Nanosecond AI for anomaly detection with decision trees on FPGA using fwXmachina

ATLAS EXPERIMENT

SMARTHEP Edge Machine Learning School

Mon, 23 Sep 2024

Ben Carlson, Isabelle Taylor, Joerg Stelzer, Kemal Ercikti, Kyle Mo, Pavel Serhiayenka, Rajat Gupta, Santiago Cane, Stephen Roche, Tae Min Hong, Yuvaraj Elangovan

WESTMONT

University of Pittsburgh

SAINT LOUIS UNIVERSITY
SCHOOL OF MEDICINE

fwX – an efficient BDT implementation on FPGAs

FW X Machina

Framework for generating nanosecond-scale inference BDTs for use in FPGAs

Anticipated areas of use: event analysis in hardware triggers in HEP experiments

Work on

- Fast event classification with BDT (Hong et al., JINST 16, P08016 (2021))
- Fast regression with deep BDT's (*) (Carlson et al., JINST 17, P09039 (2022))
- Fast anomaly detection with BDT-based auto-encoders (*) (Roche et al., accepted for publication)

* Currently being implemented in ATLAS L1 trigger

BDTs for auto-encoders

Typically constructed using neural networks

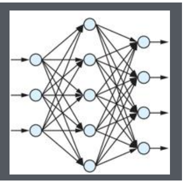
- Challenge to implement in pure digital logic on FPGA

See: Govorkova et al., *Autoencoders on field-programmable gate arrays for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider*, Nature Mach. Intell. **4** (2022) 154–161
https://doi.org/10.1038/s42256-022-00441-3

Neural Network

Popular	Been around HEP since the 80s[1]
Depth	Challenging, so ~3 on FPGA[2]
Score	$y = \Theta(\mathbb{M} \cdot x + b)$

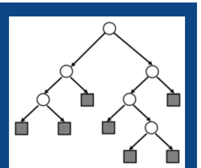
Activation (Θ) *Multiplication* ($\cdot$)



Decision Tree

Popular	Discovered the Higgs![3]
Depth	Challenging, so 4 to 8 on FPGA[4,5,6]
Score	$y = \Theta(x < \text{threshold})$

Step fn (Θ) *Comparison* ($<$)



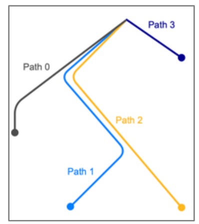
Classification performance of BDTs is often comparable

Advantages of BDT

- Technical (no multiplication)
- Philosophical (interpretable)

FWX approach:

- Goal: make evaluation of the BDT in FPGA faster while using less resources
- Achieved by parallelizing node evaluation

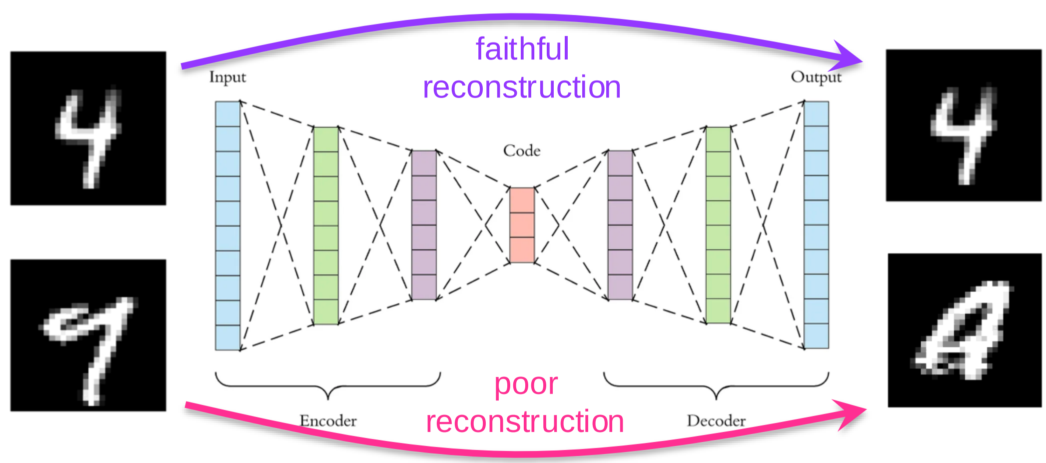


Auto-encoders for anomaly detection

Auto-encoders rely on data-compression algorithm (usually NN, fwX: BDT), trained on known, expected data (background)

