HLS4ML Integration with QICK

Mohamud Ali, Northwestern University l Advisors: **Nhan Tran**, Giuseppe G, Javi C, Fermilab

FERMILAB-POSTER-24-0220-STUDENT

I- Introduction

The QICK (Quantum Instrumentation Control Kit) integration aims to enhance the readout of superconducting qubits by leveraging machine learning (ML) techniques. These techniques offer high accuracy, increased speed, and better state preservation for qubit readouts. The integration process employs neural network algorithms to optimize system performance and achieve high accuracy rates. The QICK board uses an FPGA, allowing for efficient real-time processing and high-performance execution of machine learning algorithms.

II- Purpose of the Project

The goal of the "ML-based Qubit readout with QICK" is to develop and implement a machine learning-driven approach for the efficient and accurate readout of qubits using the Quantum Instrumentation Control Kit (QICK).

III- Tools and Material

Software: Vivado, HLS4ML, Python, VNCserver, and C++

Hardware: PC, FPGA board and QICK board

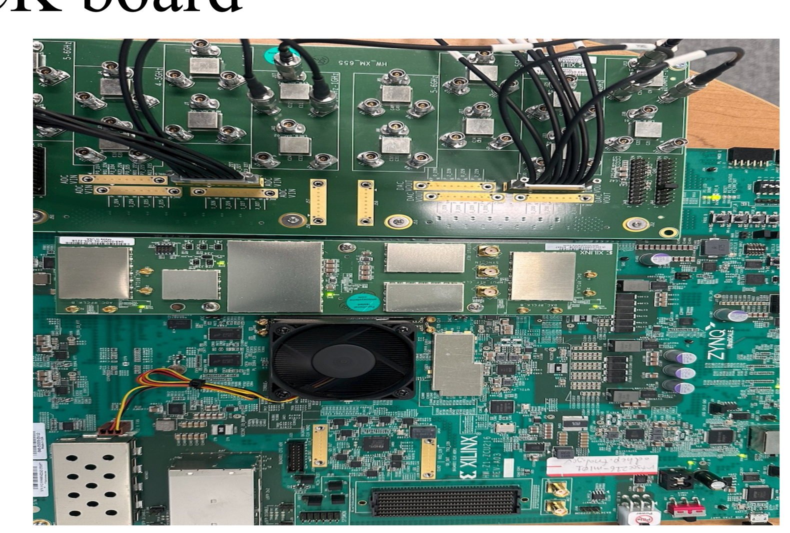

Figure 1: QICK Board Hardware

IV- Design Flow

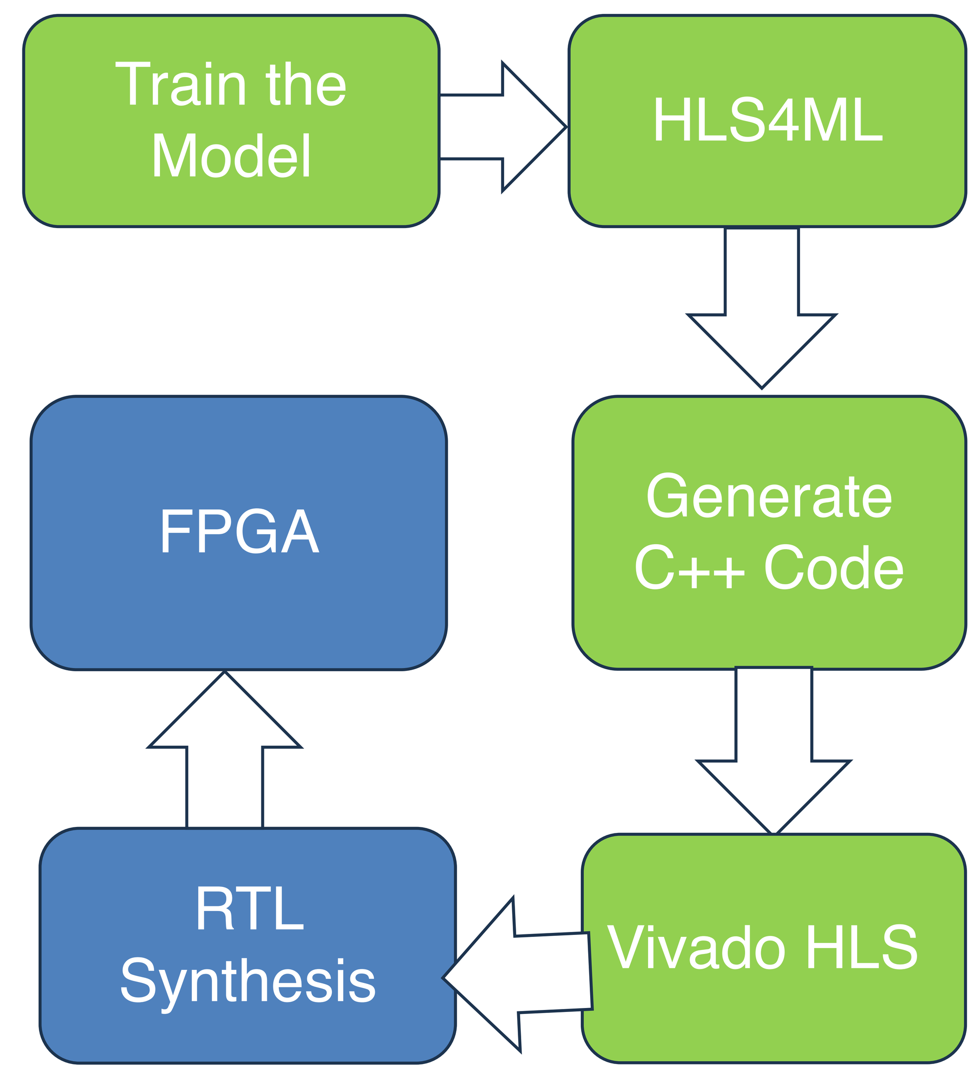

Figure 2: QICK Firmware Design Flow

V- ML Model Results

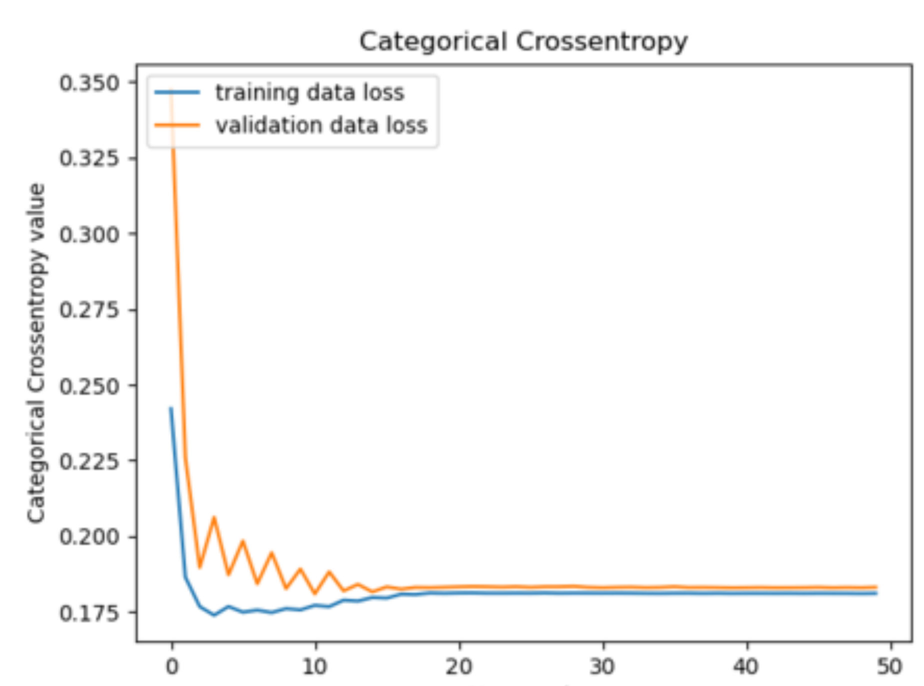

Figure 3: ML Loss During The Training

- In validation data the Categorical Cross-Entropy is quite sensitive to weight swing over first 10 epochs.
- Loss difference between training data and validation data decreases as epochs increase, two lines merge close together.

