Neural network approaches to track reconstruction in drift chambers

Y. Ichinohe[*1] and H. Baba[*1]

High-speed event reconstruction using hardware accelerators such as FPGA and GPU is receiving attention for developing next-generation data acquisition (DAQ) systems. We studied a strategy to speed up the event reconstruction of tracking detectors, particularly of drift chambers.

Drift chamber track reconstruction is usually time-consuming. For example, a common event reconstruction algorithm compares the χ^2 values for all possible combinations of hit patterns to find the most probable track for a given list of raw data associated with a single trigger. Two main factors increase the number of combinations: 1. multiple hits in a single layer increase the combination by a factor of the multiplicity. This factor increases as $\prod_i N_i$, where N_i is the hit multiplicity in the ith layer, because the multiplicity in each layer must be independently considered. 2. left-right uncertainty: the true track is tangent to the cylinders, whose axes correspond to the wires and whose radii correspond to the drift times measured for these wires. This constraint is considered through testing two possibilities for each wire hit, corresponding to the two sides (left or right) that the true track can be located with respect to the wire. This factor scales with 2^N, where N is the number of layers.

These conditions are typically implemented using the nested and variable-length loop syntax as both the hit multiplicity in each layer and the number of hit layers vary from event to event. The latency of the track reconstruction task is therefore variable depending on the event size, hindering the combination of this task with other event processing tasks that can be completed within a predetermined clock latency. We considered the use of deep learning as an alternative to traditional variable-latency algorithms to perform the track reconstruction task on variable-length input event lists with fixed latency.

We formulated the problem as inferring the track parameters (2D intersection coordinates X and Y as well as the corresponding slopes A and B) given a variable-length input list of hit information (layer ID, wire ID, and drift length). We first implemented a simple simulator to generate random hit patterns and their associated track parameters (Fig. 1) for training the neural networks.

We are currently exploring the architecture of neural network models to achieve sufficient prediction accuracy and prediction speed. Figure 2 shows some examples of model performance. The prediction performance improved when the architecture changed from multilayer perceptron (top two rows) to a graph attention network (bottom two rows). We will continue to explore the optimal model architecture and (hyper)parameters.

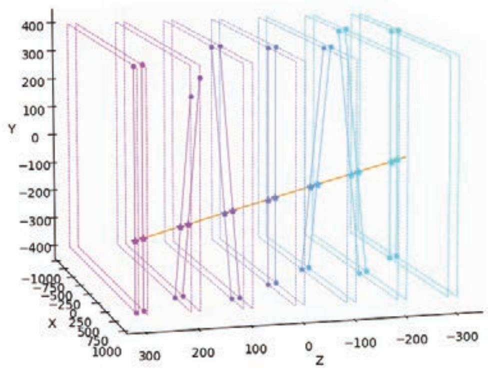

Fig. 1. Example output from event simulator. Orange line indicates the true track. Only the fired wires are visualized, and axes are in arbitrary units.

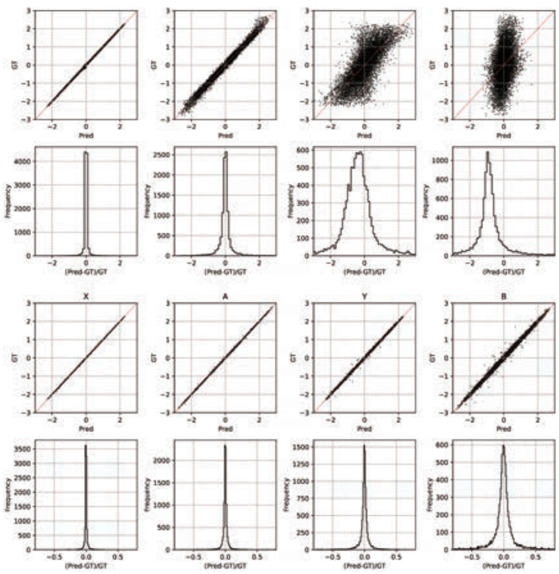

Fig. 2. Comparison between ground truth (simulated) data and neural network predictions. Multilayer perceptron was the model architecture for the top two rows and a graph attention network[1] for the bottom two rows. First and third rows: scatter plots of predicted (Pred) versus ground truth (GT) values. Axis units are arbitrary. Second and fourth rows: histograms of relative errors with respect to ground truth values. Panels correspond to parameters X, A, Y, and B, from left to right.

Reference

1) P. Veličković *et al.*, arXiv:1710.10903, (2018).

*1 RIKEN Nishina Center

DOI:10.34448/RIKEN.APR.58-159