Encoding input into latent ("code"-) space and decoding back into input space preserves objects which are similar to training sample (known data), but fails to faithfully re-construct anomalies (unknown data)

Training sample: 0..4



Poor reconstruction

large discrepancy between input and output => high anomaly score

Our approach to using BDTs for auto-encoders

Novel algorithm for using decision trees in auto-encoders for anomaly detection

- Anomaly score from comparison of input with latent space, no decoding step

Method: (a glimpse)

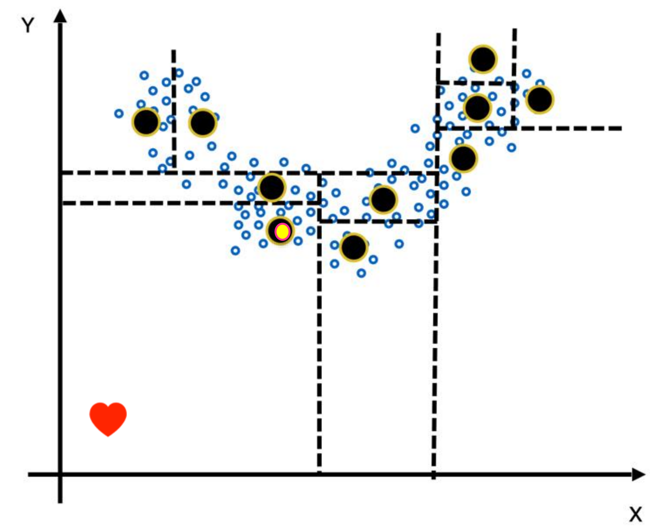
Place small boxes around locations of high event density

Encoding an event ❤:

- Return *index b* of the box the event ❤ falls into

Decoding a box index *b*:

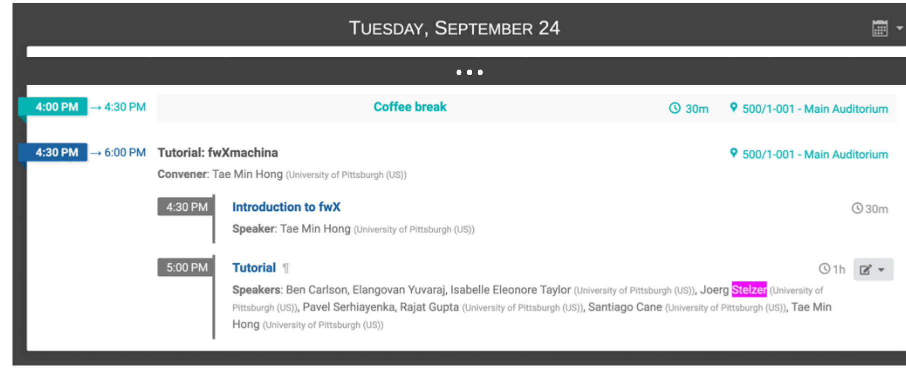
- Return the median ● of the training data in box *b*



Want to learn more – join us tomorrow

In-depth introduction to anomaly detection with FWX by Tae tomorrow afternoon @16:30.

Followed by a hands-on tutorial



Tutorial with three parts

- Training and fwX-BDT code generation (with TMVA and FWX)
- Synthesis (with Vivado)
- FPGA evaluation (simulation with Vivado)

Each part has a 10’ video (where you can work along), followed by a Q&A session

- If you like to follow the tutorial on your laptop, please make sure you have root, fwX (part 1) and vivado (parts 2+3) installed