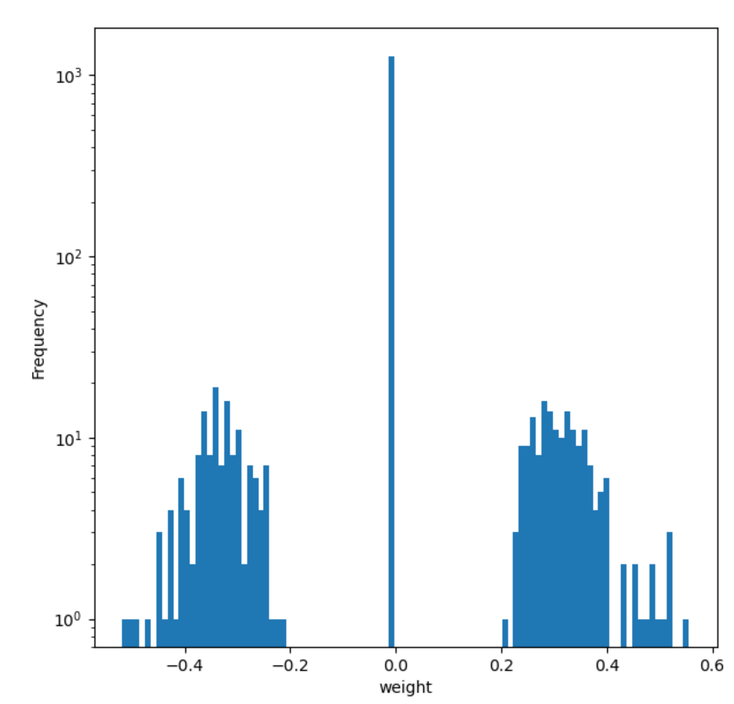

Figure 4: Pruning the model

Three different clusters of distribution

- Significant spike at zero
- Two small clusters at -0.4 & 0.4

Effect of Pruning: majority of weights have been set to zero, which is typical in model pruning to reduce model complexity and size. The remaining weights are clustered around -0.4 and 0.4, suggesting that these weights are more significant and have been retained through the pruning process.

