5-2019

# Pion Identification through Machine Learning for the MOLLER Experiment at the Thomas Jefferson National Accelerator Facility

Anne-Katherine Burns

# Pion Identification through Machine Learning for the MOLLER Experiment at the Thomas Jefferson National Accelerator Facility

A thesis submitted in partial fulfillment of the requirement
for the degree of Bachelor of Science with Honors in
Physics from the College of William and Mary in Virginia,

by

Anne-Katherine Burns

Accepted for Honors

(Honors or no-Honors)

David Armstrong

Advisor: Prof. David Armstrong, Physics

Seth Aubin for Prof. Phiala Shanahan

Advisor: Prof. Phiala Shanahan, Physics

Seth Aubin

Prof. Seth Aubin, Physics

Prof. Dan Runfola, Applied Science

Williamsburg, Virginia
May 3, 2019

# Contents

# Acknowledgments

There are quite a few people I would like to thank for helping me complete this research project. Thank you to Professor David Armstrong for helping me to devise the project, advising me throughout it, and teaching me so much about particle physics. Thank you to Professor Phiala Shanahan for giving me my first introduction to Machine Learning last year, for being willing to continue advising me remotely in this project, and for inspiring me to pursue my dreams as a woman in physics. Thank you to Professor Wouter Deconick for always being available and willing to help me through tricky coding issues and navigating the MOLLER simulation. Finally, thank you to William and Mary's Physics Department including all of its students, professors, and administrators for fostering a nurturing and encouraging environment from which I can launch the rest of my career in physics.

# List of Figures

# List of Tables

# Abstract

This report describes the implementation of a deep neural network for particle identification on the MOLLER experiment. The MOLLER experiment, currently in its early stages of design at the Thomas Jefferson National Accelerator Facility (JLab), will attempt to measure the parity-violating asymmetry present in the elastic electron-electron scattering, to a precision of 0.7 ppb. While the Standard Model precisely predicts this asymmetry, if the value measured by the MOLLER experiment were to differ significantly from the predicted value, then the experiment could provide laboratory-based evidence of physics beyhond the Standard Model (BSM) and point researchers in the right direction for its exploration. The high energy electron beam used in this experiment is predicted to generate scattered electrons as well as a background of roughly 0.13 percent pions. While the ratio of pions to electrons will be small, their presence may significantly affect the asymmetry measurement. The detected particles, predominantly pions and electrons, must thus be classified. Here, an algorithm is proposed to classify particles detected in the MOLLER experiment using deep neural networks (DNNs). Once the classification algorithm is successfully written and proven to work, the uncertainty in the classification of the particles as pions, electrons, or positrons will be determined. If successful, this classification algorithm may be used to optimize the design of the experiment hardware.

# Chapter 1

# Introduction and Theory

## 1.1 Research Objectives

The purpose of this research project is to write and implement a neural network that will perform particle classification for the MOLLER Experiment. The specific objectives of the project are the following. First, the project will determine the observables from the MOLLER Experiments GEANT-4 simulation from which a neural network could ex- tract relevant information. Second, the project will design, code, and optimize the neural network. Third, the project will determine which combination of available observables allow the network to perform most effectively and to display these results using a series of metrics. Finally, the project will use the results of the classification to aid in the design of the particle detectors for the MOLLER Experiment by giving insight as to which observables are most relevant for particle classification.

## 1.2 The Goal of the MOLLER Experiment

The purpose of the MOLLER (Measurement Of a Lepton Lepton Electroweak Reaction) experiment is to measure the parity-violating asymmetry that arises when polarized electrons are scattered off of unpolarized electrons. This measurement will serve to either further validate or invalidate the electroweak theory and will contribute to the search for physics beyond the Standard Model (BSM). The two most important motivations for conducting the experiment are the following. First, the measurement of parity violation arising due to the weak mixing angle will allow experimenters to measure the parity-violating asymmetry in polarized electron-electron scattering more precisely than ever before. Importantly, the average experimental value of this asymmetry, A$_{PV}$, is a function of the weak mixing angle, $\theta_w$, and aligns with the theoretical prediction of the Standard Model when the 126 GeV scalar resonance is assumed to be the Higgs Boson. However, the two most precise measurements that contribute to the world average value for $\theta_w$ have a remarkable statistical difference of $3\sigma$, potentially signalling the existence of physics BSM. Physicists involved with the MOLLER experiment plan to measure the asymmetry with a sensitivity of $\delta(sin^2\theta_w)$ = ± 0.00028 in hopes of further constraining the theoretical possibilities that have arisen from these measurements [4].

The second piece of the physics motivation behind the MOLLER experiment is to parameterize BSM neutral current interactions by constraining the coupling for four- fermion interactions, $\Lambda$/g. Here, $\Lambda$ is a mass scale factor and g describes the strength of the new physics interactions for these occurrences. The sensitivity of the measurement of A$_PV$ corresponds to the sensitivity of $\Lambda$/g and is currently the most sensitive proposed measurement of the CP-conserving neutral current interactions [4].

The planned experimental setup for the MOLLER experiment is the following. A beam of longitudinally polarized electrons at 11 GeV will strike a liquid hydrogen target, producing electrons that have been scattered off other electrons, referred to as Moller electrons. These electrons, along with other particles produced, namely pions, will be observed by a detector system shown in Figure 1.1 [4].

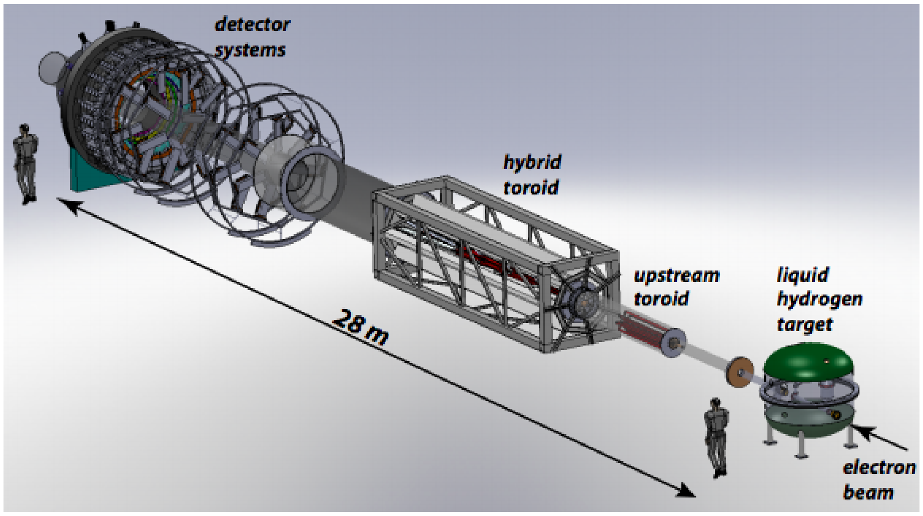

Figure 1.1: Layout of the Detector Setup for the Moller Experiment, taken from [4]. This image shows the longitudinally polarized electron beam which will hit a liquid hydrogen target. The resulting Moller electrons and other particles will then be measured by the detector system.

Figure 1.1 shows the experimental setup of the Moller experiment [4]. Figure 1.2 shows a plot of weak mixing angle measurements vs. energy scale, along with the prediction of the Standard Model. The value given for the Moller experiment is the expected SM value, with the anticipated experimental precision indicated [4].

