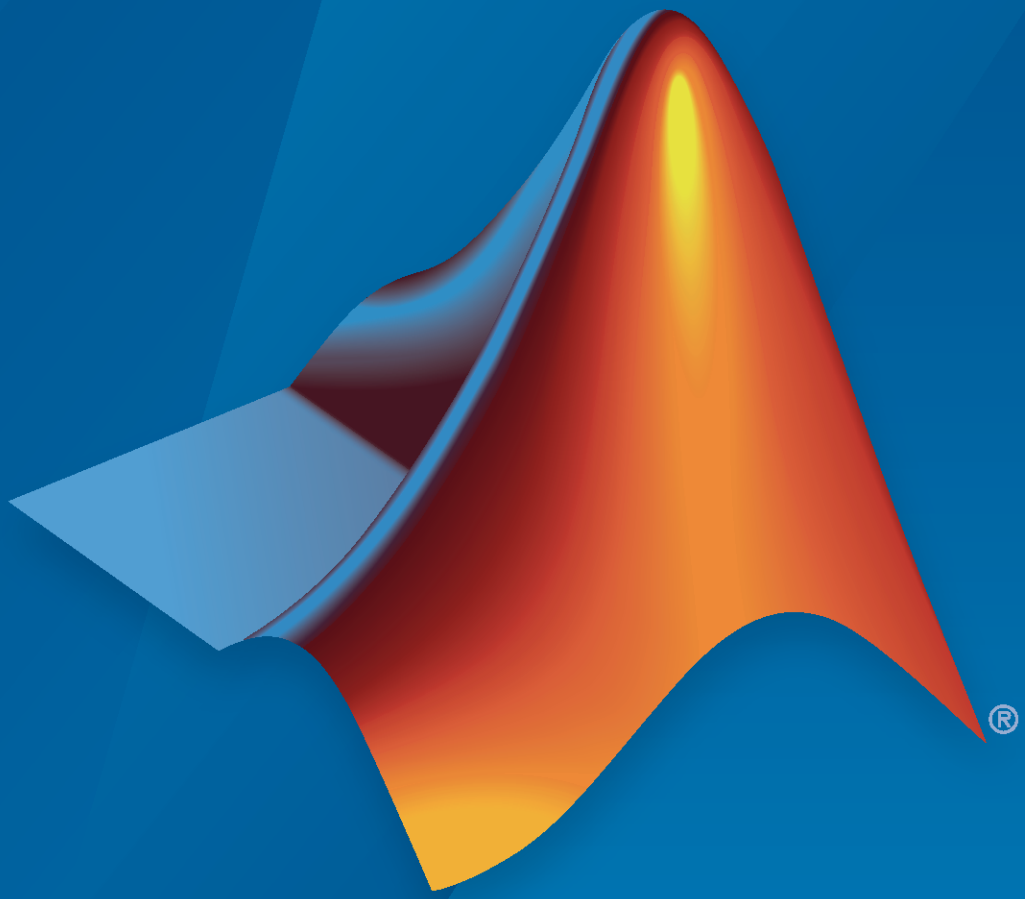
Deep Learning HDL Toolbox™ Release Notes



MATLAB®

MathWorks®

How to Contact MathWorks

Latest news: `www.mathworks.com`

Sales and services: `www.mathworks.com/sales_and_services`

User community: `www.mathworks.com/matlabcentral`

Technical support: `www.mathworks.com/support/contact_us`

Phone: 508-647-7000

The MathWorks, Inc.
1 Apple Hill Drive
Natick, MA 01760-2098

Deep Learning HDL Toolbox™ Release Notes

Contents

R2021a

R2020b

R2021a

Version: 1.1

New Features

Compatibility Considerations

Custom directed acyclic graph (DAG) network support

Compile and deploy your custom DAG networks. Retrieve predictions from the deployed network by using MATLAB®. For a list of supported networks, see “Supported Networks, Layers, Boards, and Tools”. The deep learning compiler analyzes the DAG network graph and generates the instructions, address mapping, and schedule to run the DAG network on the new deep learning processor. Deploy larger DAG networks onto FPGA boards with smaller resources by quantizing your DAG networks to use `int8` data types. See “Quantization of Deep Neural Networks”.

Performance estimation and profiling

Estimate performance by using the `estimatePerformance` function on the `dlhdl.ProcessorConfig` object before building your custom deep learning processor. For more information, see `estimatePerformance`. Retrieve the processor configuration of the shipping (reference) bitstream, by using the `dlhdl.ProcessorConfig` object. See `dlhdl.ProcessorConfig`. Perform design space exploration to find the deep learning processor configuration that fits your performance requirements by comparing the performance of your custom deep learning processor configuration to the performance of the shipping (reference) bitstream processor configuration.

