titled **USART2 Mode and Configuration** and is divided into two sections: **Mode** and **Configuration**.

Mode Section:

- Mode: Asynchronous
- Hardware Flow Control (RS232): Disable

Configuration Section:

A **Reset Configuration** button is present. Below it, there are tabs for **NVIC Settings**, **DMA Settings**, **GPIO Settings**, **Parameter Settings** (selected), and **User Constants**.

The **Parameter Settings** tab shows the following parameters:

| Configure the below parameters : | |
|----------------------------------|---------------------------|
| Search (Ctrl+F) | |
| Basic Parameters | |
| Baud Rate | 115200 Bits/s |
| Word Length | 8 Bits (including Parity) |
| Parity | None |
| Stop Bits | 1 |
| Advanced Parameters | |
| Data Direction | Receive and Transmit |

■ Select Components from Software Packs Menu



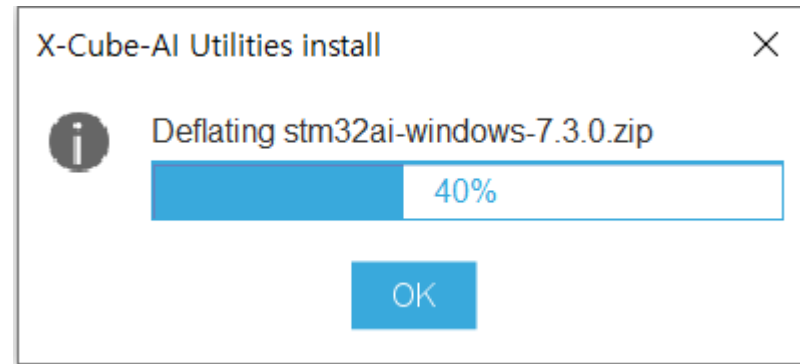
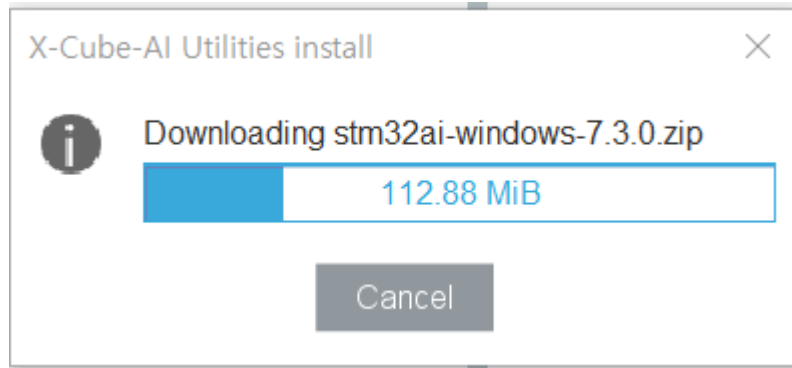
- Select Application: SystemPerformance
- Select X-CUBE-AI: Core
- Then, Click OK

MX Software Packs Component Selector

Packs

☰ ⓘ >

| Pack / Bundle / Component | Status | Version | Selection |
|---------------------------------------|--------|-----------|-------------------------------------|
| > STMicroelectronics.FP-ATR-SIGFOX1 | | 3.2.0 ⓘ | Install |
| > STMicroelectronics.FP-SNS-FLIGHT1 | | 5.0.2 ⓘ | Install |
| > STMicroelectronics.FP-SNS-MOTENV1 | | 4.3.2 ⓘ | Install |
| ▼ STMicroelectronics.X-CUBE-AI | ✓ | 8.0.1 ▼ | |
| ▼ Artificial Intelligence X-CUBE-AI | ✓ | 8.0.1 | |
| Core | ✓ | 8.0.1 | <input checked="" type="checkbox"/> |
| ▼ Device Application | ✓ | 8.0.1 | |
| Application | ✓ | 8.0.1 | SystemPerformance ▼ |
| > STMicroelectronics.X-CUBE-ALGOBUILD | | 1.3.0 ⓘ ▼ | Install |
| > STMicroelectronics.X-CUBE-ALS | | 1.0.1 ⓘ | Install |



- Click Software Packs and click X-CUBE-AI

The screenshot shows the STM32CubeMX interface with the 'Software Packs' tab selected. The left sidebar lists categories: Connectivity, Multimedia, Security, Computing, and Middleware and Software Packs. Under 'Middleware and Software Packs', a list of software packs is shown, including FATFS, FREERTOS, I-CUBE-Cesium, I-CUBE-UNISONRTOS, I-CUBE-embOS, I-CUBE-wolfSSL, I-Cube-SoM-uGOAL, LIBJPEG, LWIP, MBEDTLS, PDM2PCM, USB_DEVICE, USB_HOST, and X-CUBE-AI. The 'X-CUBE-AI' pack is highlighted. The main area shows the 'Mode' configuration for 'STMicroelectronics.X-CUBE-AI.8.0.1 Mode and Configuration', with 'Artificial Intelligence X-CUBE-AI' and 'Device Application' selected. The 'Configuration' section includes buttons for 'Reset Configuration', 'Add network', and 'Delete network'. The 'Main' tab is selected, showing a 'Platform proposal' and 'Communication' section. The 'Communication' section has a table with columns: Name, IPs or Components, Found Solutions, and BSP API. The table contains one row: 'COM Port', 'USART:Asynchronous', 'USART2', and 'Unknown'.

| Name | IPs or Components | Found Solutions | BSP API |
|----------|--------------------|-----------------|---------|
| COM Port | USART:Asynchronous | USART2 | Unknown |

■ Platform Settings

The screenshot shows the 'Configuration' window with the 'Platform Settings' tab selected. At the top, there are buttons for 'Reset Configuration', 'Add network', and 'Delete network'. Below these are tabs for 'Main', 'Platform Settings', and a '+' icon. The 'Platform Settings' section includes a 'Platform proposal' dropdown and a 'Communication' section. The 'Communication' section contains a table with columns: 'Name', 'IPs or Components', 'Found Solutions', and 'BSP API'. The table has one row for 'COM Port' with values: 'USART:Asynchronous', 'USART2', and 'Unknown'.

