Mar 31, 2025

NeuraLUT-Assemble

Hardware-aware Assembling of Sub-Neural Networks for Efficient LUT Inference

Marta Andronic
Imperial College London
marta.andronic18@imperial.ac.uk

2 ns latency

Agenda

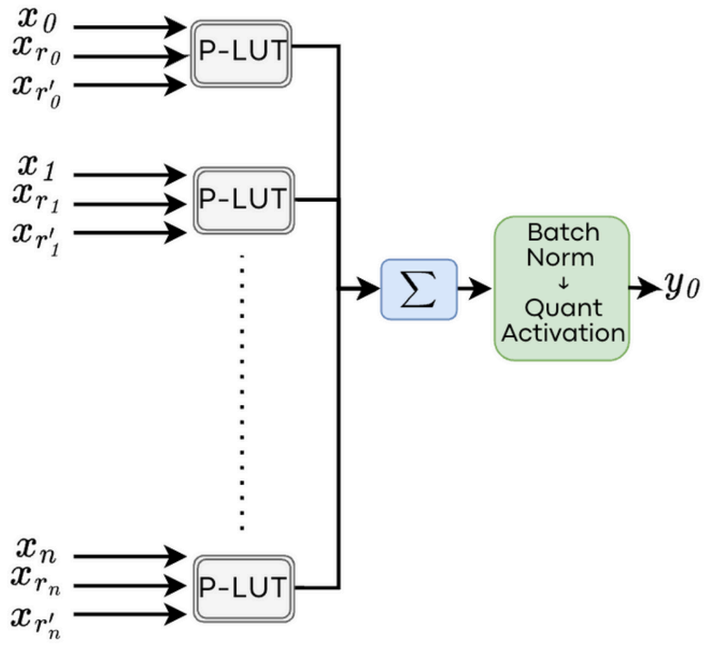
1. **Motivation and background**

- Learning K-input LUTs:
    i. LUTNet - Wang *et al.*, FCCM 2019
    ii. Differentiable Weightless Neural Networks - Bacellar *et al.*, ICML 2024
- Designing NNs for LUT circuits:
    iii. LogicNets - Umuroglu *et al.*, FPL 2020
    iv. PolyLUT, NeuraLUT - Andronic *et al.*, FPT 2023 and FPL 2024
    v. PolyLUT-Add - Lou *et al.*, FPL 2024

2. **Methodology**
3. **Experimental results**

Learning LUTs

LUTNet



BNN's XNORs replaced with learned Physical LUTs.

Characteristics:

- Lagrange interpolating polynomial
- Supports only 1-bit inputs
- Contains exposed datapaths

Scaling of parameters: Exponential

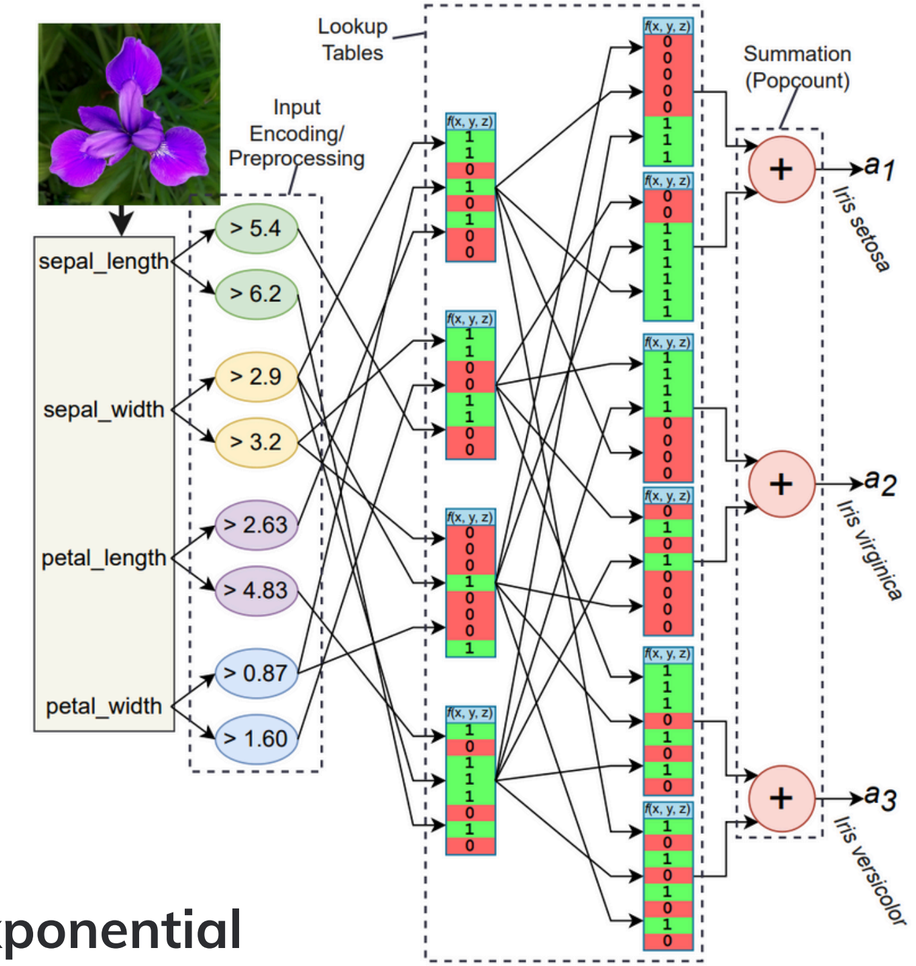
Wang *et al.*, LUTNet: Rethinking Inference in FPGA Soft Logic, FCCM 2019

