KalEdge Lite

End-To-End ML to FPGA Deplyment

Laboratory Guide

Objective

By completing this lab, you will be able to:

- Use the KalEdge-Lite graphical workflow to load the dataset and configure the input format.
- Select, configure, and train different neural network architectures (Baseline, Student, QKeras) directly from the application.
- Compare model variants using the metrics and reports generated by the platform.
- Generate an FPGA-oriented implementation through the hls4ml integration within KalEdge-Lite.
- Export an HLS project ready for hardware synthesis and deployment.

Pre-requisites

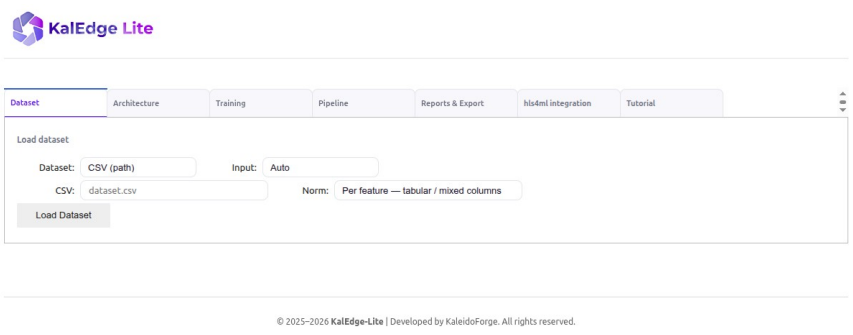
Before starting, make sure you have:

- Access to KalEdge-Lite
- A valid HyperFPGA instance
- Basic understanding of: neural networks, model training, and FPGA-based acceleration

Step 1. Launch the Application

Open KalEdge-Lite using the following link:

```
http://kaledge-lite.kaleidoforge.com/
```



Step 2. Load the Dataset

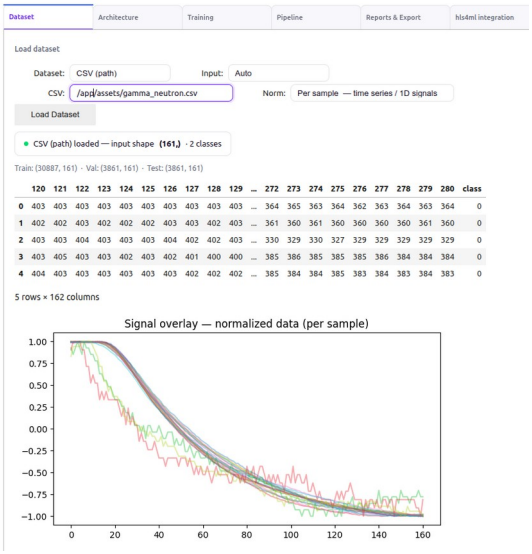
1. Open the Dataset panel.
2. From the dropdown menu, select CSV (path).
3. Enter the following path:

```
/app/assets/gamma_neutron.csv
```

> *This dataset corresponds to the gamma/neutron discrimination task.*

4. Leave the Input parameter set to **Auto**.
5. In the Norm option, select **Per sample – time series / 1D signals**

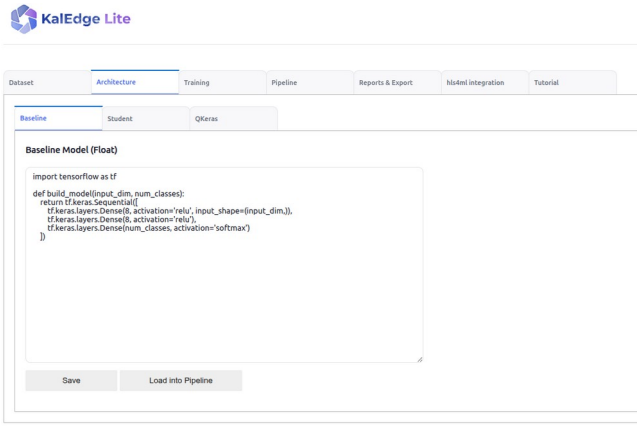
Once loaded, you should see the dataset preview with signal overlay and shape information.



Step 3. Define the Model Architecture

Go to the Architecture tab and select the predefined models. For each model:

6. Click Save
7. Then click Load into Pipeline



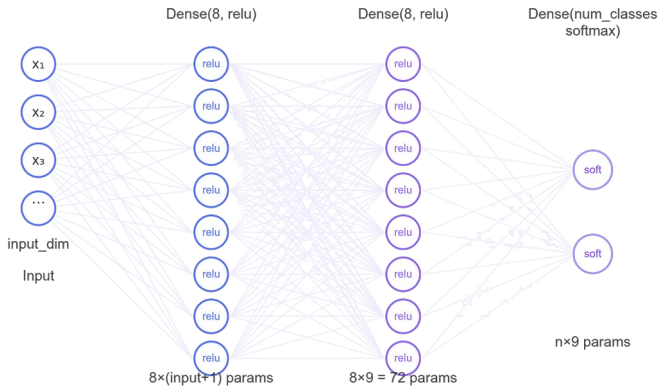
Important: For each model, you must click **Save** first, then **Load into Pipeline**. If you skip either button, training will not start.

Available options:

- Baseline model
- Student model
- QKeras model

hls4ml tip: For QKeras models, use activation="softmax" inside QDense for the last layer, not QActivation("softmax"). Supported hidden activations: quantized_relu, quantized_tanh, quantized_sigmoid.

In this lab, the baseline model has the following structure:



Step 4. Configure Training

In the Training window:

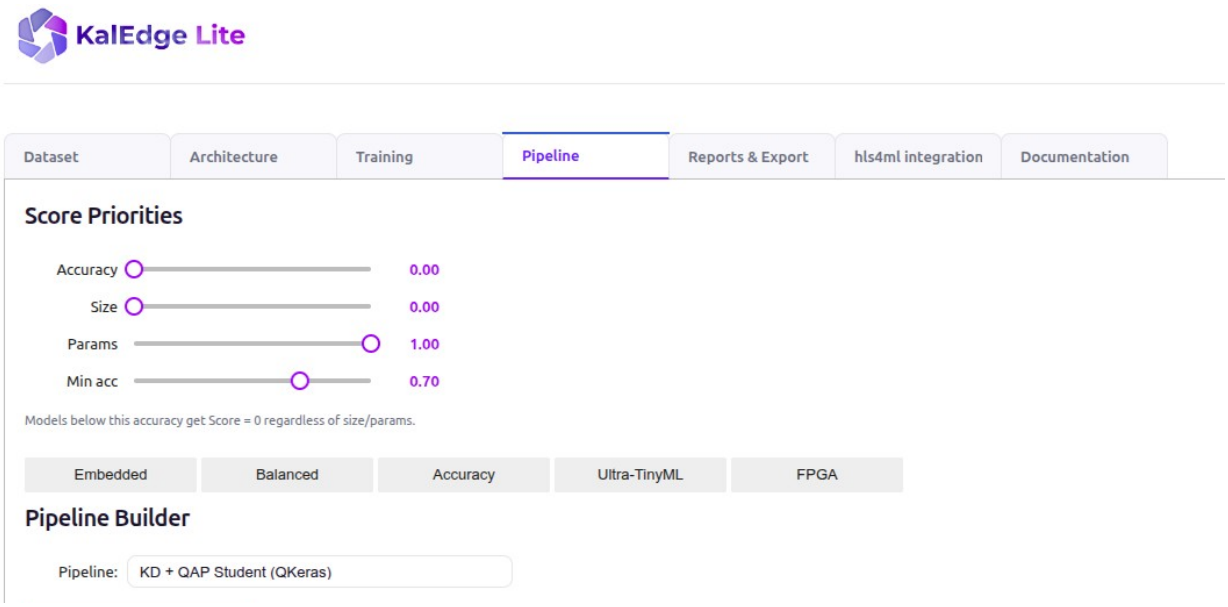
- Configure the training parameters if desired.
- You can train each model independently using the standalone buttons.
- For this tutorial, leave all parameters as default.

