ML Track Fitting in Nuclear Physics

Thomas Britton[1]*, *David* Lawrence[1]**, and *Gagik* Gavalian[1]***
on behalf of the GlueX and CLAS12 collaborations

[1]Thomas Jefferson National Accelerator Facility

Abstract. Charged particle tracking represents the largest consumer of CPU resources in high data volume Nuclear Physics (NP) experiments. An effort is underway to develop machine learning (ML) networks that will reduce the resources required for charged particle tracking. Tracking in NP experiments represent some unique challenges compared to high energy physics (HEP). In particular, track finding typically represents only a small fraction of the overall tracking problem in NP. This presentation will outline the differences and similarities between NP and HEP charged particle tracking and areas where ML learning may provide a benefit. The status of the specific effort taking place at Jefferson Lab will also be shown.

1 Introduction

The collaborative effort of applying machine learning (ML) tools and techniques to the challenge of tracking in nuclear physics is made up of members from both the GlueX[1] and CLAS12[2] experiments. The GlueX experiment is housed in Hall-D, one of Jefferson Laboratory's four experimental halls, and the CLAS12 experiment is housed in Hall-B, also at Jefferson Laboratory.

Tracks in nuclear physics experiments at Jefferson Laboratory differ from those found in high energy physics (HEP) experiments, such as those at the Large Hadron Collider (LHC), in several distinct ways. Nuclear physics tends to deal with far fewer tracks per event; on the order of 2-4 in NP versus hundreds to thousands in HEP at the LHC. The tracks in nuclear physics also tend to have more curvature than those found in higher energy domains. Additionally, the magnetic fields employed in nuclear physics tend to be less uniform than those in high energy physics. There also exist more subtle differences between the challenges facing GlueX and CLAS12. These differences result in different approaches being taken by each collaborative party. These methods and results are shared freely.

2 GlueX

Currently GlueX spends 94.4% of total event reconstruction time on tracking. This 94.4% is comprised of 69.9% of time spent on time based tracking, 22.5% of time spent on wire based tracking, and only about 2% of time spent on track finding. On a per track level this means

*e-mail: tbritton@jlab.org
**e-mail: davidl@jlab.org
***e-mail: gavalian@jlab.org

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

finding the track candidate takes approximately 1.2 ms, performing wire based tracking takes approximately 3.3 ms, and time based tracking takes about 10.3 ms. The bottom line is that track finding is not the big contributor to reconstruction times for GlueX, track fitting is.

GlueX therefore desires to surgically replace existing methods with an emphasis on obtaining performance gains via "shallow learning", which should, in principle reduce inference times. Another added benefit of moving aspects of tracking to machine learning algorithms is the ability of well established machine learning libraries to operate on, and easily port to, different hardware solutions. The newest exascale machines planned will produce a majority of their FLOPs via GPUs. With GlueX set to begin a second phase of running at higher luminosity it may not have access to a similarly scaling supply of CPU cycles which could negatively affect the total wall time needed to completely reconstruct data.

To begin research on the topic GlueX focused on a toy problem modeled after the GlueX Forward Drift Chamber (FDC) and resembling the CLAS12 design too (6 layers of 6 chambers with about 100 wires each). The toy detector was comprised of 6 chambers of 6 layers each for a total of 36 layers. Each layer was comprised of 100 wires each separated from neighboring wires by 1 cm and having 90% hit efficiency. Each layer is only allowed one hit. Each of the 6 chambers are separated from the neighboring chamber(s) by 50 cm. A neutral track begins from some z location within a 15 cm region of the z axis and may make an angle of between -10 and 10 degrees with the z-axis (labeled as either θ or ϕ depending on individual preference). A sketch of the toy problem can be found in Figure 1. The goal was to create a machine learning model using Keras[3] and Tensorflow[4] to obtain the z and angle of each track.

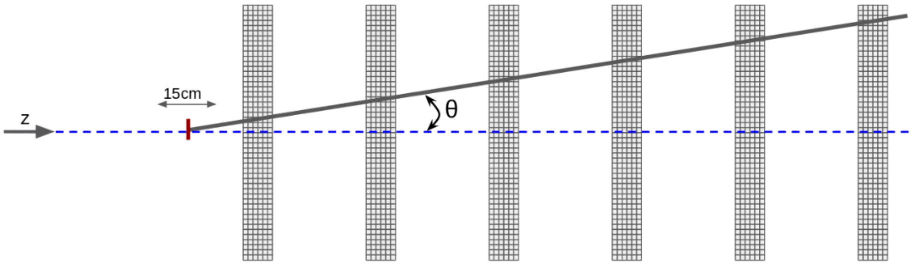

Figure 1. A sketch of the toy problem analyzed by GlueX modeled on the Forward Drift Chamber.