Weightless Neural Networks

- **Extended Finite Difference** technique for approximate differentiation of binary values
- **Thermometer Encoding**
- **Learnable Mappings**

LIMITATIONS:

- Thermometer encoding assigns distinct floating-point thresholds to each feature, leading to potentially large overhead.

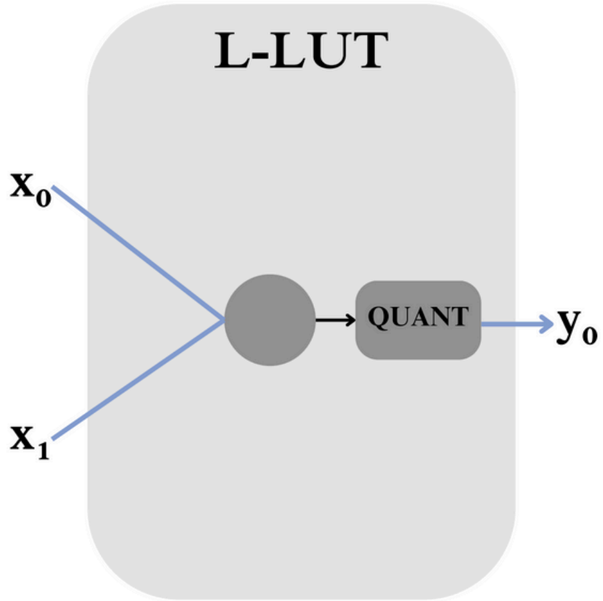


Scaling of parameters: Exponential

Bacellar *et al.*, Differentiable Weightless Neural Networks, ICML 2024

LUT-based neural networks

LogicNets



Neuron absorbed in a Logical LUT.

Supports larger bit-widths

Neural network trained in the traditional way with quantization between layers

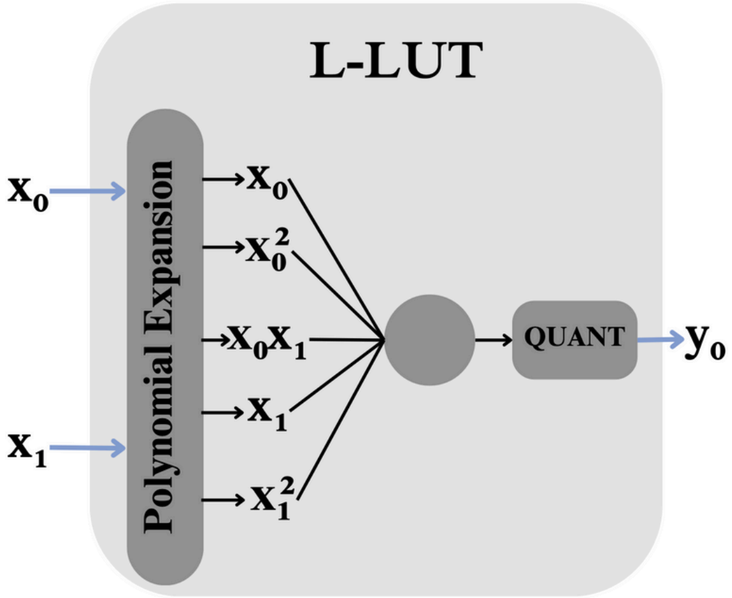
Post training the NN gets converted to a network of L-LUTs.

Logical LUTs get mapped to one or more Physical LUTs by the synthesis tools

Umuroglu et al., Co-Designed Neural Networks and Circuits for Extreme-Throughput Applications, FPL 2020

L-LUTs can hide any function

PolyLUT



L-LUT hides a multivariate polynomial.

