Particle Trajectory Classification and Prediction Using Machine Learning

Angelos Angelopoulos (CRTC), Polykarpos Thomadakis (CRTC), Gagik Gavalian (JLab), Nikos Chrisochoides (CRTC)

ODU REYES 2020, August 11

Introduction

- The **CLAS12 detector** at Jefferson Lab is used to study the structure of matter by scattering an electron beam of a proton target.
- When the electron beam hits a proton, particles are scattered.
- Particles are detected by the signals a particle leaves in wire drift chambers.
  - In the drift chambers, there are 6 layers, each layer containing 6 wires.
  - Each wire contains 112 sensors, for a total of **4032 sensors** in a drift chamber.
  - A **sensor** is **activated** if a particle passes over it.

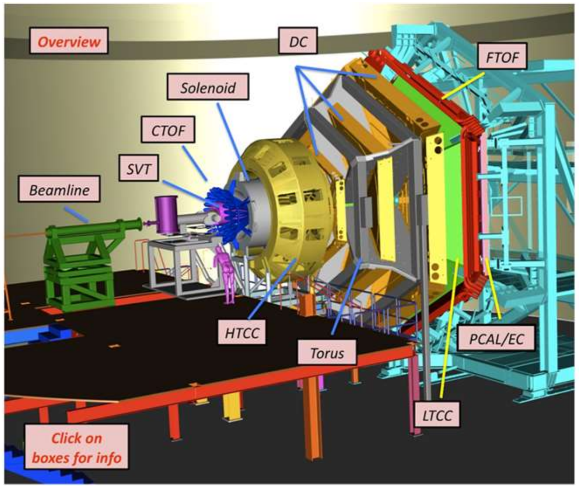


Figure 1: The CLAS12 detector in Jefferson Lab, Experimental Hall B.

The Problem

- Reconstructing particle trajectories is **expensive and time consuming**.
    - All possible combinations of segments need to be considered.
    - The one that most closely matches a line is accepted.
- A **faster and more reliable method is needed** to identify valid particle trajectories
    - Speed is crucial in the experimental pipeline.

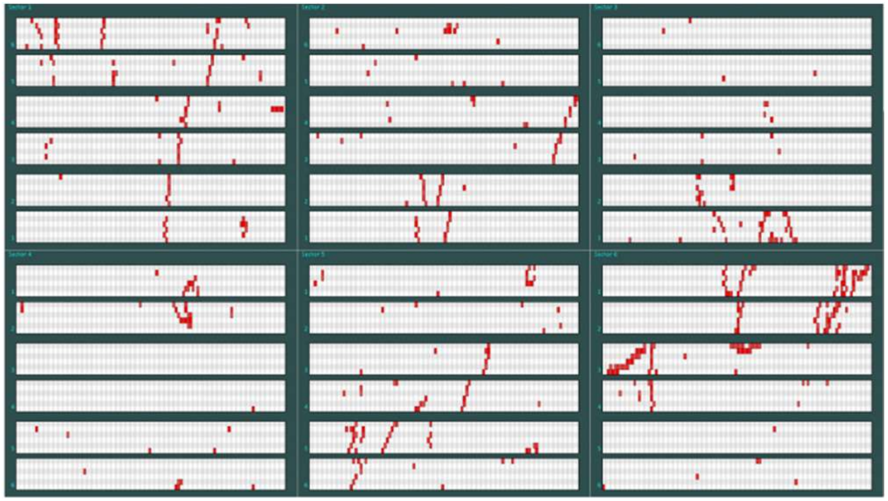


Figure 2: CLAS12 detector drift chamber showing the sensors activated (red) by traveling charged particles. The activations that form an approximate line from beginning to end are classified as valid.

The Solution

- We developed **machine learning models** that can classify incoming particle trajectory data with **high throughput.**

- The solution consists of two components:
  - **Prediction** of particle trajectories based on previous trajectory information.
  - **Classification** of trajectories as being valid (approximate lines) or not.

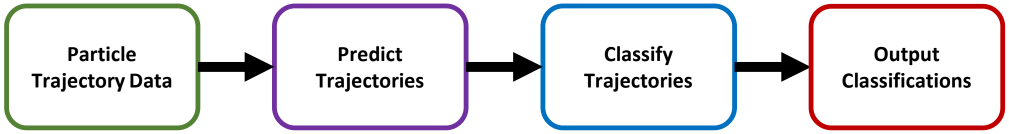
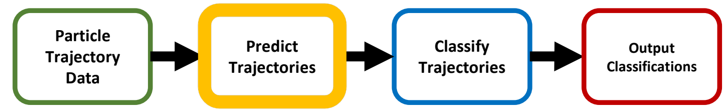


Figure 3: The filtering pipeline of the solution.

Particle Trajectory Prediction



- We developed a Recurrent Neural Network (RNN) using Gated Recurrent Unit layers (GRU).
  - Trained on only valid particle trajectories.

- Can predict the continuation of valid particle trajectories.
  - Will give incorrect predictions for invalid trajectories.
  - By measuring the spatial distance of the actual trajectory from the predicted one, we can infer whether the actual trajectory is valid or not.
  - A large distance means that the actual trajectory is invalid.

