Particle Identification in Cherenkov Detectors using Convolutional Neural Networks

THEODORE TOMALTY

Supervisor: Stefania Bordoni

I INTRODUCTION

CHERENKOV detectors are used for charged particle identification. When a charged particle moves through a medium faster than light can propagate in that medium, Cherenkov radiation is released in the shape of a cone in the direction of movement. The interior of the Cherenkov detector is instrumented with PMTs to detect this Cherenkov light. Particles, then, can be identified by the shapes of the images on the detector walls.

In neutrino experiments, water Cherenkov detectors are commonly used to distinguish between electrons and muons, which determines the flavour of the interacting neutrino in the medium. An example of this kind of detector is Super-Kamiokande [2]: a cylinder filled with 50 kilotonnes of water located underground in Japan.

Standard event identification uses likelihood fits to the PMT charge and timing information. In particular, theoretical Cherenkov cones are fitted to the raw data using different particle-type hypotheses, to see which one corresponds best. This project will test the use of a machine learning algorithm as an alternative to the standard e μ separation, and compare its performance to the methods already in practice.

Specifically, we will be using a Python machine learning framework, Tensorflow [1], to train a Convolutional Neural Network (CNN). The goal will be to achieve the highest classification performance possible on simulated electron and muon events in Super-K.

II DATA AND PROCESSING

To train a neural network, a large number of of statistics are required. The standard Super-K particle simulation was run on 400,000 events that were split into a training sample and a testing sample. An equal number of electrons and muons are generated, with vertexes homogeneously distributed in the detector volume and initial energy evenly distributed between 300MeV and 1GeV. Each event in the simulation returns a map of the detector response including PMT integrated charge, which is proportional to light intensity, and timing information. The standard analysis fitting algorithm, fiTQun [3] [4], is run on all of these events to return a particle-type prediction as well as approximations of the particle's kinematics (e.g. vertex position and momentum).

2.1 Image Production

Each PMT in a Cherenkov detector produces a measure of light intensity and has a fixed position in 3D space, much like a single pixel on a cylindrical image. Convolutional Neural Networks function by scanning square images, using different filters, to identify features. A first problem we encounter is that of generalizing flat convolution (scanning) to a cylindrical one. Another problem arises because particles can originate at any point in the detector, and move in any direction, so there are many degrees of freedom in the shape of the rings on the cylinder. They can have different sizes, elongations, and further irregularities arising in the corners of the cylinder. Such ring deformations can make learning very difficult for the algorithm.

In order to simplify the problem we focus the algorithm on the Cherenkov profiles rather than their projection onto the cylinder. This is done, as indicated in figure 1, using a conical projection of PMT charges onto a flat, transverse plane. The standard analysis reconstructed vertex is used as the point of origin for the projection. Instead of warped rings on a cylinder we will get circular rings on a flat image, so both problems described above are solved this way.

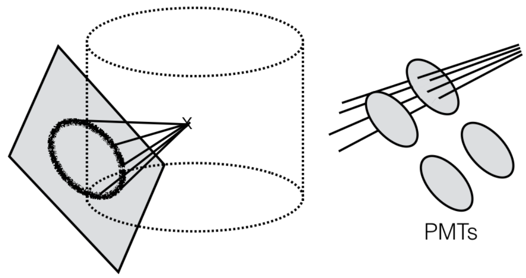

Figure 1: *Process the detector output by performing a conical projection of the PMT signals onto transverse-planar image.*

A blank 30×30 pixel image is set up in 3D space. It is centred at the intersection of the particle trajectory with the detector walls, normal to the direction of the particle, and its width and height are scaled with the distance traversed by the particle. The position of the plane is chosen so that the projected rings are centred, and so the resolution of the image can be compared to the resolution of the detector. The orientation of the image is chosen in order to ensure the rings are all circular. In addition, the image size is scaled to have a length and width proportional to the expected ring radius (so that all the rings appear the same size in the images).

Ring projections are done by considering the lines between each PMT and the particle vertex. If the line intersects the image, the corresponding PMT signal is added to the weight of the intersection pixel. This converts a complex shape on the detector wall to a circle on the image plane. It is often the case, however, that some PMTs are much closer to the vertex than others, and the resulting difference in angular resolution of the detector is enough to make the output image quite irregular. To smooth out the image, a flux of randomly distributed lines intersecting a PMT, as shown in figure 1, are set to carry a fraction of the PMT's signal. These lines are projected onto the image individually instead of a single line projection per PMT.

The question of scaling was mentioned briefly above. The image size is set proportional to vertex-to-wall distance in the direction the particle travels. This way, since Cherenkov cones all have the same angle with respect to the vertex, the rings have the same radius on the output image. However, since it is not ideal to use an image resolution that is much higher than the PMT resolution, the events are separated into four image sets with different distance-to-wall ranges. Three of those sets are shown in figure 2. Each set has a different size of ring on the image, selected to correspond with PMT resolution, but all the images in a single set have the same ring size. This way, we train a separate CNN for each set so that an individual network does not have to deal with differences in ring shape due to resolution issues.

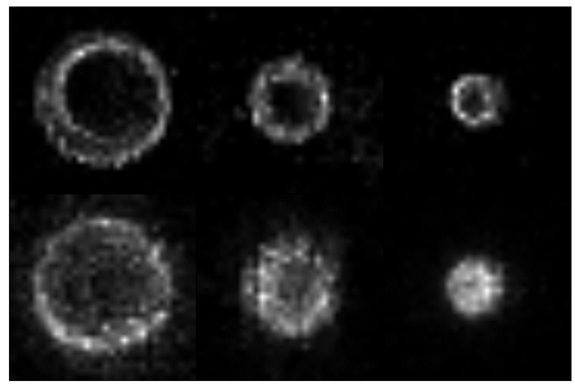

Figure 2: *Output images of the cone projection described in figure 1. Top: muon rings, Bottom: electron rings, Left to Right: data set 1, 3, and 4.*

Once this procedure returns a stream of square images of circular rings, like those in figure 2, a cut must be made before passing these to the CNN. It is standard to remove events from the analysis with vertexes closer than 2m from the detector walls, so this cut is performed using the reconstructed vertex from fiTQun. In cases of particles very close the walls, however, the fiTQun reconstruction can be quite poor due to a lack of significant PMT hits from Cherenkov radiation. To remove these events, a second cut is done on the number of PMTs hit in the detector.

III The Algorithm

Convolutional neural networks [5] are standard for image recognition problems, and will be used for this project. The algorithm takes input from 30×30 pixel images, as in figure 2, and will be tasked with classifying the rings as electron- or muon-shaped. The CNN will be designed to process the images analogous to how a human might.

To the eye the classification is easy because muon rings have sharp edges and relatively empty interiors, while electron rings have fuzzy edges with full interiors. The differences arise because electrons move through water in a scattered trajectory, whereas muons travel relatively straight through the water.

3.1 Network Architecture

Similar to the mammalian visual cortex, CNNs begin with searching an image for simple features like edge shape or colour gradients. This is done by creating $n \times n$ pixel filters, that visually represent the features in question, and doing a discrete convolution of the filter with the image. The output feature maps of the convolution are activated in regions where there is a strong correspondence between the filter and the image.

For our network, 5×5 pixel filters were used to correspond roughly with pixel scale of the ring edges. The filters were set manually in this project because the network was observed to have difficulty optimizing the filter shapes independently. The initial filters shown in figure 3 contain sharp and smooth edges, each rotated eight times for complete investigation of the circle, as well as reflection-invariant filters that will be used to probe the ring shape in general.

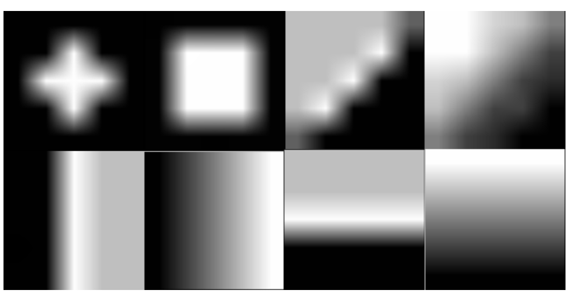

Figure 3: *Sample of the initial filters used in the convolution layer of the network. 22 filters total including 16 from smooth-sharp edges each rotated eight times.*

The feature maps associated with each filter are pooled into 10×10 images (using the maximum of each 3×3 area from the feature map) and the pixels of the pooled images are linked directly into the classification network. In more complex image recognition problems a second convolution might be performed, and then even a third, but our network does not need to perform high-level analysis of shapes aside from simple feature identification.

The classification network, then, consists of two neuron layers (with seven neurons each) before the two-channel output prediction as either muon or electron. This is a standard neural network with the purpose of converting visual information from the feature maps into a ring classification. Certain features, such as a smooth edge, can be weighted more or less strongly depending on where they occur, and the presence of two layers offers more complex logic in the classification if needed.

3.2 Network Training

The complete CNN takes input in the form of a 30×30 pixel image and outputs a classification prediction. Network loss is a continuous function that measures the deviation of prediction output from the true classification. Training a network, then, consists of taking a random sample of input images and minimizing the loss function of the network in the space of variable parameters (e.g. filter or neuron weights).

With a large training set of images, however, it is virtually impossible to minimize loss on all the data at once so each step in the minimizing procedure uses a random batch of 200 images. Step size must be chosen carefully to not overstep global topology while also not getting stuck in local fluctuations. Simple trial and error of different step sizes is done until training successfully begins reducing the average losses. Recall that there are four separate CNNs, that operate on individual image sets. Each network is trained in parallel on independent batches of images.

This stochastic method works well in learning general features that are shared between many images, but easily *forgets* important features that appear rarely. For that reason, the network parameters are saved in an averaged quantity that contains components from all previous steps (older steps are weighted less). Testing is done on the averaged parameters rather than the parameters in the most recent step. As the algorithm converges, the step size is also reduced exponentially so that the parameters can settle in a true local minimum.

IV Results

The total accuracy of the CNN is shown in figure 5. But, for context, it is useful to compare the performance of the CNN to the standard analysis (fiTQun) in the case of single ring e μ separation. It is first necessary evaluate the performance of fiTQun.

4.1 Standard Analysis

FiTQun fits theoretical Cherenkov cones to the PMT time and charge information using different particle hypotheses. The Cherenkov cone with the largest maximum likelihood is selected to represent the event, providing a reconstructed particle type, vertex position, energy, etc.

The simplest fiTQun configuration, to compare with the CNN, uses two single ring fits: one electron fit and one muon fit. The accuracy of this regime is plotted in figure 4 in red. As expected, events in the second, third, and fourth image sets perform subsequently worse due to reduced information in the PMT output. There is, however, a significant dip in the performance for muons in the first image set. Investigation reveals that this region is populated by images with more than one ring, most likely from muons that have a segmented trajectory due to scattering.

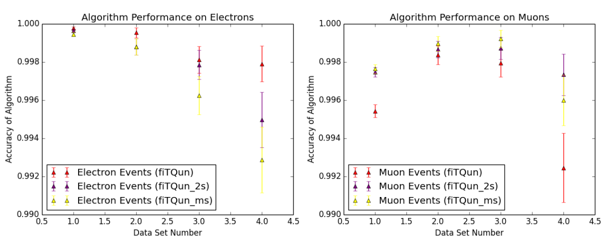

Figure 4: *Performance of three fiTQun regimes for electron and muon events and each image set. Red: default fiTQun with single ring electron and muon fits (99.77%), Purple: fiTQun with two-ring segmented muon fit (99.846%), Yellow: fiTQun with three-ring segmented muon fit (99.836%).*

To improve the identification performance on these cases, an augmented muon fit strategy is used that searches for secondary rings. The algorithm performances using two- and three-ring searches are plotted in figure 4 in purple and yellow respectively. It can be observed that the two-ring segmented regime represents the best total classification performance of the standard analysis.

4.2 Performance Comparison

Once the CNN is trained on two runs of 1500 batches, the results are plotted in figure 5 with the best fiTQun regime for comparison. The two algorithms are similar in performance except in the first and last muon bins. There are far more events in the first bin, however, so in total the CNN recovers roughly 30% of the standard analysis classification performance. In reality, however, the CNN structure is most comparable to a regime of fiTQun that does not use timing information and which is not tuned for segmented rings. In that case it recovers close to 70% of the performance.

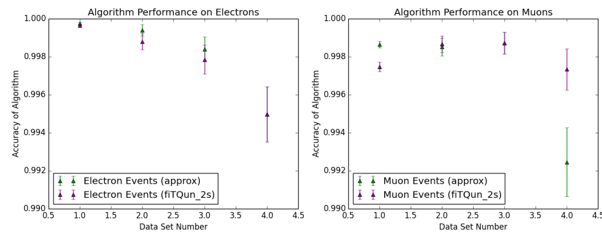

Figure 5: *Comparison of CNN classification (Green: 99.89%) to fiTQun with two-ring segmented muon fit (Purple: 99.846%).*

V Conclusion

Particle identification in water Cherenkov detectors is done by comparing the Cherenkov profiles of different rings. Since the PMTs act as pixels in an image, and ring shape is a qualitative attribute, it is a natural solution to implement image recognition machine learning methods to classify them. We tested this approach using simulated events in the Super-K detector.

The Super-K standard analysis fitting algorithm is spcifically tuned for the problem of electron muon separation, with complex modelling of Cherenkov light moving and scattering in the detector. The CNN, however, has the advantage that it requires no prior knowledge of Cherenkov physics, and is only tuned in a few generic ways. In fact, all the ring details, besides the initial filters, are learned automatically by the algorithm. The comparable performance of the CNN to the standard particle identification is promising, and these results may be of interest for future analysis.

References

[1] TensorFlow: *Large-scale machine learning on heterogeneous systems*, 2015. Software available from tensorflow.org

[2] *The Super-Kamiokande detector* Nucl. Instrum. Meth. A501 (2003) 418-462

[3] *Improving The T2K Oscillation Analysis With fiTQun*. Poster by Andrew Missert.

t2k.org/docs/poster/076

[4] Abe K. et al. (the T2K collaboration), *Measurements of Neutrino Oscillation in appearance and disappearance channels by the T2K experiment with* 6.6×1020 *protons on target*. Phys. Rev. D 91, 072010 (2015).

[5] Notes from the Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition:

cs231n.github.io/convolutional-networks/