The approach used was to treat the problem as a classification problem with each class representing a "bin" for the desired variable (starting z location and angle). A weighted average of the confidence of the model in each "bin" can then be taken to obtain scalar answers for both variables. As an additional cross-check the confidence distribution is fit with a Gaussian. Results for the resolution of the track's angular value as well as a comparison between truth and inference can be found in Figure 2. The fact that the resolution of the angle, when using the weighted average approach, was smaller than the estimation of the optimal resolution flummoxed the group. After more careful investigation it was found that the estimation of the "optimum" resolution was based purely on the wire geometries and too simplistic. In actuality, the resolution has an angular component. Similar to a crystal lattice, there exist some specific angles where angle determination becomes difficult and other "magic" angles where the determination can be made more precisely. Taking this into account a more faithful comparison was made and can be found in Figure 3. The model was determined to faithfully reproduce the angular dependence on resolution.

With this toy problem representing a simple proof-of-concept GlueX plans on looking at actual data with a focus on the state vector, always seeking optimal models with low inference times and optimum errors. Once these algorithms have been proven functional GlueX plans on adapting these tools and techniques to other portions of the GlueX code base in search of performance gains and code adaptability.

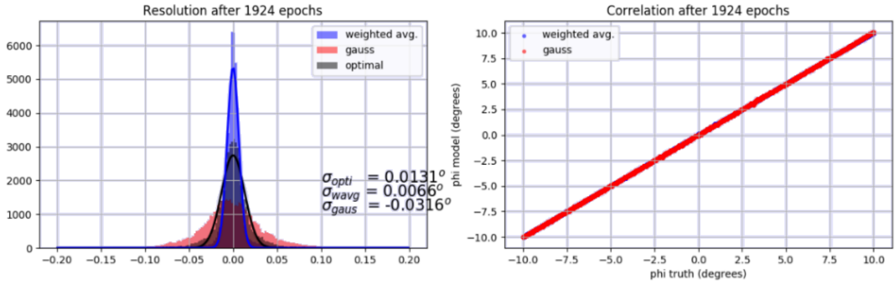

Figure 2. On the left the resolutions are shown for the weighted average method, a Gaussian fit method and a theoretical optimum. The right plot shows the comparison between the true value of the angle and the value obtained by the model for the two methods of value extraction; it shows good agreement between all methods.

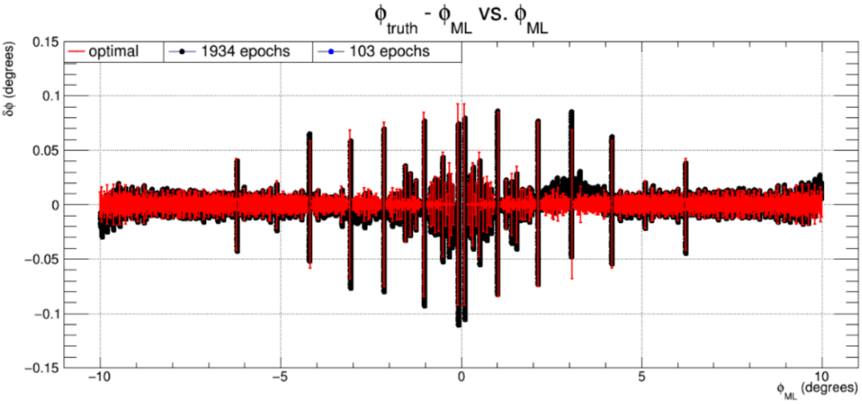

Figure 3. The difference between inference and truth versus the predicted value which shows how the difference is dependent on track angle and that the model faithfully reproduces this dependence.

3 CLAS12

At CLAS12 tracks are found through an iterative process. As CLAS12 moves to higher luminosity it is expected events will contain more hits, which in turn, will lead to an increase in the number of track segments and thus more combinations which must be iterated through. For example, the increase in hits can lead to wasted computational cycles as the software looks for tracks that don't exist (see Figure 4). It is hoped that machine learning driven signal cleaning can decrease the number of iterations needed to obtain a valid track and thus produce a faster total tracking algorithm.

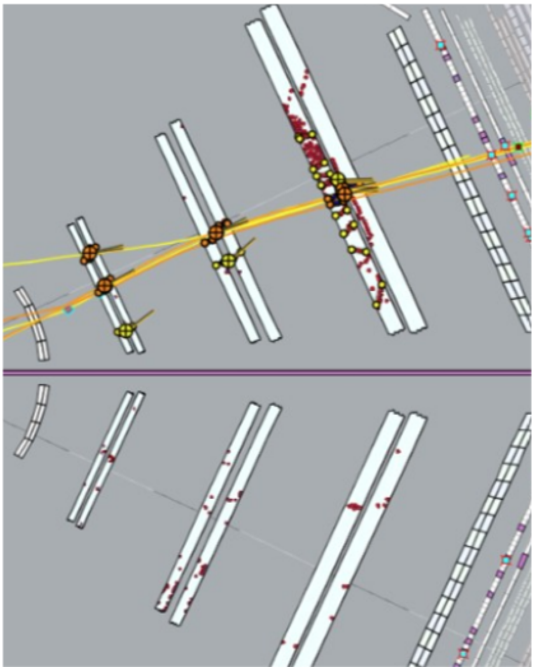

Figure 4. An example of how spurious hits increase combinatorics leading to, in this case, four track combinations being considered even though there are actually none.