| Name | IPs or Components | Found Solutions | BSP API |
|----------|--------------------|-----------------|---------|
| COM Port | USART:Asynchronous | USART2 | Unknown |

■ Add network

The screenshot shows the 'Configuration' window with the 'network' tab selected. At the top, there are buttons for 'Reset Configuration', 'Add network', and 'Delete network'. Below these are tabs for 'Main', 'Platform Settings', 'network', and a '+' icon. The 'network' section includes a dropdown menu with 'keras' selected and 'Saved model' next to it. Below this is a 'Model:' label followed by a text input field containing 'C:\emwork\mnist_mlp_model.h5' and a 'Browse...' button. There is another empty text input field with a 'Browse...' button below it. At the bottom, there are dropdown menus for 'Compression' (set to 'Low'), 'Optimization' (set to 'Balanced'), 'Validation inputs' (set to 'Random numbers'), and 'Validation outputs' (set to 'None'). A 'Show graph' button is located at the bottom right.

Model: C:\emwork\mnist_mlp_model.h5

Compression: Low Optimization: Balanced

Validation inputs: Random numbers

Validation outputs: None

Show graph

- Click Analyze and check memory

Configuration

Reset Configuration

Add network

Delete network

Main

Platform Settings

network

+

Keras

Saved model

Model: C:\lemwork\mnist_mlp_model.h5

Browse...

Browse...

Compression: Low

Optimization: Balanced

Validation inputs: Random numbers

Validation outputs: None

Complexity: 670880 MACC

Used Flash: 686.89 KiB (686.89 KiB over 1024.00 KiB Internal)

Used Ram: 7.12 KiB (7.12 KiB over 192.00 KiB Internal)

Achieved compression: 3.9

Analysis status: done

Show graph

Analyze

Validate on desktop

Validate on target

■ Validate on desktop

MX Please wait...

i

Validation on desktop

NOTE: the output of the reference model is used as ground truth/reference value

acc=100.00%, rmse=0.006813521, mae=0.001394981, l2r=0.022370711

10 classes (10 samples)

| | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|
| C0 | 0 | . | . | . | . | . | . | . | . |
| C1 | . | 0 | . | . | . | . | . | . | . |
| C2 | . | . | 6 | . | . | . | . | . | . |
| C3 | . | . | . | 0 | . | . | . | . | . |
| C4 | . | . | . | . | 0 | . | . | . | . |
| C5 | . | . | . | . | . | 0 | . | . | . |
| C6 | . | . | . | . | . | . | 3 | . | . |
| C7 | . | . | . | . | . | . | . | 1 | . |
| C8 | . | . | . | . | . | . | . | . | 0 |
| C9 | . | . | . | . | . | . | . | . | . |

Evaluation report (summary)

| | | | | | |
|------------|---------|-------------|-------------|-------------|--|
| Mode | acc | rmse | mae | l2r | tensor |
| X-cross #1 | 100.00% | 0.006813521 | 0.001394981 | 0.022370711 | dense_3_nl, ai_float, [(1, 1, 10)], m_id=[4] |

X-cross (l2r) #1 error : 2.23707110e-02 (expected to be < 0.01)

Creating txt report file C:\Users\lmdj\stm32cube\stm32cube\stm32cube\network_validate_report.txt

elapsed time (validate): 10.767s

OK

Generate Code and Build

workspace_1.6.0 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

The screenshot displays the STM32CubeIDE interface. On the left, the Project Explorer shows a project named 'ai' with various components like Binaries, Includes, Core, Drivers, Middlewares, USB_HOST, X-CUBE-AI, Debug, and firmware files. The main workspace is divided into three tabs: Pinout & Configuration, Clock Configuration, and Project Manager. The Pinout & Configuration tab is active, showing a list of categories (System Core, Analog, Timers, Connectivity, Multimedia, Security, Computing) and a search bar. The Project Manager tab shows the 'Software Packs' section with 'STMicroelectronics X-CUBE-AI 6.0.0 Mode and Configuration' selected. Below this, the 'Configuration' section has tabs for Main, Platform Settings, and network. The network tab is active, showing a 'Model' field with the path 'C:\emwork\mnist_mlp_model.h5' and a 'Browse...' button. Other settings include 'Compression: 4', 'Validation inputs: Random numbers', and 'Validation outputs: None'. At the bottom, the Console shows the build output for 'ai', including commands like 'arm-none-eabi-size ai.elf' and 'arm-none-eabi-objcopy -O binary ai.elf "ai.bin"', and the final status: '12:32:26 Build Finished. 0 errors, 0 warnings. (took 16s.305ms)'.

ai.loc main.c

Pinout & Configuration Clock Configuration Project Manager

Software Packs Pinout

STMicroelectronics X-CUBE-AI 6.0.0 Mode and Configuration

Configuration

Reset Configuration Add network Delete network

Main Platform Settings network +

Model: C:\emwork\mnist_mlp_model.h5 Browse... Browse...