VI- Keras vs. HLS Model

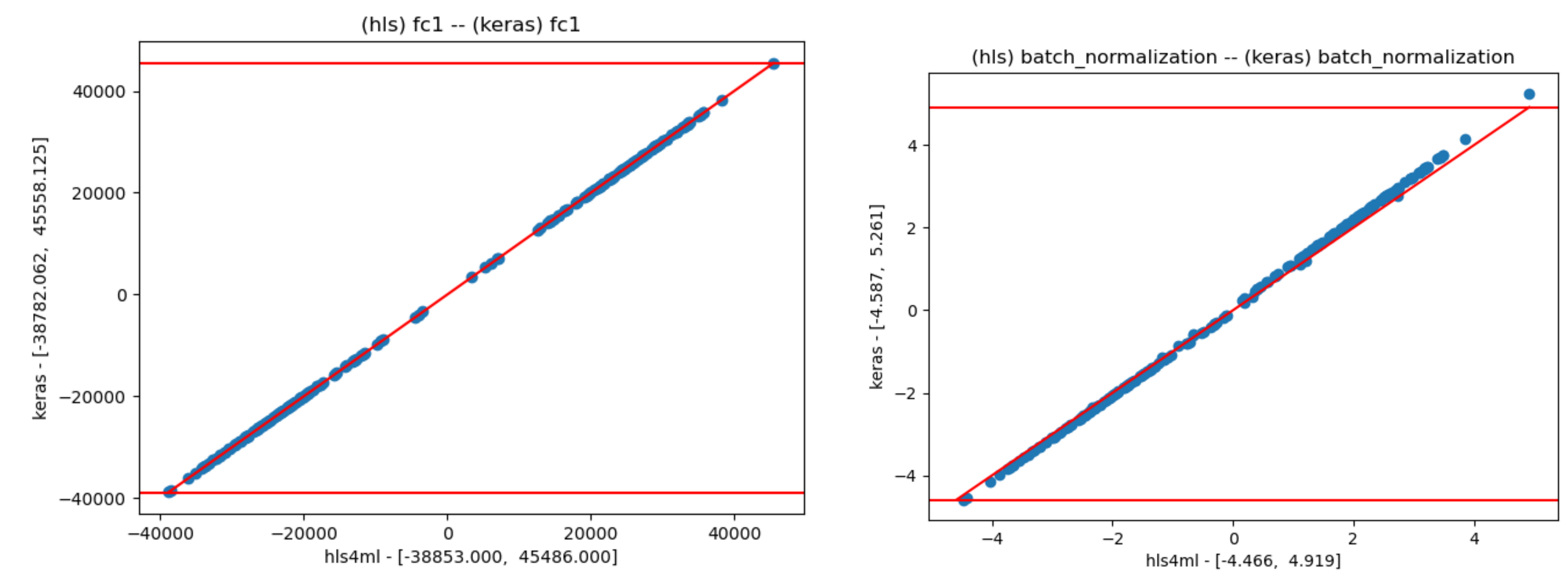

Figure 5: Comparison of Qkeras and HLS Model Output

- The data points are closely aligned along the diagonal line, showing that both the fully-connected layer (fc1) and batch normalization outputs are very similar between the two models. This indicates strong correlation between the two models.
- The data points clustering along the diagonal line in both graphs indicate that the HLS model effectively replicates the Qkeras model's behavior, ensuring consistent performance across different layers.

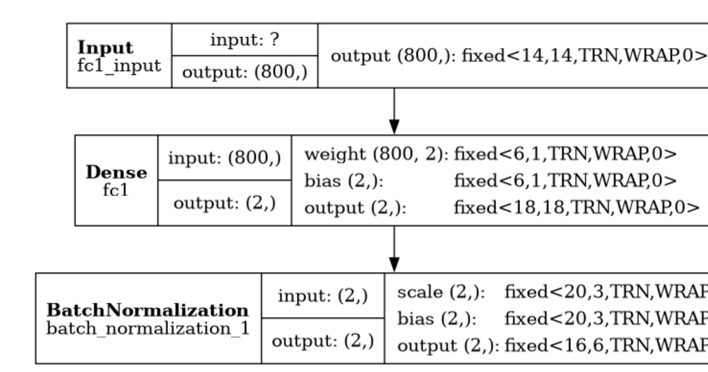

Figure 6: Dual layer NN Architecture Flow

- **Fixed-Point Arithmetic**: Ensures efficient hardware implementation with specific truncation (TRN) and wrap-around (WRAP) handling for overflow.

	Keras	HLS
Accuracy	95.405%	95.407%

Figure 7: Accuracy measure of model training

VII – Future work

Future work will include RTL synthesis, implementation on FPGA and QICK board to ensure Qbit readout data is correct with the expected accuracy level, and further verification to validate the design's real-world performance.

VIII- References

1. **HLS4ML Tutorials access to this link** " Document"
2. **HLS4ML GitHub** https://github.com/hls-fpga-machine-learning/hls4ml
3. **QICK Reference "https://arxiv.org/pdf/2110.00557"**

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics