Understanding Misclassification in a Five Particle CNN

Jonathan Karp
8/04/2016

Abstract
In this paper, I present the results of my research as an REU student working on the MicroBooNE experiment at Fermilab. I first give an introduction to the MicroBooNE experiment. I then introduce the topic of deep learning and describe the collaboration’s attempt to use convolutional neural networks (CNNs) to perform particle identification. Finally, I discuss my particular research concerning our attempt to understand the CNN’s particle misclassification, with a particular focus on separating muons from pions. I show that muon network classification score correlates with the length of a straight line feature in the collection plane image.

1. Introduction to MicroBooNE

MicroBooNE is a liquid argon time projection chamber (LArTPC) located on the booster neutrino beamline (BNB) at Fermilab. As a follow up to the MiniBooNE experiment, one of the main goals of MicroBooNE is to investigate the possible existence of so called “sterile neutrinos” [1], neutrinos that do not undergo weak interactions. The MiniBooNE experiment attempted to verify the LSND anomaly [2], and it found an excess of electron neutrinos, but only in the low energy range. However, MiniBooNE’s data alone is not enough to establish the existence of sterile neutrinos because the low energy excess may be due to greater background than expected. The MiniBooNE experiment utilizes a Cherenkov detector, which cannot distinguish between electrons and photons that are possibly produced from $\pi^0$ decays. It is therefore possible that the excess is due to a greater production of photons than expected instead of a greater production of electron neutrinos than expected. One of the primary goals of MicroBooNE is to identify the nature of the excess of low energy events. MicroBooNE will be able to distinguish electrons from photons because it uses a LArTPC, which is ideal for identifying which particles pass through the detector [1]. In fact, my project involves particle identification.

In addition to being a follow up to the MiniBooNE experiment, MicroBooNE will be used to measure the cross sections of neutrinos in liquid argon. It will also be instrumented to measure neutrinos coming from a supernova burst in case such an event occurs. Furthermore, MicroBooNE is the first in a series of large LArTPCs planned to be built on the booster neutrino beamline (BNB) at Fermilab, so it will act as R&D for these other detectors.

The MicroBooNE detector consists of two subdetector systems, a TPC and an array of 32 photomultiplier tubes (PMTs). Using a right handed Cartesian coordinate system, we align the $z$ axis along the neutrino beam. The detector then has $xyz$ dimensions of $2.5m \times 2.3m \times 10.4m$. A potential difference is applied across the $x$ direction, with three wire planes at the anode in the $yz$ plane. The furthest...
plane from the cathode is called the $Y$ plane, and it consists of 3456 wires oriented vertically. The other two planes are called the $U$ and $V$ planes, and they consist of 2400 wires that are oriented at $\pm 60^\circ$ from the $y$ axis. The wires in the $U$ and $V$ planes sense a current that is inducted by electrons drifting past, and the wires on the $Y$ plane collect the drifting electrons. Behind the planes is an array of PMTs that serve as light collectors. When a charged particle passes through the detector, it gives off light and leaves a trail of ionized electrons. The light is collected by the PMTs on the nanosecond scale, and it triggers the TPC to record data. The ionized electrons drift towards the wire planes because of the applied electric field, and they reach the wire planes on a millisecond scale. The electrons interact with wires on the planes, producing signals in the wires, and a series of electronic devices record the times and particular wires with which the charged particles interact. [3]

The raw data from the detector for a particular event essentially consists of three images, each of which is a two dimensional map of drift time vs. wire number. A mark on the image corresponds to ionization electrons reaching a particular wire at a particular time. The intensity of the pixel corresponds to the charge deposited on the wire at that time, which is proportional to the energy deposited lost by the charged particle per unit distance. Each image represents a two dimensional projection of a particle’s trajectory. Figure 2 shows an example image of a cosmic ray passing through the detector.

The MicroBooNE group hopes to use these images to reconstruct the information about the events, such as the identities of the particles producing the tracks in the images and the particles’ energies. [3] Since the raw data from a LArTPC is a collection of images and MicroboooNE is first in a series of LArTPCs that will be used for neutrino detection experiments, developing image processing techniques plays a central role in the MicroBooNE experiment. I was involved with a group that is attempting to employ CNNs for automatic reconstruction.

2. Introduction to CNNs:

In general, a neural network is tasked with taking an image as input and classifying it as one of $C$ classes. The network has a score function that acts on the image and outputs a $C$ dimensional score vector, which can be converted into a measure of confidence level for each class. The network trains on a group of images for which the actual class is known. While training, the score function acts on the images, and a loss function is used to measure how much the score vector differs from the truth. Next, the gradient of the loss function is taken with respect to the parameters of the score function, and the parameters of the score function are updated accordingly. Eventually, the network will have “learned”
the features of the images that correspond to particular classes, and it will be able to accurately classify images. [4]

In a traditional neural network, the image is first decomposed into a single vector. The score function is a series of layers, each of which contains a two dimensional matrix and a nonlinear activation function. The matrix acts on the image vector and outputs another vector, which is referred to as the next layer of neurons. These layers of the network are referred to as “fully connected” layers because each neuron is connected to every neuron in the previous layer through a particular entry in the matrix. The activation function acts as a nonlinearity that allows the layers to be independent of each other. [4] A major disadvantage of normal neural networks is that fully connected layers have an immense number of parameters, requiring an enormous amount of memory. Additionally, the network ends up looking at the image as a whole instead of looking for local features. [4]

CNNs address these issues because their architecture allows for the detection of local features while cutting down on required memory. [3] In a CNN, the input image is not decomposed into a single vector. The CNN is composed of multiple convolutional layers that do not merely connect each output neuron to every input neuron. Instead, a convolutional layer is composed of multiple filters that are smaller than the input image. For each filter, an inner product is taken between the filter and each region of the input, producing a feature map of the input. The feature maps from each filter are stacked on top of one another, and these filters are passed to the next layer. The network terminates with fully connected layers that turn the filter stacks into a score vector. [4] The use of these filters is advantageous because the convolutional layers require much fewer parameters than fully connected layers, allowing CNNs to be efficiently composed of several convolutional layers. Additionally, the use of convolutional layers enables each filter to recognize a particular local feature in images, and the filters all operate independently of each other, so the network learns to recognize multiple features in the images. [3] CNNs are particularly useful for our data because we do not have to tell the networks which features to look out for. However, a disadvantage is that we do not know which features the network ends up picking up on.

3. Single particle Network

In investigating the ability of CNNs to classify particle event images, the MicroBooNE deep learning group tested CNNs on single particles simulated in the TPC. For each event, one of five particles, $e^-, \gamma, \mu^-, \pi^-, p^+$, is simulated in the TPC and an image representing the data that would be generated by the detector is produced. Figures 4, 5, 6 and 9 show some examples of cropped collection plane images. After training, the networks classify most of these particles correctly, but they are not 100% accurate. [3]

My project involves understanding why the network misclassifies particles, particularly why muons and pions are misclassified as each other. The data sample I use is a single particle Monte Carlo sample. The initial energies of the particles are distributed uniformly from 100 MeV to 1 GeV, except the protons which have initial kinetic energy ranging from 100 MeV to 788 MeV. The initial position of the particle is distributed uniformly throughout the TPC, and the initial direction of momentum is distributed isotropically. The images used are just taken from the collection plane, and they are cropped to a $576 \times 576$ pixel region containing at least part of the particle track. The network used is AlexNet, a CNN with 8 total layers.
4. Muon Misclassification as Pion

When muons are misclassified, they are most often misclassified as pions [3]. Figure 3 shows the classification of muons as a function of initial kinetic energy. High energy muons have very high accuracy, but low energy muons are much more likely misclassified as pions. Seemingly, the network thinks that low energy muons look like pions.

In order to understand why the network thinks that low energy muons look like pions, we must first see what the network thinks a muon looks like and what it thinks a pion looks like. We do this by looking at images of muons with high muon scores and pions with high pion scores. To prevent bias in choosing images with particular features, we choose the images at random. For example, figure 4 shows randomly selected muons with muon scores greater than 0.9, where the score has a maximum value of 1. These muons seem to be characterized by their long, straight tracks.

Conversely, figure 5 shows randomly selected muons with muon scores less than 0.1. These have much shorter and less straight tracks than the high energy muons.

Figure 3: Muon classification vs. initial kinetic energy. The fraction classified as muons is in red and the fraction classified as pions is in cyan. At high energies, accuracy is about 90%. Below about 320 MeV, accuracy drops drastically, and the low energy muons have a significant probability of being misclassified as pions.

Figure 4: muons with high muon scores. Both have long, straight tracks with relatively low intensity. The track on the left has a gap in the track, but that does not seem to have a major effect on the score.
Figure 5: Muons with low muon scores. The one on the left has a shorter track than the high energy muons and has a kink in it, and the one on the left is short and wiggly.

Figure 6 shows pions with high pion scores. As opposed to the high score muons, these pions have short tracks with kinks in them.

Figure 6: Pions with pion scores > 0.9. Both have very short tracks with kinks.
5. Muons with Michel Electrons

To explain why low energy muons are misclassified as pions, we study the features of muons in the sample that give off Michel electrons in the TPC. A muon tends to decay and gives off a low energy electron called a Michel electron. However, most muons escape the detector region before decaying. In our sample, less than 10% of the muons give off Michel electrons. There are two important pieces of information about these muons that give off Michels. The first is that while the energy distribution of all muons is relatively uniform, the muons that give off Michels tend to be at lower energies (figure 7). The second is that the pion scores of these muons that emit Michels tend to be higher than for muons in general, while the overall accuracy of muons that give off Michels is lower (figure 8). These two facts combined must give at least a partial explanation for why low energy muons are misclassified as pions. Low energy muons are more likely to give off Michel electrons, and giving off a Michel electron makes a muon more likely to be misclassified as a pion, so low energy muons are more likely to be misclassified as pions. To see why this is so, we look at a couple of randomly selected muons that emit Michels (figure 9). The Michel electrons have low energies, so they do not form electromagnetic showers. While they have somewhat of a wiggly track, the tracks may still be confused for straight tracks, and the point where a muon gives off a Michel electron looks like a kink. The overall effect is that the muons giving off Michel electrons look qualitatively similar to the pions in figure 5.

Figure 7: Histogram of energies of all muons (green) overlaid with a histogram of the energies of just the muons that give of Michel electrons (blue). The energy distribution of all muons is relatively uniform, while that of muons that give off Michels is biased towards lower energies.

Figure 8: histograms of muon and pion scores for muons that do not emit Michels in the TPC (left) and Muons that emit Michels in the TPC (right). The pion scores are higher and overall accuracy lower when the muons give off Michels.
Figure 9: Muons that give off Michel electrons in the TPC. They have short tracks and kink-like features that make them look like pions.

Even though the reduction in classification accuracy due to muons that emit Michels is a partial explanation for the dip in accuracy for low energies, it is definitely not the whole story. Figure 10 shows the plot of accuracy vs. energy for muon that do not give off Michel electrons. The accuracy still drops dramatically for low energies, although slightly less than with the muons that emit Michels. This graph shows that muons giving off Michels at low energy do not completely explain why low energy muons are misclassified as pions.

6. Houghlines

Even though muon-Michels are not the whole story on why low energy muons are misclassified as pions, they could provide insight into why other muons might be misclassified as pions. Just like muons that give off Michel electrons tend to have short tracks with kink-like features, we might expect other misclassified muons to have similar features. We need a parameter that quantifies how much an image is a long straight track without any kinks. In order to do this, I make use of OpenCV [5] feature detection algorithms. One algorithm that I find particularly helpful is the HoughlinesP algorithm [6], which takes and image and returns the endpoints of any straight lines that it finds. No matter how straight the tracks are, there are still some small deflections, which cause the algorithm to find many straight lines along a single track. Additionally, having a gap in the track does not seem to negatively affect the muon score.
(Figure 4 left), so in order to correctly represent the features seen by the network, the algorithm must return straight lines even if they are broken. The result is that several straight lines are found for each image. In order to find a single parameter that classifies the straightness of a track, I take the longest line in a particular image and write a custom algorithm to stitch onto the line any nearby lines with similar slopes to it (see appendix for details). I make sure to validate the custom algorithm using a unit test. The final result is a single line that classifies the tracks. Figure 11 shows the original image points, Houghlines, and stitched line overlaid on the original image for an image with a high muon score and an image with a low muon score.

Muon score: 0.940

Muon score: 0.154

Figure 11: results of HoughlineP and stitching algorithm for a muon with a high muon score (left) and one with a low muons score(right). The top shows the points of the image that have a pixel value over 50 ADC (the threshold used by the network). The middle shows the Houghlines overlaid on the original image. The Houghlines are in black with their endpoints as black circles. The bottom shows the result of the stitching algorithm overlaid on the original image. For the image on the left, the image is a long straight line, so the stitched line is a long straight line that covers most of the image. For the image on the right, the image has a shorter track followed by a kink, and the stitched line only covers the pre-kink track and not post-kink tracks.

7. Length-Score Correlation

The length of the stitched Houghline has a strong correlation with muon score, as shown in the two plots in figure 12. The first is a 2D histogram of stitched line length and muon score. There are plenty of events with low line length and low muon score, but hardly any with length over 300 pixels and low muon score. This makes sense because the cropped image is 576 × 576 pixels, so a straight line of 300-400 pixels would cover most of the image. Each horizontal bar on the histogram can be thought of as a probability density function of muon score for a single sliver of length. The plot shows that the peak of this function increases as length increases. Similarly, the plot on the right shows the probability density function of muon score for particular ranges of length. As the length increases, the peak gets higher and moves further to the right. For small lengths there is significant probability of having a low muon score, but that probability is negligible for high muon scores.
Figure 13 shows the correlation of line length and initial energy, and it shows that line length tends to increase as energy increases. This combined with the fact that muon score correlates with line length explains why low energy muons are misclassified as pions. It seems that the fact that the tracks are shorter for low energy muons makes the tracks look more similar to the pions in figure 5 that have short tracks.

It is still noteworthy that muons with short line length still on average have higher muon scores than pion scores. It seems that for low track lengths there is some other factor dictating whether or not they have a high muon score.

8. Conclusion

We see that muons are much more likely to be misclassified as pions by CNNs when they have lower initial energies, and I show two reasons why this is true. The first is that low energy muons are more likely to give off Michel electrons, leading to lower muon scores. The second is that low energy muons tend to have short track lengths, which correlates with low muon score. It seems that short track length alone is not enough of an explanation for why muons are misclassified as pions, and some other factor must be added on, which should be the basis for future work. Additionally, similar analyses should be done for other particles to explain why they are sometimes misclassified. These studies will help us understand the performance of CNNs for particle identification, which is important given the crucial role of particle identification in MicroBooNE.
A. Appendix

The HoughlinesP algorithm returns the endpoints of many straight line segments for a particular image. The endpoints are always ordered such the one with the smaller x coordinate is first. Any line shorter than 5 pixels is discarded. The remaining lines are made instantiations of a class called “Line”, which stores the endpoints of the line and has functions to return the length and slope of the line.

In order to stitch the lines together, my algorithm first finds the line with the longest length. It then creates a variable called “stitched_line” that is an instantiation of the “Line” class, and initializes it with the endpoints of the longest line and remove the longest line for the list of lines. It then loops over all of the remaining lines. Each line is first checked whether its slope is within 0.3 of the slope of “stitched_line.” If it is, then the first endpoint of the line is projected onto a line that passes through “stitched_line.” If the projection of the endpoint is within 20 pixels of the endpoint, the line is stitched on to “stitched line” if one of the four following conditions are met (figure 14):

1. If the line is entirely to the right of “stitched_line”, then if the right endpoint of “stitched_line” is within 60 pixels of the left endpoint of the line, the right endpoint of the line is set as the right endpoint of “stitched_line.”
2. If the line is entirely to the left of “stitched_line”, then if the left endpoint of “stitched_line” is within 60 pixels of the right endpoint of the line, the left endpoint of the line is set as the left endpoint of “stitched_line.”
3. If the line overlaps with “stitched_line” on the right, in that the right endpoint of the line is to the right of the right endpoint of “stitched_line” but the left endpoint of the line is to the left of the right endpoint of “stitched_line”, then the right endpoint of the line is set as the right endpoint of “stitched_line.”
4. If the line overlaps with “stitched_line” on the left, in that the left endpoint of the line is to the left of the left endpoint of “stitched_line” but the right endpoint of the line is to the right of the left endpoint of “stitched_line”, then the left endpoint of the left is set as the right endpoint of “stitched_line.”

In each case, the line is deleted from the list of lines after being stitched on to “stitched_line.” The algorithm continues looping over the list of lines until no lines are stitched on to “stitched_line” in a single iteration. Finally, the algorithm returns the endpoints of “stitched_line.”

B. Acknowledgments

I would like to thank Dr. Mike Shaevitz for all of his guidance and for providing me with the amazing opportunity of being part of his research group. I would also like to thank Kazu Terao, Victor Genty, David Caratelli, and David Kaleko for taking the time out of their busy schedules to teach and guide me.
this summer. I would also like to thank the MicroBooNE collaboration at Fermilab for letting me be a part of the team for this summer. Additionally, I would like to thank John Parsons and Amy Garwood for making the Nevis REU program happen. Finally, I would like to thank the National Science Foundation for funding the REU program, affording me the opportunity of a lifetime.

Works Cited