Compression: 4

Validation inputs: Random numbers

Validation outputs: None

Show graph Analyze Validate on desktop

Complexity: 670880 MACC

Console

CDT Build Console [ai]

```
arm-none-eabi-size ai.elf
arm-none-eabi-objdump -h -S ai.elf > "ai.list"
arm-none-eabi-objcopy -O binary ai.elf "ai.bin"
text data bss dec hex filename
764040 2224 22128 788392 c07a8 ai.elf
Finished building: default.size.stdout

Finished building: ai.bin
Finished building: ai.list

12:32:26 Build Finished. 0 errors, 0 warnings. (took 16s.305ms)
```

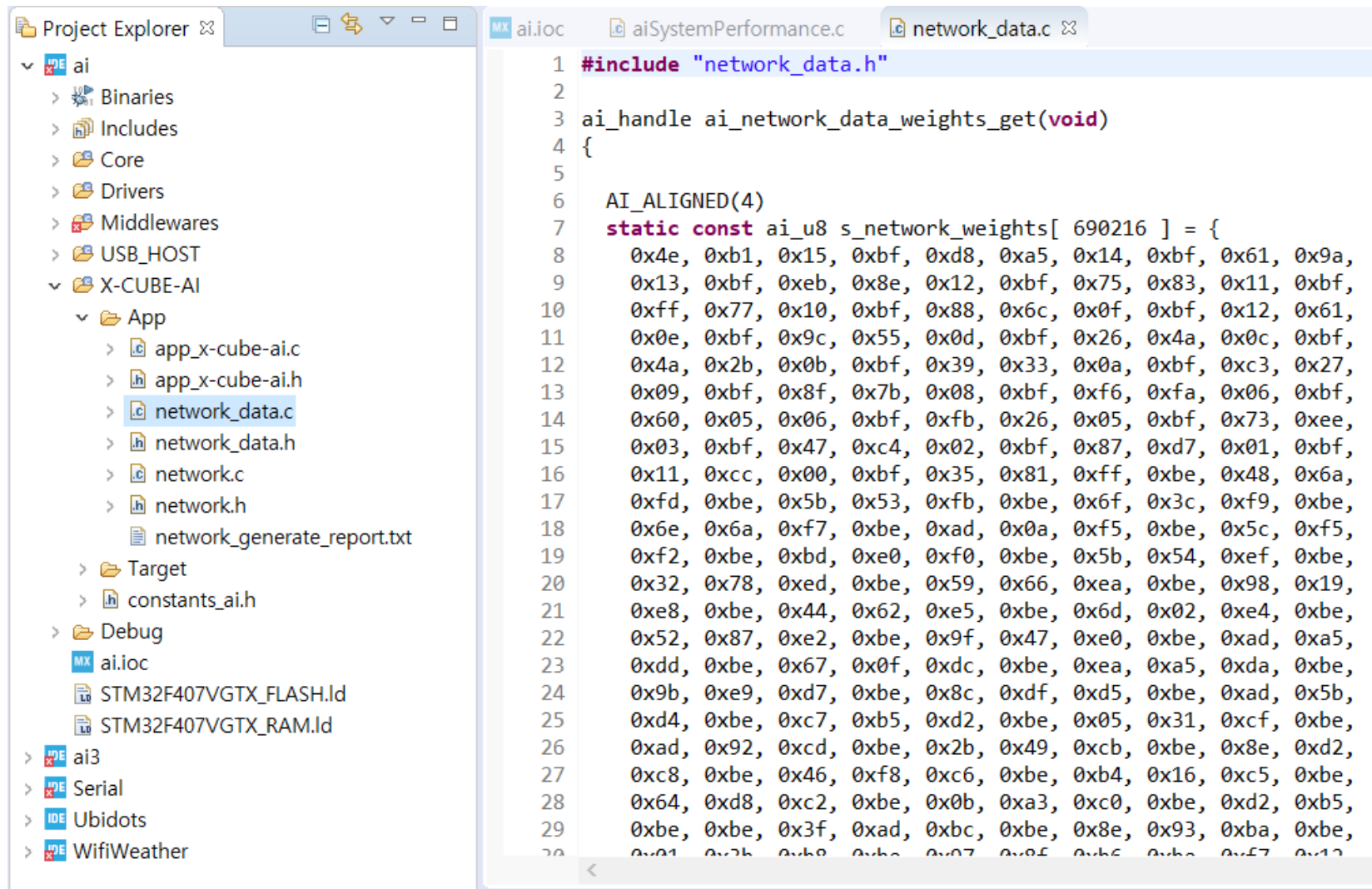
- 터미널을 열어서 다음과 같이 실행되는지 확인

```
SmarTTY - Raw Terminal
Connected to COM13 (115200 bps) Baud rate: 115200
cycles/MACC : 8.18 (average for all layers)
used stack  : 368 bytes
used heap    : 0:0 0:0 (req:allocated,req:released) max=0 cur=0 (cf
g=3)
observer res : 120 bytes used from the heap (6 c-nodes)

Inference time by c-node
kernel : 5.771ms (time passed in the c-kernel fcts)
user   : 0.006ms (time passed in the user cb)

c_id  type      id      time (ms)
-----
0      DENSE     0       4.847  83.99 %
1      NL         0       0.012   0.22 %
2      DENSE     2       0.816  14.15 %
3      NL         2       0.012   0.22 %
4      DENSE     4       0.066   1.16 %
5      NL         4       0.014   0.26 %
-----
                                5.771 ms
```

