B-Lunch Talk

NOVEL DEEP LEARNING METHODS FOR TRACK RECONSTRUCTION

6th November 2018

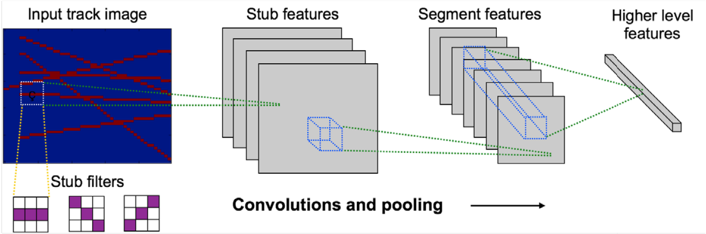
Michael Eliachevitch | ETP – KIT

DEEP LEARNING, THE FUTURE OF TRACK RECONSTRUCTION?

- HL-LHC: expected pileup of 200, with O(10k) particles and O(100k) hits
- existing track finding (combinatorial seed finding and track buidling) struggles
  - scales badly with $O(N^2)$ or worse
  - inherently serial
- new HEP.TrkX project: explore, whether machine learning can help
  - models non-linear dependencies
  - learn effective representations on high-dimensional data through training
  - parallizes easily, e.g. on GPU's

INITIAL IDEA: BORROW FROM IMAGE RECOGNITION

- deep learning has been applied very successfully computer vision and image recognition
- fixed number of features, aligned in euclidean grid
- learn higher order features via convolutional neural networks
- first approach: use image-like representation of tracking data [1]

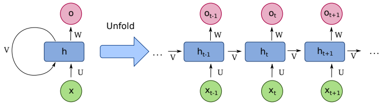


BACK TO SPACE POINTS

- image recognition algorithms don't scale to realistic HL-LHC conditions with high dimensionality and sparsity
  - loss of information due to image representation
  - real detector geometries not "grids"
- $\rightarrow$ need algorithms which can use the full precision of 3D space points ("hits")
- two candidates:
  1. Recurrent neural networks (RNN's)
  2. Graph neural networks (GNN's)

HIT PREDICTION WITH RECURRENT NEURAL NETWORKS

- RNN's useful for predicting/classifying the next item in a sequence, when it depends on the previous items
  - e.g. very successfully applied prediction of next word, speech recognition etc.



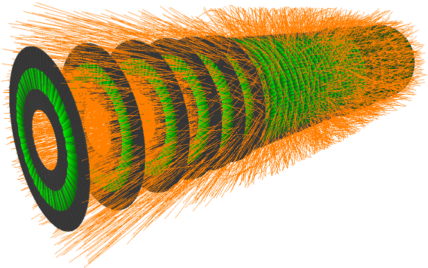
- based on previous hits in sequence, predict next one (regression)
- architecture: long short-term memory (LSTM) layer and fully conctected (FC) layer
- trained with mean-squared-error loss function

Simple hit predictor model

$$\boxed{\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{N-1}} \rightarrow \boxed{\text{LSTM}} \rightarrow \boxed{\text{FC}} \rightarrow \boxed{\hat{\vec{r}}_1, \hat{\vec{r}}_2, \ldots, \hat{\vec{r}}_N}$$

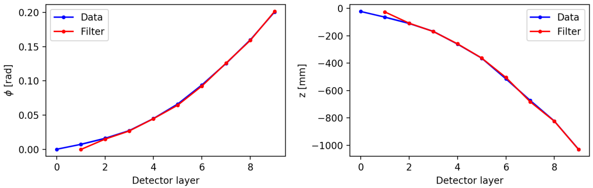
INTERLUDE: DATA USED FOR TESTS

- MC data from ACTS [2] with "generic"HL-LHC detector (image below)
- events with one hard QCD scattering process and an avarage of 10 soft QCD pileup interactions
- only tracks with $p > 1$ GeV
- require that all barrel 10 layers are hit, remove duplicate layer hits

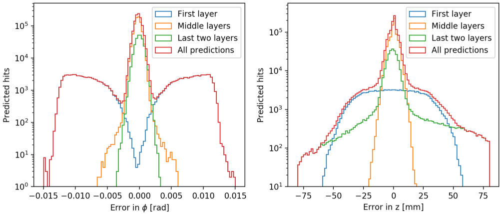


RESULTS OF SIMPLE HIT PREDICTOR

- prediciton of next hit for example track



- residual distributions



GAUSSIAN MODEL

- outputs are gaussians PDF's with mean $\hat{\vec{r}}$ and covariance Σ
- trained with max. log-likelihood loss function
  $L(r, \hat{r}, \Sigma) = \log|\Sigma| + (r - \hat{r})^T \Sigma^{-1} (r - \hat{r})$
- learns to predict own uncertainty
  - important for scoring hits candidates in candidates
  - intrinsic to Kalman filter

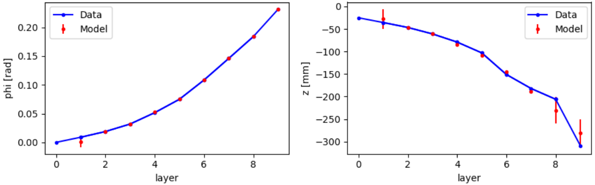
Gaussian hit predictor model

$$\boxed{\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{N-1}} \rightarrow \textbf{LSTM} \rightarrow \textbf{FC} \rightarrow \boxed{(\hat{\vec{r}}_1, \Sigma_1), (\hat{\vec{r}}_2, \Sigma_2), \ldots, (\hat{\vec{r}}_N, \Sigma_N),}$$

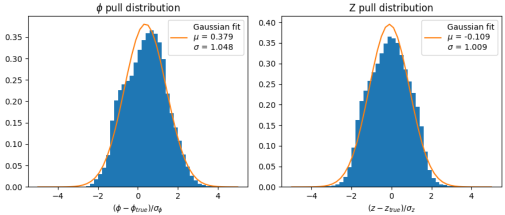
$$\vec{r} = (r, \phi, z) \qquad \hat{\vec{r}} = (\hat{\phi}, \hat{z}) \quad \Sigma = \begin{pmatrix} \sigma_\phi^2 & \sigma_{\phi z}^2 \\ \sigma_{\phi z}^2 & \sigma_z^2 \end{pmatrix}$$

GAUSSIAN MODEL RESULTS

- predictions of gaussian model for example track

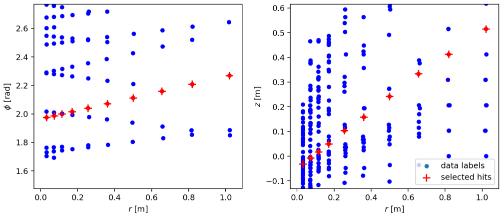


- pull-distribution: good prediction, but non-gaussian features



TRACK BUILDING PROOF-OF-CONCEPT

- simple topology: no B-field, low-occupancy, particle-gun
- seed of three true hits
- predict next hit with RNN, select closest measured hit to track

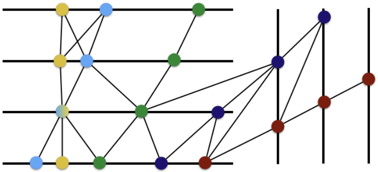


- combinatorial tree search algorithm needed for proper tracking

(like CKF with RNN instead of Kalman)

GRAPH NEURAL NETWORKS (GNN'S)

- part of Geometric Deep Learning [3]: exploit true geometry of problem domain instead of euclidean grid approach
- represent hits (space points) as nodes in graph
- connections (edges) can be restricted with geometric constraints/preprocessing
- two applications developed
  1. hit classification: Which nodes belong to some track?
  2. segment classification: Which edges correspond to same-track hits?



USED GNN ARCHITECTURE

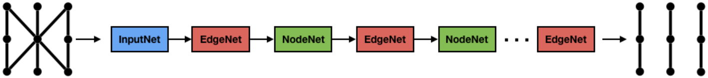
EdgeNetwork

For each edge, computes the weight based on the features of the two nodes which it connects.

NodeNetwork

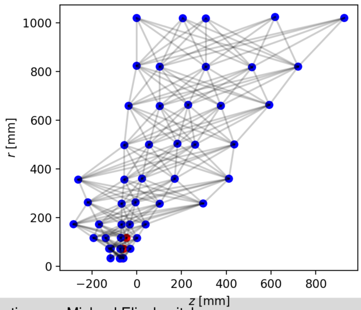
For each node, aggregates featuress of the connected nodes in the previous and next layer according to the edge weights, and the current node features.

- two-layer multi-layers perceptrons (MLP’s)
- applied in alternation
- information propagates through graph
- edges strengthened/weakened according to importance



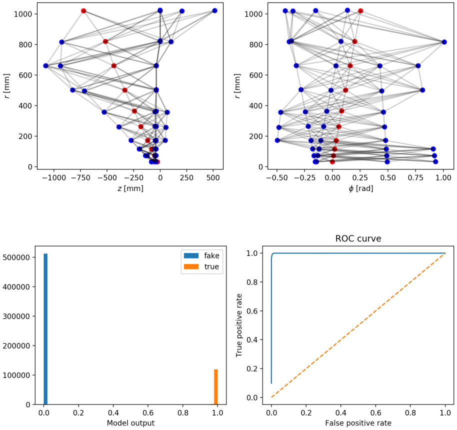
GRAPH HIT CLASSIFICATION

- starting from seed (three true hits), classify other hits whether they belong to track
- ideally, find all hits of single true track
- graph construction
  - take four hits in each layer around true track
  - connect all hits on adjacent layers
- use seven graph iterations with one final node classification layer



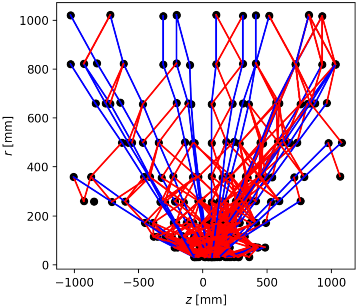
HIT CLASSIFICATION RESULTS

- 99.2% purity, 97.9% efficiency, 99.4% accuracy
- ROC-Curve: excellent separation



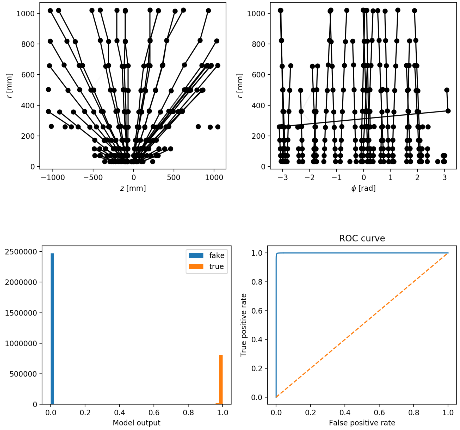
SEGMENT CLASSIFICATION

- classify edges, whether they connect two hits of same track
- graph construction
  - 45° cut on ϕ
  - 300 mm cut on z
- use four graph iterations and one final application of edge network



SEGMENT CLASSIFICATION RESULTS

- 99.5% purity, 98.7% efficiency, 99.5O% accuracy
- ROC-Curve: excellent separation



SUMMARY

- two methods to apply deep learning to tracking with exact space point hit presentations
- Recurrent Neural Networks similar to Kalman filter, use for track following
- Graph Neural Networks learn graph presentation of hit data
  - excellent results on toy data make hope that they scale for more realistic data
  - "most promising"deep learning solution to address HL-LHC tracking challenge
- TODO
  - built RNN into combinatorial track tree search akin to CKF and test track-building with full-occupancy data
  - turn the GNN's into actual track finders, also scale up to realistic data

REFERENCES AND FURTHER READING

- Steven Farrell et al.
  The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking.
  EPJ Web Conf., 150:00003, 2017.
- C Gumpert, A Salzburger, M Kiehn, J Hrdinka, N Calace, and ATLAS Collaboration.
  Acts: from atlas software towards a common track reconstruction software.
  Journal of Physics: Conference Series, 898(4):042011, 2017.
- Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst.
  Geometric Deep Learning: Going beyond Euclidean data.
  IEEE Sig. Proc. Mag., 34(4):18–42, 2017.
- Steven Farrell et al.
  Novel deep learning methods for track reconstruction.
  In *4th International Workshop Connecting The Dots 2018 (CTD2018)*

BACKUP

KALMAN FILTER

- Kalman filter principle: