



DesignNews

Machine Learning Application Design using STM32 MCU's

DAY 5 : Running an Inference on Target

Sponsored by



Webinar Logistics

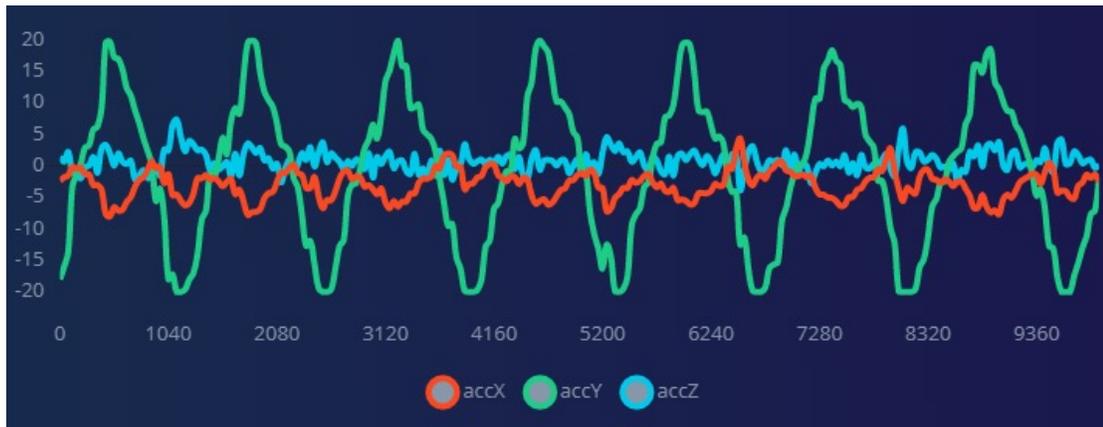
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

Course Sessions

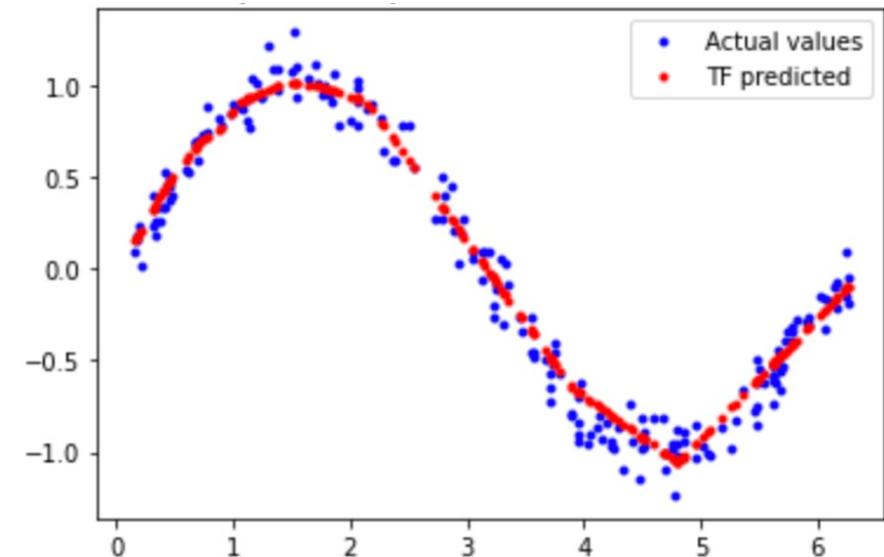
- Introduction to Machine Learning on MCU's
- Capturing, Cleaning and Digital Signal Processing Data
- Training a Neural Network Part 1
- Training a Neural Network Part 2
- **Running an Inference on Target**

Two Models to Deploy

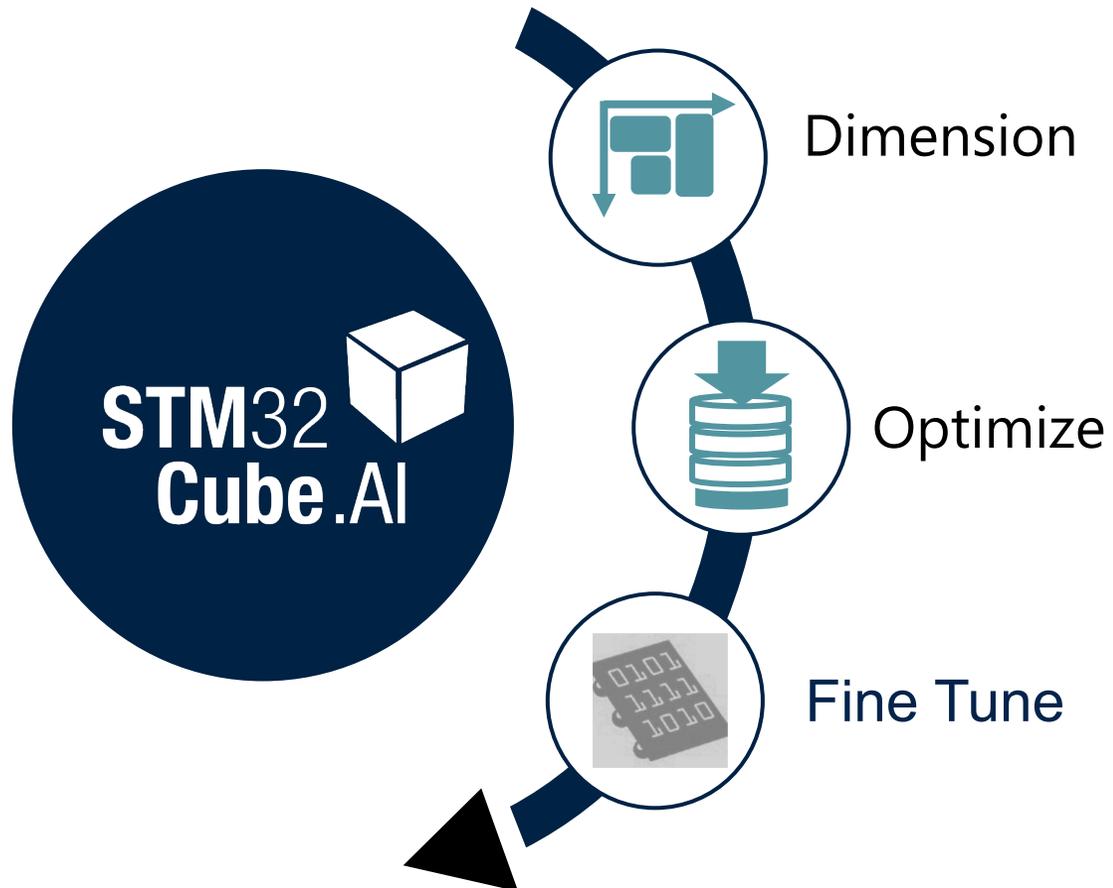
1) Gesture Detection



2) Hello World – Sine Wave



STM32Cube.AI Overview



- ✓ Quickly assess model footprint requirements
- ✓ Select and configure MCU in STM32CubeMX
- ✓ Review model layers in STM32Cube.AI

- ✓ Generate C-code for pre-trained model
- ✓ Support quantized models to reduce RAM, flash and latency with minimal loss of accuracy
- ✓ Use light run-time libraries
- ✓ Optimize for performance

- ✓ Optimize memory allocation
- ✓ Fine control of weight mapping
- ✓ Split between internal and external memory
- ✓ Update model without full FW update

And quickly iterate thanks to on-target validation

Gesture Detection Deployment

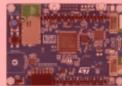


Exporting an STM32 Binary

- Data acquisition
- Impulse design
 - Create impulse
 - Spectral features
 - NN Classifier
 - Anomaly detection
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment**

Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.



