Generating Cherenkov rings with convolutional neural networks

NN group meeting

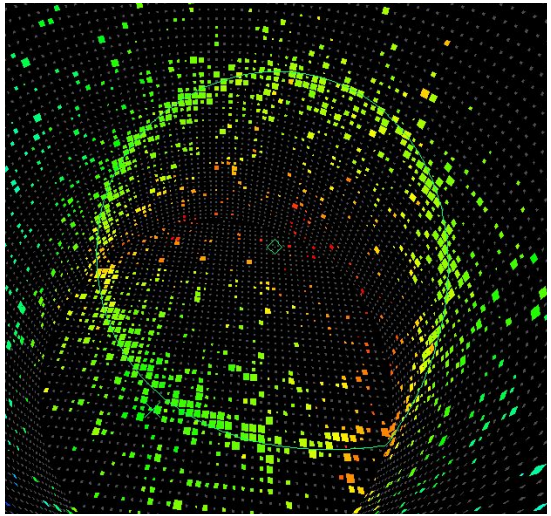
July 11 2019

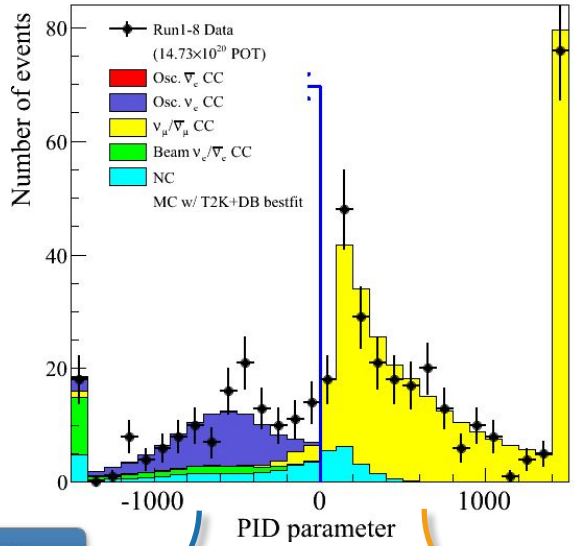
Cristóvão Vilela

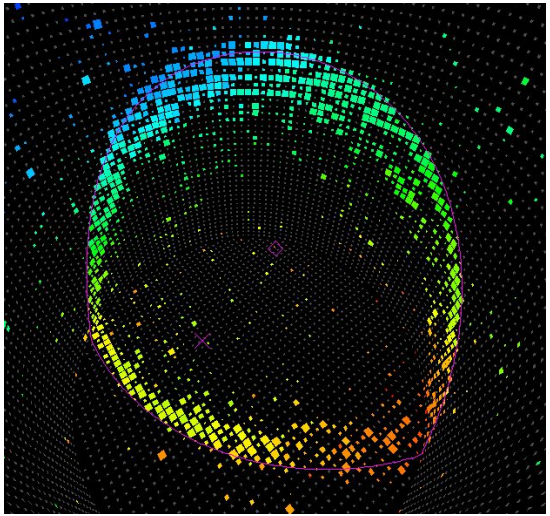
Introduction

- There is an effort to investigate using deep-learning techniques for water Cherenkov event reconstruction.
  - Mostly based in Canada/TRIUMF and focused on Hyper-K.
  - Kick-off workshop held at UVic in April.
- At Stony Brook, we've been thinking of ways to improve the FiTQun likelihood function for a while, possibly using machine learning techniques.
  - E.g.: looking into replacing 6D look-up table with boosted decision tree in order to further increase dimensionality.
- After attending the workshop, I've been looking into developing a maximum-likelihood reconstruction algorithm (i.e., like FiTQun) using convolutional neural networks to capture the likelihood function.