Supports larger bit-widths

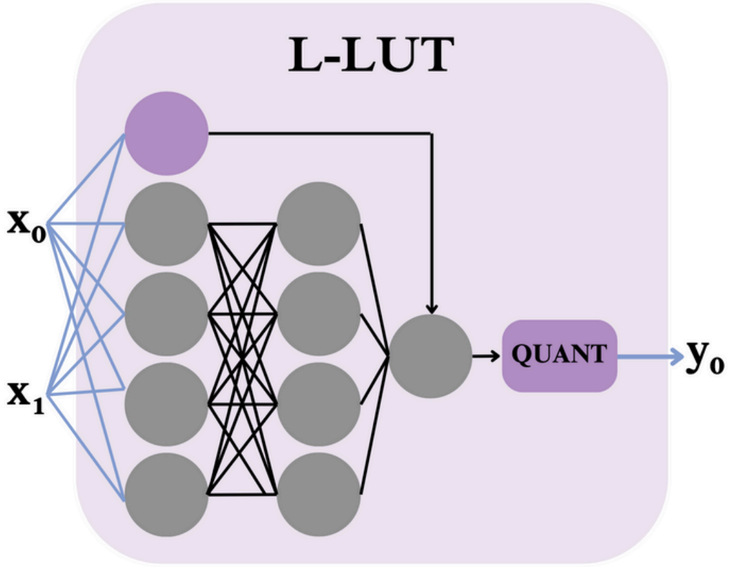
Expansion of the feature vector x with all of its monomials up to a parametric degree D

Post training the NN gets converted to a network of L-LUTs .

The number of trainable parameters can be controlled by tuning D

Andronic et al., PolyLUT: Learning Piecewise Polynomials for Ultra-Low Latency FPGA LUT-based Inference, FPT 2023

NeuraLUT



Neuron hides MLP with skip connections.

Supports larger bit-widths

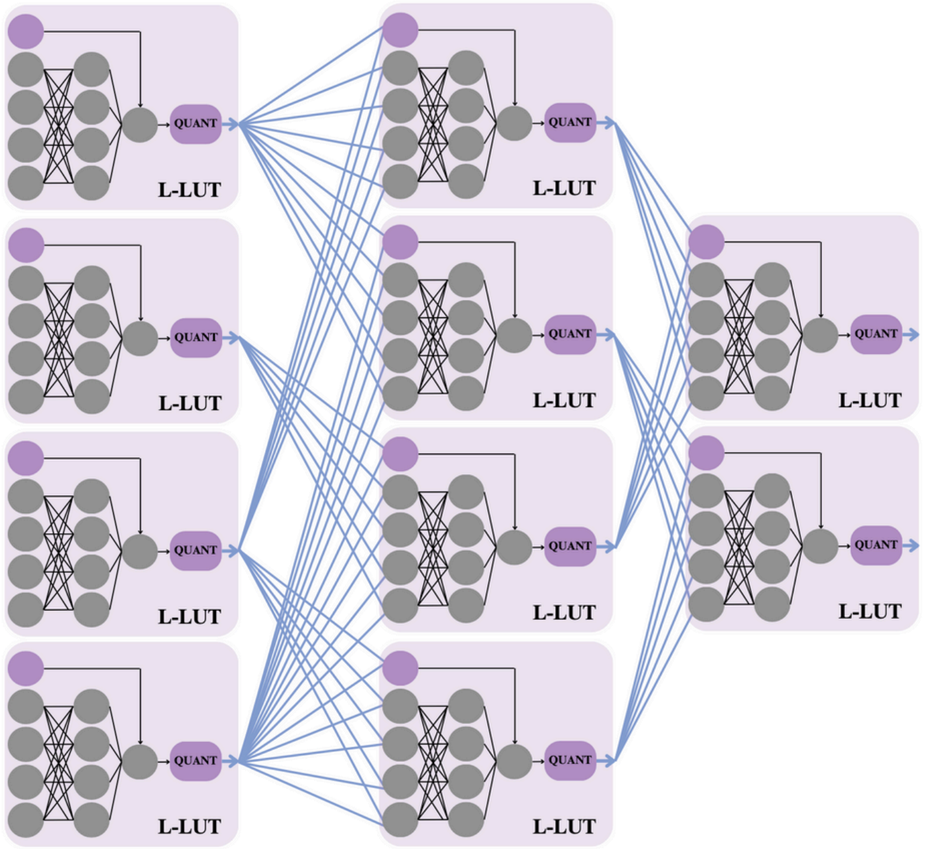
Hiding dense and full precision sub-networks

Integrating skip connections facilitates the flow of gradients, promoting stable and efficient learning without extra hardware.

The number of trainable parameters can be controlled by tuning the width and depth of the sub-network.

Andronic et al., NeuraLUT: Hiding Neural Network Density in Boolean Synthesizable Functions, FPL 2024