ST IoT Discovery Kit



Arduino Nano 33 BLE Sense



Eta Compute ECM3532 AI Sensor



SiLabs Thunderboard Sense 2



Himax WE-I Plus



Nordic nRF52840 DK + IKS02A1



Nordic nRF5340 DK + IKS02A1



Linux boards

Exporting an STM32 AI pack

EDGE IMPULSE

- Dashboard
- Devices
- Data acquisition
- Impulse design
 - Create impulse
 - Spectral features
 - NN Classifier
 - Anomaly detection
- Retrain model
- Live classification
- Model testing
- Versioning
- Deployment**

DEPLOYMENT (BENINGO-PROJECT-1)

Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

Create library

Turn your impulse into optimized source code that you can run on any device.

 C++ library	 Arduino library	 Cube.MX CMSIS-PACK
 WebAssembly	 TensorRT library	

Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.

What method do you prefer for testing?

- Using the prebuilt binary
- Using the pack
- C++ library
- other

Apply Optimization(s)

Available optimizations for NN Classifier

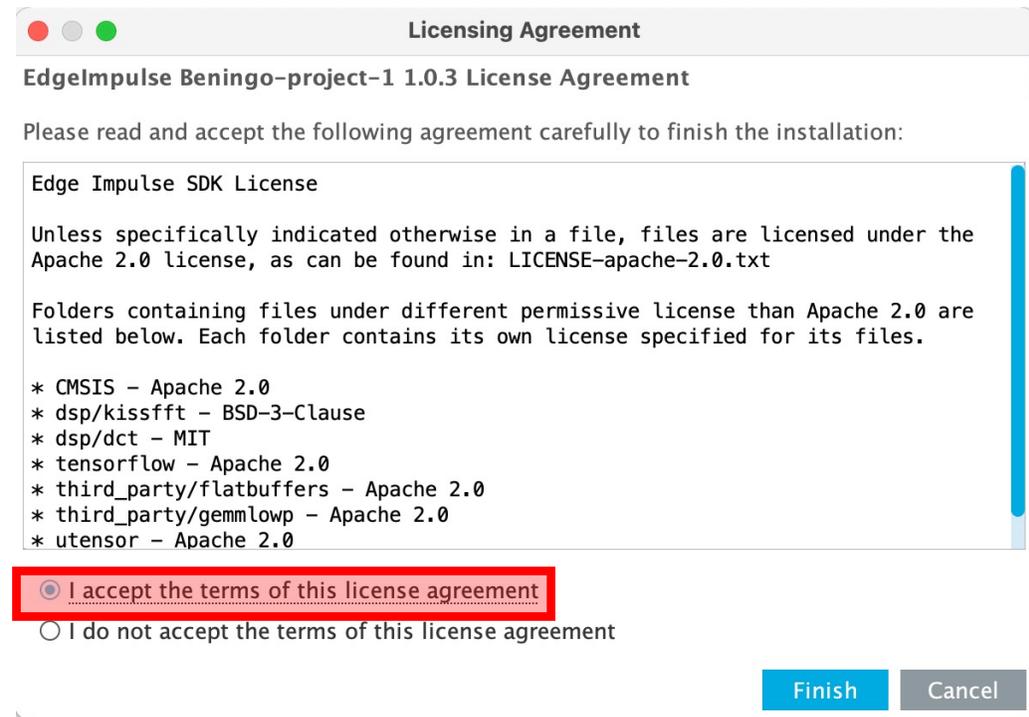
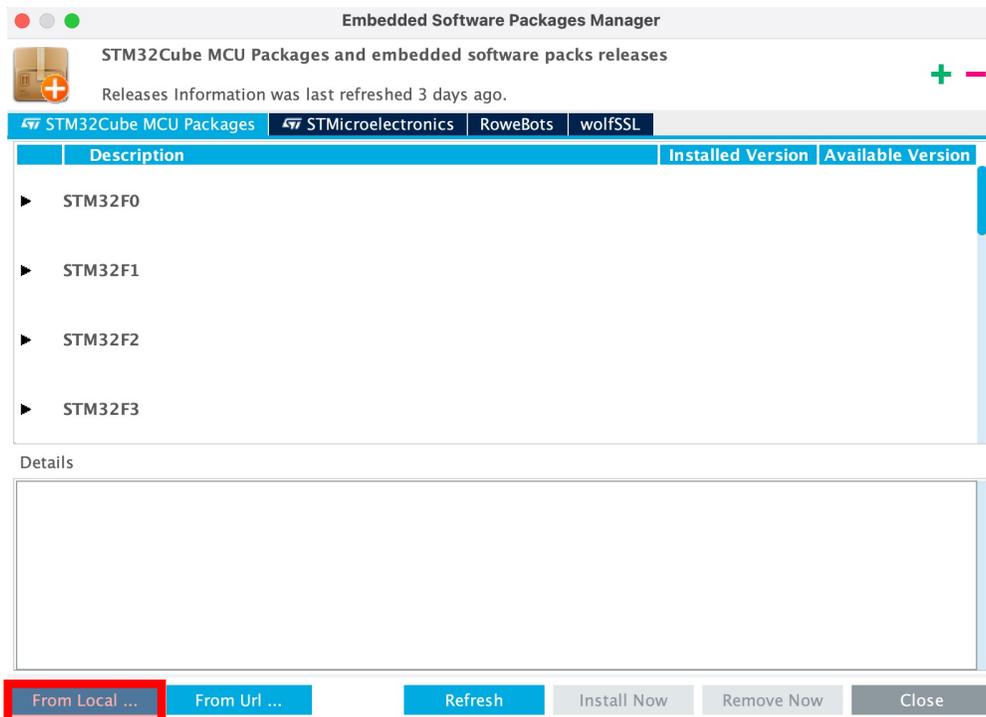
<p>Quantized (int8) ★</p> <p>Currently selected</p> <p>This optimization is recommended for best performance.</p>	<p>RAM USAGE 1.5K</p> <p>ROM USAGE 15.4K</p>	<p>LATENCY 1 ms</p> <p>ACCURACY 95.68%</p>	<p>CONFUSION MATRIX ?</p> <table border="1"> <tr> <td>87.8</td> <td>7.0</td> <td>0</td> <td>0</td> <td>5.2</td> </tr> <tr> <td>0</td> <td>99.2</td> <td>0</td> <td>0</td> <td>0.8</td> </tr> <tr> <td>0</td> <td>0</td> <td>100</td> <td>0</td> <td>0</td> </tr> <tr> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> </table>	87.8	7.0	0	0	5.2	0	99.2	0	0	0.8	0	0	100	0	0	-	-	-	-	-
87.8	7.0	0	0	5.2																			
0	99.2	0	0	0.8																			
0	0	100	0	0																			
-	-	-	-	-																			
<p>Unoptimized (float32)</p> <p>Click to select</p>	<p>RAM USAGE 1.5K</p> <p>ROM USAGE 17.7K</p>	<p>LATENCY 1 ms</p> <p>ACCURACY 95.88%</p>	<p>CONFUSION MATRIX ?</p> <table border="1"> <tr> <td>87.6</td> <td>6.6</td> <td>0</td> <td>0</td> <td>5.8</td> </tr> <tr> <td>0</td> <td>100</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>100</td> <td>0</td> <td>0</td> </tr> <tr> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> </table>	87.6	6.6	0	0	5.8	0	100	0	0	0	0	0	100	0	0	-	-	-	-	-
87.6	6.6	0	0	5.8																			
0	100	0	0	0																			
0	0	100	0	0																			
-	-	-	-	-																			