Water Cherenkov event reconstruction





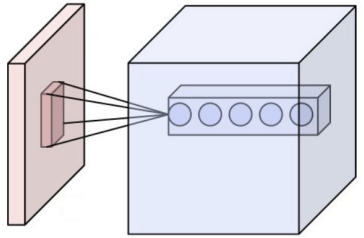


e-like ring

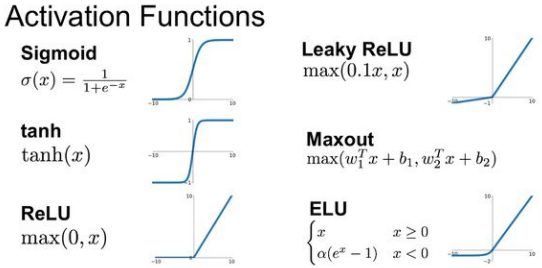
μ-like ring

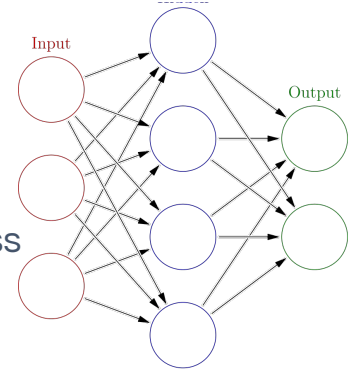
- Given event (observed q, t for each PMT) want to estimate:
  - How many particles? Particle type, momentum and position for each particle.
  - End-goal is to classify event according to topologies of interest:
    - nueCC0pi, numuCCpi, nueCC1pi$^+$, proton -> e$^+$+pi^0, ...

Neural networks, in < nutshell



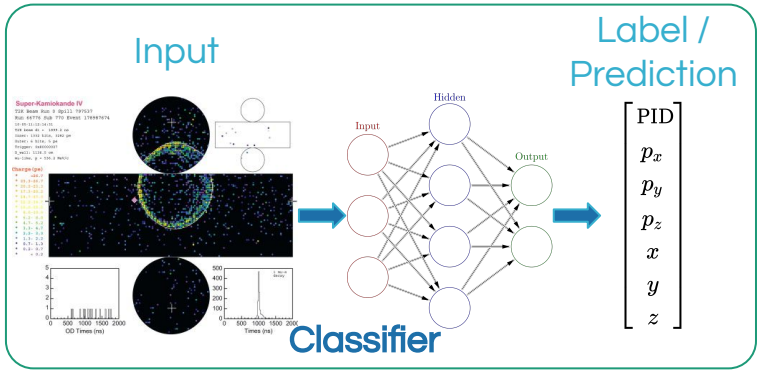
- Machine learning technique that mimics biological brains.
- Development exploded in recent years (~2012) mainly due to success in overcoming “trainability” issues.
- Network made of layers of “nodes”.
  - Layers connected to each other:
    - Fully connected.
    - Convolutional (space-aware).
    - …
  - “Connections” are linear transformations of preceding layers outputs followed by an “activation function”.
    - Non-linear!
- “Training” updates linear transformation parameters in order to minimize a loss function.
  - Choice of loss depends on the problem to be solved.
  - Work backward from the loss using “backpropagation”, i.e., the chain rule.

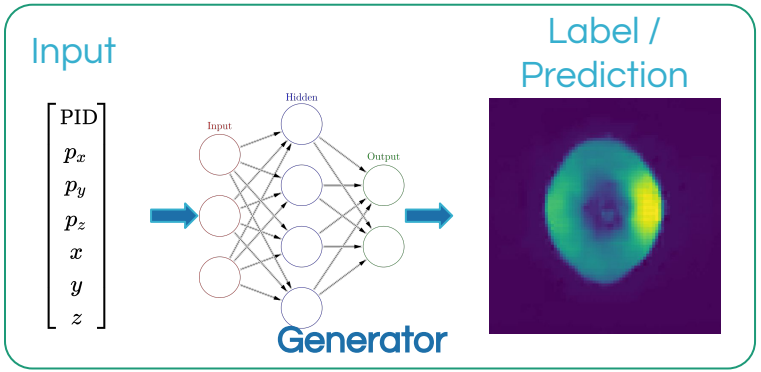




Reconstruction with CNNs

- We are exploring an alternative approach to the more traditional “end-to-end” CNN event classification for reconstruction of water Cherenkov events.





- A CNN is trained to predict the hit charges and times for a given set of track parameters.
- This Cherenkov ring generator is incorporated into a maximum-likelihood estimation framework to form an event reconstruction algorithm.
  - This method is analogous to FiTQun reconstruction: the CNN replaces the parameterized charge and time pdf prediction.
- While I don’t necessarily expect this method to outperform the end-to-end CNN classifier’s accuracy, it has potential advantages in the context of physics analyses.