NeuraLUT

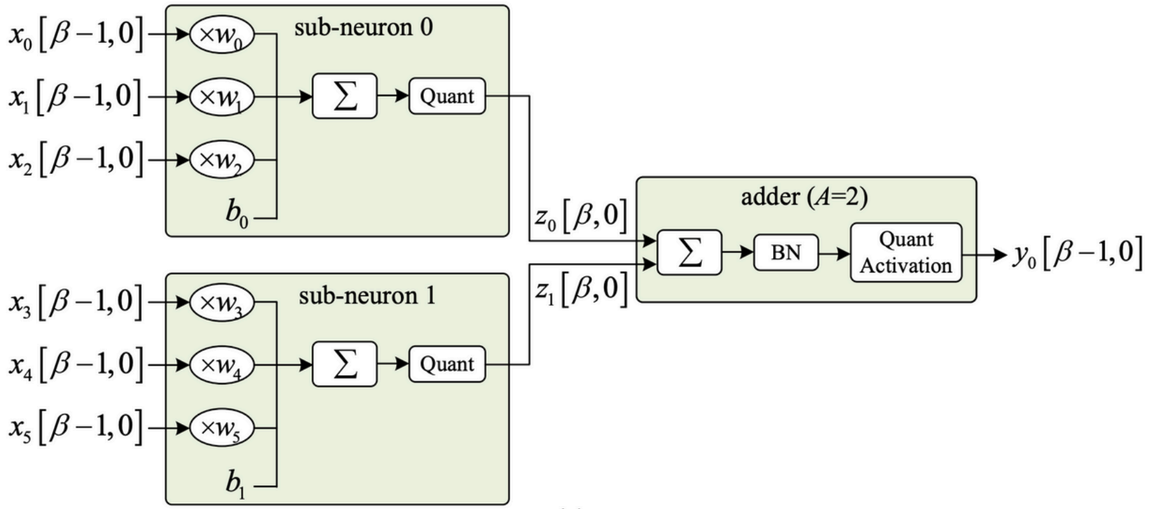


LIMITATIONS

This type of neural networks suffer from accuracy degradation due to fan-in constraints.
Result of the exponential scaling of LUT resources with the input width.

Andronic et al., NeuraLUT: Hiding Neural Network Density in Boolean Synthesizable Functions, FPL 2024

PolyLUT-Add



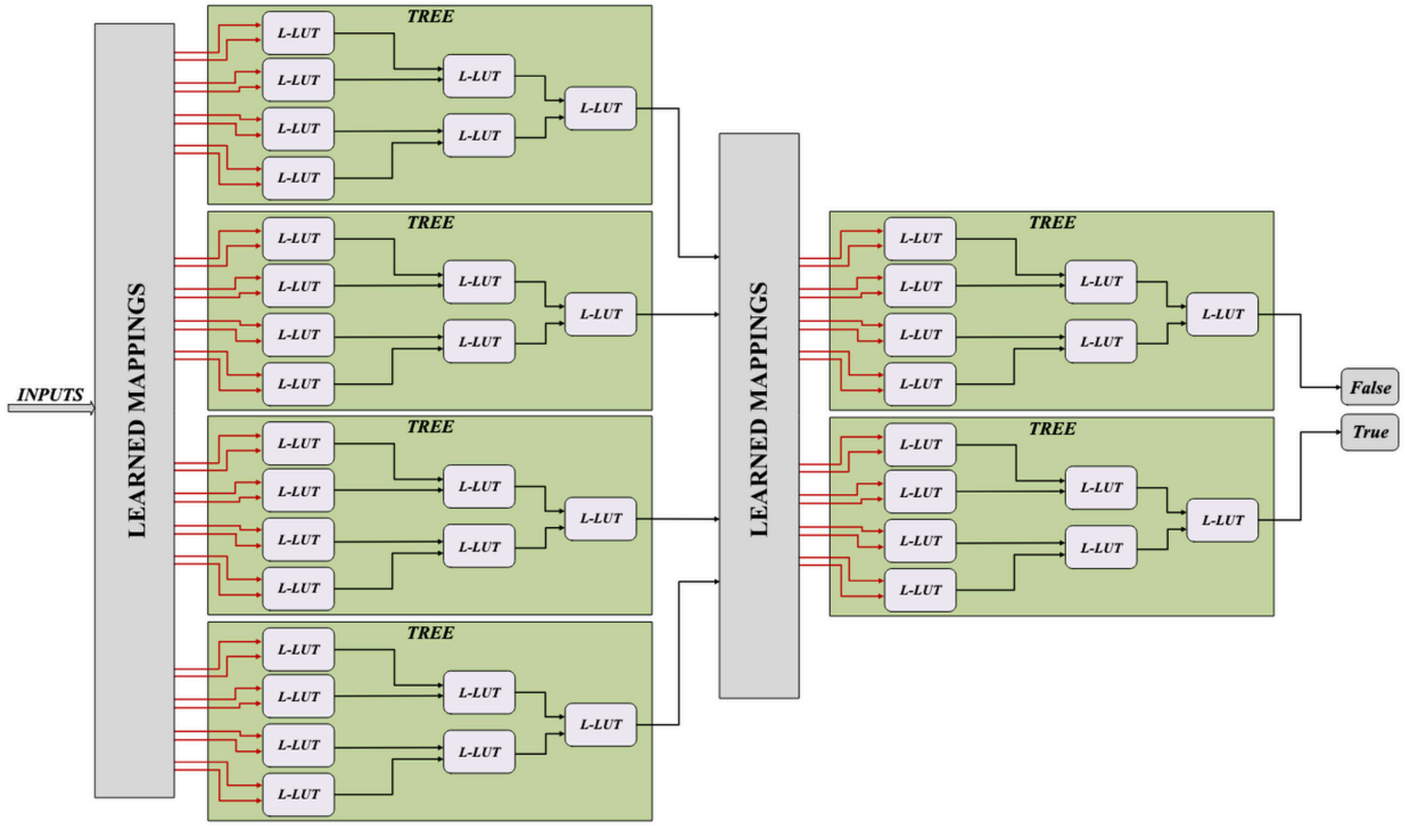
LIMITATIONS

LUTs are utilized for the sum which are also restricted by their fan-in.
LUTs can be leveraged for more than just a sum operation.

Lou et al., PolyLUT-Add: FPGA-based LUT Inference with Wide Inputs, FPL 2024

Methodology

NeuraLUT-Assemble

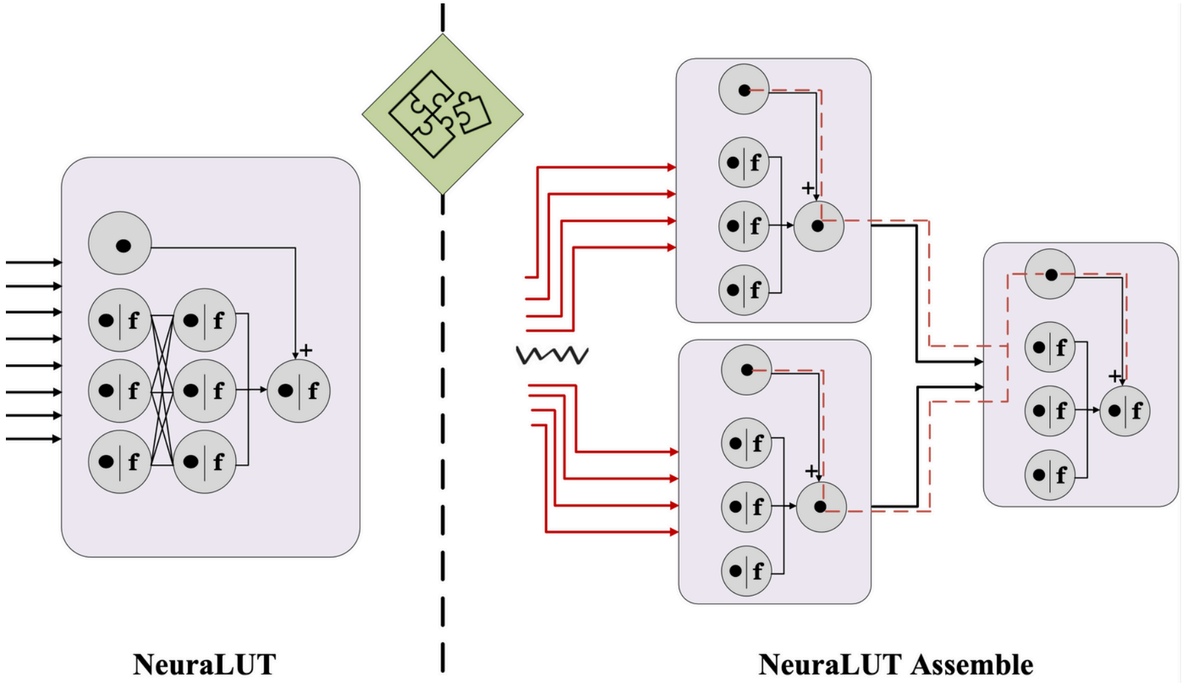


Fully-parametrizable framework that assembles multiple NeuraLUT neurons as tree structures with larger fan-in.

Directly addresses the exponential scaling challenge.

The grouping of connections at the input of the tree structure is guided by a hardware-aware pruning strategy.

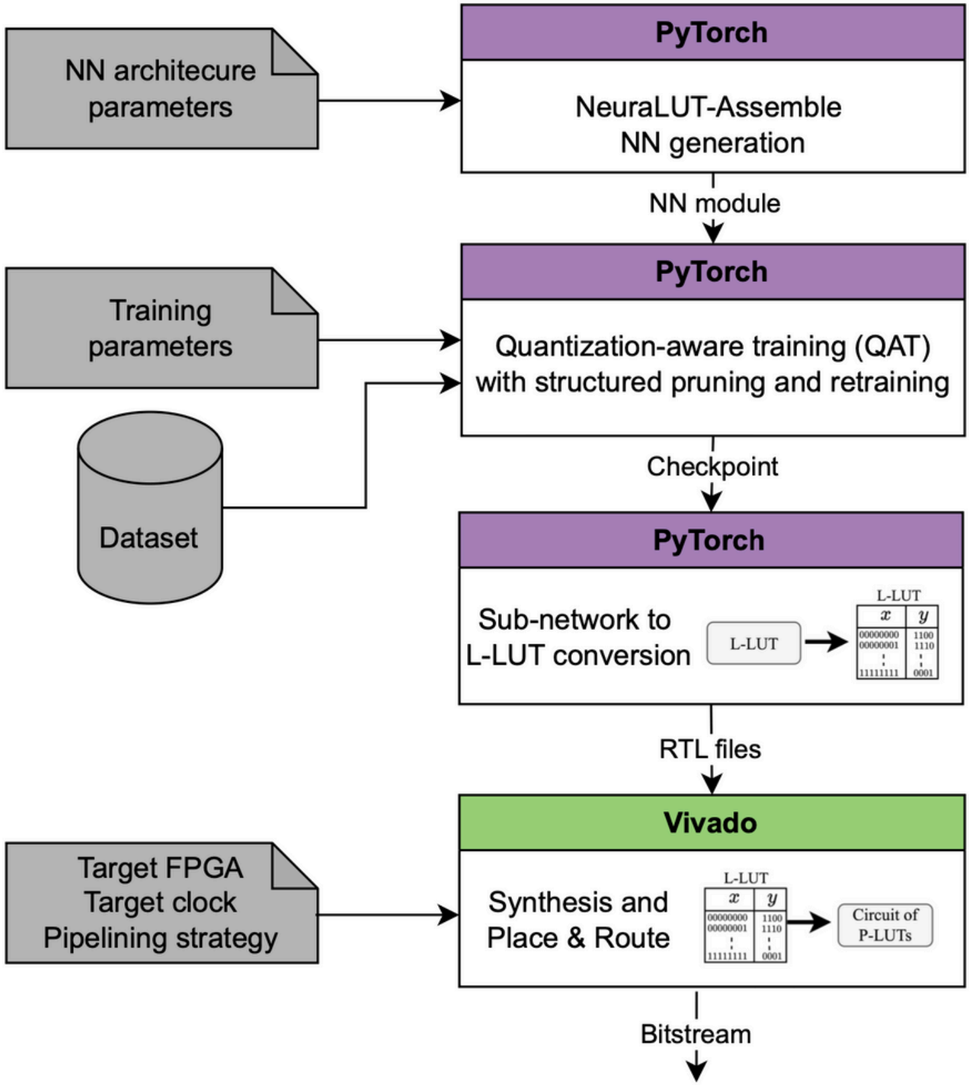
NeuraLUT-Assemble



Skip-connections are embedded within the L-LUTs, promoting smooth gradient flow throughout the entire tree structure.

These models train fundamentally different functions, therefore the tree structure is trained from scratch.

NeuraLUT-Assemble



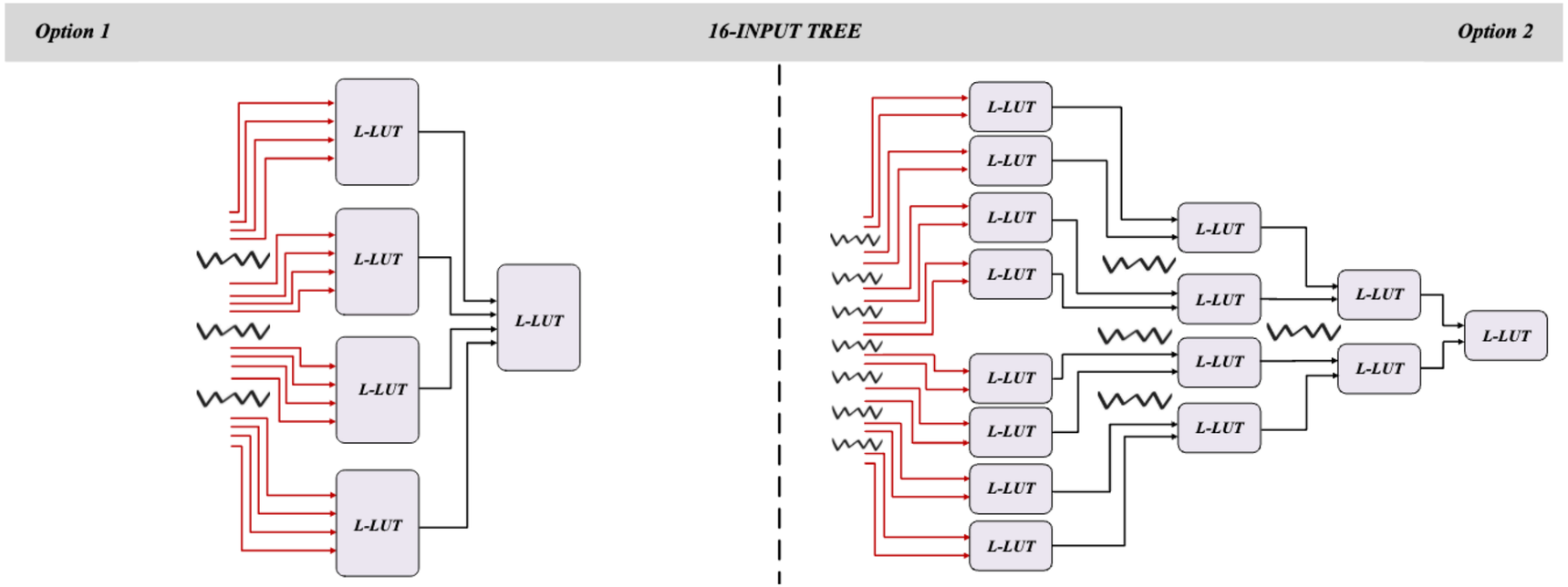
Toolflow Overview
Supports fully parametrizable tree-based structures and unique hidden NNs.

Quantization-Aware Training (QAT)
PyTorch-based training with quantization via Brevitas.

Sub-network to L-LUT Conversion
Automatic transformation of trained sub-networks into L-LUTs.
Inference-based lookup table generation.

RTL Generation & Hardware Compilation
Verilog RTL automatically generated with L-LUTs as distributed ROMs.
Out-of-context synthesis and place & route.

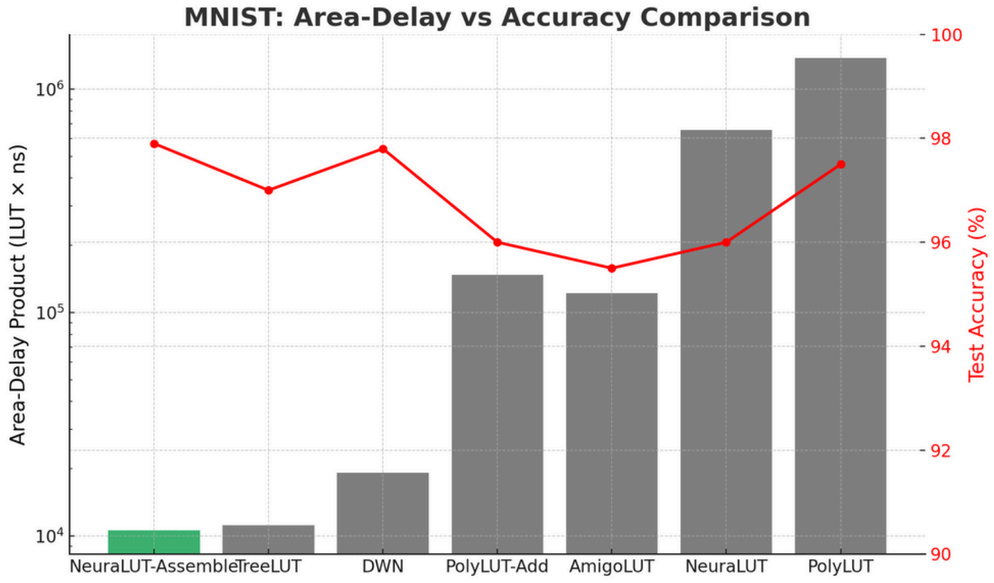
NeuraLUT-Assemble

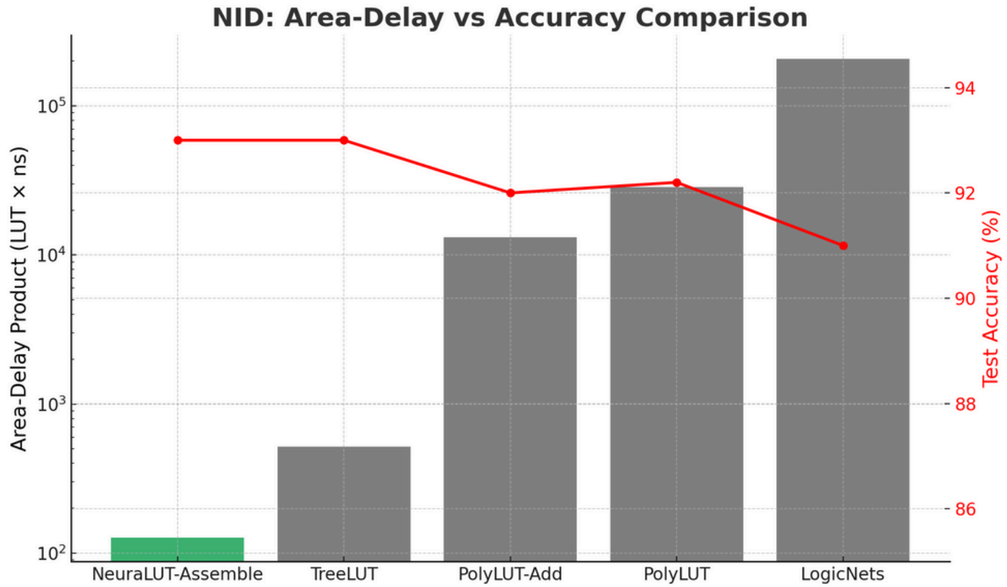


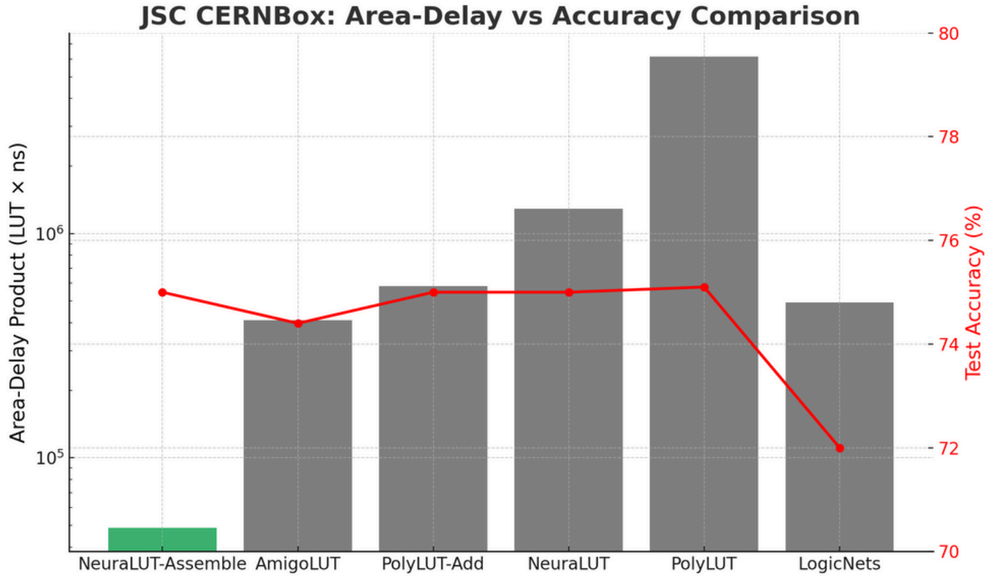
Our fully customizable framework allows users to construct tree structures tailored to their needs, balancing the trade-off between the number of L-LUTs and their size.

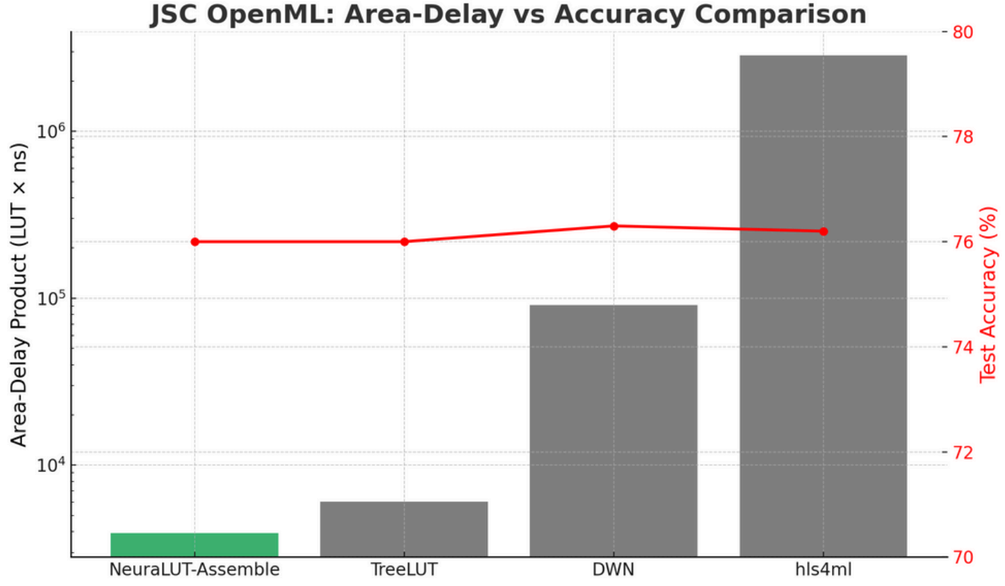
Two different NeuraLUT-Assemble configurations for a 16-input tree.

NeuraLUT-Assemble