## 1.3 The Standard Model

The Standard Model is an elegant and highly accurate model of the known fundamental particles in the universe and the ways in which they interact with one another. The Standard Model takes elementary particles to be point-like objects that are not made up of more fundamental constituents. These elementary particles are split into two categories: half-integer spin particles, or fermions, and integer spin particles, or bosons. Fermions are further classified into leptons and quarks. Leptons interact

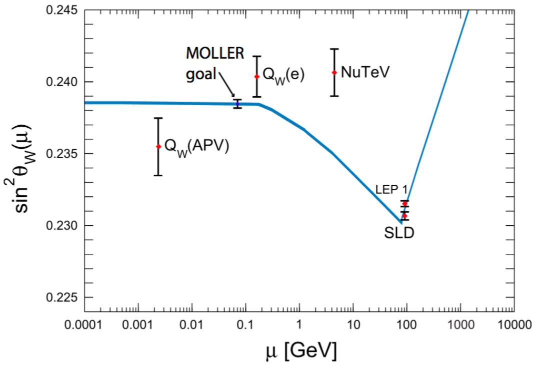

Figure 1.2: The weak mixing angle vs. energy scale for various HEP experiments, taken from [4]. The measurement the MOLLER Experiment hopes to make is shown with the proposed error bar and the proposed energy scale. Its central value is taken from the Standard Model prediction. The blue line on this plot is the Standard Model prediction.

through the weak force, gravity, and the electromagnetic force and include electrons, muons, taus and their corresponding neutrinos. Quarks interact with each of these forces as well, and also with the strong force, and thus, have a color charge. In nature, quarks are only found in composite particles with quantum numbers of either three quarks (baryons) or quark-antiquark pairs (mesons). Other combinations, in principle, are allowed by the Standard Model; however, evidence for their existence is not yet definite [7].

The second type of particle is the boson. The strong force and the electroweak force are both mediated by the exchange of bosons. The strong force is mediated by eight types of massless bosons called gluons, the weak force is mediated by the $W^{\pm}$ and Z bosons, and the electromagnetic force by the massless photon. The gluons and $W^{\pm}$ and Z bosons are all self-interacting [7].

### 1.3.1 Symmetry in the Standard Model

The SM has two types of symmetries, both of which play important roles in quantum field theory. The first type covers continuous symmetries in which the values of the parameters are continuous. This type includes space-time symmetries such as rotation and translation and internal symmetries, the symmetry of some physical quantity. The SM is described by the gauge symmetry, a continuous symmetry, SU$(3)_C$ x SU$(2)_L$ x U$(1)_Y$ where SU$(3)_C$ corresponds to strong force interactions and SU$(2)_L$ x U$(1)_Y$ corresponds to electroweak force interactions. The $W^{\pm}$ and Z bosons are not massless and therefore SU$(2)_L$ x U$(1)_Y$ is not a symmetry of the vacuum, the Standard Model must be allowed a mechanism through which spontaneous symmetry breaking occurs [7]. This mechanism is the Higgs Mechanism and is governed by the Higgs Boson which gives mass to those particles that couple with the Higgs. The Higgs was confirmed experimentally by researchers at the LHC in 2012 and was reported to have a mass of 126 ± 0.4 ± 0.4 GeV, which is within SM predictions [1].

The second type of symmetry, and the type that the MOLLER experiment is interested in probing, encompasses discrete symmetries. As the name suggests, the parameters in discrete symmetries have discrete values. The CPT theorem says that interactions must be invariant under transformations that include charge conjugation (C), parity (P), and time reversal (T). Among other transformations, the weak force is not invariant under parity transformation, and thus, violates this symmetry. These three symmetries act on the wave function of a particle in the following ways. Charge conjugation converts particles to anti-particles and vice-versa. Parity takes the spatial components of the wave function and makes them negative in this way: $P\Psi(x, y, z) \rightarrow \Psi(-x, -y, -z)$. Time reversal makes the time component of the wave function negative in this way: $T(x, y,z,t) \rightarrow \Psi(x,y,z,-t)$ [7].

### 1.3.2 Parity Violation

In practice, parity violation in the weak force can be measured by changing the spin of the particles that interact with one another via the weak force and then comparing the interactions of the particles for which the dot product of their spin and momentum is positive and for which this dot product is negative. Any difference in scattering probability indicates the presence of parity violation, described by the following equations:

$$N_R = \sigma \cdot p = + \tag{1}$$

$$N_L = \sigma \cdot p = - \tag{2}$$

$$A_{PV} = (N_R - N_L)/(N_R + N_L) \tag{3}$$

where $N_R$ is the number of right handed particles, $N_L$ is the number of left handed particles, $\sigma$ is the cross section of the interaction, $p$ is the momentum, and $A_{PV}$ is the asymmetry. In the MOLLER experiment this is done by rapidly changing the polarity of the electron beam. It is known from experiment that parity is not conserved and therefore nature prefers particles whose spin and momentum vectors are anti-aligned, also known as left-handed particles, and anti-particles whose spin and momentum vectors are aligned, known as right-handed particles.

## 1.4 Physics Beyond the Standard Model

Because of its high sensitivity, the MOLLER experiment will have the ability to put constraints on future BSM phenomena as might be observed by higher-energy experiments like the LHC. This new physics, if observed, will likely involve parity violating coupling for electrons with a force carrier whose mass is either much greater than or much less than that of the Z boson in the Standard Model and will conserve flavor and CP symmetry. Such bosons arise in many theories including some grand unified theories, dark-Z models, and others. This BSM physics will also have similar sensitivities to such new physics interactions in current and anticipated LHC data.

## 1.5 Background Processes Present in the MOLLER Experiment

A roughly 0.1 percent signal contamination from pions and other hadrons is expected in the measurement of the Moller peak. For the parity-violating asymmetry expected to be present in the measured cross-section, the hadronic signal will correspond to a necessary correction of about three percent to the measurement of $A_{PV}$ for electron-electron scattering. For this reason, it is very important to differentiate between hadrons and electrons in the signal [10].

## 1.6 Binary Classification through the use of Neural Networks

In the context of machine learning, binary classification is the process of classifying a data set of observables into one of two categories (here, given as Type A and Type B). The output of a binary classification network is, for each data point, the probability that the data point is Type A and the probability that it is Type B. In most approaches, these two numbers will add to one. For example, a network output for a given sample could be (0.6, 0.4) corresponding to a 60% probability that the sample is Type A and 40% probability the sample is Type B. The input data set for these networks gives the observables used in a series of columns where the rows are then the individual data points. Once training data is input, the network processes the training data. It does this using a series of matrix multiplications with non-linear activations in which free parameters are optimized to minimize a loss function, calculated by contrasting the network predictions to a separate validation data set. Each observable in which the network successfully identifies a pattern will increase the accuracy of the networks ability to predict the type of each data point in

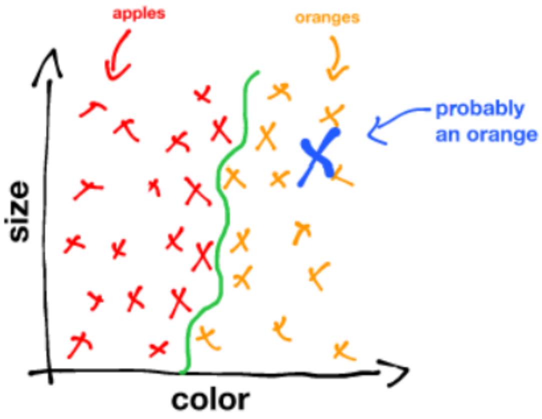

Figure 1.3: Binary classification using two observables, taken from [8]. This cartoon illustrates that when given $n$ observables, in this case two, the network will draw an $n-1$ dimensional shape around the data, in this case a one dimensional line, in order to classify the two types of data.

unlabeled data. The more distinguishable an observable is between one data type to the other, the more useful it is to the networks prediction. The cartoon in Figure 1.3 shows a binary classification of apples and oranges using two observables [8]. Here, the network draws a one dimensional line through the plot. As is made clear by this example, in this type of network architecture, a network using $n$ observables to classify data will draw an $n-1$ dimensional shape around the training data in order to determine not just patterns, but correlations and patterns simultaneously.

### 1.6.1 Metrics Used

There are multiple ways to represent the output of a binary classifier. Here, I will discuss output histograms and ROC curves. Output histograms divide the output from the neural network into percentage bins and plot the Type A and Type B predictions next to each other. In the following example plot from my previous research with the ATLAS experiment, Figure 1.4, there are ten bins representing groups of ten percent. The far left bin represents the number of data points with a zero to ten percent likelihood of being Type A objects, in this case, signal jets.

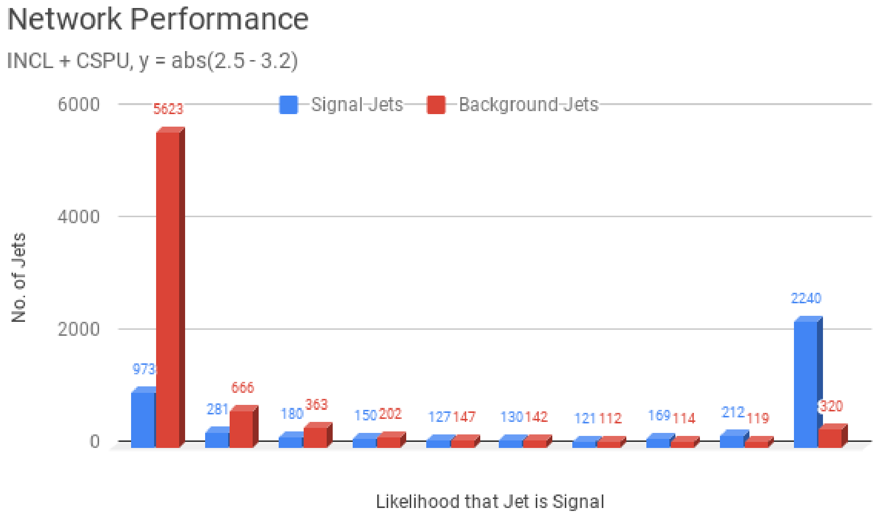

Figure 1.4: Example Output Histogram, taken from previous research (personal files). The classification of signal and background jets is shown. The plot is split up into bins of ten percent. The far left bin indicates a zero to ten percent likelihood that the jet is a signal jet. As is evident from the fact that the majority of the background jets are in this bin, the algorithm has a high success rate for background jet classification. Likewise for the signal jets, the majority of signal jets are in the far right ninety to one hundred percent likelihood bin, again indicating network success.

The blue column is the population of signal jets with this likelihood and the red column is the population of background jets (Type B). As is evident, this is an example of a successful training algorithm as the majority of the jets are in the zero to ten percent bin or the ninety to one hundred percent bin and are, for the most part, correctly classified within these bins. Receiving Operator Characteristic (ROC) curves are another tool for measuring the performance of binary classifiers. ROC curves plot the true positive rate, or in the case of this example, the efficiency of predicting Type A data points correctly vs. the false positive rate or the likelihood that the Type B data points are predicted incorrectly for a range of threshold values on a discrimination value. It is expected that the ROC curve of a network that is performing well will be exponential in shape. The smaller the area under the curve, the more effective the network is at predicting data point type [6]. Figure 1.5 shows two ROC curves from my aforementioned previous research. This ROC curve on the left is evidence of effective network performance. The ROC curve on the right shows less effective network performance. Again, here, Type A objects are signal jets and Type B objects are background jets.

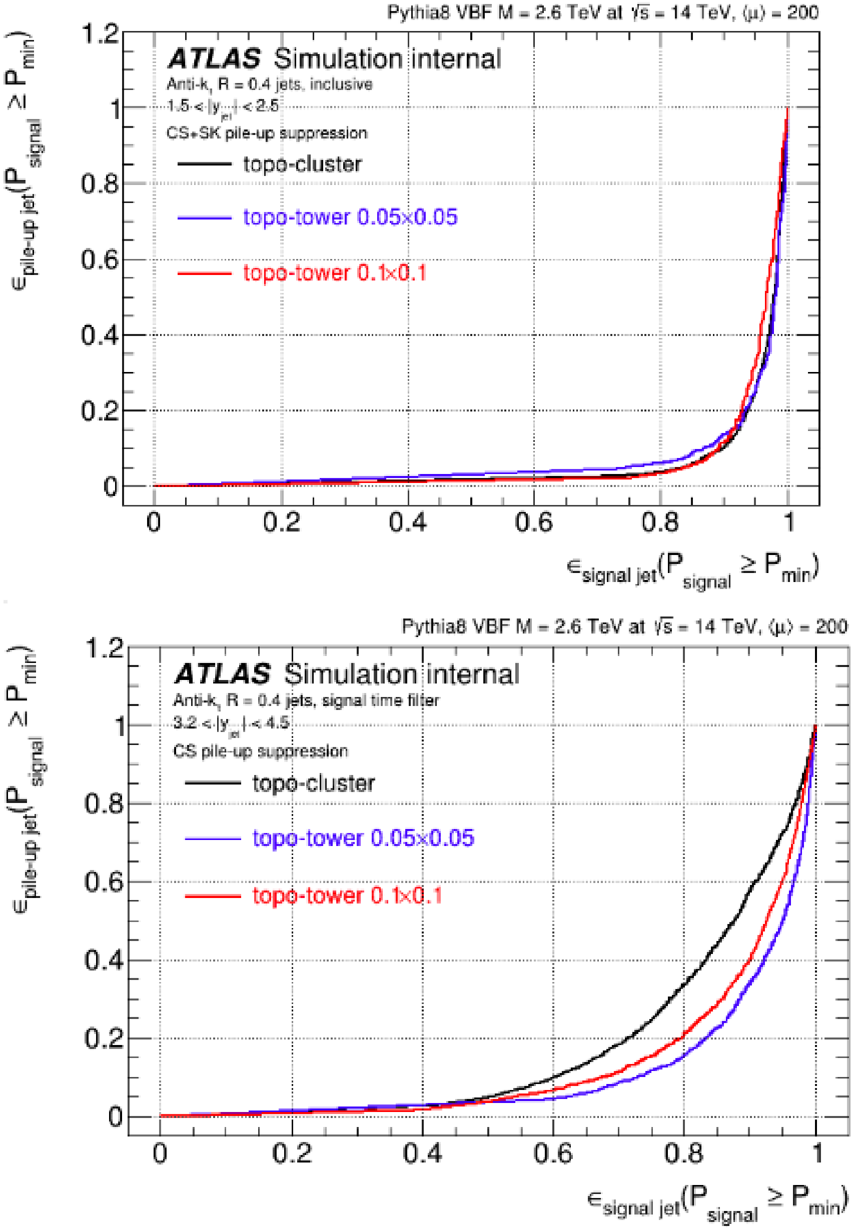

Figure 1.5: Example ROC Curves, taken from previous research (personal files). The area under the curve for the data in the top plot is smaller than that for the data in the bottom plot. This indicates better network performance for the data sets in the plot on the top. I have two hypotheses for this difference. The first is the data sets used to produce the right ROC curves have fewer data points, thus, the network has less information to learn from. Second, the data sets used to produce the left ROC curves are in a region of the ATLAS detector that gives better distinguishably between signal and background jets, thereby leading to better network performance for this region.

# Chapter 2

# Experimental Technique

## 2.1 Machine Learning Approach

### 2.1.1 Tensor Flow

TensorFlow is an open source machine learning data analysis system created by Google. It allows its users to be flexible in their Machine Learning (ML) approach and, while most developed for the use of running data through Deep Neural Networks (DNNs), its use is not limited to this application. Neural networks, like the ones implemented by TensorFlow, use multidimensional arrays or tensors to find patterns in data for both classification problems, like the one described here, and regression problems [2]. In this research project I have used Tflearn, a deep learning library built on top of TensorFlow. The Tflearn dnn function allowed me to train, validate, and evaluate the performance of the network. Appendix A shows the network code used. Figure 2.1, adapted from [5] shows the current neural network setup. Figure 2.2 shows a schematic for the way in which data is processed by Tflearn from [2].

### 2.1.2 Parameters

**Activation**

Using a series of neural networks a set of parameters was implemented that allowed the network to produce the most accurate classification results. In the last fully con-

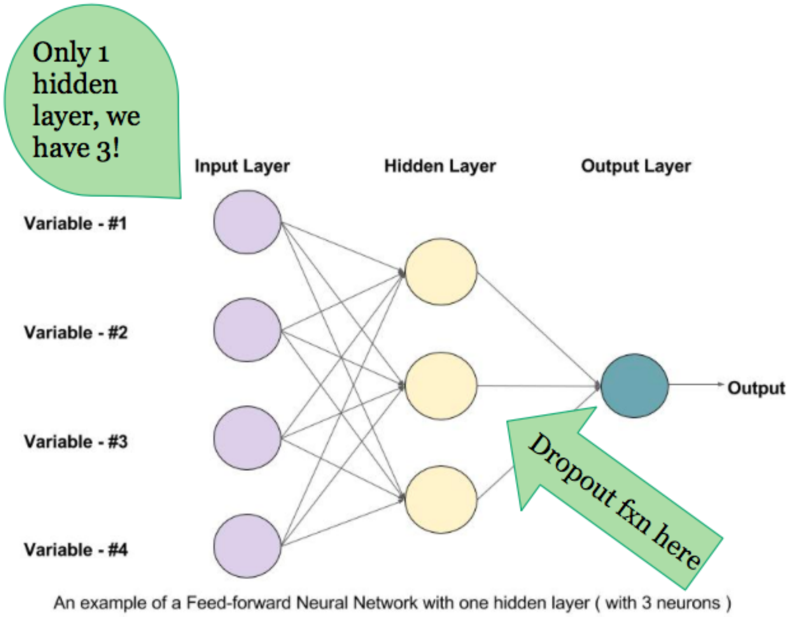
Only 1 hidden layer, we have 3!
Input Layer
Hidden Layer
Output Layer
Variable - #1
Variable - #2
Variable - #3
Variable - #4
Output
Dropout fxn here
An example of a Feed-forward Neural Network with one hidden layer ( with 3 neurons )

Figure 2.1: Initial network configuration, adapted from [5]. This figure shows a network with a single hidden layer and no dropout function. The network coded for this project had four hidden layers and one dropout function, indicated by the arrow, but is otherwise identical.

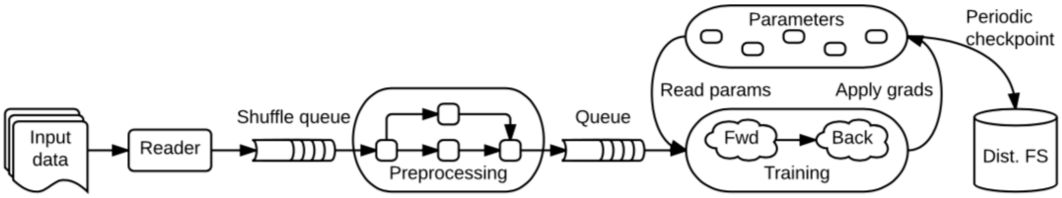

Figure 2.2: Data Flow Schematic, taken from [2]. This schematic show the way in which data is processed by TfLearn. After it is read in and shuffled, it is preprocessed. Then, using the structure chosen, the network is trained and validated.

nected layer a softmax activation was used. The softmax activation assigns decimal probabilities to the output of a multiclass network, i.e. a pion electron classification network, that must add to one. Unlike a standard normalization, the softmax activation will minimize cross-entropy, between predicted and true values and give outputs between [0, 1]. Here, cross-entropy refers to a loss function that describes the loss between the two probability distributions. This is the most useful output for pion identification. The cross-entropy function, f is below:

$$\text{f} = -(y \cdot log(p)) + (1-y) \cdot log(1-p) \tag{4}$$

Here, $y$ is the zero or one indicator for a particle being a pion or electron and $p$ is the likelihood that the data point in question is of class $y$ . Log refers to log base 10 [12].

### Optimizer

The learning rate is a parameter that indicates the step size taken in numerical multivariable optimization methods. A learning rate of 0.5 was used in the adam optimizer to maximize the efficiency of the learning in the initial configuration. Within the DNN function provided by TfLearn, beta1 of 0.9 and beta2 of 0.999 are parameters which control the rates at which the uncentered variance and uncentered variance squared are optimized, and relate to the distribution of probabilities. An epsilon value of 0.1 was used to prevent division by zero in the implementation of the optimizer. By setting the value to 0.1, ten percent of the time the network will explore random classification paths it has not yet considered before eventually settling on its decision. When a significantly smaller learning rate was used within the adam optimizer, not only did the loss plateau as iterations were run but the program was unable to distinguish between the particles as clearly when training data was analyzed. Similarly, the program was again unable to distinguish between the particles when the learning rate was very high, above 0.9, however, rather than plateauing the loss tended to decrease and then fluctuate irregularly.

### 2.1.3 Dropout Function

One dropout layer is included in the network to prevent over fitting. This function tells the training network to dropout a randomized set of connections making the network less sensitive to the specific weights of the individual data and allowing it to most effectively generalize validation data. [1]

[1] Section 2.1.2 is adapted from an unpulished report by Mary Robinson and myself written for our research with Professor Phiala Shannahan. Taken from personal files.

# Chapter 3

# Results

## 3.1 Summary of Results

In the first part of this project, the goal of the research was to develop the tools necessary to classify pions and electrons for the MOLLER Experiment. After the MOLLER simulation file was completed, testing was able to start. Once the data was extracted from the simulation file, distinguishability was calculated and the data was run through the neural network using a series of different network configurations. Network variation included both variation in the architecture and in the optimization of the results. In general, accuracy for electron identification was fairly good, however, the networks did poorly at identifying pions. Further results and discussion on this discrepancy can be found below.

### 3.1.1 The Data Set

The data set used in this analysis was developed using GEANT-4, a platformed developed by CERN which allows experimenters to simulate high energy particle and nuclear physics experiments using Monte Carlo methods. While the simulation file contains a huge amount of information including momentum, timing information, etc. for each detector hit, only a small amount of this information will be available to experimenters once the experiment is running. This available data includes position data and the number of photo-electrons in the pion and Shower-Max detectors. Because z data is trivial, representing the position of the detector only, all of the position data can be geometrically related to x and y. Furthermore, the simulation data includes many zeroes. In the position data, zeroes indicate that no particle track intercepted the detector, thus referred to as a null data point. Therefore, data points containing a zero in x or y were dropped. In the photo-electron data the data points with zeroes in both the pion detector data and the Shower-Max detector data were considered to be null data points and were dropped under these conditions. Table 3.1 shows the lengths of the data sets with different amounts of data removed.

| Data Rejected | Length of Data Set | |
|---|---|---|
| | Pions | Electrons |
| None | 100,000 | 100,000 |
| x = 0 OR y = 0 | 29,717 | 78,439 |
| x = 0 OR y = 0 OR (p-e$_{SM}$ = 0 AND p-e$_{\pi}$ = 0) | 6,731 | 29,921 |

Table 3.1: This table shows the lengths of data sets with different amounts of null data dropped. Here, p-e$_{SM}$ refers to the number of photo-electrons in the Shower-Max detector and p-e$_{\pi}$ refers to the number of photo-electrons in the pion detector. This information is relevant because, in general, more data leads to higher accuracy in network performance.

The histograms in Figure 3.1 show the number of zeroes in the photo-electron data for the Shower-Max detector in the unfiltered data in which the null data points are left. The histograms in Figure 3.2 show the number of zeroes in the photo-electron data for the pion detector in the unfiltered data with the null data points left in. These plots show that the majority of the photo-electrons corresponding to electrons are absorbed in the Shower-Max detector and the majority of the photo-electrons corresponding to the pions are absorbed by the pion detector.

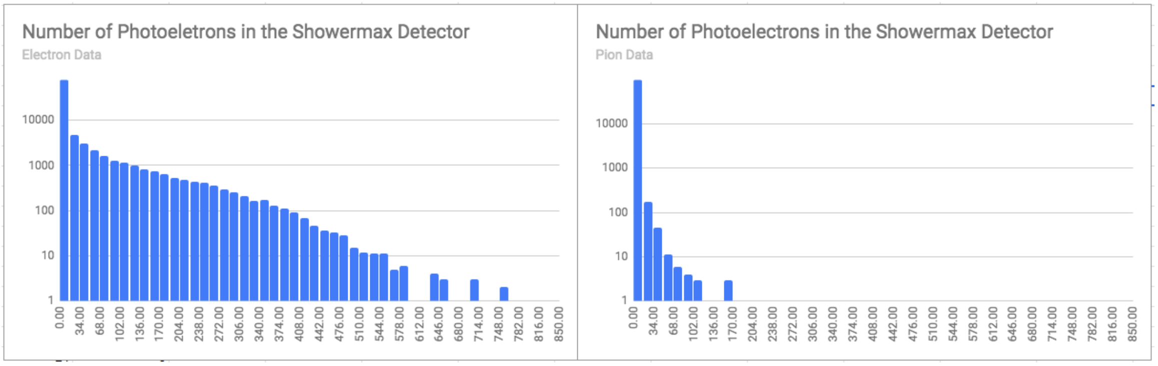

Figure 3.1: This figure shows histograms of the number of photo-electrons in the Shower-Max detector for electrons (left) and pions (right). These histograms are plotted on a log scale so that data points not equal to zero can be seen. The data shown here includes null data points.

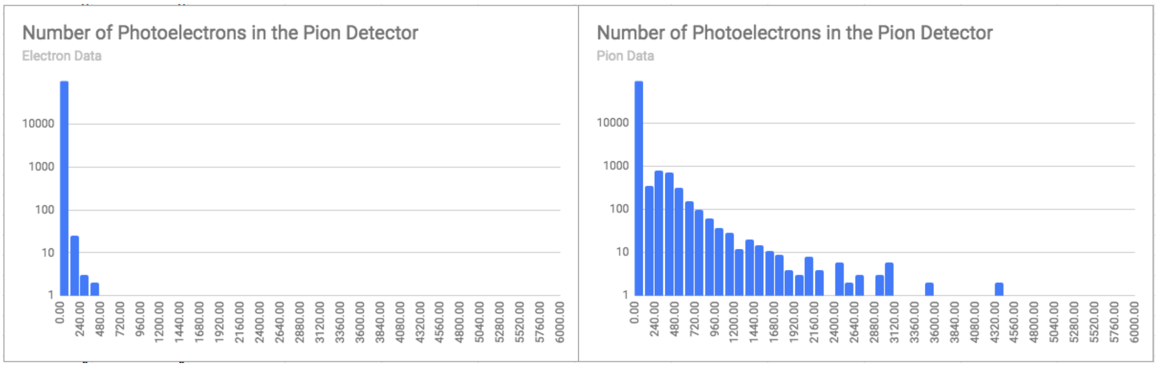

Figure 3.2: This figure shows histograms of the number of photo-electrons in the pion detector for electrons (left) and pions (right). These histograms are plotted on a log scale so that data points not equal to zero can be seen. The data shown here includes null data points.

### 3.1.2 Observable Plots

The observables used for classification were distance from the beamline to the position of the hit (r, Figure 3.3), the x and y coordinates, and the number of photo-electrons in the pion and Shower-Max detectors. It can be quantitatively stated from the plot below that the r variable has good distinguishability between pions and electrons indicating that the network should be able to identify the differing patterns present. Histograms for x and y data as well as filtered Shower-Max and pion detector data can be seen in Appendix B.

### 3.1.3 Quantitative Distinguishability

Quantitative distinguishability of the observables was calculated using the coefficient of variation. No cuts were made on the observables before feeding them into the network, however, the quantitative disinguishability of each observable was calculated to determine the oder in which the network was given data in testing. The coefficient of variation (CV) is calculated by dividing the standard deviation by the mean for both pions and electrons. The distinguishability was then calculated by subtracting $\text{CV}_{pion}$ from $\text{CV}_{electron}$ and taking the absolute value. As is clear from this formulation, if the means and standard deviations of the pions and electrons are similar, this number will be small. Likewise, if the means, standard deviations, or both are different this number will be large, and the bigger the difference the larger the distinguishability. It was recently pointed out that in some cases, while the means and standard deviations could be very different, the ratio of the two could be similar, leading to a small quantitative distinguishability. This insight should be explored more in future research. Similarly to the figure shown in Section 3.1.1, Figure 3.4 shows electron and pion data for the x coordinate overlaid to exemplify the distinguishability. Table 3.2 shows relevant distinguishability data.

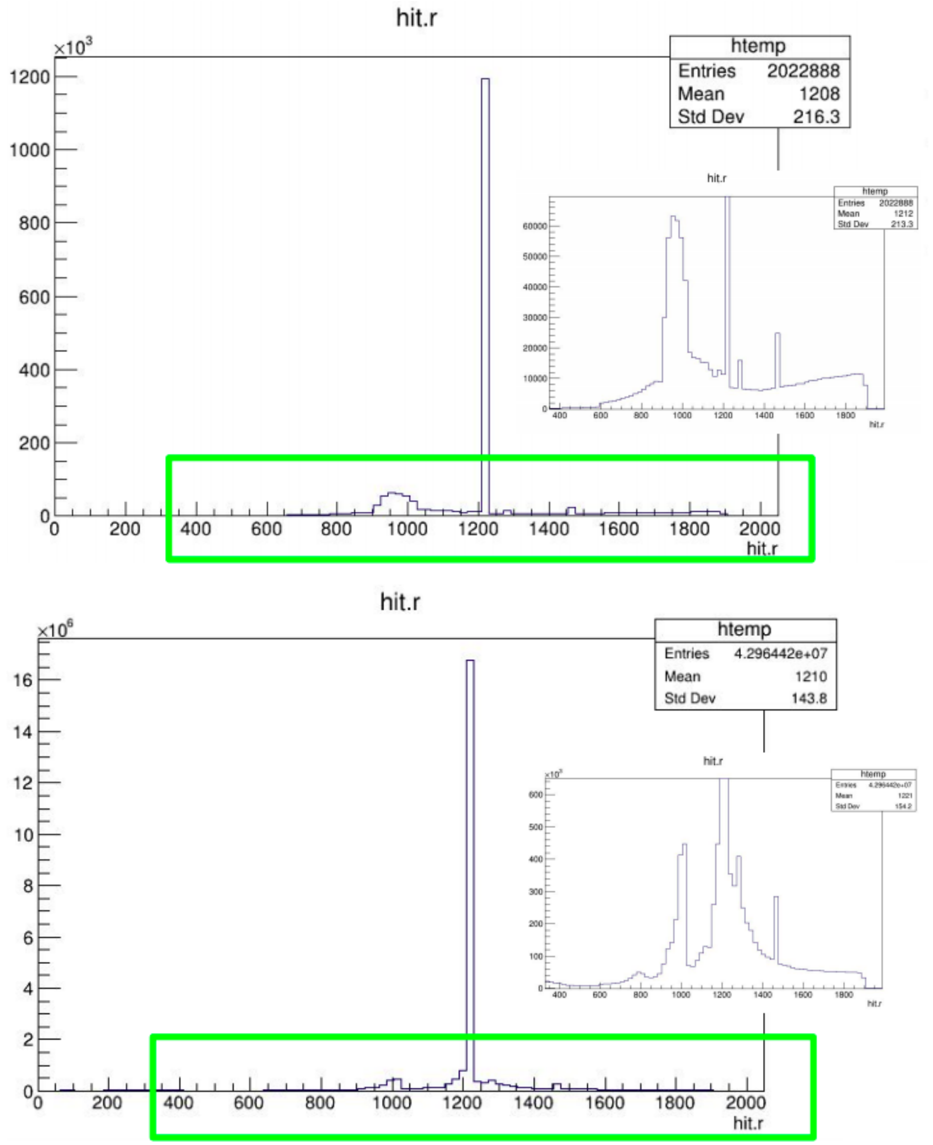

Figure 3.3: This figure shows histograms of the r variable (defined above) for pions and electrons. The y-axis represents number of events and the x-axis is distance in millimeters. The scales on the larger plots are different, however, the scales on the smaller plots are the same in order to be able to clearly see the difference between the shapes of the distributions of the pions and electrons. The top plot shows the pion data and the bottom plot shows electron data.

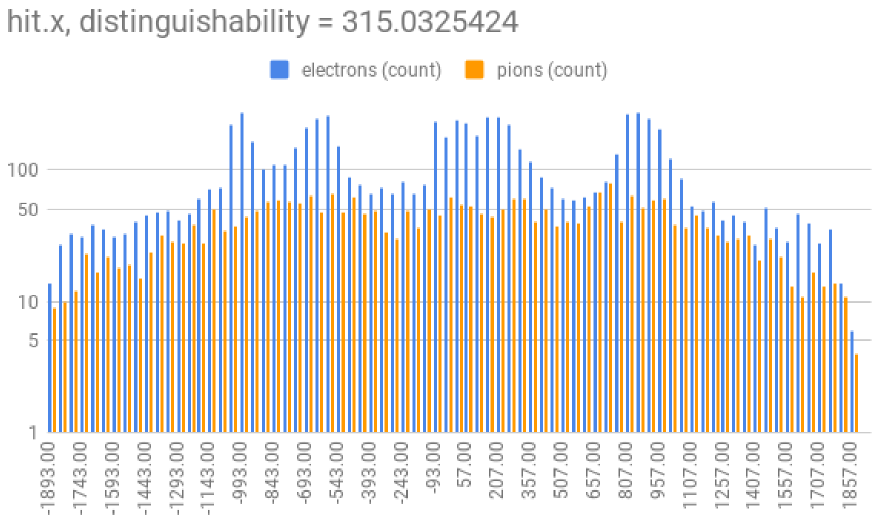

Figure 3.4: This figure shows histograms of the x variable overlaid for pions and electrons. The y-axis represents number of events and the x-axis is distance in millimeters. Here, the distinguishability is high indicating a difference in the means and standard deviations of x data for pions and electrons. This indicates that when run through the neural network, the network should be able to effectively learn from this observable. Null data points have been removed from data shown.

| Observable | Distinguishability |
|---|---:|
| x | 315.0325424 |
| y | 295.4429294 |
| r | 0.9770266829 |
| Number of photo-electrons in the pion detector | 9.19215353 |
| Number of photo-electrons in the Shower-Max detector | 2.38805432 |

Table 3.2: The distinguishability of each relevant observable is shown here. The distinguishability = $\mid CV_{electron} - CV_{pion} \mid$. Larger values indicate more usefulness for the network. Here, x has the highest distinguishability and is thereby the observable that is hypothesized to be most useful to the network in learning the difference between pions and electrons. The data presented does not include null data points.

### 3.1.4 Network Results

**Variation in Optimization Procedure**

In order to achieve the most effective neural network, three parameters were varied in each run. The first is number of epochs. The number of epochs determines the number of times the network runs through the data, for example, if the number of epochs is ten, the network with look at the data ten times before settling on a decision about the likelihood that the data point fed in is an electron or pion. The second parameter is batch size. Batch size is the number of data points the network looks at at one time. For example, if the batch size is ten and the total number of data points is one-thousand, the network will look at one-hundred batches of data with ten data points in each batch. The final parameter is cutoff. The cutoff indicates the likelihood of a particle being a moller electron that experimenters count as being a moller electron. For instance, the most intuitive cutoff would be fifty percent, or .5. This would mean that when the network output data is analyzed, if the network gives a fifty percent or higher likelihood of a particle being a moller electon, it is counted as a moller electron. This parameter can be as high as 0.999999 and as low as 0.0001 to optimize results. Thus far, a total of 238 combinations of these parameters and 16 data sets have been tested.

In testing, the network was first run using only x data which has the highest distinguishability. Next y data was added, then timing data. It was later determined that timing data is not a usable observable because, while the simulation has fine timing data, the timing data from the actual experiment will be too coarse to use. Next, the timing data was taken out and photo-electron data was added. R data was not added to the network because r = $\sqrt{x^2 + y^2}$, a correlation the network would have determined and therefore, adding r to the data set would not have added new information. 10% of the data was used as a validation set. Finally, photo-electron data in the Shower-Max and Pion detectors was added and null data points were removed, as described above.

At each iteration of data sets, a batch size of ten and an epoch of ten was shown to be the most effective. The most effective cutoff varied from .5 to .9 and likelihoods below 50% don't exist in most runs of the network. This indicates that the network favored labeling data points as electrons. This is probably because there is more moller electron data than pion data because all rows having a zero in the data set were dropped (null data). Interestingly, adding the timing data actually decreased the accuracy of the network. Even though the distinguishability of the timing data is high at 1644.6023, the null rows in the timing data took out more data than it added information.

The Figure 3.5 compares network performance for different network configurations using only x and y data. As is shown on the plot, the configuration with the highest accuracy was using an epoch of ten, a batch size of ten, and a cutoff of .68. This configuration resulted in about 73% accuracy for inputted particles. As is shown on the plot, in general, accuracy increases with cutoff as discussed above.

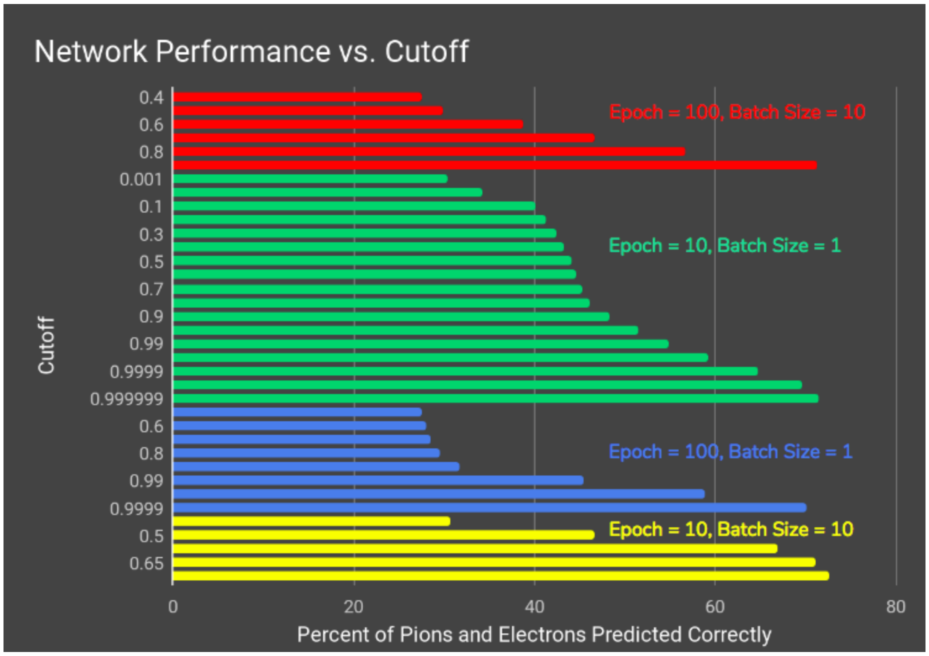

Figure 3.5: This figure shows network performance vs. cutoff for multiple network configurations. Each color on the plot represents a different combination of epoch and batch size. The x-axis is the percentage of particles correctly identified. The y-axis is the cutoff, elaborated on above.

**Full Data Set Results**

Table 3.3 shows the results for the full data set network runs. In this data, points were dropped where x = 0, y = 0, or both the number of photo-electrons in the pion and Shower-Max detector = 0. In general, electron identification accuracy hovers just above 70% while pion identification accuracy is much lower, between 2% and 3%. The particle identification efficiency was optimized for the full data sets. Rather than testing various cutoffs by hand, a code was written that found the optimal cutoff for identifying the maximum number of pions and electrons. The uncertainty of this optimization is ± 5%. All further results presented were generated using the optimal cutoff unless otherwise stated.

| Epochs | Batch Size | Cutoff | Percentage Identified Correctly | |
|---|---|---|---|---|
| | | | Electrons | Pions |
| 100 | 10 | 0.1 | 75.014 | 1.449 |
| 10 | 10 | 0.1 | 73.061 | 1.870 |
| 100 | 1 | 0.1 | 68.872 | 2.941 |
| 10 | 1 | 0.1 | 68.644 | 2.916 |

Table 3.3: This table shows the network results for four different configurations of number of epochs and batch size with optimized cutoff. The accuracy is low for pions and as is clear from the cutoff, the network labels most data points as electrons

The low pion identification accuracy could be attributed to a number of factors. The most likely explanation is that when the null data points are removed from the data set, the pion data only comprises about 18% of the tested statistics. Because 10% of the data is used for network training and the other 90% is used for testing, the low pion accuracy is likely a result of low statistics. With more data to train the network on, ideally at least 10,000 data points for both pions and electrons (a 200,000 point data set, total), the accuracy would be expected to increase. Unfortunately, this type of data set was unavailable for this project. Other explanations include the distinguishability between observables not being high enough and there not being enough observables available for testing. Taking a closer look at the network test in which the number of epochs and batch size were both 10, even at a cutoff of .9, the network only identified about 5% of the pions accurately.

### Variation in Network Architecture

In order to attempt to increase the accuracy of pion identification, a more in depth network structure change was implemented. These networks configurations were tested using 10 epochs and a batch size of 10 because this yielded the highest accuracy in the original configuration. First the number of network layers was tested. Of the networks tested, the network structure that yielded the highest accuracy was a network with four fully connected layers. Next, different numbers of dropout functions with different values were tested. A single dropout function with a dropout value of .9 got the best results. Next, in the first fully connected layer, a variance scaling weight initialization was found to be the most effective weight initialization. Also in this layer, a bias was more effective than the layer used without bias. Finally, a softplus activation in the last fully connected layer was determined to give the highest accuracy compared to other activations. The final network code can be seen in Appendix A.

While this most effective network configuration resulted in about 81.4% of the electrons being correctly identified, the pion accuracy was still very low; less than .1% at the optimized cuttoff and only about 1.5% at a cutoff of .9. The network performance plot, Figure 3.6, discussed above, is shown here. While the network is clearly effective at classifying electrons, it is ineffective at classifying pions.

Because the pion accuracy is so low, the ROC curves made are not worth showing here as they do not give any useful information about the network performance that

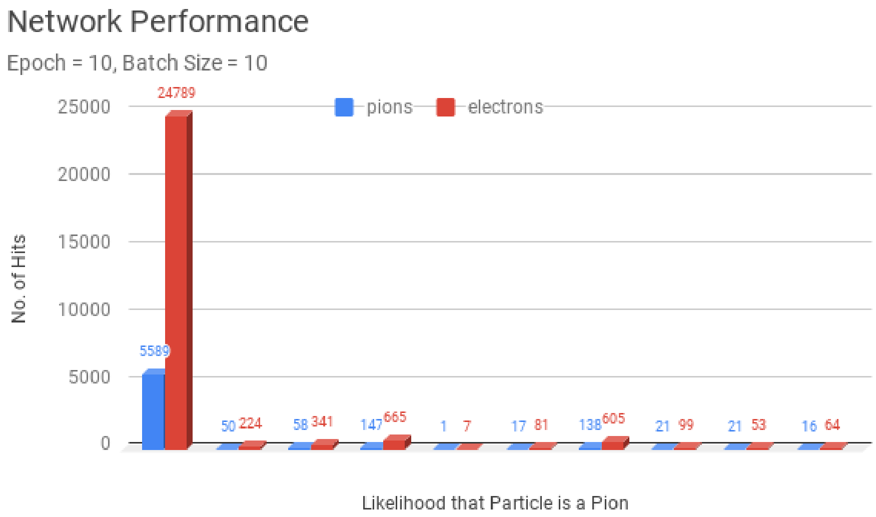

Figure 3.6: This figure shows the network performance plot for the final network configuration. The left most collumns are the 0 - 10% likelihood the particle is a pion bin which corresponds to 90 - 100% likelihood the particle is an electron. The fact that most of the electrons are in this bin is positive, however, the majority of the pions are in this bin as well. This plot shows that the network identifies the majority of particles as electrons.

cannot be gained from the clearer network performance plot above.

# Chapter 4

# Conclusion

## 4.1 Hardware Design Applications

It is the goal of this experiment to aid in the design of the pion detectors by determining which observables would be most beneficial to network performance. In the future, these observables may be prioritized in the design of the detectors. From the results presented here, the recommendation from this project is that the following observables be prioritized: distance from the beamline to the position of the hit and number of photo-electrons in the Shower-Max and pion detectors. Figure 3.7 shows the current detector setup [4].

## 4.2 Further Testing

Students in the future wishing to continue this projects have a couple of avenues through which to do so. The first would be to focus on the development of the neural network. While initial optimal results were obtained by changing the epochs, batch size, and cutoffs, activation, etc. a completely different network architecture could be tried to maximize network efficiency. Such differences in architecture could include a different network shape or different numbers of hidden layers. Alternatively, students could use the results presented above to contribute to the hardware design of the

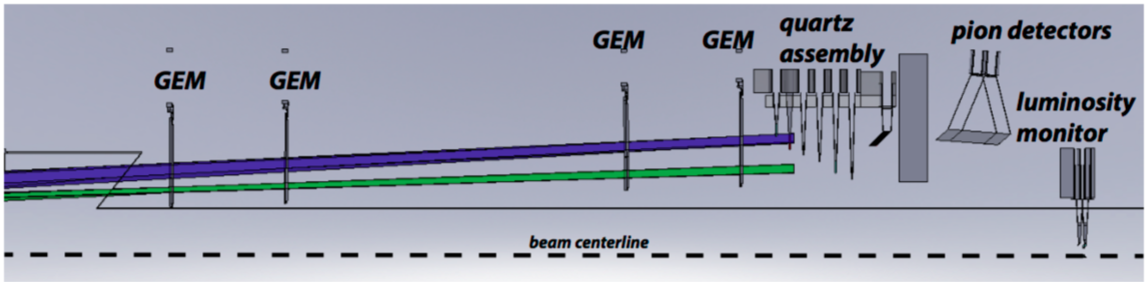

Figure 4.1: This figure shows the layout of the main tracing and integrating detectors. The predicted electron trajectories are shown in blue and green. The green track is the predicted track for electrons that are elastically scattered off of target protons and the blue track is the predicted track for electrons elastically scattered off of target electrons (Moller Electrons). Taken from [4].

MOLLER experiment. While the initial design phase of the experiment is coming to a close, there is still plenty of hardware work to be done that could possibly be influenced by the results of this project.

## 4.3 Final Summary

This thesis details the first attempt at classifying pions and electrons for the MOLLER experiment using a binary classifier deep neural network. While the network was fairly successful at identifying electrons, the pion identification accuracy was very low, regardless of the network configuration. Because of this, it is not advisable to use this specific machine learning approach in the analysis of the MOLLER experiment without further testing. In order to increase pion identification accuracy more statistics are needed as well as perhaps a different neural network structure. Currently, these results cannot be compared against other classification schemes for the MOLLER experiment because no other have been tried as of April 2019.

# Appendix A

# Code Sample

The following is the neural network code found to be the most effective.

```
import numpy as np
import tflearn
import csv


# Load CSV file, indicate that the first column represents labels
from tflearn.data_utils import load_csv
data, labelsA = load_csv(’/data/remoll/akburns/config/hit_b_train.csv’, target_column=4, categori
input, labelsB = load_csv(’/data/remoll/akburns/config/hit_b_tst.csv’, target_column=4, categoric
# Build neural network
#Data has 2 features - change each time a new observable is added

net = tflearn.input_data(shape=[None, 4])
net = tflearn.fully_connected(net, 10, activation=’tanh’, regularizer=’L2’, weight_decay=0.001, b
net = tflearn.fully_connected(net, 32)
dropout1 = tflearn.dropout(net, 0.9)
net = tflearn.fully_connected(dropout1, 10, activation=’tanh’, regularizer=’L2’, weight_decay = (
net = tflearn.fully_connected(dropout1, 2, activation=softplus)
net = tflearn.regression(net)

# Define model
model = tflearn.DNN(net, tensorboard_verbose=3)
# Start training (apply gradient descent algorithm)
model.fit(data, labelsA,  n_epoch=10, batch_size=10, show_metric=True, validation_set=0.1)

pred = model.predict(input)
print("Moll likelihood:", pred)

with open(’/data/remoll/akburns/evaluate/hit_b_rsults.csv’,’wb’) as f:
csv_writer = csv.writer(f, delimiter=’,’)
for x in pred:
csv_writer.writerow(x)
```

# Appendix B

# Observable Plots

These plots show histogram of x (Figure B.1), y (Figure B.2), the number of photo-electrons in the pion detector (Figure B.3), and the number of photo-electrons in the Shower-Max detector (Figure B.4). These plots give readers an idea of the differences in the shapes of the distributions in pions and electrons. In each of these figures, the electron histograms are shown on the right and the pion histograms are shown on the left.

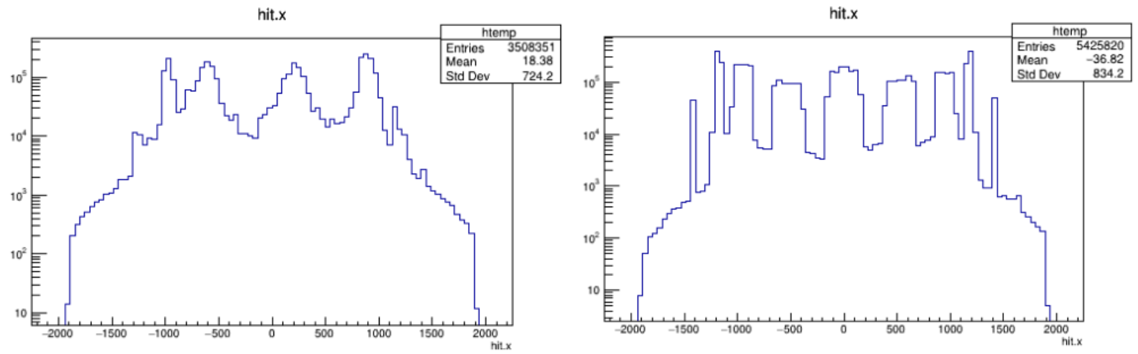

Figure B.1: This figure shows histograms of the x variable for electrons (left) and pions (right). The y-axis represents number of events and the x-axis is distance in millimeters. The scales on the plots are the same in order to be able to clearly see the difference between the shapes of the distributions of the pions and electrons.

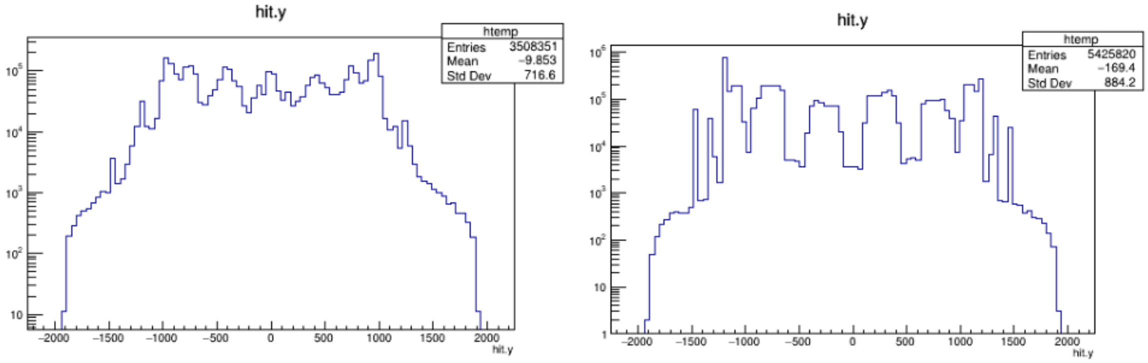

Figure B.2: This figure shows histograms of the y variable for electrons (left) and pions (right). The y-axis represents number of events and the x-axis is distance in millimeters. The scales on the plots are the same in order to be able to clearly see the difference between the shapes of the distributions of the pions and electrons.

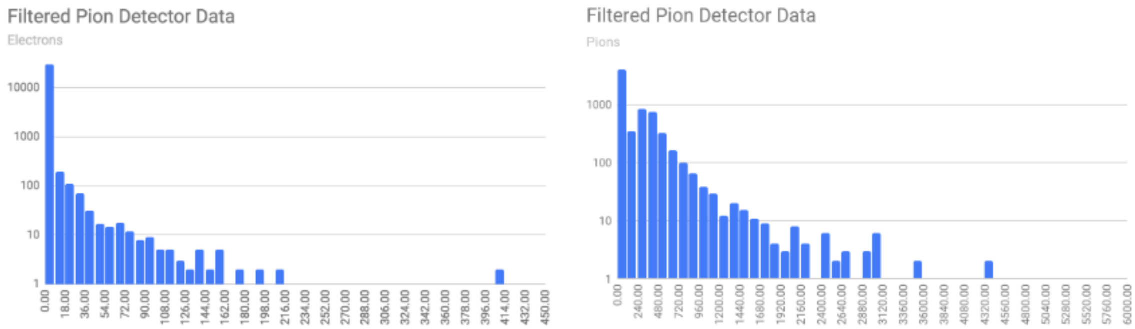

Figure B.3: This figure shows histograms of the number of photo-electrons in the pion detector using filtered data for electrons (left) and pions (right). The scales on the plots are different in order to be able to see the non-zero data points on both plots. The data is plotted on a log scale because non-zero data points on the electron plot are otherwise not visible.

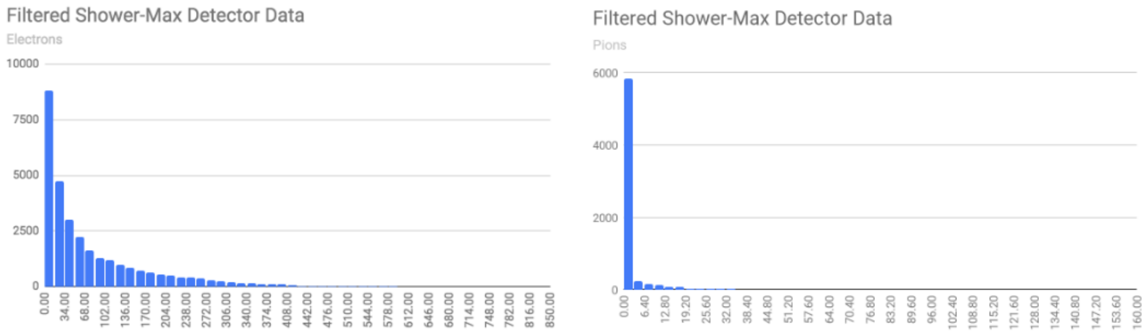

Figure B.4: This figure shows histograms of the number of photo-electrons in the Shower-Max detector using filtered data for electrons (left) and pions (right). The scales on the plots are different in order to be able to see the non-zero data points on both plots on a linear scale.

# Bibliography

[1] Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC - ATLAS Collaboration (Aad, Georges et al.) Phys.Lett. B716 (2012) 1-29 arXiv:1207.7214 [hep-ex] CERN-PH-EP-2012-218

[2] Abadi, M., Barham, P., Chen, J., et. al. (n.d.). TensorFlow: A System for Large-Scale Machine Learning. Retrieved from https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[3] P.L. Anthony, et al. [SLAC E158 Collaboration], ”Precision measurement of the weak mixing angle in Moller scattering” Phys.Rev.Lett. 95 (2005) 081601 DOI: 10.1103/PhysRevLett.95.081601 arXiv: hep-ex/0504049.

[4] J. Benesch, et al. [MOLLER Collaboration], ”The MOLLER Experiment: An Ultra-Precise Measurement of the Weak Mixing Angle Using Moller Scattering,” arXiv:1411.4088 [nucl-ex].

[5] Gupta, Vikas. Home. Learn OpenCV, 9 Oct. 2017, www.learnopencv.com/understanding-feedforward-neural-networks/.

[6] Hanley, J. A., Mcneil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology, 143(1), 29-36. doi:10.1148/radiology.143.1.7063747

[7] M. Herrero ”The Standard Model” NATO Sci. Ser. C **534**, 1 (1999) doi:10.1007/978-94-011-4689-01

[8] KDnuggets. KDnuggets Analytics Big Data Data Mining and Data Science, www.kdnuggets.com/2017/05/machine-learning-crash-course-part-1.html.

[9] Jacob McCormick, ”GEANT4 Simulation of Pion Detectors for the MOLLER Experiment”, Senior thesis, Physics Dept, College of William and Mary, May 2017 (unpublished)

[10] The MOLLER Experiment: Measurement Of a Lepton Lepton Electroweak Reaction, Pre-Conceptual Design Report, J. Benesch, et al. [The MOLLER Collaboration], Dec. 2016 (unpublished).

[11] P. Souder, K.D. Paschke, ”Parity violation in electron scattering” Front.Phys.(Beijing) 11 (2016)111301 DOI: 10.1007/s11467-015-0482-0.

[12] Loss Functions. Loss Functions - ML Cheatsheet Documentation, ml-cheatsheet.readthedocs.io/en/latest/loss$_{f}unctions.htmlcross - entropy$.