Step 5. Configure the Pipeline

8. Go to the Pipeline tab.
9. In Score Priorities, select: **FPGA**.

Step 6. Pipeline Builder

Within the **Pipeline Builder** section, leave all options at their default values, except for the **Pipeline** option, where **KD + QAP Student (Qkeras)** should be selected.



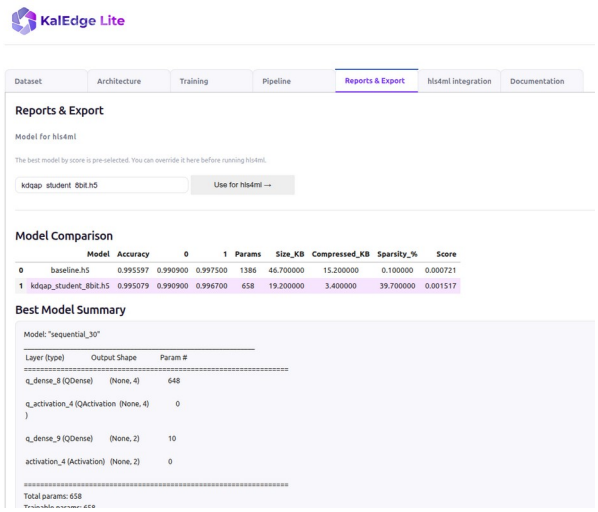
Step 7. Run the Pipeline

Click **Run Pipeline** and wait until the process completes.

Step 8. Results

Once finished:

- A message "Report generated" will appear.
- Navigate to the **Reports & Export** tab to inspect the generated report.
- Use the model selector dropdown to choose which model to send to hls4ml.



Step 9. HLS Integration

Click on the HLS4ML Integration tab and configure the following settings:

HLS / FPGA Settings

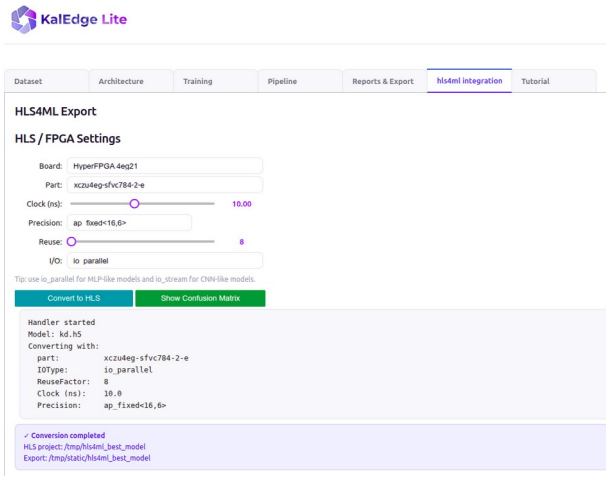
10. **Board:** Target board
    - HyperFPGA 4eg21
    - HyperFPGA 3be11

> *Must match the FPGA instance assigned in HyperFPGA.*

11. Clock period: 10ns
12. Precision: ap_fixed<16,6>
13. Reuse factor: 8
14. I/O type: io_parallel

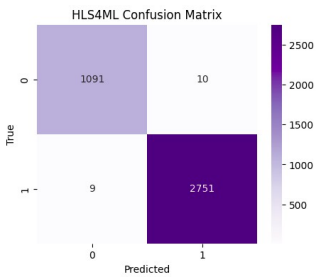
Step 10. Generate HLS Project

Click Convert to HLS and wait until the conversion process finishes.



Step 11 — Visualization

Click Show Confusion Matrix to visualize the classification performance of the converted model.



Step 12 — Download

Finally:

- Download the HLS project.

Interaction with the HyperFPGA

1. Access the HyperFPGA platform at: https://hyperfpga.sti.ictp.it/

2. Continue with the lab in the HyperFPGA.