A deep learning approach to PID and alignment of the DIRC

C. Fanelli

User Group Annual Meeting at JLab, June 2018
The GlueX detector in Hall D and the DIRC

DIRC will improve GlueX PID capabilities
(current π/K separation limited to 2 GeV/c)
Transportation and installation

Milestone:

- 6/5/2018 all of the DIRC bar boxes in Hall D
- No issues mechanically and optically
- 2 installed in the lower part
Optical Box

Particle track

Cherenkov photons

supporting bracket
MaPMTs
window
3-segmented mirror
flat mirrors
mirror support
steel box
3-seg parameters
θx, θy, θz
yoff, zoff
mirror support
BaBar bar box
Cherenkov Photon “Ring” in PMT plane

On average about ~30 photons detected per particle
Misalignment

- After installation the optical box will be filled by distilled water (refraction index close to bars).
- Optical box made by several components, system for calibration.
- During data-taking this becomes a black-box problem with many non-differentiable terms.
  - relative alignment of the tracking system with the location and angle of the bars
  - mirrors shifts cause parts of the image change
  - other offsets
- These aspects make seemingly impossible to analytically understand the change in PMT pattern
A complex inverse problem

- A grid search with high dimensionality seems less suitable.
- Need of dedicated algorithms for alignment optimization using pure samples of pions from Ks.
Hit Patterns

- 3D (x,y,t) readout and this allows to separate spatial overlaps.
- Patterns take up significant fractions of the PMT in x,y and are read out over 50-100 ns due to propagation time in bars.
- H12700 PMTs have a time resolution of O(200 ps) and read-out electronics giving time information in 1 ns buckets.

DIRC rings for π⁺ plotted with time on the z-axis.

Credits:
J. Hardin, PhD thesis
J. Hardin and M. Williams, JINST 11.10 (2016)
Bayesian Optimization

- BO is a strategy for global optimization of black-box functions.
- After gathering evaluations BO builds a posterior distribution used to construct an acquisition function.
- This determines what is next query point.
- BO is agnostic to what is optimized.

Select high purity sample of incoming charged particles \((p, \theta, \varphi)\), e.g. \(\pi\)'s.
Bayesian Optimization

- BO worked amazingly well for tuning MC (Ilten, Williams, Yang [1610.08328]).
- BO could align the DIRC detector leveraging current reconstruction algorithms.
Bayesian Optimization

Toy model
- 3 parameters:
  - 3-seg mirror $\theta_x$, $\theta_y$, $\theta_z$

Each call based on high purity sample of (only) 100 pions

true:
(0.50, 1.0, -0.50) deg

found:
(0.48, 0.9, -0.44) deg

Grid search
true offsets $<\theta_x, \theta_y, \theta_z>$:
(0.50, 1.0, -0.50) deg

BO

Select high purity sample of incoming charged particles ($p, \theta, \phi$), e.g. $\pi$'s
Detector Optimization

- Optimization of detector design is quite complex problem that can be accomplished with BO

- Multi-purpose detector like EIC requires large-scale simulations of the main processes to make decision

- Goal: satisfy detector requirements and minimize cost R&D

The BO is agnostic to what is optimized

POSTERIOR MODEL

OBJECTIVE FUNCTION

(e.g. maximize Objective Function)

exploration/exploitation trade-off

new detector configuration

(geometrical, type, ...)

HEP events simulation

Detector simulation

observations

(detector response)

Hyperparameters space
Reconstruction Algorithms and PID

1. Creation of the LUT: store directions at the end of the radiator for each hit pixel
2. Direction from the LUT for the hit pixels are combined with the track directions (from tracking)


J. Hardin and M. Williams, JINST 11.10 (2016)

Fast tracing mapping straight lines through a tiled plane
1. Generation - 2. Traces through bars - 3. Traces through expansion volume

KDE-based

 basically a trade-off memory/CPU usage
faster reconstruction/hit pattern
better resolution in regions with high overlap

LUT-based geometrical

Fast tracing mapping straight lines through a tiled plane
1. Generation - 2. Traces through bars - 3. Traces through expansion volume

KDE-based

 basically a trade-off memory/CPU usage
faster reconstruction/hit pattern
better resolution in regions with high overlap

https://github.com/jmhardin/FasDIRC
Can we use the same technology that Facebook uses to recognize faces for doing particle ID?
Any technique enabling computers to mimic human behaviour
Artificial Intelligence

A subset of AI based on statistical methods to enable machines to improve with experiences

Machine Learning
Artificial Intelligence

Machine Learning

Deep Learning

A subset of ML which makes the computation of multi-layer Neural Network feasible

when applied to massive datasets (as particle physics experiments) and giving massive computer power it outperforms all other models most of the time
● “Hottest” field in AI, and it’s everywhere

● use of ML (and DL) in HEP is becoming ubiquitous
How does it work?

- The real magic about NN is the result of an optimization technique: back-propagation (how a NN works to improve its output over time)
- DL (more hidden) nets are good in learning non-linear functions (heavy processing tasks)
- Based on old school NN revitalized by augmented capabilities (e.g. GPU) and a plethora of new architectures (RNN, CNN, autoencoders, GAN, etc.)
Why does it work?

- Debatable and may seem a dark art (e.g., pruning/dropout neurons, transfer learning)
- No doubts it works…

Forward Propagation

Backward Propagation

Error Estimation

Layer $L_3$

$a_1^{(2)}$, $a_2^{(2)}$, $a_3^{(2)}$

$h_{W,b}(x)$

Google DeepMind

Deep Q-learning playing Atari Breakout

Mnih et al., 1312.5602

Example PID: NN vs Likelihood Approach

- LHCb uses NNs trained on 32 features from all subsystems each of which is trained to identify a specific particle type.

- Standard candles are used as calibration samples to characterize performance of NN.

Typically getting ~3x less misID background per particle.
Deep Learning

- In particle physics, our goal is often (if not always) that of distinguishing signal from background.
- We need to build high-level features (e.g. invariant masses) to accomplish this task.
- DNN does not need our “help” and can learn from low-level features.

The DNN is able to learn all that it needs in this case, as providing high-level features results in no gains. DNN using low-level features outperforms any selection based only on high-level features.
Convolutional Neural Network

- CNN architecture is inspired by human visual cortex. An image can be thought as a group of numbers each describing an intensity value. This can be input in a NN for classification in output.

- The neurons in a CNN look for local examples of translationally invariant features. This is done using convolutional filters to locate patterns producing maps of simple features, then build complex features using many layers of simple feature maps.

CNN “scans the image”
http://scs.ryerson.ca/~aharley/vis/conv/
CNNs for neutrinos

- MicroBoone has managed to train CNNs that can locate neutrino interactions within an event (draw bounding boxes), identify objects and assign pixels to them arXiv 1611.05531

Similar work ongoing at other ν-experiments (see, e.g., NOvA [1604.01444]), and also at colliders in the area of jet physics [1511.05190], [1603.09349], …)

DNN at NOvA led to an impressive improvement for ν_\text{e}-detection
(equivalent to ~ 30% more in exposure than previous PID techniques: $$$)
Generative Adversarial Network

Fast Simulations

- Detailed simulation of detector response is provided by amazing tools like Geant, which is slow and often prohibitive for generating large enough samples.
- Cutting-edge application of deep learning uses GAN for fast simulation.
- 2-NN game, one model maps noise to images, the other classifies the images if real or fake.
- The goal is to confuse the discriminator.

- CALOGAN: Paganini, de Oliveira, Nachman 1705.02355
- jet images production: 1701.05927

CALOGAN can generate the reconstructed CALO image using random noise, skipping the GEANT and RECO steps.
Generative Adversarial Network

arXiv:1406.2661

Fast Simulations

- Detailed simulation of detector response is provided by amazing tools like Geant, which is slow and often prohibitive for generating large enough samples.
- Cutting-edge application of deep learning uses GAN for fast simulation.
- 2-NN game, one model maps noise to images, the other classifies the images if real or fake.
- **The goal is to confuse the discriminator.**

CALOGAN can generate the reconstructed CALO image using random noise, skipping the GEANT and RECO steps.

- CALOGAN: Paganini, de Oliveira, Nachman 1705.02355
- jet images production: 1701.05927
Deliverables and timeline

- DIRC alignment with BO. ~ ½ year
- Explore deep nets architectures for GlueX DIRC/PID; determination of best approaches enhanced by BO hyperparameter tuning (on a longer term ~ 1 year).
- Collaboration: MIT/DIRC group.
- Resources: deep learning workstation.
Summary

- Use of ML became ubiquitous in particle physics. Deep Learning is starting to make an impact.

- A lot of exciting cutting-edge work not discussed, e.g., DKF, DNNs on FPGAs, automatic anomaly detection, compression using autoencoders, etc.

- Systematics are vital in our field: we are developing systematics aware ML algorithms and we are characterizing “black boxes” (e.g. DIRC optical box, EIC detector design).

- Beyond the issue of systematics, our data have other interesting features from a CS perspective: sparse data, irregularities in detector geometries, heterogeneous information, physical symmetries and conservation laws (e.g. recursive NN), etc.

We just began to scratch the surface when it comes to using tools like BO and DL and recent achievements in our field (e.g. NOvA, LHCb, ...) suggest it’s worth exploring these strategies.
BACKUP
Deep jet tagging

- A jet in LHC is a spray of hadrons from shower initiated by some fundamental particle
- Define set of features with distributions depending on the jet nature
- Train NN using a sample of jets whose nature is known
Deep jet tagging

- DNN based on high-level features (jet masses, multiplicities, energy correlation functions, etc.)

16 inputs
64 nodes
activation: ReLU
32 nodes
activation: ReLU
32 nodes
activation: ReLU
5 outputs
activation: SoftMax

J. Duarte et al arXiv:1804.06913v2
D. Guest et al arXiv:1607.08633v2
Intuitively understanding CNN

![Diagram](image)

![Matrix](image)

![Values](image)
Other Architectures: Recurrent & Recursive NN

- Example: jet physics.
- More efficient than image-based networks.

G. Louppe, K. Cho, C. Becot, and K. Cranmer 1702.00748v1
Other Architectures: Recurrent & Recursive NN

Figure 2. Typical tree structures for 1 TeV gluon jet (left) and quark jet (right).

T. Cheng [1711.02633v1]
## Frameworks

### Deep learning libraries:
**Accumulated GitHub metrics**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Popularity</th>
<th>Library/Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>172.29</td>
<td>tensorflow/tensorflow</td>
</tr>
<tr>
<td>#2</td>
<td>89.78</td>
<td>BVLC/caffe</td>
</tr>
<tr>
<td>#3</td>
<td>69.70</td>
<td>fchollet/keras</td>
</tr>
<tr>
<td>#4</td>
<td>53.09</td>
<td>dmlc/mxnet</td>
</tr>
<tr>
<td>#5</td>
<td>38.23</td>
<td>Theano/Theano</td>
</tr>
<tr>
<td>#6</td>
<td>29.86</td>
<td>deeplearning4j/deeplearning4j</td>
</tr>
<tr>
<td>#7</td>
<td>27.99</td>
<td>Microsoft/CNTK</td>
</tr>
<tr>
<td>#8</td>
<td>17.36</td>
<td>torch/torch7</td>
</tr>
<tr>
<td>#9</td>
<td>14.43</td>
<td>baidu/paddle</td>
</tr>
<tr>
<td>#10</td>
<td>13.10</td>
<td>pfnet/chainer</td>
</tr>
<tr>
<td>#11</td>
<td>12.37</td>
<td>NVIDIA/DIGITS</td>
</tr>
<tr>
<td>#12</td>
<td>10.42</td>
<td>tflearn/tflearn</td>
</tr>
<tr>
<td>#13</td>
<td>9.20</td>
<td>pytorch/pytorch</td>
</tr>
</tbody>
</table>
Geometrical Reconstruction

BABAR-like

Has two stages:

1. Creation of the look-up table (LUT): store directions at the end of the radiator for each hit pixel

2. Directions from the LUT for the hit pixels are combined with the track direction (from the tracking system)

Geometrical Reconstruction

Kaons and pions with $p = 4$ GeV/c, $\theta = 11^\circ$, $\phi = 90^\circ$

Log likelihood is based on the Cherenkov angle and the number of detected photons
Timing information is used to cut out some solutions

Reconstructed Cherenkov angle

Separation between kaons and pions = 4.95 s. d.

Design goal: $\geq 3$ s.d. between pions and kaons for momenta up to 4 GeV/c

$\sigma^2_c = \sigma^2_{tr} + \frac{\sigma^2}{N}$
Time-based imaging

Calculate log likelihoods for each particle hypothesis directly from the time spectrum in each hit pixel

In advance need time spectra for each particle type and the given track configuration (momentum, direction)

Such time spectra can be simulated or calculated analytically

kaons 2 GeV/c, $\theta=1.2^\circ$, $\phi=90^\circ$

pions 2 GeV/c, $\theta=1.2^\circ$, $\phi=90^\circ$
Time-based imaging

4 GeV/c pions and kaons, theta = 1.2°, phi = 90°

\[ \log \mathcal{L}_h = \sum_{i=1}^{N} \log \left( \frac{S_h(x_i, y_i, t_i) + B(x_i, y_i, t_i)}{N_e} \right) + \log p_N(N_e) \]

Right now we simulate PDFs
Belle II TOP uses **analytical PDFs**, and we plan to try this method
FastDIRC - KDE based Reconstruction

- Use Kernel Density Estimation to compare particle hypothesis patterns
- Use fast tracing to implement KDE
  - $O(1)$ speed vs $O(100)$ bounces
- Compare Log Likelihoods

$$P(x) \approx \sum_{i}^{n} K(x - s_i)$$

See the FastDIRC paper (J Hardin, M Williams 1608.01180)
Github: https://github.com/jmhardin/FastDIRC
FastDIRC - KDE based Reconstruction

\[ \text{Fraction Worse} = \frac{\text{LUTres} - \text{FastDIRCres}}{\text{LUTres}} \]