ModusToolbox™

Machine Learning Configurator guide

About this document

Version

1.20

Scope and purpose

The ModusToolbox™ Machine Learning (ML) Configurator is used in ML applications for adapting a pretrained learning model to an Infineon target platform. The tool accepts a pretrained ML model and generates an embedded model (as a library), which can be used along with your application code for a target device. The ModusToolbox™ ML Configurator also lets you fit the pretrained model of choice to the target device with a set of optimization parameters.

Intended audience

This document helps application developers understand how to use the ML Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- ML – Machine Learning
- NPZ – NumPy array in zipped format. When unzipped, it provides validation data as NumPy arrays used by the tool.
- MAE – Maximum Absolute Error
- MACC – Machine-learned ASAS Classification Catalog

Reference documents

Refer to the following documents for more information as needed:

- [ModusToolbox™ Machine Learning user guide](#)
- [Machine Learning Inference library readme file](#)
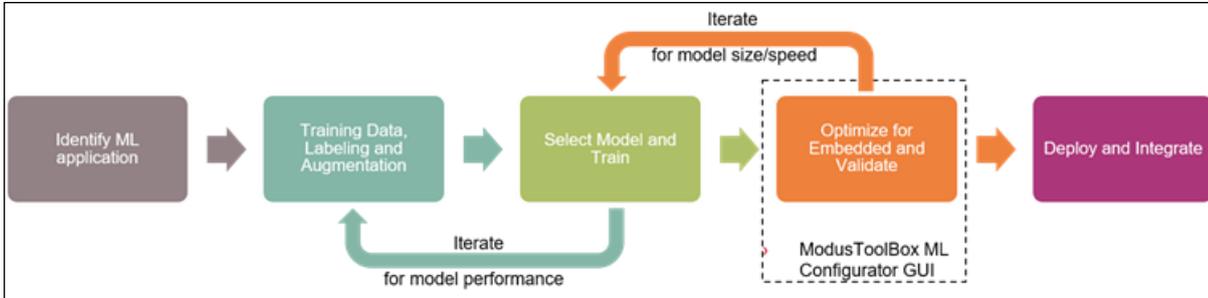- [ModusToolbox™ user guide](#)

User Guide

www.infineon.com

Please read the Important Notice and Warnings at the end of this document

page 1 of 15

002-32756 Rev. *B

2021-09-08

Table of Contents

1 Overview

The following shows the design flow for a typical application. The ModusToolbox™ ML Configurator GUI forms a vital part in fitting the model to the target platform.

The ModusToolbox™ ML Configurator is required to support the machine learning tool ecosystem. This tool forms the central asset that brings together other assets of the machine learning tool ecosystem, including Core tools, Inference engine, etc. as described in the ModusToolbox™ Machine Learning user guide. You can create a new application using the ML code example, or you can add the ML library to an existing application using the Library Manager.

1.1 Supported library

Name	Version	Link
Machine Learning Inference Engine library	1.2	https://github.com/Infineon/ml-inference

2 Launch the ML Configurator

There are numerous ways to launch the ML Configurator, and those ways depend on how you use the various tools in ModusToolbox™ software.
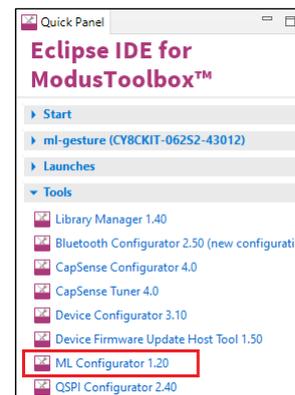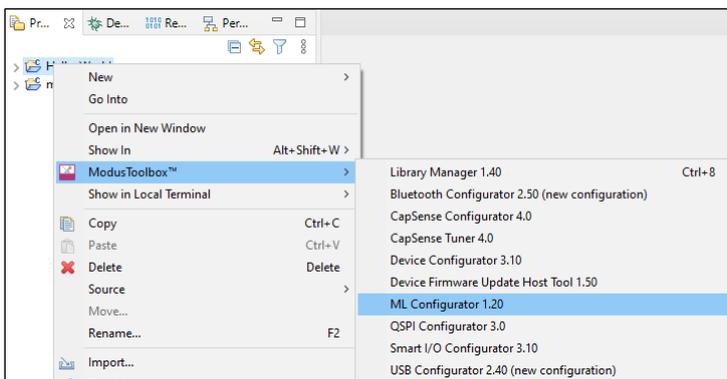
2.1 make command