- Allows us to eliminate many samples where there are no valid trajectories.

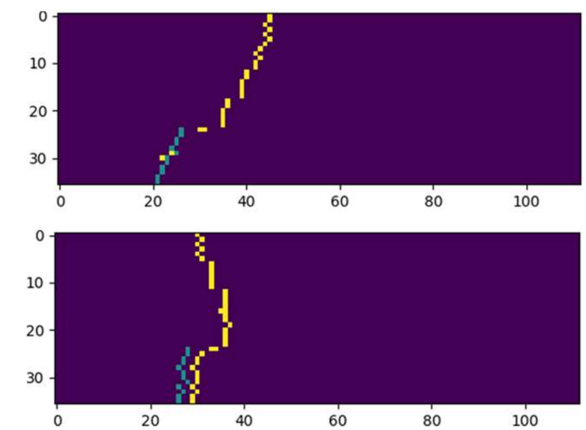


Figure 4: Shows two separate valid actual particle trajectories (yellow) and the predictions of the RNN for part of them (blue).

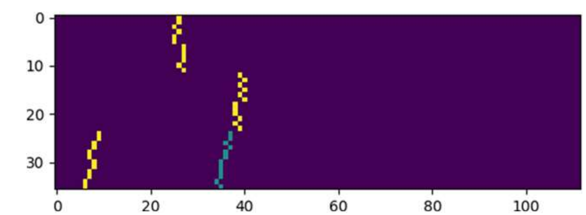


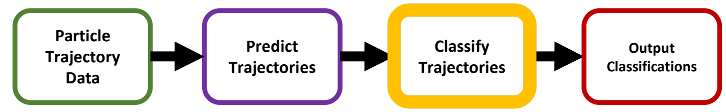
Figure 5: Shows one invalid actual particle trajectory (yellow) and the prediction of the RNN for part of it (blue).

Trajectory Prediction Results

Model Type	Loss (Mean Absolute Error)	Training Time	Inference Time / sample
RNN/GRU	**1.18**	374 sec	688 µs

Table 1: Metrics for the RNN for particle trajectory prediction. The unit of the loss is one sensor (a loss of 1.18 means a mean distance of 1.18 sensors between the actual and predicted particle tracks). The RNN executed on one Tesla V100-SXM2-16GB GPU.

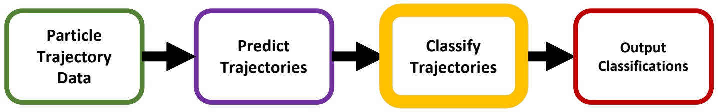
Particle Trajectory Classification



- We developed three separate models: extremely randomized trees (ERT), a multi-layer perceptron (MLP), and a convolutional neural network (CNN).
  - Trained on tens of thousands of particle trajectory samples.
  - Samples included multiple trajectories, of which one was valid.
- The models classify a trajectory as valid or not.
  - A valid trajectory is approximately a line.

Figure 6: CLAS12 detector drift chamber showing the sensors activated (red) by traveling charged particles. The activations that form an approximate line from beginning to end are classified as valid. The valid trajectory is marked with a green box.

Classification Accuracy Metrics



- In order to determine the accuracy of the models we devised and used several metrics:

1. **A1**: The percentage of samples where the valid particle trajectory was detected.
2. **Ac**: The percentage of *A1* for which there were invalid tracks misclassified as valid ones (false positives).
3. **Ah**: The percentage of *A1* for which the valid particle trajectory had the highest probability of being valid out of all tracks in a sample.
4. **Af**: The percentage of samples where the valid trajectory was not detected (false negatives). This metric was critical to minimize as we don't want to miss valid trajectories.

Trajectory Classification Results

Model Type	A1 Metric	Ac Metric	Ah Metric	Af Metric	Training Accuracy	Training Time	Inference Time / sample	Training Samples	Training Rows
MLP	**99.96 %**	10.77 %	98.88%	0.04 %	99.65 %	254 sec	120 µs	29,810	291,367
ERT	**100.00 %**	6.14 %	100.00 %	0.00 %	100.00 %	19 sec	306 µs	29,810	291,367
CNN	**96.11%**	28.11%	90.16%	3.89%	94.26%	457 sec	1.2 ms	4,818	17,311

Table 2: Metrics for the three models for particle trajectory classification. MLP and ERT executed on a multi-core CPU while the CNN executed on one Tesla V100-SXM2-16GB GPU.

Conclusion

- The machine learning models have high accuracy and throughput.

- By employing them in the experimental pipeline for CLAS12, we can **save 6 times more time and energy** compared to current trajectory filtering methods used and ultimately increase the accuracy of experiments

Acknowledgements

- This work is in part funded by Jefferson Lab, SURA grant No. CNF-19-04, NSF grant No. CCF-1439079, and Richard T. Cheng endowment.

- Any subjective views or opinions that might be expressed in the presentation do not necessarily represent the views of the United States Government.