Estimate for Cortex-M4F 80MHz (ST IoT Discovery Kit)

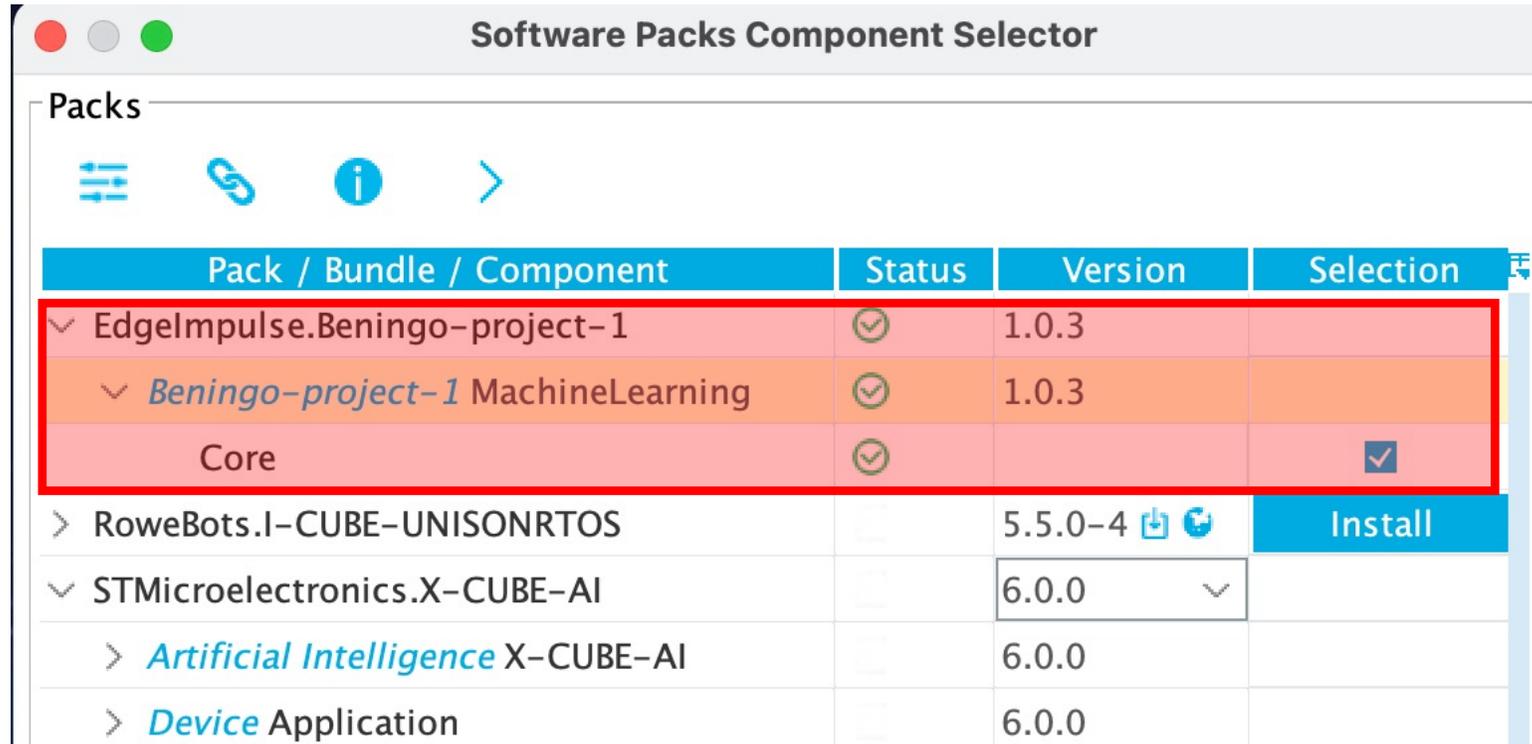
Build

Install Pack into STM32Mx Project

Help -> Manage Embedded Software Packages

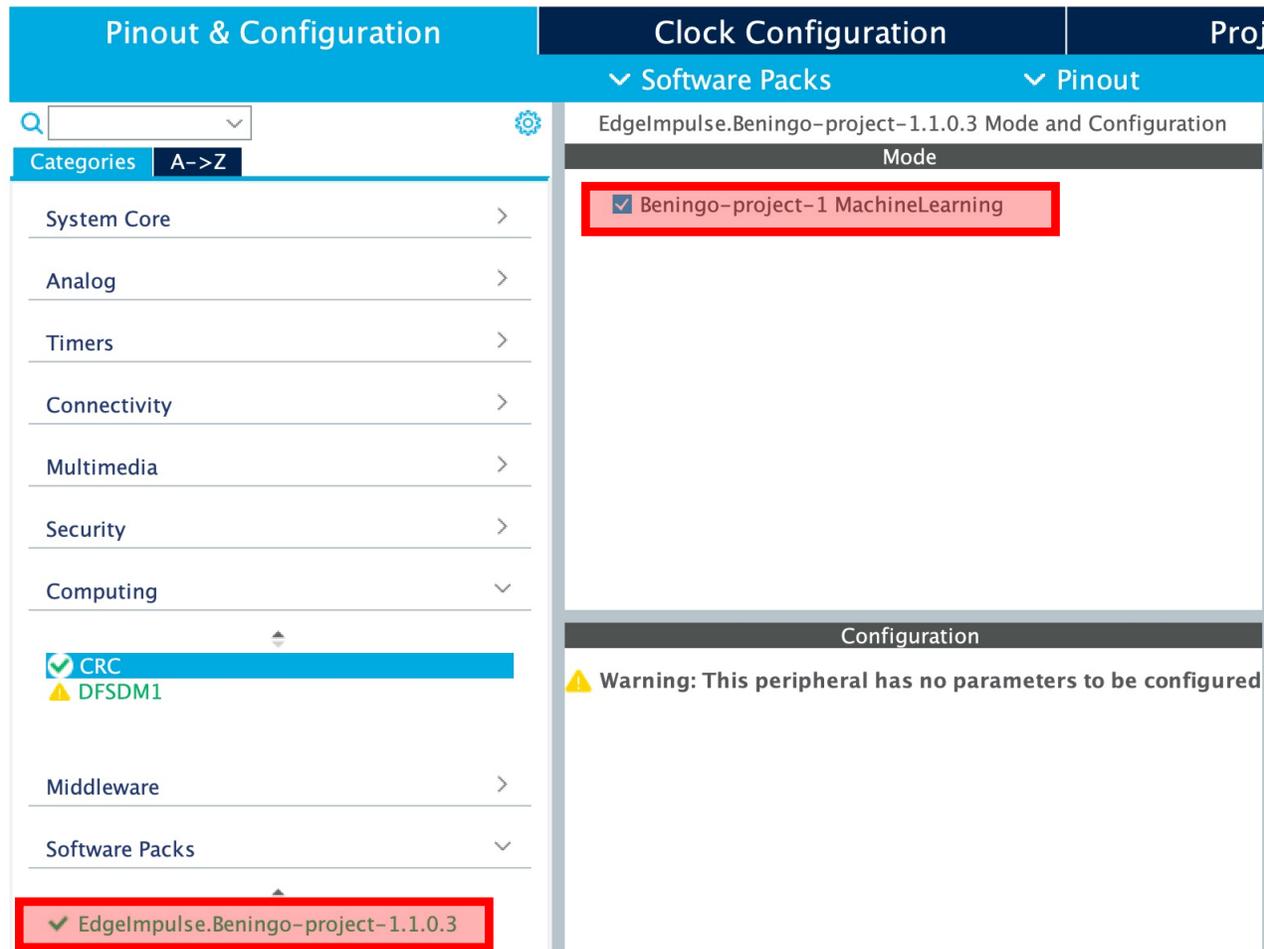


Install Pack into STM32Mx Project



Pack / Bundle / Component	Status	Version	Selection
EdgImpulse.Beningo-project-1	✓	1.0.3	
Beningo-project-1 MachineLearning	✓	1.0.3	
Core	✓		✓
> RoweBots.I-CUBE-UNISONRTOS		5.5.0-4  	Install
STMicroelectronics.X-CUBE-AI		6.0.0 	
> Artificial Intelligence X-CUBE-AI		6.0.0	
> Device Application		6.0.0	

Install Pack into STM32Mx Project



The screenshot displays the STM32CubeMX interface with the 'Pinout & Configuration' tab selected. The 'Software Packs' section is expanded, showing a list of available packs. The pack 'EdgImpulse.Beningo-project-1.1.0.3 Mode and Configuration' is selected, and its 'Mode' section is expanded to show 'Beningo-project-1 MachineLearning' with a checked checkbox. The 'Configuration' section below shows a warning: 'Warning: This peripheral has no parameters to be configured'. The 'Software Packs' list at the bottom shows 'EdgImpulse.Beningo-project-1.1.0.3' selected.

Pinout & Configuration	Clock Configuration	Proj
Search: <input type="text"/>	Software Packs	Pinout
Categories: A->Z	EdgImpulse.Beningo-project-1.1.0.3 Mode and Configuration	
System Core >	Mode	
Analog >	<input checked="" type="checkbox"/> Beningo-project-1 MachineLearning	
Timers >		
Connectivity >		
Multimedia >		
Security >		
Computing >		
✓ CRC	Configuration	
⚠ DFSDM1	⚠ Warning: This peripheral has no parameters to be configured	
Middleware >		
Software Packs >		
✓ EdgImpulse.Beningo-project-1.1.0.3		

Install Pack into STM32Mx Project

Home > STM32L475VGTx - B-L475E-IOT01A1 > STM32_ML_Gesture.ioc - Project Manager > **GENERATE CODE**

Pinout & Configuration | Clock Configuration | **Project Manager** | Tools

Project

Code Generator

Project Settings

Project Name
STM32_ML_Gesture

Project Location
/Users/beningo

Application Structure
Advanced Do not generate the main()

Toolchain Folder Location
/Users/beningo/STM32_ML_Gesture/

Toolchain / IDE
STM32CubeIDE Generate Under Root

Modify, Build, Deploy
<https://bit.ly/32ESC3N>

Running the Model

In a terminal, run the command: `edge-impulse-run-impulse`

```
Starting inferencing in 2 seconds...
Sampling... Storing in file name: /fs/device-classification.116
Predictions (DSP: 14 ms., Classification: 1 ms., Anomaly: 1 ms.):
Circle: 0.99609
Updown: 0.00000
Wave: 0.00000
anomaly score: -0.026
Finished inferencing, raw data is stored in '/fs/device-classification.116'. Use AT+UPLOADFILE to send back to Edge Impulse.
```

```
Starting inferencing in 2 seconds...
Sampling... Storing in file name: /fs/device-classification.121
Predictions (DSP: 15 ms., Classification: 0 ms., Anomaly: 2 ms.):
Circle: 0.00000
Updown: 0.00000
Wave: 0.99609
anomaly score: -0.132
Finished inferencing, raw data is stored in '/fs/device-classification.121'. Use AT+UPLOADFILE to send back to Edge Impulse.
```

Running the Model

```
Starting inferencing in 2 seconds...
Sampling... Storing in file name: /fs/device-classification.118
Predictions (DSP: 15 ms., Classification: 0 ms., Anomaly: 2 ms.):
  Circle: 0.01172
  Updown: 0.98828
  Wave: 0.00000
  anomaly score: -0.141
Finished inferencing, raw data is stored in '/fs/device-classification.118'. Use AT+UPLOADFILE to send back to Edge Impulse.
Starting inferencing in 2 seconds...
Sampling... Storing in file name: /fs/device-classification.119
Predictions (DSP: 14 ms., Classification: 1 ms., Anomaly: 1 ms.):
  Circle: 0.21094
  Updown: 0.78906
  Wave: 0.00000
  anomaly score: -0.164
Finished inferencing, raw data is stored in '/fs/device-classification.119'. Use AT+UPLOADFILE to send back to Edge Impulse.
```

What methods can be used to improve classification ?

- Running average on the output
- Monitor the anomaly value
- Set a minimum classification percentage
- All the above
- Other

Hello World Deployment



Create a New STM32CubeMx Project

1

I need to :

Start My project from MCU

[ACCESS TO MCU SELECTOR](#)

Start My project from ST Board

[ACCESS TO BOARD SELECTOR](#)

Start My project from Example

[ACCESS TO EXAMPLE SELECTOR](#)

2

Board Filters

Commercial
Part Number

Vendor

Type

B-G474E-DPOW1
B-L072Z-LRWAN1
B-L462E-CELL1
B-L475E-IOT01A1
B-L475E-IOT01A2
B-L455I-IOT01A

3



Initialize all peripherals with their default Mode ?

[Yes](#)

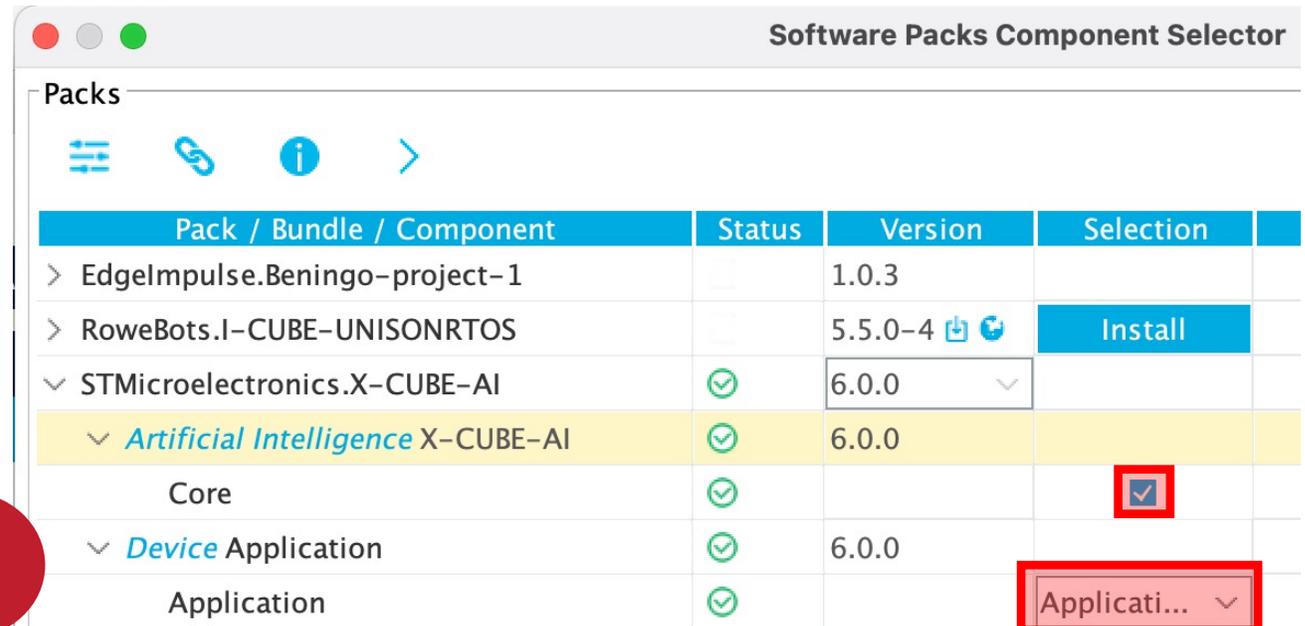
[No](#)

Add the AI Pack to the Project

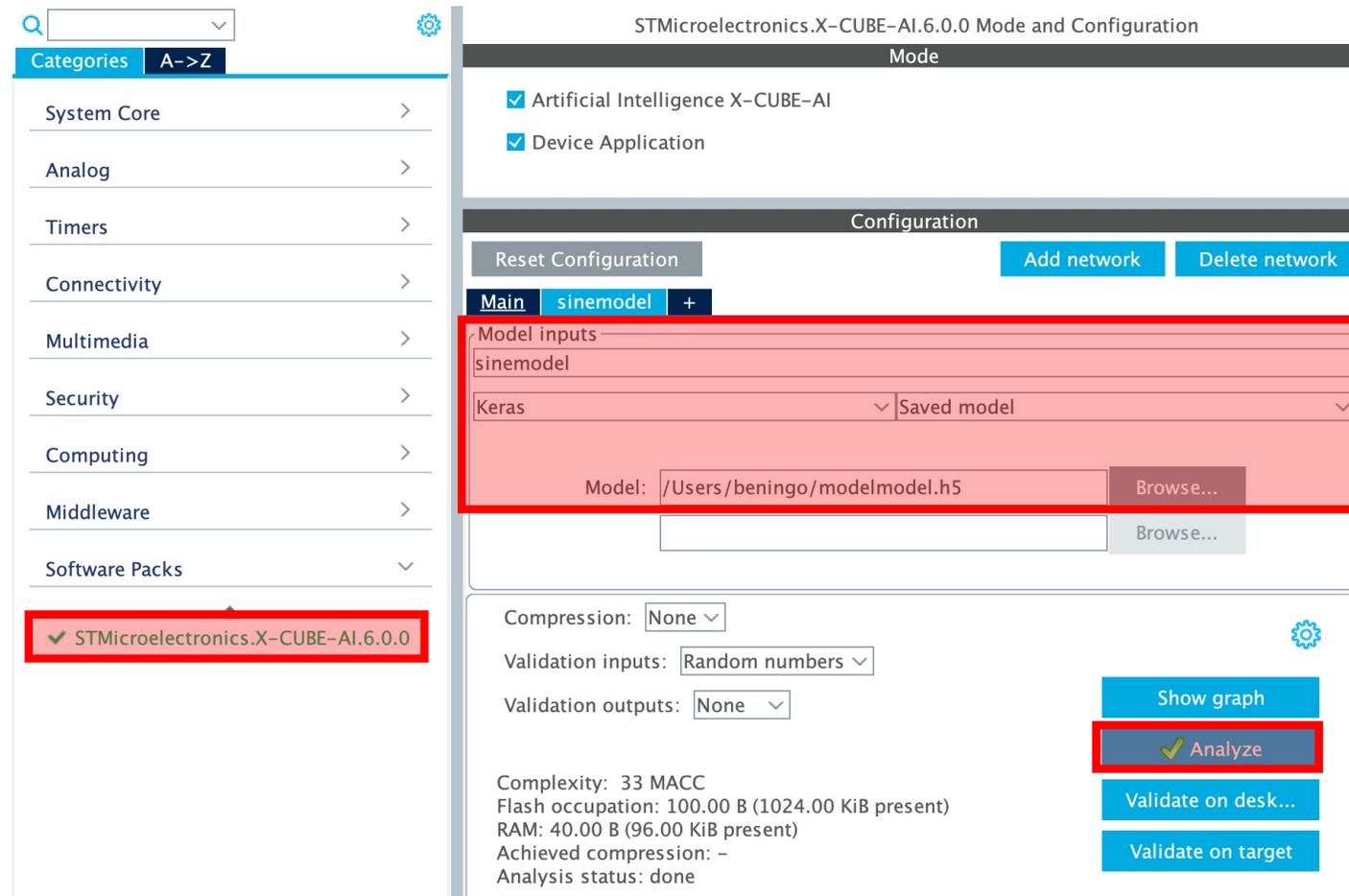
1



2



Setup and Analyze the Keras Model



The screenshot displays the STM32Cube.AI web interface for configuring and analyzing a Keras model. The interface is divided into a left sidebar and a main configuration area.

Left Sidebar: A search bar is at the top. Below it, a "Categories" menu shows "A->Z" selected. A list of categories includes System Core, Analog, Timers, Connectivity, Multimedia, Security, Computing, Middleware, and Software Packs. At the bottom, "STMicroelectronics.X-CUBE-AI.6.0.0" is selected and highlighted with a red box.

Main Configuration Area: The title is "STMicroelectronics.X-CUBE-AI.6.0.0 Mode and Configuration".

- Mode:** Includes checkboxes for "Artificial Intelligence X-CUBE-AI" and "Device Application", both of which are checked.
- Configuration:** Contains buttons for "Reset Configuration", "Add network", and "Delete network".
- Model Setup:** A section titled "Model inputs" is highlighted with a red box. It contains:
 - A text input field with "sinemodel".
 - A dropdown menu set to "Keras" and another dropdown menu set to "Saved model".
 - A text input field for the model path: "/Users/beningo/modelmodel.h5", with a "Browse..." button to its right.
 - A second "Browse..." button below the path field.
- Validation and Analysis:**
 - "Compression:" is set to "None".
 - "Validation inputs:" is set to "Random numbers".
 - "Validation outputs:" is set to "None".
 - A "Show graph" button is present.
 - An "Analyze" button is highlighted with a red box.
 - Other buttons include "Validate on desk..." and "Validate on target".
- Complexity and Status:** At the bottom, it shows "Complexity: 33 MACC", "Flash occupation: 100.00 B (1024.00 KiB present)", "RAM: 40.00 B (96.00 KiB present)", "Achieved compression: -", and "Analysis status: done".

Keras Model Analysis

Analyzing Network

```
workspace_dir : /private/var/folders/t3/4x_x6z9x0zX47b1mZr05zVzW0000gn/T/mxAl_workspace1734668119781397559350431621580:
output_dir   : /Users/beningo/.stm32cubemx

model_name   : modelmodel
model_hash   : de2ced29f4c4bafd7d1dcf4c08240154
input        : input_0 [1 items, 4 B, ai_float, FLOAT32, (1, 1, 1)]
inputs (total) : 4 B
output       : dense_1 [1 items, 4 B, ai_float, FLOAT32, (1, 1, 1)]
outputs (total) : 4 B
params #     : 25 items (100 B)
macc         : 33
weights (ro) : 100 B (100 B)
activations (rw) : 32 B (32 B)
ram (total)   : 40 B (40 B) = 32 + 4 + 4
```

Model name - modelmodel ['input_0'] ['dense_1']

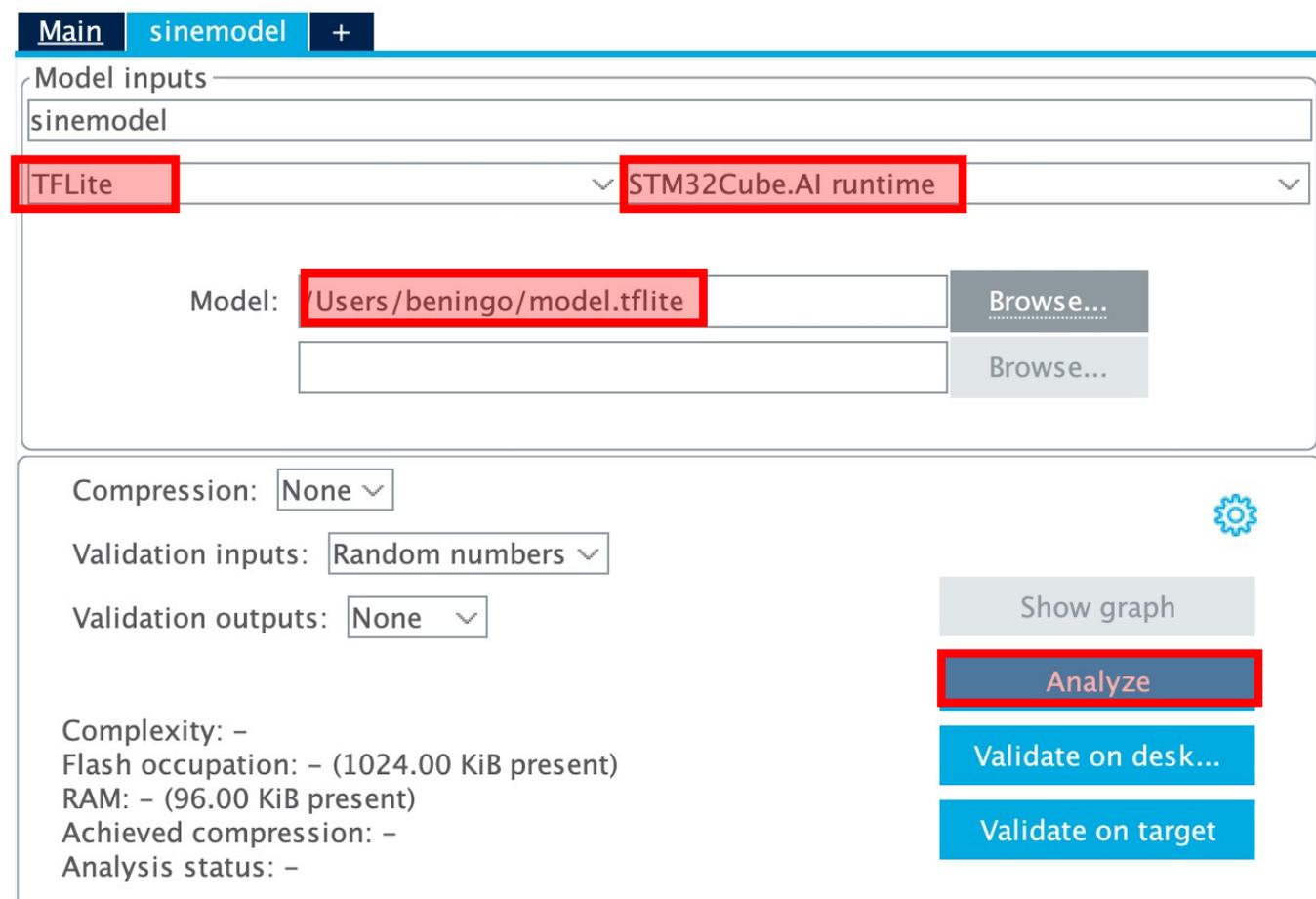
id	layer (type)	shape	param/size	macc	connected to	c_size	c_macc	c_type
0	input_0 (Input)	(c:1)						
	dense (Dense)	(c:8)	16/64	16	input_0			dense()[0]
	dense_nl (Nonlinearity)	(c:8)		8	dense			nl()[1]
1	dense_1 (Dense)	(c:1)	9/36	9	dense_nl			dense()/o[2]

model/c-model: macc=33/33 weights=100/100 activations=--/32 io=--/8

Complexity report per layer - macc=33 weights=100 act=32 ram_io=8

id	name	c_macc	c_rom	c_id
0	dense		48.5%	64.0% [0]
0	dense_nl		24.2%	0.0% [1]
1	dense_1		27.3%	36.0% [2]

