Optimizing Your Anomaly Detection Models

Jack Cleeve[1], Seokju Chung[1], Akshay Malige[2]

[1]Columbia University, [2]Brookhaven National Laboratory

Anomaly Detection for High Energy Physics Workshop
June 17, 2025

Outline

- Introduction (1-2 minutes)
- Installing Your Environment (20 minutes)
- Training Your Models (15 minutes)
- Converting to hls and Inference (5-10 minutes)

Introduction

Yesterday Andrew spoke to you all about utilizing 2D CNNs for anomaly detection.

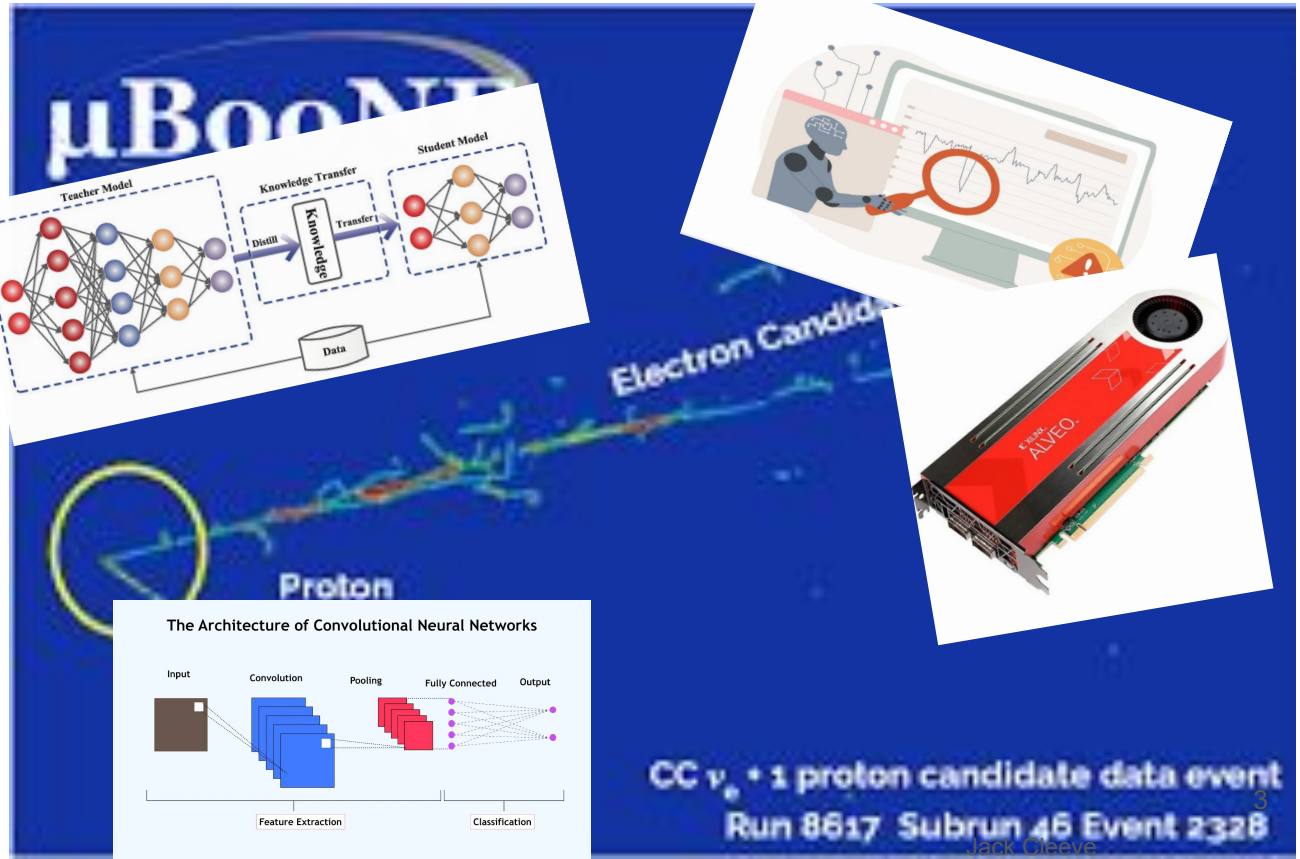
Today we'll discuss **optimizing the latency and performance** of those networks, as well as **synthesizing them to be used on FPGAs**!!

Highly valuable for applying algorithms at the **trigger level**, where you must deal with:

- *large data rates*

- *limited hardware resources*

- *require extremely fast decisions (microsecond scale)*



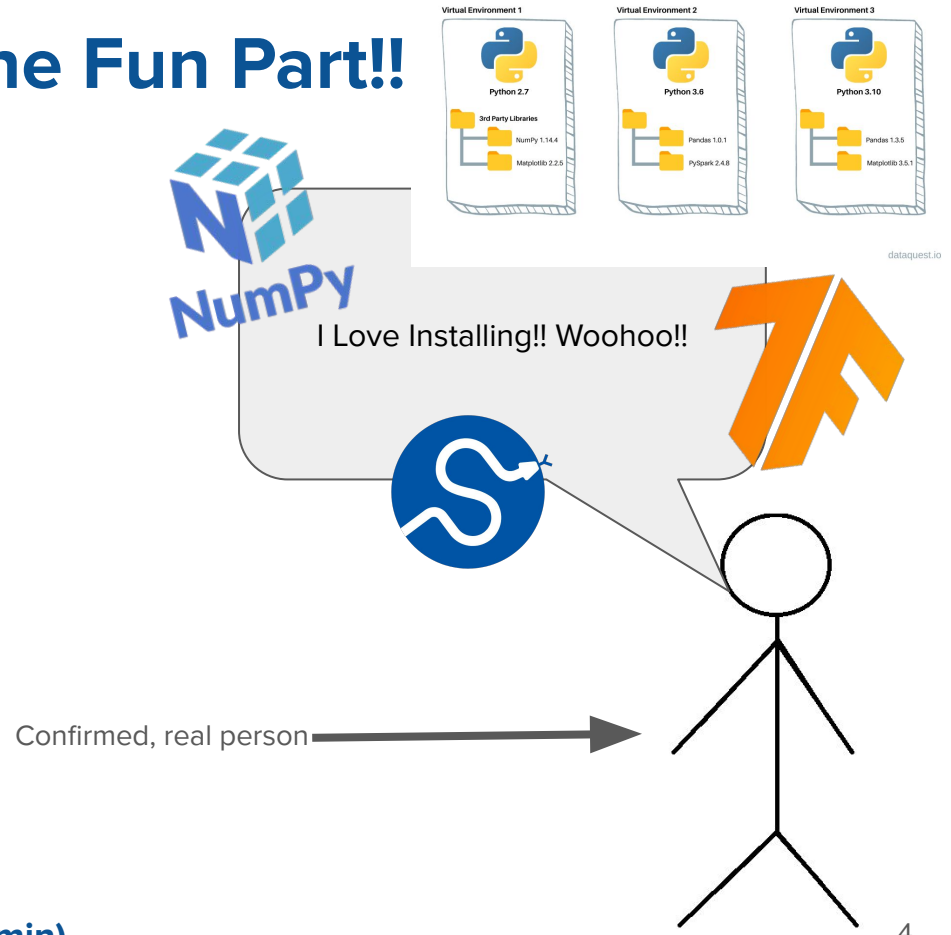
Installing: The Fun Part!!

Please go to the github and clone the repo:

https://github.com/jhc2191/AD4HEP_2025_Tutorial

After that follow the instructions in the notebook to create a python virtual environment and install the necessary dependencies.

Please let us know if you're having any trouble at all.

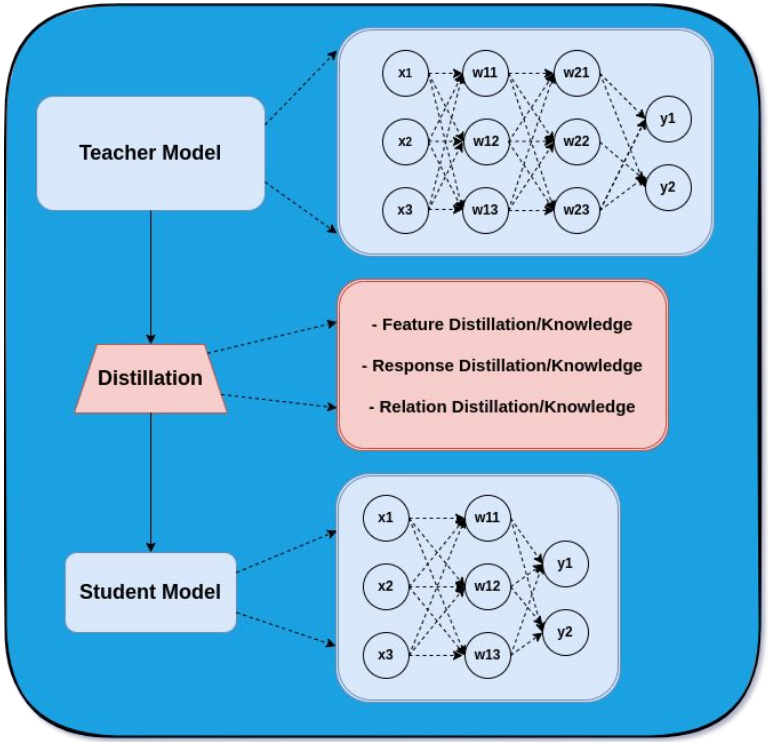


(20 min)

Building and Training the Models

We'll use the simple MNIST dataset as an example on which to train and test our models.

Thanks to **Adrian Pol** for providing us with some existing teacher and student CNNs (maybe you can adjust/add some models yourself…)



Knowledge Distillation

(15 min)

hls4ml

Hls4ml: Python package for machine learning inference in FPGAs.

- They create firmware implementations of machine learning algorithms using high level synthesis language (HLS).
- They translate traditional open-source machine learning package models into HLS that can be configured for your use-case!

Who's Got the Best Model????

We want 2 things:

1. We want our hls synthesized models to **match** almost identically to the regular models
2. We want a model to have great **performance** (AUC) without sacrificing too much latency (we'll measure with number of parameters here, obviously not perfect)

Whoever has the best model wins...!