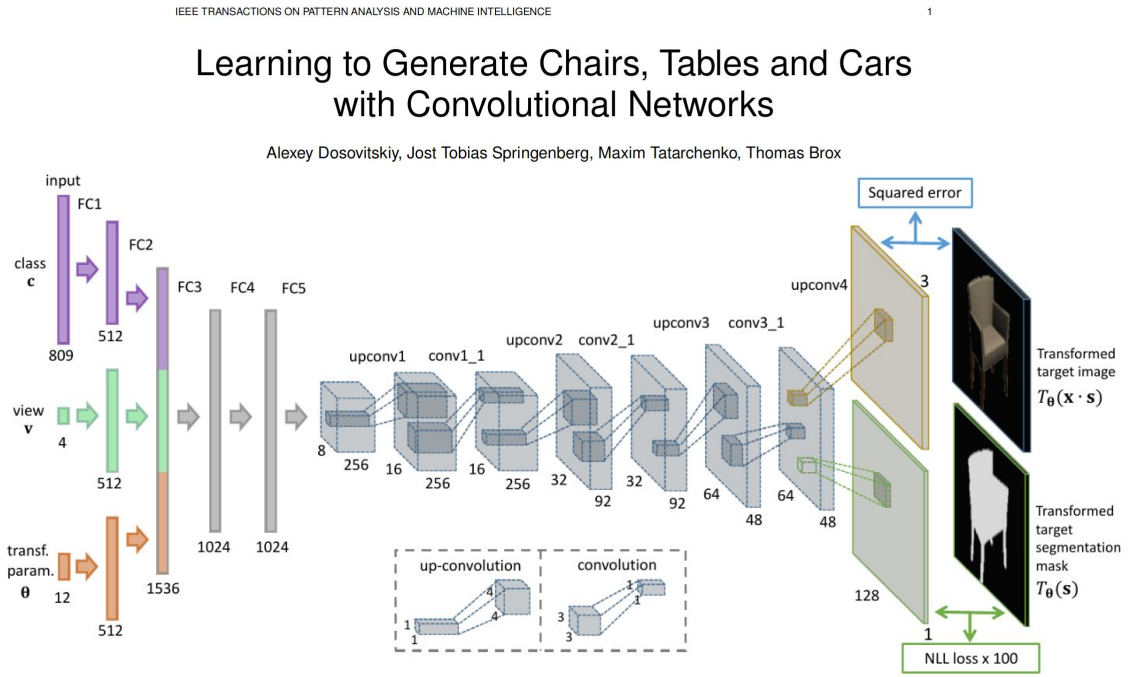
Why this might be interesting

- Single-ring predictions can be combined to predict arbitrary event hypotheses.
  - E.g.: in FiTQun mean predicted charges at each PMT are added up and time pdfs are combined, weighted by charge.
- Neural network can be trained on single-particle MC:
  - A priori not relying on problematic neutrino interaction and secondary interaction models.
  - Avoid multi-particle final states combinatorics.
- “Interesting” event topologies do not need to be defined at training stage.
  - Analyzers have flexibility to produce very specific event hypotheses out of single-ring predictions without having to retrain the neural network.
    - E.g.: proton decay to kaon and neutrino analysis with FiTQun specifies event with single de-excitation gamma followed (12 ns) by mono-energetic muon.
- This reconstruction approach would be a drop-in replacement for FiTQun.
  - Could be used with current analysis and systematic uncertainty estimation techniques.
  - Could be a useful first step in the move towards end-to-end ML reconstruction.

Generating images with CNNs

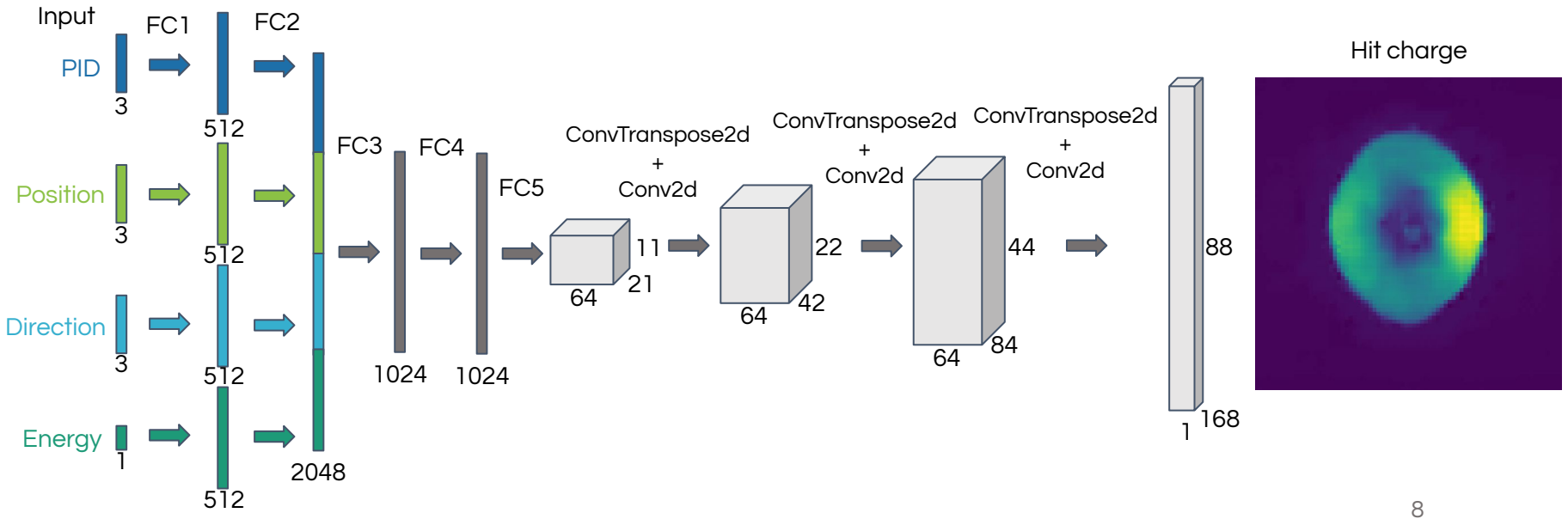
- First iteration of a Cherenkov ring generator neural network follows approach in

IEEE Trans. Pattern Anal. Mach. Intell. 39(4): 692-705, Apr 2017 (arXiv:1411.5928 [cs.CV]).



Generating rings with CNNs

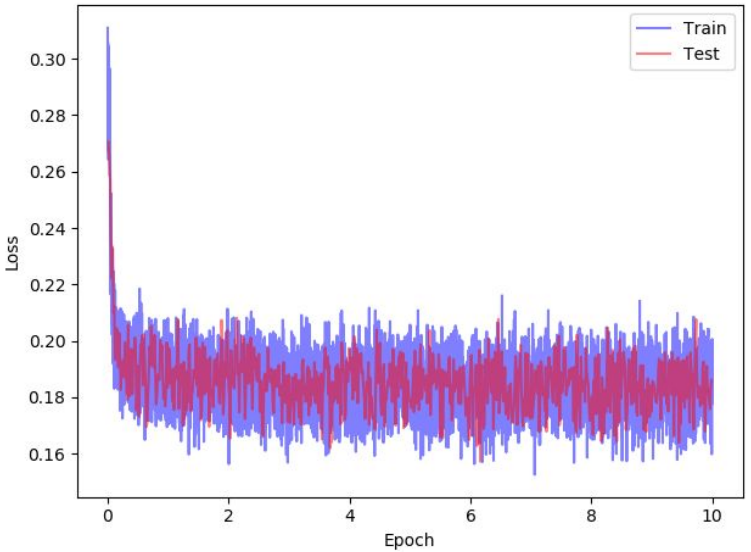
- Follow network architecture described in the paper as close as possible, output is the observed (mean) charge at each PMT in the barrel.
  - Almost certainly not optimal, just want to see if it works.
  - Implemented in PyTorch, based on Kazu's examples from the workshop.



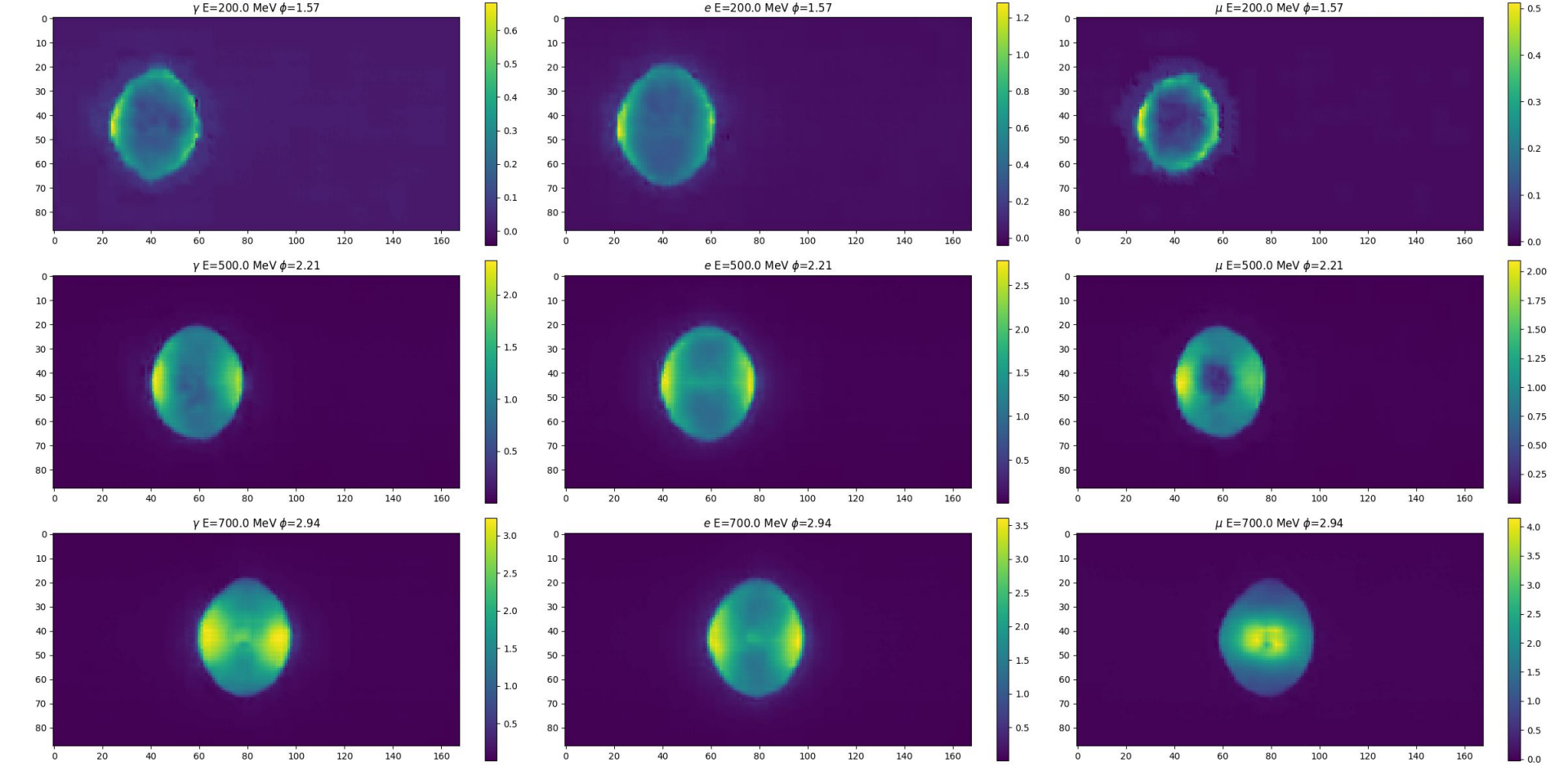
Training

- Used training sample prepared by Nick for the workshop:
  - 1M of each: electrons, gammas and muons.
  - Batch size: 200, train for 10 epochs (~day using PC w/ GPU)
  - SmoothL1Loss (Huber)
  - Adam optimizer

- Hitting loss "floor" due to variation in the samples themselves?
  - Move from predicting mean charge to predicting the charge pdf?
    - E.g.: Output of the network is a Gaussian mixture model?
  - Capture event-by-event variation with additional "latent" input parameters? Something along the lines of a VAE? Not sure if feasible…
- Maybe just terrible network architecture…



CNN-generated rings



Plans

- Ramp up on this work at Stony Brook over the coming weeks.
  - Focus will likely be toward T2K and Super-K reconstruction
- Short/medium term tasks:
  - Generate large training Super-K training sample using SKDETSIM.
  - Investigate neural network architecture further:
    - Effects of layer size and number.
    - Try to implement pdf output, rather than mean charges.
    - Look into introducing latent features on the input.
      - Conditional generative adversarial network?
    - Add hit times to the output - will need some kind of pdf output first, I don't think mean hit times will work...
  - Run basic checks of how this would look like as a reconstruction tool.
    - Start with simple likelihood scans, using FiTQun to pre-process events.