

# ML (RNN) in MT

Sadman Ahmed Shanto

2020-11-03

## NOTES FROM RNN PAPER:

### Problems:

- Parallelization
- Quadratic Scaling
- ROI determination
- RNN > Kalman Filter

### Track Building:

#### RNN (regression)

- Makes great use sequential data
- reposes next-step hit predictions as regression problem
- Uses LSTM network for powerful non-linear sequence modeling capabilities

**Working principle:** Given a sequence of hit coordinates, the model produces for every element a prediction of the position of the next hit conditioned on its position and the preceding hit positions. The learning problem is thus cast as a multi- target regression problem and the model is trained with a mean-squared-error loss function.

We have to train model on tracks that hit all 4 detector layers.

#### RNN (gaussian)

- predictions in form of probability distribution
- same philosophy as regression
- The outputs of the network include the mean-value predictions as well as the parameters of a covariance matrix
- learning problem now is cast as a maximum likelihood estimation

### Building Tracks

- using three hits of a true track as a track “seed”
- methodology
  1. get track seed
  2. use RNN to make forward predictions
  3. select highest scoring hit in the event on each successive layer

### Track Finding:

#### Graph Neural Networks (GNN):

- Precondition: representation of tracking data with points is as a graph of connected hits
- A GNN model can learn on this representation and solve tasks with predictions over the graph nodes, edges, or global state
- The inputs to these models are the node features (the 3D hit coordinates) and the connectivity specification.

**Creating the Graph:** The graph can be constructed by connecting plausibly-related hits using geometric constraints or some kind of pre-processing algorithm like the Hough Transform.

### **Binary Hit Classification Model**

- learns to identify one track in a partially-labeled graph by classifying the graph nodes.

### **Binary Segment Classification Model**

- learns to identify many tracks at once by classifying the graph edges (hit pairs).

### **GNN Architecture:**

- EdgeNetwork: computes weights for every edge of the graph using the features of the start and end nodes.
- NodeNetwork: computes new features for every node using the edge weight aggregated features of the connected nodes on the previous and next detector layers separately as well as the nodes' current features.

Both the EdgeNetwork and NodeNetwork are implemented as Multi-Layer Perceptrons (MLPs) with two layers each and hyperbolic tangent hidden activations. The full Graph Neural Network model consists of an input transformation layer followed by recurrent alternating applications of the EdgeNetwork and NodeNetwork.

### **Graph Hit Classification**

The hit classification model performs binary classification of the nodes of the graph using labels to specify three seed hits. The graphs are constructed by taking four hits on each detector layer in the region around the true track location and connecting all hits together on adjacent layers. The model uses seven graph iterations followed by a final classification layer with sigmoid activation that operates on every node to predict whether the nodes belong to the target track or not.

## **TO INCLUDE:**

-[] CERN Presentation Material -[] Image Segmentation