Contribution ID: **175** Type: **talk**

Tree Tensor Network predictors implemented on FPGA for ultra-low latency inference.

Wednesday 22 January 2025 17:45 (15 minutes)

Tree Tensor Networks (TTNs), a loopless type of tensor network, are commonly used to represent and simulate many-body quantum systems, but they can also be exploited for several applications in Machine Learning (ML).
They rely on the factorization of high-order tensors into networks of smaller tensors, effectively overcoming the "curse of dimensionality"by moving the computational complexity from an exponential to a polynomial scaling with the number of inputs. Originally designed for studying weakly entangled quantum states, TTNs can also be leveraged in several ML tasks, exploiting their quantum-inspired characteristics to gain useful insights into the distribution of learned information. Entanglement entropy and quantum correlations can help optimize data representation, eliminate redundancies, and reduce the number of active parameters. Moreover, due to the linearity of the operations involved in their algorithms, TTNs happen to be well-suited for implementation on hardware like Field Programmable Gate Arrays (FPGAs), which excel in fast parallel computations.
Starting from the statements above, this study focuses on the development of TTNs on FPGAs, intending to deploy them in ultra-low latency environments, such as High Energy Physics (HEP) experiments, where rapid decision-making is crucial. To do this, various TTN architectures are studied as binary classifiers, starting with simple benchmark datasets (Iris and Titanic) and moving to the more complex case of a classifier for $b/\bar{b}$ jet flavor tagging, exploiting LHCb open data.
The TTNs are trained using tensor-specific optimization techniques ("sweeping" algorithms) rather than classical ML optimizers, reducing computational costs and avoiding issues like barren plateaus. Additionally, the architectures are initialized with unsupervised learning methods to improve training stability and avoid falling into local minima. For the inference algorithm offloaded in hardware, the full contraction process iteratively combines feature vectors with network weights, producing output values representing the probabilities of a single sample to belong to each final class.
Two different hardware strategies for tensor contraction are produced as VHDL firmware for FPGA, allowing resource usage and latency analyses to support a wide range of TTN topologies. The Full Parallel (FP) implementation performs most operations simultaneously and it achieves the lowest latency, but it uses significant hardware resources, making it suitable for environments with high resource availability. The Partial Parallel (PP) implementation instead reduces Digital Signal Processors (DSPs) usage by sequentially computing tensor-weight multiplications, happening to be more suitable for resource-constrained environments where latency is less critical. Both methods use pipelining for efficient computation of multiple input samples, and the flexibility of DSPs is leveraged to adapt tensor operations to FPGA constraints.
Regarding the firmware precision, the numbers in the TTN implementations are represented using 16-bit fixed-point precision and normalized to the range $[-2,2]$. However, different TTN architectures might require varying numeric precision, therefore quantization studies are essential for optimizing resource usage since reducing the number of bits used to represent values in firmware can significantly lower the number of needed DSPs.
This project is developed on XCKU115 FPGA, which is pre-programmed with fixed hyperparameters; the trained tensor weights are instead stored in on-chip memory (BRAM). Communication between the host PC

and FPGA is handled using AXI Lite and AXI Stream protocols for weight programming and data streaming, respectively. Eventually, inference results are validated by comparing FPGA outputs with software outputs, ensuring consistency. For all the involved networks, hardware outputs closely match software results, ensuring identical classification labels in hardware and software and accurate FPGA deployment of TTNs for real-time classification tasks.
The LHCb TTN classifier for b/$\bar{b}$ jet flavor tagging is successfully implemented on FPGA, achieving sub-microsecond inference latency, demonstrating its potential for real-time use in HEP trigger systems. With this work, it is confirmed the feasibility of deploying TTNs on FPGA with efficient resource usage and high-speed performance, making them suitable for low-latency applications like HEP experiments.
For the future developments of this project, several possibilities are being considered. Currently, this TTN deployment is being compared to classic NN implementations available in hls4ml, considering LHC jet tagging datasets. The comparisons are in terms of accuracies, readability, and resource consumption, to identify the most suitable tool that can be used for ultra-low latency classification at the lower levels of the trigger pipelines of HEP experiments, where the available signals are not yet constituting complex physical objects but better trigger primitives. If the two approaches would report completely orthogonal characteristics, the joint usage of TTNs and NNs for learning the dataset's relevant information for classification could lead to the production of a new software tool for quantum-inspired ML. In addition to this, the transposition from VHDL to hls4ml of the firmware generative code is being considered, also possibly enlarging the current libraries with additional TN ansatzes: this could allow more users to exploit the perks of tensor network methods.

Email Address of submitter

lorenzo.borella.1@phd.unipd.it

Short summary

Authors: Dr COPPI, Alberto (Università di Padova); Prof. STANCO, Andrea (Università di Padova); TRIOSSI, Andrea (Universita e INFN, Padova (IT)); PAZZINI, Jacopo (Università e INFN, Padova (IT)); BORELLA, Lorenzo (Universita e INFN, Padova (IT)); ZANETTI, Marco (Universita e INFN, Padova (IT))

Presenter: BORELLA, Lorenzo (Universita e INFN, Padova (IT))

Session Classification: Computing and Algorithms