Machine Learning For Track Finding at PANDA FTS

Waleed Esmail, Tobias Stockmanns, and James Ritman

On Behalf of the PANDA Collaboration

Institut für Kernphysik (IKP)
Forschungszentrum Jülich
Outlines:

- PANDA Detector.
- Forward Tracking System.
- Tracking Model.
- Artificial Neural Networks.
- Addition of Skewed Layers.
- Recurrent Neural Networks.
- Conclusion and outlook.
PANDA Detector:

antiProton ANnihillation at DArmstadt

- Hadron Spectroscopy
- Hadron Structure
- Hypernuclei
- $4\pi$ Acceptance
- Good Tracking
- Good PID

Beam Momentum: 1.5 GeV/c to 15 GeV/c
PANDA Detector: 

Forward Tracking System (FTS):

- Straw tubes, same as in the barrel, vertically arranged in double layers

- 3 stations with 2 chambers each
  - FTS1&2: Before the Field
  - FTS3&4: In the Field (2Tm)
  - FTS5&6: After the Field

- 8 double layers per chamber.
- Orientations $0^\circ/ +5^\circ/-5^\circ/0^\circ$ per chamber

- Tracks are defined by distance of closest approach to the wire (Isochrones)
Tracking Model:

The current approach

I. Create Track Segments by using k-menas.

II. Create Track Segments by using a deep neural network.

(FST1+FST2)
(FST3+FST4)
(FST5+FST6)

III. Interpolate Track Segments by using a recurrent neural network

<table>
<thead>
<tr>
<th>TrackSeg 1</th>
<th>TrackSeg 2</th>
<th>TrackSeg 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrackSeg 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrackSeg 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrackSeg 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create Track Segments by using k-menas.

✓ **k-means** algorithm works with unlabeled multidimensional dataset.

✓ **k-means** to group the samples based on their *feature similarities*.

✓ The Algorithm:

1. Randomly pick *k* clusters (*centroids*) from the sample points as initial cluster centres.

2. Assign each sample to the *nearest* centroid. $\mu_j: j \in \{1, ..., k\}$.

3. Move the centroids to the centre of the samples that were assigned to it.

4. **Iterations**, Repeat steps 2 and 3 until the cluster assignments do not change or maximum number of iterations is reached.

✓ how do we measure *similarity* between samples?!

  - $d(x, y)^2 = \sum_{j=1}^{m} (x_j - y_j)^2$.
  - Minimization of *cluster inertia*:

$$\text{Inertia} = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{i,j} \|x^i - \mu^j\|^2$$
How to specify the k centroids:

1. **Elbow** method is performance metric for clustering algorithms

   ✓ The *idea* behind the elbow method is to identify the value of \( k \) where the *distortion* begins to decrease most rapidly.

2. **Silhouette analysis** is a measure of how tightly grouped the samples in the clusters are. This can be done by defining *silhouette coefficient*:

   \[
   s^{(i)} = \frac{b^{(i)} - a^{(i)}}{\max\{b^{(i)}, a^{(i)}\}}
   \]

   ✓ \( a \) is *cluster cohesion*.

   ✓ and \( b \) is *cluster separation*.  

![Distortion vs Number of Clusters](image)
How to specify the k centroids:

1. **Elbow method** is a performance metric for clustering algorithms.
   - The idea behind the elbow method is to identify the value of \( k \) where the distortion begins to decrease most rapidly.

2. **Silhouette analysis** is a measure of how tightly grouped the samples in the clusters are. This can be done by defining the silhouette coefficient:
   - \( a \) is cluster cohesion.
   - \( b \) is cluster separation.

Fails in case of 1 track per event.
Artificial Neural Networks:

Application to the FTS:

✓ Create all possible combinations of hit pairs (adjacent layers).
✓ Train the network to predict if hit pairs are on the same track or not.

✓ Input observables:
  1) Hit pair positions in x-z projection (vertical layers).
  2) Drift radii (Isochrones).
  3) Distance between hits.

✓ Output:
  1) Probability that hit pair are on the same track.

✓ Connect hits that pass the probability cut (threshold).
  e.g.  probability(h₁-h₂) > threshold, and
         probability(h₂-h₃) > threshold, so
         h₁, h₂, h₃ are on the same track.
Artificial Neural Networks: Application to the FTS:

✓ **Network Architecture:**
  - 5 hidden layers
    (400, 300, 200, 100, 50)
  - Drop-out layers with 50%
  - Tanh activation
  - Last layer “Sigmoid” activation

✓ **Training data:** Particle gun
  - Momentum Range 1 - 10 GeV/c
  - Polar Angle 0.5° – 10°
  - 5μ per event
Artificial Neural Networks:

Network Accuracy:

FTS 1, 2, 5, 6

FTS 3, 4
Artificial Neural Networks:

Pictorial Representation (example found track):

FTS 1,2

FTS 3,4

FTS 5,6
Artificial Neural Networks:

Some Results:

✓ Data are generated by particle gun
 ✓ 1000 events (5 muons per event).

✓ Criteria:

1. If found track has less than 4 hits, do not count the track.
2. Find the matching particle (the matching particle is the one to which the majority of the tracks points belong).
3. Calculate purity: \( \frac{n_{\text{correct}}}{n_{\text{all}}} \) if purity > 0.8 count reconstructed track.

<table>
<thead>
<tr>
<th></th>
<th>FTS 1 &amp; FTS 2</th>
<th>FTS 3 &amp; FTS 4</th>
<th>FTS 5 &amp; FTS 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.96</td>
<td>0.95</td>
<td>0.96</td>
</tr>
</tbody>
</table>
Addition of Skewed Layers:

✓ The y-z track motion is extracted from the skewed layers.
✓ Thus x-z projection candidates are used as "seed" for such task.

1) **Fitting** (Orthogonal Distance Regression).

![Graphs of FTS 1,2, FTS 3,4, and FTS 5,6 showing linear and parabolic fits.]

Member of the Helmholtz Association

03.April.2019
Addition of Skewed Layers:

2) Using the fit predict the true x position of the skewed layers.

3) The distance between the skewed layers measurements and the predicted x position allows to identify a y measurement.

4) Collect all hits that have the same slope (y/z).

\[ \Delta y = \frac{\Delta x}{\tan \theta} \]

\[ \Delta x = x_{\text{pred}} - x_{\text{measured}} \]
Addition of Skewed Layers:

✓ The fitting provides the correct hit positions (tangent to isochrones).
✓ Linear fitting in y-z plane.
**Recurrent Neural Networks (RNN):**

**Long Short-Term Memory (LSTM):**

- LSTMs are a special kind of RNN, capable of learning long-term dependencies.
- LSTMs also have the same chain-like structure as simple RNN, but the repeating module has a different structure. Instead of a simple neuron, it is a cell.
- The key to LSTMs is the cell state.

![Diagram of LSTM](http://colah.github.io/posts/2015-08-Understanding-LSTMs)
Recruent Neural Networks (RNN):

Long Short-Term Memory (LSTM):

✓ All possible combinations of track segments in FTS(1,2), FTS(3,4), FTS(5,6).

✓ Input observables:
  1) Corrected hit positions (x,y,z).

✓ Network Architecture:
  - 3 hidden layers (Bidirectional LSTM)
    (300, 200, 100)
  - Drop-out layer with 50%.
  - Last layer “Sigmoid” activation.
Recurrent Neural Networks (RNN):

Results:

- Graph 1: Efficiency vs. Momentum [GeV/c] for 1 Track, 3 Track, and 5 Track.
- Graph 2: Purity vs. Momentum [GeV/c] for 1 Track, 3 Track, and 5 Track.
Conclusion and outlook:

✓ Proof of concept implementation.
✓ Results from ANN as well as RNN are promising.
✓ Deep learning methods are in principle applicable to tracking finding problem, at least in PANDA FTS.

✓ Test on real data.
✓ Track Fitting (RNN?!).
✓ Move the implementation to PandaRoot (C++).
Thank you for your Attention