Setup and Analyze the TensorFlow Lite Model



The screenshot shows the TensorFlow Lite Model Analyzer interface. At the top, there are tabs for 'Main', 'sinemodel', and a '+' icon. Below the tabs, the 'Model inputs' section contains a text field with 'sinemodel'. The 'TFLite' dropdown menu is highlighted in red, and the 'STM32Cube.AI runtime' dropdown menu is also highlighted in red. Below these, the 'Model:' field contains the path '/Users/beningo/model.tflite', which is highlighted in red, and a 'Browse...' button. There is another empty 'Browse...' button below it. The 'Compression:' dropdown is set to 'None'. The 'Validation inputs:' dropdown is set to 'Random numbers'. The 'Validation outputs:' dropdown is set to 'None'. A gear icon is visible to the right of the 'Validation inputs:' dropdown. Below these settings, there are three buttons: 'Show graph', 'Analyze' (highlighted in red), and 'Validate on desk...'. At the bottom, there are three more buttons: 'Validate on target', 'Complexity: -', 'Flash occupation: - (1024.00 KiB present)', 'RAM: - (96.00 KiB present)', 'Achieved compression: -', and 'Analysis status: -'.

TFLite Model Analysis

```

model_name      : model
model_hash     : 1c2d5a21b889b8e12b4284a72cfb10fb
input          : serving_default_dense_2_input0_int8 [1 items, 1 B, ai_i8, scale=0.024573976173996925, zero_point=-128]
inputs (total) : 1 B
output        : dense_2 [1 items, 1 B, ai_i8, scale=0.008472034707665443, zero_point=4, (1, 1, 1)]
outputs (total): 1 B
params #      : 321 items (420 B)
macc         : 321
weights (ro)  : 420 B (420 B)
activations (rw) : 32 B (32 B)
ram (total)   : 34 B (34 B) = 32 + 1 + 1

```

Model name - model ['serving_default_dense_2_input0_int8'] ['dense_2']

id	layer (type)	shape	param/size	macc	connected to
0	serving_default_dense_2_input0_int8 (Input)	(c:1)			
	dense_0 (Dense)	(c:16)	32/80	32	serving_default_dense_2_input0_int8
	nl_0_nl (Nonlinearity)	(c:16)		16	dense_0
1	dense_1 (Dense)	(c:16)	272/320	272	nl_0_nl
	nl_1_nl (Nonlinearity)	(c:16)		16	dense_1
2	dense_2 (Dense)	(c:1)	17/20	17	nl_1_nl

model/c-model: macc=353/321 -32(-9.1%) weights=420/420 activations=--/32 io=--/2

Complexity report per layer - macc=321 weights=420 act=32 ram_io=2

id	name	c_macc	c_rom	c_id
0	dense_0		10.0%	19.0% [0]
1	dense_1		84.7%	76.2% [1]
2	dense_2		5.3%	4.8% [2]

Cross Platform Validation Reports

Keras

Cross accuracy report #1 (reference vs C-model)

NOTE: the output of the reference model is used as ground truth/reference value
NOTE: ACC metric is not computed ("--classifier" option can be used to force it)

acc=n.a., rmse=0.000000128, mae=0.000000104, l2r=0.000000225

Evaluation report (summary)

Mode	acc	rmse	mae	l2r	tensor
X-cross #1	n.a.	0.000000128	0.000000104	0.000000225	dense_5, ai_float, [(1, 1, 1)], m_id=[2]

X-cross (l2r) #1 error : 2.25302543e-07 (expected to be < 0.01)

TF Lite

Cross accuracy report #1 (reference vs C-model)

NOTE: the output of the reference model is used as ground truth/reference value
NOTE: ACC metric is not computed ("--classifier" option can be used to force it)

acc=n.a., rmse=0.000000000, mae=0.000000000, l2r=0.000000000

Evaluation report (summary)

Mode	acc	rmse	mae	l2r	tensor
X-cross #1	n.a.	0.000000000	0.000000000	0.000000000	dense_2, ai_i8, [(1, 1, 1)], m_id=[2]

X-cross (rmse) #1 error : 0.00000000e+00 (expected to be < 0.01)

Creating txt report file /Users/benigo/.stm32cubemx/sinemodel_validate_report.txt
elapsed time (validate): 1.453s

Generate the Model

1

Pinout & Configuration | Clock Configuration | Project

Project

Code Generator

Advanced Settings

Project Settings

Project Name
MachineLearning_HelloWorld

Project Location
/Users/beningo/ML_HelloWorld

Application Structure
Advanced Do not generate the main()

Toolchain Folder Location
/Users/beningo/ML_HelloWorld/MachineLearning_HelloWorld/

Toolchain / IDE
STM32CubeIDE Generate Under Root

Linker Settings

Minimum Heap Size

Minimum Stack Size

Mcu and Firmware Package

Mcu Reference
STM32L475VGTx

Firmware Package Name and Version
STM32Cube FW_L4 V1.17.0

Use Default Firmware Location

2

GENERATE CODE

The Application Template

```

122  /* Initialize all configured peripherals */
123  MX_GPIO_Init();
124  MX_CRC_Init();
125  MX_DFSDM1_Init();
126  MX_I2C2_Init();
127  MX_QUADSPI_Init();
128  MX_SPI3_Init();
129  MX_USART1_UART_Init();
130  MX_USART3_UART_Init();
131  MX_USB_OTG_FS_PCD_Init();
132  MX_X_CUBE_AI_Init();
133  /* USER CODE BEGIN 2 */
134
135  /* USER CODE END 2 */
136
137  /* Infinite loop */
138  /* USER CODE BEGIN WHILE */
139  while (1)
140  {
141      /* USER CODE END WHILE */
142
143      MX_X_CUBE_AI_Process();
144      /* USER CODE BEGIN 3 */
145  }
146  /* USER CODE END 3 */
147  }

```

```

..
78 /* Private user code -----*/
79 /* USER CODE BEGIN 0 */
80 int _write(int file, char *ptr, int len)
81 {
82     HAL_UART_Transmit(&huart1, (uint8_t *)ptr, len, 0xffff); // send message via UART
83
84     return len;
85 }
--
254 /* 2 - main loop */
255 do {
256     /* 1 - acquire and pre-process input data */
257     res = acquire_and_process_data(in_data);
258     /* 2 - process the data - call inference engine */
259     if (res == 0)
260     {
261         res = ai_run(in_data, out_data);
262     }
263
264     /* 3- post-process the predictions */
265     if (res == 0)
266     {
267         res = post_process(out_data);
268     }
269 } while (res==0);

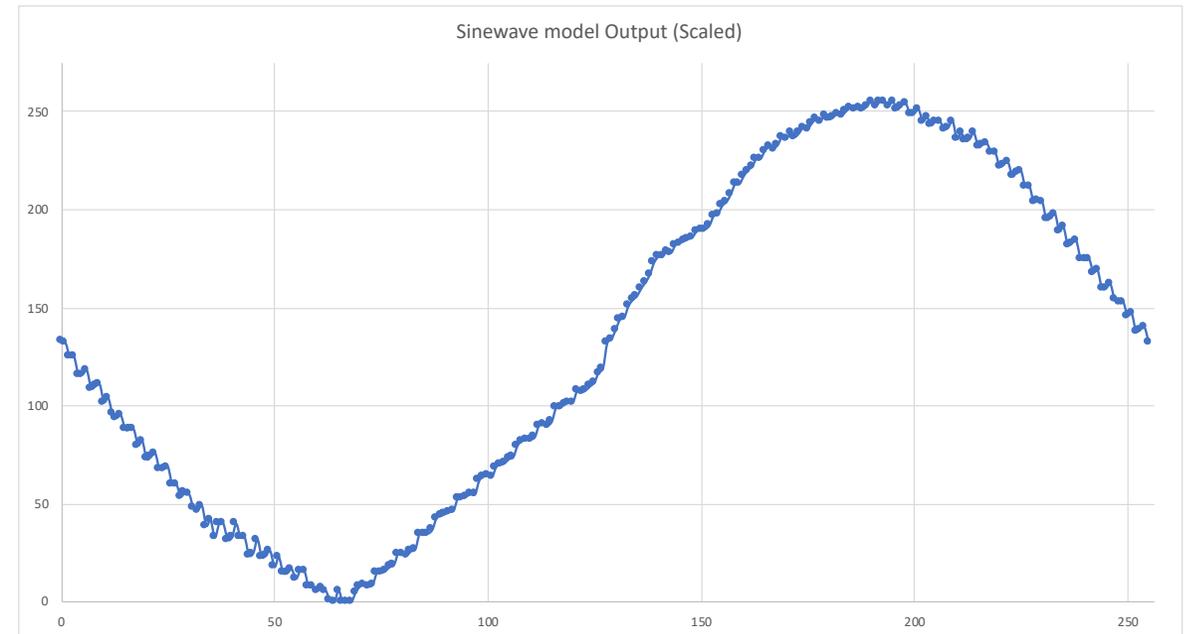
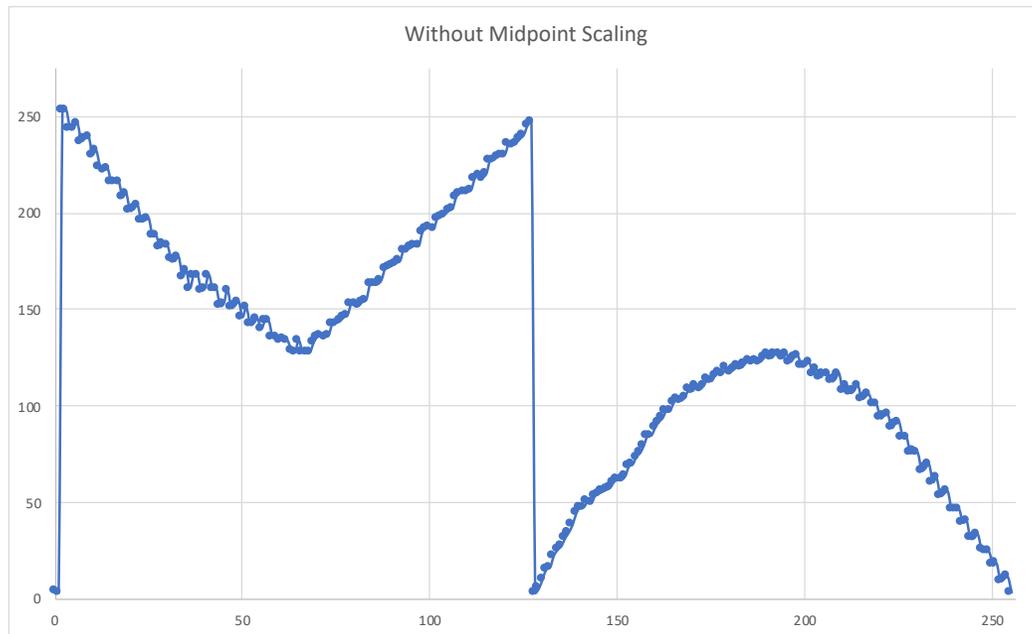
```

The Details

```
174 /* USER CODE BEGIN 2 */
175 int acquire_and_process_data(void * data)
176 {
177     static uint8_t position = 0;
178     uint8_t * Value = data;
179
180     * Value = position;
181
182     position++;
183
184     return 0;
185 }
```

```
187 int post_process(void * data)
188 {
189     uint8_t * Value = data;
190
191     if(*Value >= 128)
192     {
193         *Value -= 128;
194     }
195     else
196     {
197         *Value += 128;
198     }
199
200     return 0;
201 }
```

Results



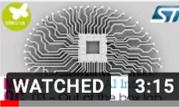
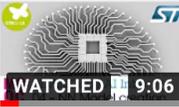
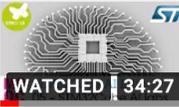
Next Steps

- Connect the output to a PWM LED channel
- Setup a DAC and drive an output voltage
- Configure the rate at which the inference runs (frequency control)
- Try and compare the Keras model behavior
- Improve the training model to provide a more accurate sine wave

Going Further

 FP-AI-SENSING1		People activity recognition Audio scene classification
 FP-AI-NANOEDG1		Condition-based monitoring
 FP-AI-VISION1		Person presence detection Food classification
 FP-AI-FACEREC1		Face recognition

<https://bit.ly/3nf99EZ>

1		Introduction to STM32Cube.AI - 1 Marketing introduction STMicroelectronics WATCHED 10:23
2		Introduction to STM32Cube.AI - 2 Theory of AI STMicroelectronics WATCHED 5:24
3		Introduction to STM32Cube.AI - 3 Out of the box lab STMicroelectronics WATCHED 3:15
4		Introduction to STM32Cube.AI - 4 NN Model creation using Keras STMicroelectronics WATCHED 9:06
5		Introduction to STM32Cube.AI - 5 STM32Cube.AI labs STMicroelectronics WATCHED 34:27

What would you like to learn more about?

- Developing Keras based models
- How to develop ML test cases
- Building more complex ML edge applications
- other

Thank you for attending

Please consider the resources below:

- www.beningo.com
 - Blog, White Papers, Courses
 - Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>



From www.beningo.com under

- Blog > CEC – Machine Learning Application Design using STM32 MCUs



Thank You

Sponsored by