Tracks in CLAS12 are reconstructed from "track segments" from different layers[5], with each layer contributing a single segment to a potential track even though a layer may contain more than one segment. This is easier to picture when one considers a two layer system with each layer containing two track segments; in this system there are exactly 4 track candidates where perhaps there may have only been one true track. Labeled data is obtained by taking all track candidates, that is to say, all combinations of track segments through all layers and putting them in to either the "positive sample", which contains valid tracks, or the "negative sample", which contain segment combinations that do not make a valid track. Examples from both of these samples can be found in Figure 5. Using VGG16[6], a convolutional neural network, brings the time of track finding down from about 15 ms using traditional means to approximately 3 ms. For single track events ML based algorithms beat traditional algorithms by a factor of 2.2. Two track events show an even larger effect with ML methods beating traditional methods by a factor of 6. It is important to note that these performance gains are driven purely by the ability of the machine learning algorithm to reduce combinatorics.

CLAS12, additionally, has formed an additional collaborative effort with Old Dominion University and obtained funding. This effort is in using recurrent neural networks to take the track segments from cleaner layers (roughly 2/3 of all the hits) and predict the hits in the noisier sections. In this way data can be more efficiently cleaned and segments more easily found. The output from this pass would feed in to other neural networks to reduce combinatorics. When complete, CLAS12 plans to turn its attention to estimation of the initial state vector.

In the intervening time between CHEP 2019 and now three different network types were investigated to assess the accuracy of track classifications. The three networks were namely a

Convolutional Neural Network (CNN), an Extremely Randomized Tree (ERT) and a Multy-Layer Perceptron (MLP). The results from each network can be seen in Table 1.

Network	Training Accuracy	Validation Accuracy	Inference Time
CNN	94.26%	90.15%	1.2 ms
ERT	99.99%	91.90%	5 μs
MLP	98.41%	99.2%	4 μs

Table 1. Performance of the three Neural Networks investigated for track classification. Namely, convolutional (CNN), Extremely Randomized Tree (ERT), and the Multi-Layer Perceptron (MLP)

Encouraged by the use of ML techniques in high energy physics, these methods have begun to find their way into the nuclear physics. The challenges are different enough, however, that careful study is warranted. A collaborative effort has begun between members of GlueX and CLAS12, with CLAS12 having entered into a funded partnership with Old Dominion University. These parallel research and development tracks hold the promise of reduced reconstruction times and increased code portability for nuclear physics.

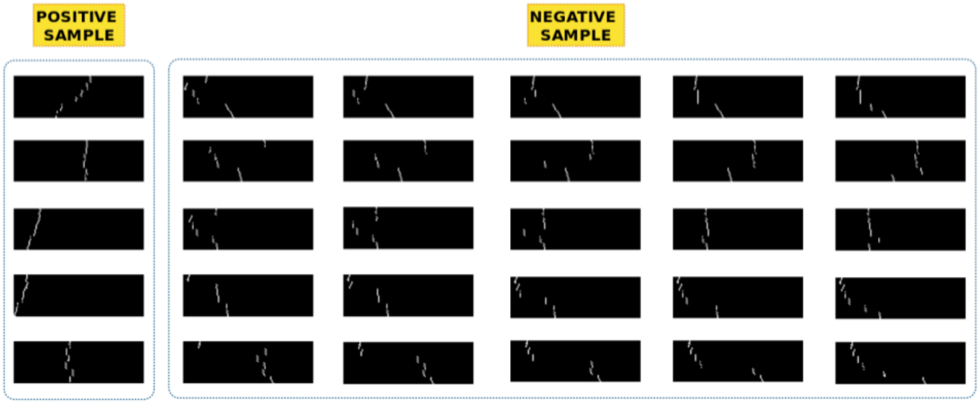

Figure 5. Some representatives of both the positive and negative samples

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics under contract DE-AC05-06OR23177.

References

[1] S. Adhikari et al. (2020), `2005.14272`

[2] V. Burkert, L. Elouadrhiri, K. Adhikari, S. Adhikari, M. Amaryan, D. Anderson, G. Angelini, M. Antonioli, H. Atac, S. Aune et al., Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **959**, 163419 (2020)

[3] F. Chollet et al., *Keras*, https://github.com/fchollet/keras (2015)

[4] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin et al., *TensorFlow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org, `https://www.tensorflow.org/`

[5] M. Mestayer et al., Nucl. Instrum. Meth. A **959**, 163518 (2020)

[6] K. Simonyan, A. Zisserman, *Very deep convolutional networks for large-scale image recognition* (2014), 1409.1556