You cannot estimate performance by using the `estimate` method for the `dlhdl.Workflow` object. For more information, see “Functionality being removed or changed” on page 1-3.

Resource estimation

Estimate resource utilization by using the `estimateResources` function on the `dlhdl.ProcessorConfig` object before building your custom deep learning processor. For more information, see `estimateResources`. Retrieve resource utilization of shipping (reference) bitstreams by using the `getBuildInfo` function on the `dlhdl.Workflow` object. See `getBuildInfo`. Perform design space exploration to find the deep learning processor configuration that fits your FPGA resource budget by comparing the resource utilization of your custom deep learning processor configuration to the resource utilization of the shipping (reference) bitstream.

Updated supported layers

Deep Learning HDL Toolbox now provides support for these layers:

- Addition layer
- Depth-wise separable convolution layer
- Depth concatenation layer

For `int8` data type quantization, Deep Learning HDL Toolbox now provides support for these layers:

- Average pooling layer
- Global average pooling layer
- Addition layer
- Clipped ReLU layer
- Leaky ReLU layer
- Depth-wise separable convolution layer

See “Supported Networks, Layers, Boards, and Tools”.

MATLAB Emulation for validate method of dlquantizer object

Validate the performance of your quantized network by comparing the prediction accuracy of your quantized network to that of your nonquantized network, without the need for hardware by using MATLAB emulation. See `validate`.

dlhdl.Workflow name-value argument pair update

For the list of name-value pair arguments that have been removed from `dlhdl.Workflow`, see “Functionality being removed or changed” on page 1-3.

Updates to supported software

Deep Learning HDL Toolbox has been tested with:

- Xilinx® Vivado® Design Suite 2020.1
- Intel® Quartus® Pro 18.1

Functionality being removed or changed

estimate function for dlhdl.Workflow object has been removed
Errors

This function has been removed.

'ProcessorConfig' option in dlhdl.Workflow has been removed
Errors

The 'ProcessorConfig' name-value pair for `dlhdl.Workflow` has been removed.

R2020b

Version: 1.0

New Features

Introducing Deep Learning HDL Toolbox: Prototype and implement deep learning networks on FPGAs and SoCs

With Deep Learning HDL Toolbox, you can prototype and implement deep learning networks on FPGAs and SoCs. Deploy and run deep learning networks on supported Xilinx and Intel FPGA and SoC devices. Improve deep learning network design, performance, and resource utilization by using profiling and estimating tools to explore tradeoffs and customize the network. Using HDL Coder™, you can generate HDL and an IP core to target FPGAs or SoCs.

Prototype on FPGAs

Use MATLAB and fixed bitstreams to compile, deploy, and run inference for pretrained series networks on target Intel and Xilinx FPGA and SoC boards. For more information, see Prototype Deep Learning Networks on FPGA.

Custom series network support

Compile and deploy your custom series networks using the same fixed-bitstreams as the pre-trained networks. For more information, see Prototype Deep Learning Networks on FPGA and SoCs Workflow.

Portable Verilog and VHDL code

Generate portable Verilog® and VHDL® code from your series deep learning network.

Tune user-configurable parameters

Customize your deep learning network implementation by tuning user-configurable parameters such as Thread Number, Input, and Output Memory Size. For more information, see Custom Processor Configuration Workflow.

Custom board support

Integrate the code generated from your customized design into your reference design for deploying to your custom board. For more information, see Generate Custom Processor IP.

Performance estimation and profiling

Gather layer-level latency and throughput estimates for your series networks. For more information, see estimate.

Hardware Support

Prototype and deploy deep learning networks to Intel and Xilinx FPGA boards. Use Ethernet based LIBIIO to rapidly deploy your series deep learning networks to your target Intel and Xilinx FPGA and SoC boards. For more information, see LIBIIO/Ethernet Connection Based Deployment.

Support Package for Intel FPGA and SoCs

You can use the Deep Learning HDL Toolbox Support Package for Intel FPGA and SoC Devices to communicate with, deploy series networks, and retrieve inference results from target Intel FPGA and SoC platforms. To download the support package, use the Add-on Explorer. For more information, see Deep Learning HDL Toolbox Support Package for Intel FPGA and SoC Devices.

Support Package for Xilinx FPGA and SoCs

You can use the Deep Learning HDL Toolbox Support Package for Xilinx FPGA and SoC Devices to communicate with, deploy series networks, and retrieve inference results from target Xilinx FPGA and SoC platforms. To download the support package, use the Add-on Explorer. For more information, see Deep Learning HDL Toolbox Support Package for Xilinx FPGA and SoC Devices.