As described in the [ModusToolbox™ user guide](#) build system chapter, you can run numerous make commands in the application directory, such as launching the ML Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make open CY_OPEN_TYPE=ml-configurator
```

This command opens the ML Configurator GUI for the specific application in which you are working.

2.2 Eclipse IDE

If you use the Eclipse IDE for ModusToolbox™, you can launch the ML Configurator for the selected application. In the Project Explorer, right-click on the project and select **ModusToolbox™ > ML Configurator <version>**. You can also click the ML Configurator link in the IDE Quick Panel.

Similar to the make command method, launching the ML Configurator using the Eclipse IDE opens the tool for the selected application. Refer to the [Eclipse IDE for ModusToolbox™ user guide](#) for details about the IDE.

2.3 Executable (CLI)

The ML Configurator executable can be run from the command line, and it also has a "cli" version of the executable as well. Running configurator executables from the command line can be useful as part of batch files or shell scripts to re-generate the source code based on the latest configuration settings. The exit code for the executable is zero if the operation is successful, or non-zero if the operation encounters an error. For more information about the command-line options, run the executable using the –h option.

2.4   Executable (GUI)

If you don't have an application or if you just want to see what the configurator looks like, you can launch the ML Configurator GUI by running its executable as appropriate for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, it is installed here:

<install_dir>/ModusToolbox/tools_<version>/ml-configurator-<version>

When launched this way, the ML Configurator opens without any settings configured. You can either open a specific configuration file or create a new one. See [Menus](#) for more information.

3 Quick start

This section provides a simple workflow for how to use the ML Configurator.

1. Create a new application based on a code example. For example:

    https://github.com/Infineon/mtb-example-ml-profiler

2. Launch the ML Configurator GUI.

3. Enter/select Project elements and Model elements.

4. Click **Generate Source** and specify the configuration (*.mtbml*) file, as needed. See the Generate source button section.

5. Click **Validate in Desktop** to open the Validation dialog, and use the various setting to visualize the output data.

4   GUI description

The ML Configurator GUI contains menus and various fields to generate data for the selected project.
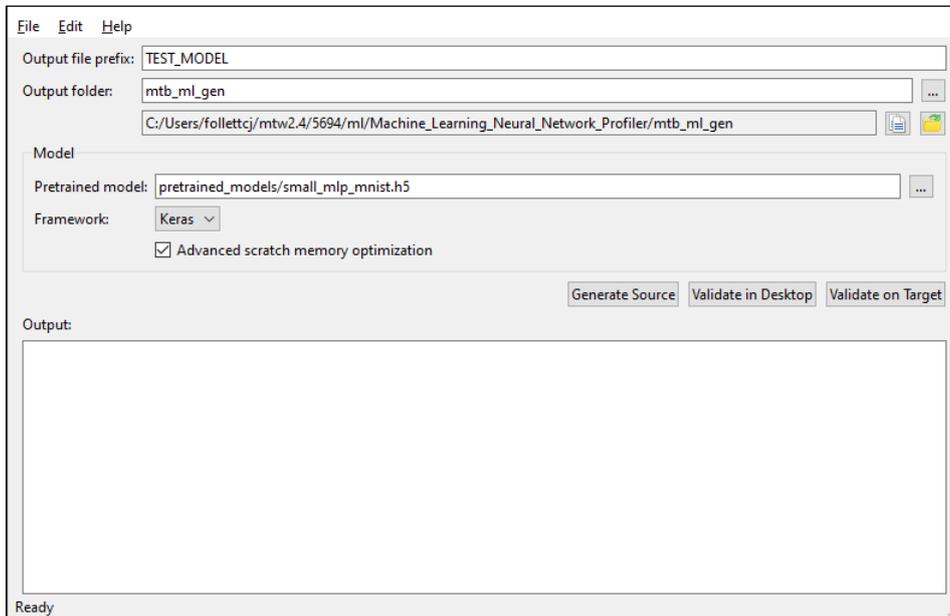
4.1   Menus

4.1.1   File

- **New** – Creates a new file with new configuration.
- **Open** – Opens the configuration file.
- **Close** – Closes the configuration file.
- **Save** – Saves the existing file.
- **Save As** – Saves the existing file under a different name.
- **Open in System Explorer** – This opens your computer's file explorer tool to the folder that contains the *.mtbml* file.
- **Settings**
  - **Automatically generate code after save** – Select this check box to automatically generate code upon saving the configuration file. By default, this check box is not selected.
- **Exit** – Closes the configurator.

4.1.2   Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

4.1.3   Help

- **View Help** – Opens this document.
- **About ML Configurator** – Opens the About box for version information.

4.2   Project elements

4.2.1   Output file prefix

This is the name used to generate files from this tool.

4.2.2   Output folder

This the name and location where the generated files are placed, relative to the location of the saved configuration file.

4.3   Model elements

4.3.1   Pretrained model

This parameter lets you select a pretrained model file from 3rd party ML frameworks.

Note: For this version, the ML Configurator supports the Keras framework; hence, only h5 type file formats are available. Subsequent versions of the tools will support for other frameworks and file formats.

4.3.2   Framework

This lists the framework that was used to create the pretrained model file being input. Typically, this information can be read from the pretrained model file itself and hence will be auto populated in the field.

Note: For this version, the ML Configurator only supports the Keras framework.

4.3.3   Advanced scratch memory optimization

This check box is selected by default. When selected, this option reduces the amount of scratch memory needed with minimal to no impact on the accuracy of the model.

4.4   Generate source button

The main purpose of this button is to analyze and validate the pretrained model, as well as generate source data. The output of this analysis displays as output messages. The configurator also generates an output file in the specified Output file location.

When you click this button for the first time without an existing *.mtbml* file , the configurator asks you to specify its file name and location. When you press this button on subsequent occasions after already specifying the *.mtbml file, the configurator skips this step and begins generating source data.

4.5   Validate in Desktop button

This button opens the Validation dialog with options to run validation using the computer. As with the **Generate Source** button, this button checks to see if there is a *.mtbml* file and asks you to specify the file name and location if one does not exist.

4.6 Validate on Target button

This button opens the Validation dialog with options to run validation using the connected device. As with the **Generate Source** button, this button checks to see if there is a *.mtbml* file and asks you to specify the file name and location if one does not exist.

Note: *Ensure that you build the streaming validation firmware on the same host OS you are using to run the ML Configurator.*

4.7 Output messages

The output messages section displays various messages about the model and the generation of source.

```
Output:

2021-03-01 08:40:37.184643: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions
that this TensorFlow binary was not compiled to use: AVX2

[INFO] Printing model summary:

Model: "sequential"

Layer (type)              Output Shape              Param #
=================================================================
conv2d (Conv2D)           (None, 128, 3, 8)         104

max_pooling2d (MaxPooling2D) (None, 42, 1, 8)        0

conv2d_1 (Conv2D)         (None, 42, 1, 16)         528
```

4.8 Status bar

The status bar displays various information about the status of the tool.

4.9 Validation dialog

The Validation dialog provides controls to evaluate and visualize the output data. The dialog's appearance and functionality vary depending on whether it was launched from the **Validate in Desktop** or the **Validate on Target** button.
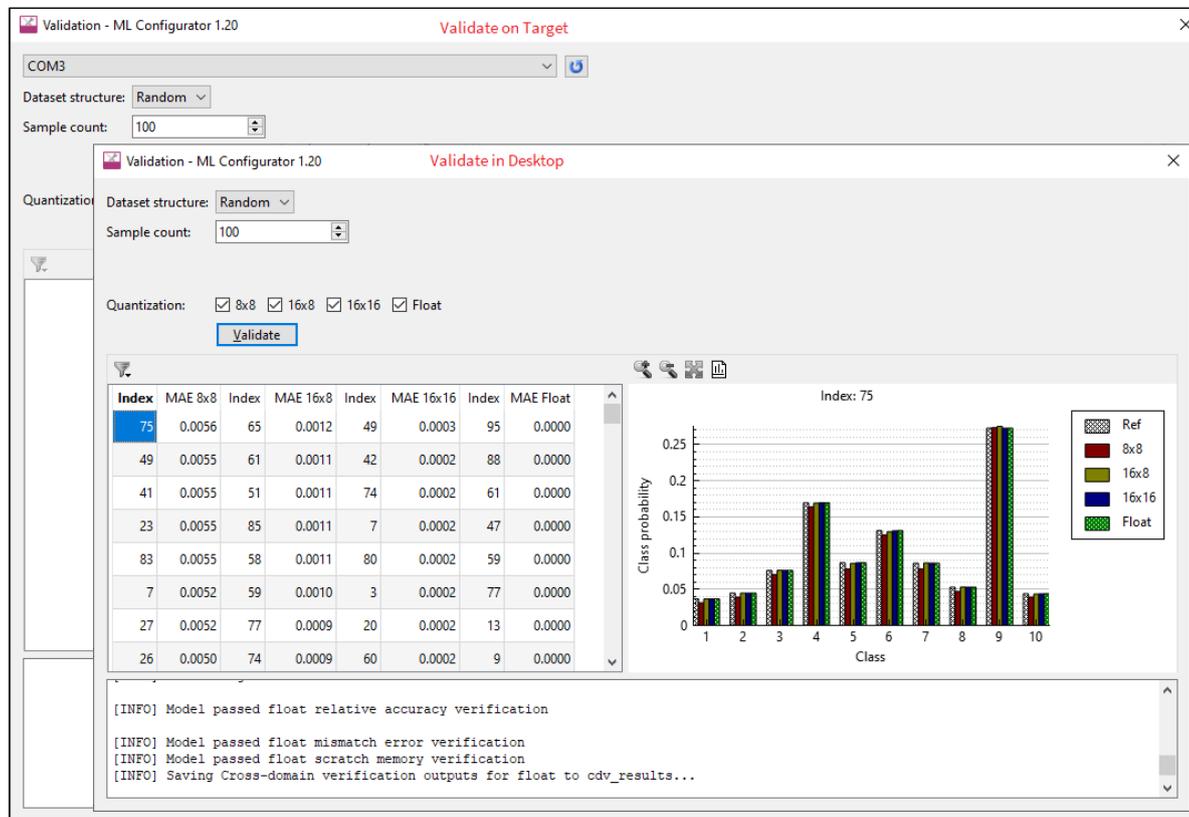
4.9.1 COM port

This pull-down menu lets you specify the COM port that connects to the target device.

4.9.2 Dataset structure

This pull-down lets you select the type of data to use as validation input. This can be random data generated by the ML Configurator or an input file that is typically a data capture from an actual application source. Options include:

- **Random** – Random data generated by the configurator.

- **NPZ** – Validation data as NumPy arrays in zipped format.
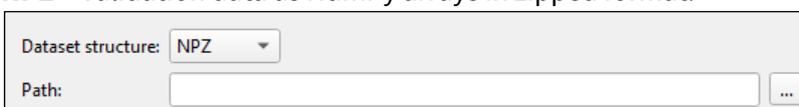
- **Folder** – Validation data as folders containing JPEG files.
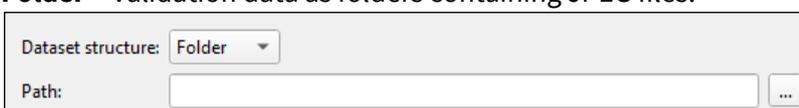
- **ML** – ML Format: A CSV file used by ml-coretools that contains only numeric data, with no header and no sample ID columns.
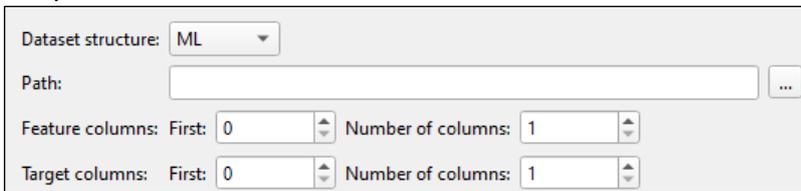
4.9.3   Sample count

If you set the **Dataset structure** field to "Random," this field displays to specify the number of rows of data to count for the sample.

4.9.4   Path

If you select a **Dataset structure** other than "Random," this field displays to specify the file name and location to use for data validation. When **Dataset structure** is set to "Folder," the path should specify a folder location.

4.9.5   Feature columns/number of columns

If you set the **Dataset structure** field to "ML," this field specifies the first and number of feature columns. The first column is 0.

4.9.6   Target columns/number of columns