■ Trained weight



The screenshot displays an IDE interface. On the left, the 'Project Explorer' shows a project named 'ai' with a tree structure including 'Binaries', 'Includes', 'Core', 'Drivers', 'Middlewares', 'USB_HOST', and 'X-CUBE-AI'. Under 'X-CUBE-AI', the 'App' folder is expanded, showing files like 'app_x-cube-ai.c', 'app_x-cube-ai.h', 'network_data.c' (highlighted), 'network_data.h', 'network.c', 'network.h', and 'network_generate_report.txt'. The main editor on the right shows the 'network_data.c' file. It includes a header '#include "network_data.h"' and defines a function 'ai_handle ai_network_data_weights_get(void)'. Inside the function, a large static array 's_network_weights' of type 'ai_u8' is declared, containing 690216 elements. The array is populated with a long sequence of hexadecimal values, each preceded by '0x'. The values are listed in groups of 16 per line, starting from '0x4e' and ending with '0x13' on the final line.

```
1 #include "network_data.h"
2
3 ai_handle ai_network_data_weights_get(void)
4 {
5
6     AI_ALIGNED(4)
7     static const ai_u8 s_network_weights[ 690216 ] = {
8         0x4e, 0xb1, 0x15, 0xbf, 0xd8, 0xa5, 0x14, 0xbf, 0x61, 0x9a,
9         0x13, 0xbf, 0xeb, 0x8e, 0x12, 0xbf, 0x75, 0x83, 0x11, 0xbf,
10        0xff, 0x77, 0x10, 0xbf, 0x88, 0x6c, 0x0f, 0xbf, 0x12, 0x61,
11        0x0e, 0xbf, 0x9c, 0x55, 0x0d, 0xbf, 0x26, 0x4a, 0x0c, 0xbf,
12        0x4a, 0x2b, 0x0b, 0xbf, 0x39, 0x33, 0x0a, 0xbf, 0xc3, 0x27,
13        0x09, 0xbf, 0x8f, 0x7b, 0x08, 0xbf, 0xf6, 0xfa, 0x06, 0xbf,
14        0x60, 0x05, 0x06, 0xbf, 0xfb, 0x26, 0x05, 0xbf, 0x73, 0xee,
15        0x03, 0xbf, 0x47, 0xc4, 0x02, 0xbf, 0x87, 0xd7, 0x01, 0xbf,
16        0x11, 0xcc, 0x00, 0xbf, 0x35, 0x81, 0xff, 0xbe, 0x48, 0x6a,
17        0xfd, 0xbe, 0x5b, 0x53, 0xfb, 0xbe, 0x6f, 0x3c, 0xf9, 0xbe,
18        0x6e, 0x6a, 0xf7, 0xbe, 0xad, 0x0a, 0xf5, 0xbe, 0x5c, 0xf5,
19        0xf2, 0xbe, 0xbd, 0xe0, 0xf0, 0xbe, 0x5b, 0x54, 0xef, 0xbe,
20        0x32, 0x78, 0xed, 0xbe, 0x59, 0x66, 0xea, 0xbe, 0x98, 0x19,
21        0xe8, 0xbe, 0x44, 0x62, 0xe5, 0xbe, 0x6d, 0x02, 0xe4, 0xbe,
22        0x52, 0x87, 0xe2, 0xbe, 0x9f, 0x47, 0xe0, 0xbe, 0xad, 0xa5,
23        0xdd, 0xbe, 0x67, 0x0f, 0xdc, 0xbe, 0xea, 0xa5, 0xda, 0xbe,
24        0x9b, 0xe9, 0xd7, 0xbe, 0x8c, 0xdf, 0xd5, 0xbe, 0xad, 0x5b,
25        0xd4, 0xbe, 0xc7, 0xb5, 0xd2, 0xbe, 0x05, 0x31, 0xcf, 0xbe,
26        0xad, 0x92, 0xcd, 0xbe, 0x2b, 0x49, 0xcb, 0xbe, 0x8e, 0xd2,
27        0xc8, 0xbe, 0x46, 0xf8, 0xc6, 0xbe, 0xb4, 0x16, 0xc5, 0xbe,
28        0x64, 0xd8, 0xc2, 0xbe, 0x0b, 0xa3, 0xc0, 0xbe, 0xd2, 0xb5,
29        0xbe, 0xbe, 0x3f, 0xad, 0xbc, 0xbe, 0x8e, 0x93, 0xba, 0xbe,
30        0x01, 0x2b, 0xb8, 0xb8, 0x07, 0x8f, 0xb6, 0xb8, 0xf7, 0x13
```


- Open aiSystemPerformance.c
- Find from Edit menu: “input tensors” or go to the line 489

IDE workspace_1.12.1 - ai/X-CUBE-AI/App/aiSystemPerformance.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer ×

- IDE ai
 - Binaries
 - Includes
 - Core
 - Drivers
 - Middlewares
 - USB_HOST
 - X-CUBE-AI
 - App
 - ai_stm32_adaptor.h
 - aiSystemPerformance.c
 - aiSystemPerformance.h
 - aiTestHelper.c
 - aiTestHelper.h
 - aiTestUtility.c
 - aiTestUtility.h
 - app_x-cube-ai.c
 - app_x-cube-ai.h
 - lc_print.c
 - lc_print.h
 - network_config.h
 - network_data_params.c
 - network_data_params.h
 - network_data.c

main.c aiSystemPerformance.c ai.ioc

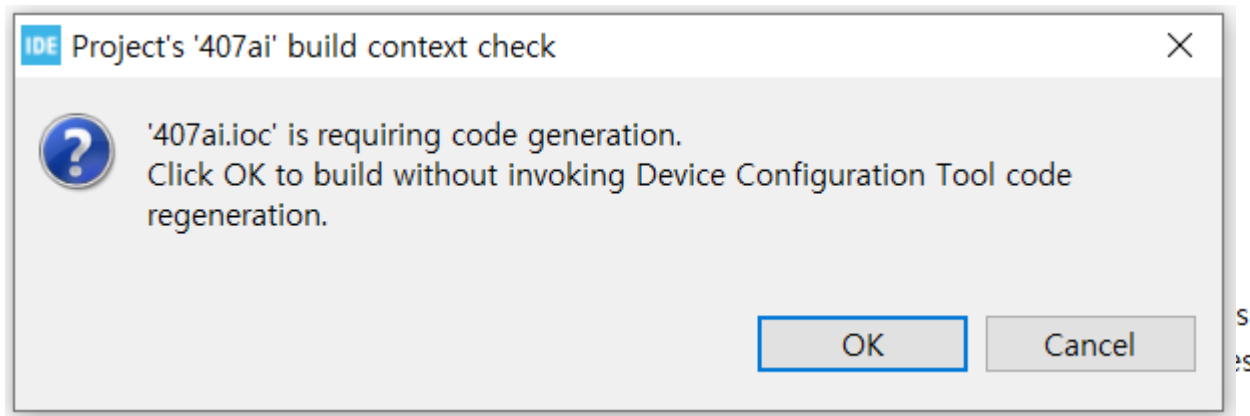
```
482 #endif
483
484 MON_ALLOC_RESET();
485
486 /* Main inference loop */
487 for (iter = 0; iter < niter; iter++) {
488
489     /* Fill input tensors with random data */
490     for (int i = 0; i < net_exec_ctx[idx].report.n_inputs; i++) {
491         const ai_buffer_format fmt = AI_BUFFER_FORMAT(&ai_input[i]);
492         ai_i8 *in_data = (ai_i8 *)ai_input[i].data;
493         for (ai_size j = 0; j < AI_BUFFER_SIZE(&ai_input[i]); ++j) {
494             /* uniform distribution between -1.0 and 1.0 */
495             const float v = 2.0f * (ai_float) rand() / (ai_float) RAND_MAX - 1.0f;
496             if (AI_BUFFER_FMT_GET_TYPE(fmt) == AI_BUFFER_FMT_TYPE_FLOAT) {
497                 *(ai_float *) (in_data + j * 4) = v;
498             }
499             else {
500                 if (AI_BUFFER_FMT_GET_BITS(fmt) >= 8) {
501                     in_data[j] = (ai_i8)(v * 127);
502                     if (AI_BUFFER_FMT_GET_TYPE(fmt) == AI_BUFFER_FMT_TYPE_BOOL) {
503                         in_data[j] = (in_data[j] > 0)?(ai_i8)1:(ai_i8)0;
504                     }
505                 }
506             }
507         }
508     }
509 }
```

Modify aiSystemPerformance.c

```
/* Fill input tensors with random data */
for (int i = 0; i < net_exec_ctx[idx].report.n_inputs; i++) {
    unsigned char string[28 * 28][3];
    ioRawReadBuffer((unsigned char*)string, 28 * 28 * 3);
    for (ai_size j = 0; j < 28 * 28; j++) {
        if (string[j][0] == ' ') string[j][0] = '0';
        if (string[j][1] == ' ') string[j][1] = '0';
    }
    const ai_buffer_format fmt = AI_BUFFER_FORMAT(&ai_input[i]);
    ai_i8 *in_data = (ai_i8 *)ai_input[i].data;
    for (ai_size j = 0; j < AI_BUFFER_SIZE(&ai_input[i]); ++j) {
        /* uniform distribution between -1.0 and 1.0 */
        //const float v = 2.0f * (ai_float) rand() / (ai_float) RAND_MAX - 1.0f;
        const float v = (100.0f*(ai_float)(string[j][0] - 0x30) + 10.0f*(ai_float)(string[j][1] - 0x30) +
            (ai_float)(string[j][2] - 0x30)) / 255.0f;

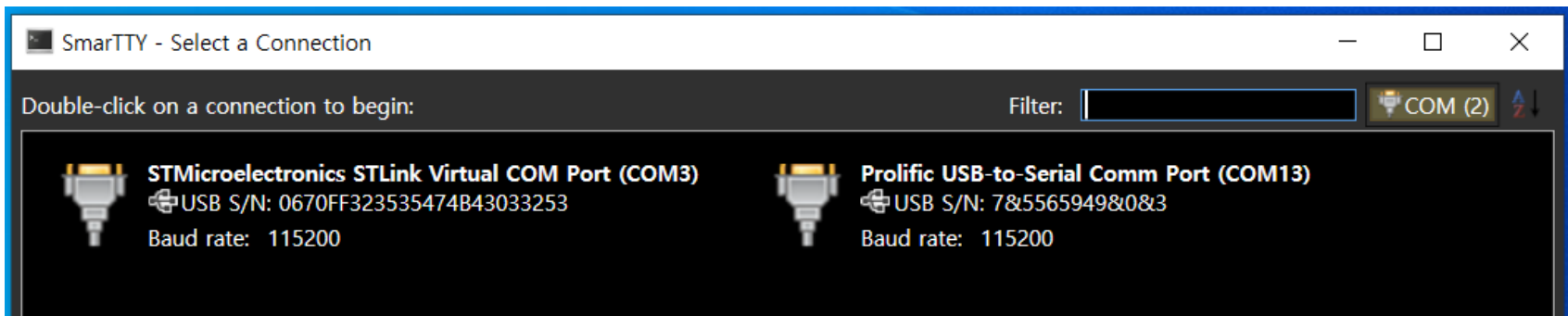
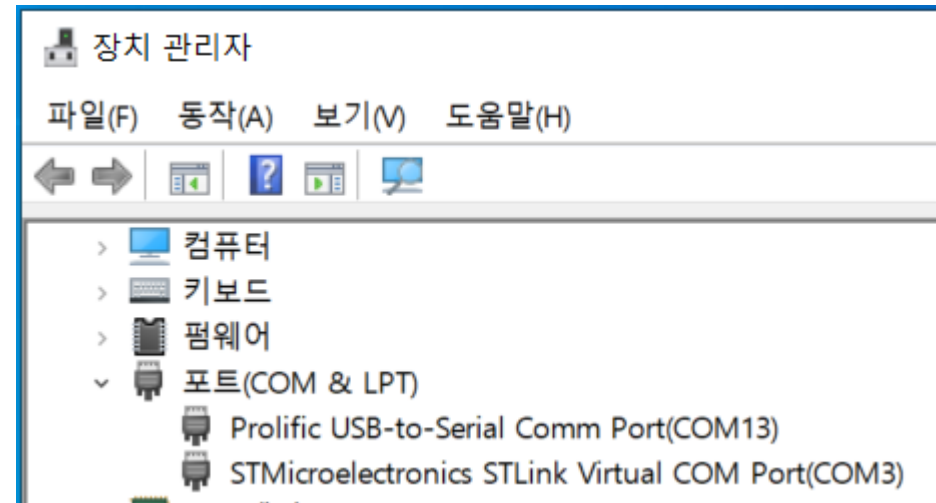
        in_data[j] = (ai_i8)v;
    }
    batch = ai_mnetwork_run(net_exec_ctx[idx].handle, ai_input, ai_output);
    if (batch != 1) {
        aiLogErr(ai_mnetwork_get_error(net_exec_ctx[idx].handle),
            "ai_mnetwork_run");
        break;
    }
    unsigned char recognized_digit;
    ai_float out_data_float[10];
    for (int j = 0; j < 10; j++) out_data_float[j] = *(ai_float *) (ai_output[0].data + j * 4);
    recognized_digit = 0;
    for (int j = 0; j < 10; j++) if (out_data_float[j] > out_data_float[recognized_digit]) recognized_digit = j;
    printf("%d", recognized_digit);
    tend = cyclesCounterEnd();
}
```

- Build and Download to the target board
- Click OK



- > Drivers
- ▼ Middlewares
 - > ST
- > USB_HOST
- ▼ X-CUBE-AI
 - > App
 - > Target
 - > constants_ai.h
- > Debug
 - 407ai Debug.launch
 - 407ai.ioc [Code Generation is required]

- Connect USB-to-serial cable and find COM port number
- Do not open com port



Run send_test.py

- Press RESET button(black button) and run send_test.py
- Change port number

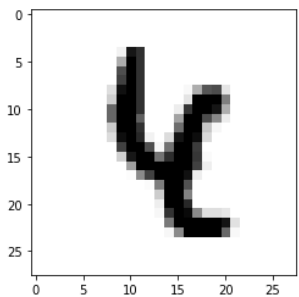
```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\work\Anaconda\send_test.py
Wemwork mnist_mlp.py send_test.py - C:\work\Anaconda
2 from keras.datasets import mnist
3 import serial
4
5 port = "COM3"
6 baud = 115200
7
8 ser = serial.Serial(port, baud, timeout=1)
9 # open the serial port
10 if ser.isOpen():
11     print(ser.name + ' is open...')
12
13 # 1. 데이터셋 생성하기
14
15 # 훈련셋과 시험셋 불러오기
16 (x_train, y_train), (x_test, y_test) = mnist.load_data()
17
18 print(x_test.shape)
19
20 for k in list(range(0,16)):
21     digit= x_test[160+k]
22     import matplotlib.pyplot as plt
23     plt.imshow(digit, cmap=plt.cm.binary)
24     plt.show()
25     #print( digit)
26
27     for i in list(range(0,28)):
28         for j in list(range(0,28)):
29             digit_string="{:3d}".format(digit[i][j])
30             ser.write(digit_string.encode('ascii'))
31             #print(digit[i][j])
32
33     out = ser.read(2)
34     print('Recognized digit:',out.decode('utf-8'))
35
36 out = ser.read(2000)
37 print(out.decode('utf-8'))
38
```