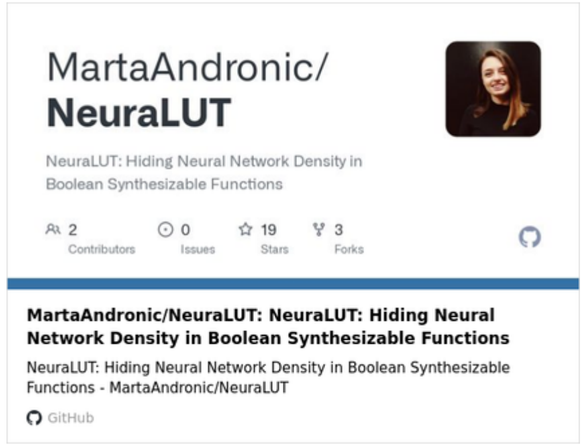
Compared to **NeuraLUT**:

- **62×** reduction in area-delay product on MNIST

Compared to **SOTA**:

- up to **8.42×** reduction in the area-delay product

Main Contributions



- We introduce **NeuraLUT-Assemble**, an **open-source toolflow** that leverages the FPGA architecture by embedding **dense, full-precision sub-networks within tree-structures of synthesizable LUTs**.
- We develop a **fully-parametrizable framework** to increase connectivity by training **larger fan-in tree structures of smaller L-LUT units**, where connection grouping is determined post-initial training.
- We develop a resource-efficient approach that embeds **skip-connections within L-LUTs**, which span the entire assembled tree structure.
- We assess NeuraLUT-Assemble on four standard tasks used in the low-latency DNN research.