If you set the **Dataset structure** field to "ML," this field specifies the first and number of target columns. The first column is 0.

4.9.7   Quantization

These check boxes let you chose 8-bit, 16-bit, and/or float quantization options for the validation results to display in the table and graph, as well as for one or more output files to be generated. For the **Validate in Desktop** option, you can select multiple options; for the **Validate on Target** option, you can only select one.

4.9.8   Validate button

This button generates the validation result based on the selected data input and the model.

4.9.9   Table

The Validation dialog contains a table that displays the validation result data with the input Index and its corresponding MAE value. The table may have up to three columns, depending on the selected quantization and filter options.

- Each of the table column headings provides a sorting option to view the data in ascending or descending order.
- A **Filter** button above the table allows you to select and deselect the quantization options, and thus show or hide the corresponding table.
- When you select each of the Indexes and their corresponding MAE value, the data displays in graph form on the right side of the dialog.

4.9.10   Graph

The graph is a simple representation of the data to allow for easy viewing. The graph includes several buttons as follows:

- Zoom in – Make the graph bigger.
- Zoom out – Make the graph smaller.
- Zoom to fit – Grow or shrink the graph to fit the size of the graph area.
- Export to file – Save the graph as currently presented to a file. Formats include png, jpg, and bmp.

You can also click and drag your mouse to pan the graph in any direction.

4.9.11   Results

This area displays the analysis results, which includes memory usage and usage of MACCs, if these exist in hardware.

5 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.10	Added **Open in System Explorer** in **File** menu. Changed how the target/feature columns are specified in Validation dialog.
1.20	Added **Edit** menu and **Undo/Redo** commands. Added **Advanced scratch memory optimization** check box. Added support for Validate on Target.

Revision history

Revision	Date	Description
**	03/16/2021	New document.
*A	04/30/2021	Updated to version 1.10.
*B	09/08/2021	Updated to version 1.20.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.