Variable explorer File explorer Hel

IPython console

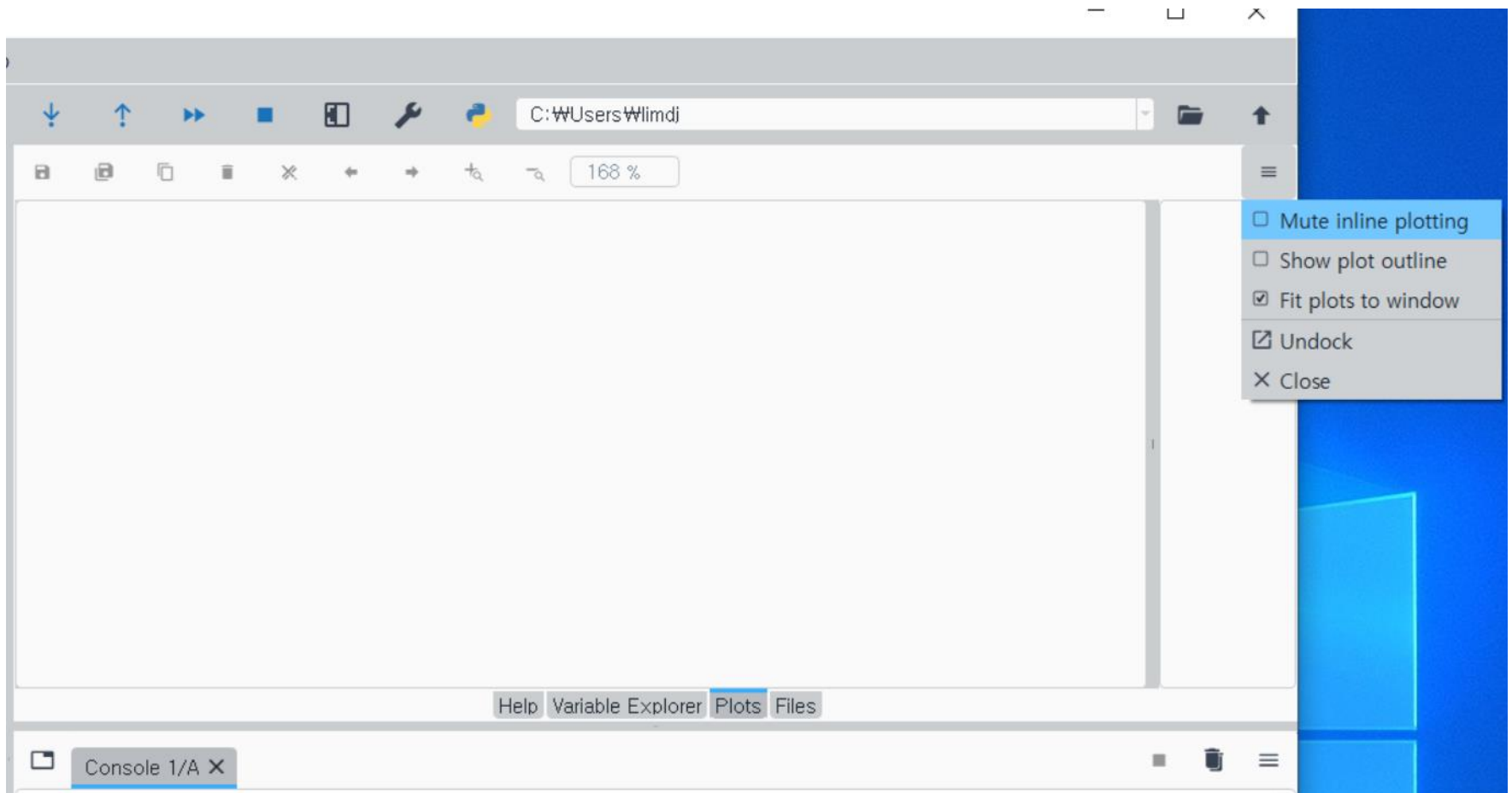
Console 1/A

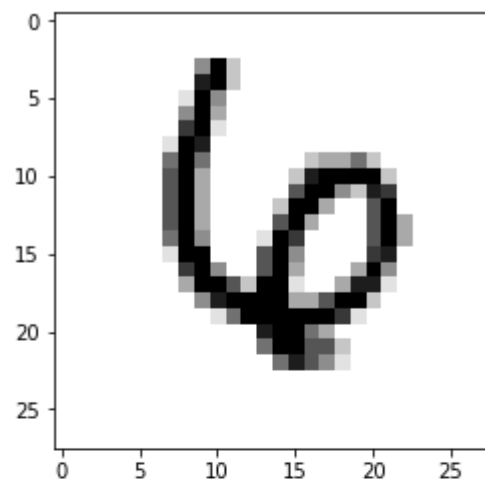
In [28]: runfile('C:/work/Anaconda', args='send_test.py')
COM3 is open...
(10000, 28, 28)



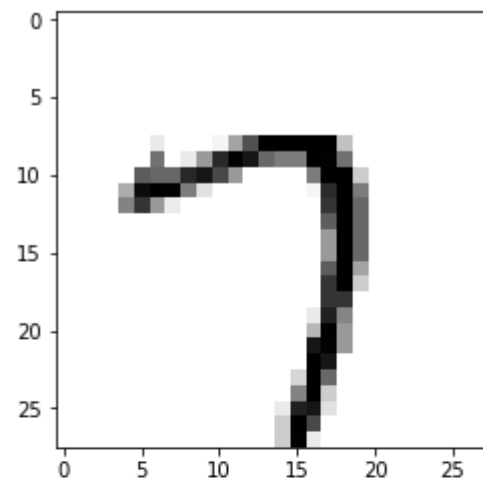
Recognized digit: 4.

- Spyder에서 Plots 메뉴에서 Mute inline plotting을 해제

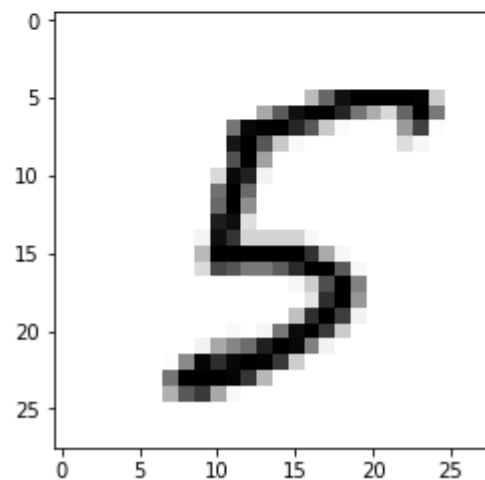




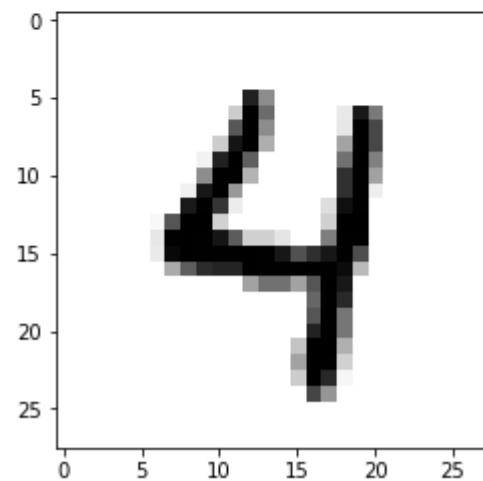
Recognized digit: 6.



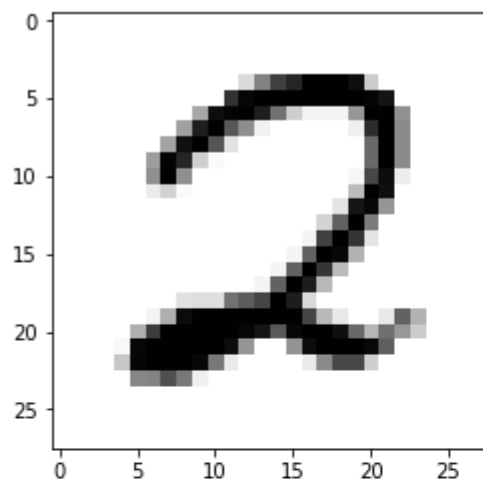
Recognized digit: 7.



Recognized digit: 5.



Recognized digit: 4.



Recognized digit: 2.

