Performance and Resource Analysis of MLP Forward Pass on LabVIEW FPGA

Magellan Scholar
McNair Junior Fellowship

Tiffany Yu[1], Dr. Rasha Karachi[1], Dr. Austin Downey[2], Joud Satme[2], Trotter Roberts[2]
Department of Computer Engineering[1], Department of Mechanical Engineering[2]
tyu@email.sc.edu, karakchi@cec.sc.edu, austindowney@sc.edu

Graduation with
Leadership Distinction in Research

Problem

- Neural networks are typically run on CPUs/GPUs, which are power-hungry and introduce latency unsuitable for real-time, embedded, or edge applications
- FPGAs offer parallel, low-latency, low-power computation — attractive for ML inference where speed and efficiency matter
- However, implementing a neural network on FPGA hardware isn't a direct port of a software model:
  - Math must be reworked into fixed-point arithmetic
  - Pipeline timing must be carefully managed
  - Overflow and precision loss must be avoided
  - All while meeting strict hardware timing constraints

Approach

- Target: NI cRIO-9033 Xilinx FPGA, developed in LabVIEW FPGA
- Built a matrix multiplication pipeline using an XNode block inside a Single-Cycle Timed Loop (SCTL)
- Extended this into a MLP with a 75-50-20-2 architecture, entirely in fixed-point arithmetic
- Used Flat Sequence Structure between SCTLs, and measured computation latency with gated Tick Count delta measurements

Acknowledgements

- This project is supported by McNair Summer Program through the Molinaroli College of Engineering and Computing at the University of South Carolina
- This project is based upon work supported by the National Science Foundation grant numbers CCF - 1956071, CCF-2234921, and CPS - 2237696. Any opinions, findings conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation

Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA

Motivation

- **Long-term goal**: full-scale 128-32-16-2 MLP for classification (% healthy vs. % unhealthy) on FPGA hardware
- Original 128-32-16-2 MLP ran at 0.609 ms/inference, which is too slow for real-time use
- Adopted a Xilinx block-based pipeline architecture to drastically cut latency within current hardware limits
- Current on-chip resource limits constrain the implementable size to 75-50-20-2

What is an MLP?

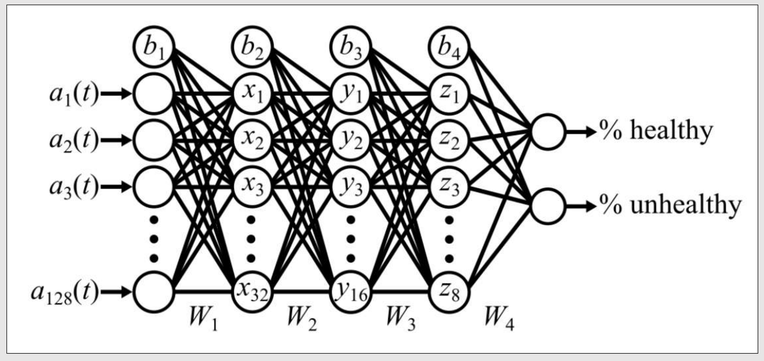

Figure 1: 128-32-16-8-2 MLP

- A multi-layer perceptron (MLP) is a type of forward propagation neural network made of stacked layers of nodes
- Data flows through a series of matrix multiplications followed by the addition of it bias values
- Mathematically, each layer computes:
  - $z = Wx + b$; where x is input vector, W is weight matrix and b is bias vector
- This structure lets the network learn complex, nonlinear mappings from inputs to outputs by adjusting weights during training

Resources

- ARTS-Lab. Ultra-High-Rate-State-Estimation GitHub, 2026, github.com/ARTS-Laboratory/Paper-2026-Ultra-High-rate-State-Estimation

Results

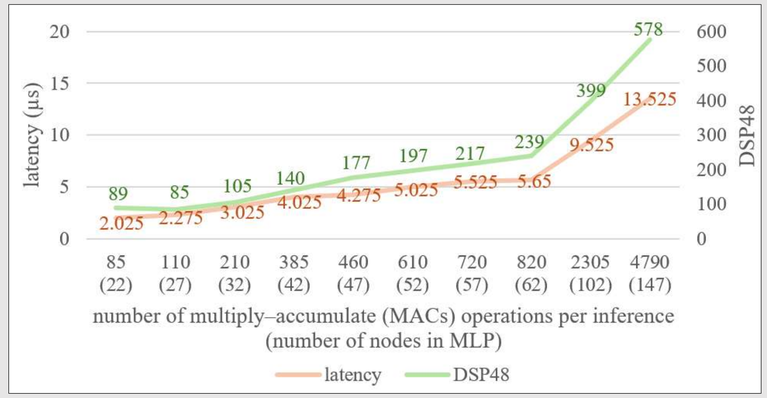

Figure 2: Latency and DSP48s for Different Sized MLPs

- Latency grows from 2.025 µs (85 MACs) to 13.525 µs (4790 MACs) — a ~6.7x increase for a ~56x increase in MACs.
- Growth is roughly linear/gradual up to ~820 MACs, then steepens noticeably at the 2305 and 4790 MAC points — signaling diminishing efficiency as the network scales toward hardware limits

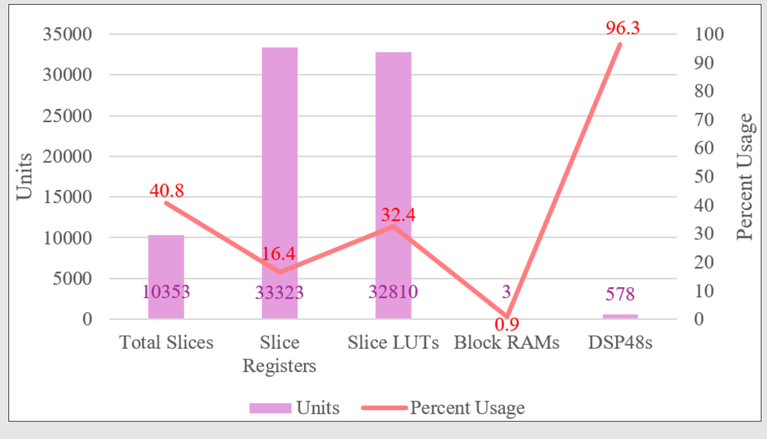

Figure 3: Resource Utilization for 4790 MACs, 147 Nodes

- DSP48s are the clear bottleneck, at 96.3% utilization (578 of 600 available), clearly far higher than any other resource