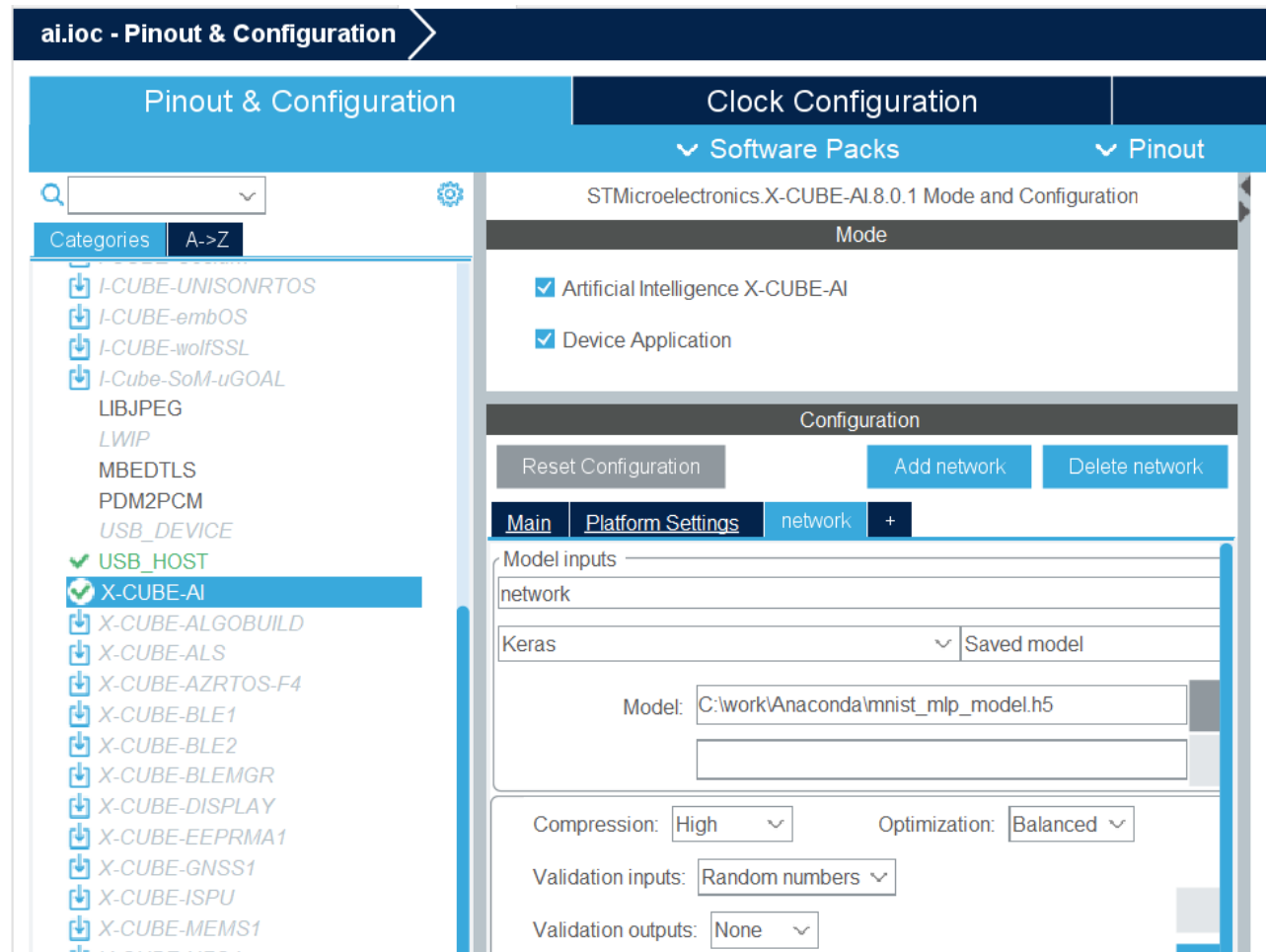
Results for "network", 16 inferences @168MHz/168MHz (comple
duration : 52.216 ms (average)
CPU cycles : 8772339 -75/+87 (average, -/+)
CPU Workload : 5%
cycles/MACC : 13 (average for all layers)
used stack : NOT CALCULATED
used heap : 0:0 0:0 (req:allocated,req:released) cfg=0

Exercise 1

- 주어진 예제를 실행하여 메모리 사용량 및 속도를 검토하고 실제 인식 성능을 확인한다.
- 모델에서 코드 생성 시 컴프레션을 하는 이유는 메모리 사용량을 줄여서 프로그램 메모리에 탑재 가능하도록 하기 위함이다.

The screenshot shows a web-based configuration interface for a neural network model. The title bar is "Configuration". Below it are buttons for "Reset Configuration", "Add network", and "Delete network". The main content area has tabs for "Main", "Platform Settings", "network", and a "+" button. The "network" tab is selected. Under "Model inputs", there is a text input field containing "network". Below that, a dropdown menu shows "Keras" and a "Saved model" dropdown. A "Model:" label is followed by a text input field containing "C:\\emwork\\mnist_mlp_model.h5" and a "Browse..." button. Below this is another empty text input field with a "Browse..." button. Further down, there are dropdowns for "Compression:" (set to "None") and "Optimization:" (set to "Balanced"). Below these are "Validation inputs:" (set to "Random numbers") and "Validation outputs:" (set to "None"). At the bottom left, there is a summary of resource usage: "Complexity: 670880 MACC", "Used Flash: 2.57 MiB (2.57 MiB over 1024.00 KiB Internal)", and "Used Ram: 7.12 KiB (7.12 KiB over 192.00 KiB Internal)". On the right side, there are four buttons: "Show graph", "Analyze" (with a green checkmark icon), "Validate on desktop", and "Validate on target".

- Compression을 High로 선택해서 메모리 사용량이 얼마나 줄어드는지 확인하고 체감 인식 성능에 변화가 있는지 알아본다.



Exercise 2

- MNIST dataset에 대해서 CNN 모델을 이용해서 Exercise 1과 동일한 실습을 진행한다.

Exercise 3

- DL_Lab6에서 진행 했던 CIFAR-10 dataset에 대한 CNN 모델에 대해서 앞의 예제와 동일한 실습을 진행한다.