How accurately can a neural network be trained to use data from the TRD detector in ALICE to identify electrons from pions.

Author: Andre Barreiros, Supervisor: Prof. Thomas Dietel

September 23, 2018

Abstract

In this experiment it was found that the neural network used was able to reduce the amount of pions in a pion-electron sample by 99.1% at the cost of simultaneously reducing the amount of electrons in the sample by 78.5%. This was done by using ALICE's particle identification to make a reference data then, using TPC data to verify the quality of this data; we were able to find that the reference data had a contamination of 0.484 ± 0.001 %.of pions.

Contents

1 Introduction

In this section, I would like to provide a short introduction in to some of the important concepts which are relevant and were used in this experiment.

1.1 Background

CERN is a European research organization dedicated to the study of particle and nuclear physics, with one of its instruments being the Large Hadron Collider (LHC). The LHC is located near Geneva Switzerland; about 100 metres under the French-Swiss border and is home to many research groups specializing in different fields, one of which is relevant to this project, ALICE.

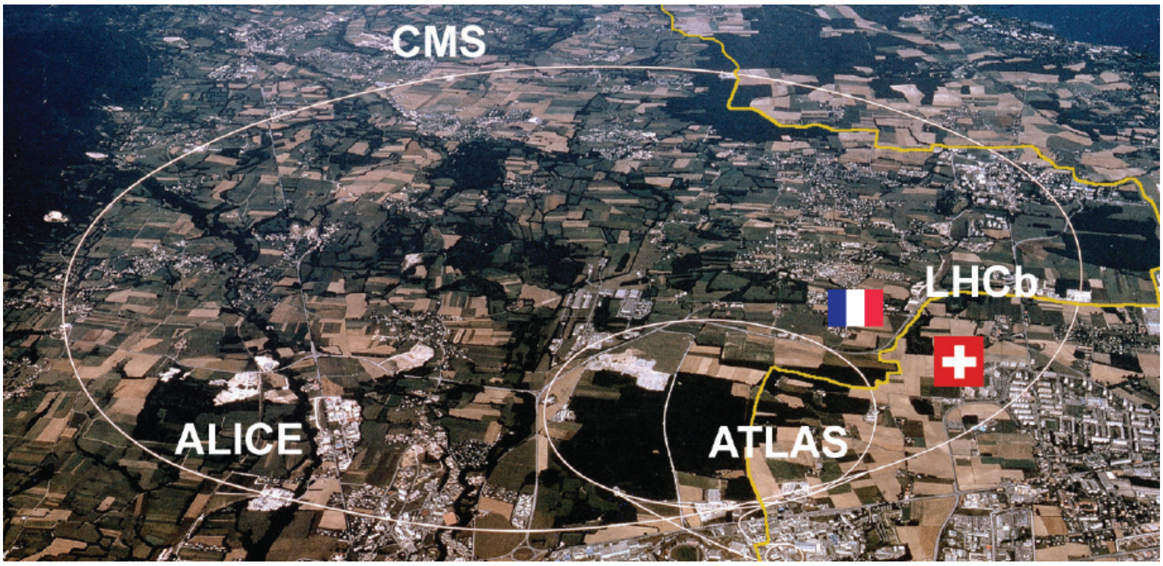


Figure 1.1.1: This aerial view of the LHC under the French-Swiss border, the relevant experiment group ALICE, is show in the bottom-right of the picture. The collider has a circumference of 27 km .and can accelerate particles to 13 TeV. This image was obtained from [1].

ALICE is an acronym for 'A Large Ion Collider Experiment', and as the name suggests, is primarily focused with colliding heavy ions at relativistic speeds. The collisions are encouraged to occur within the detectors, shown below (Figure 1.1.2). Due to the amount of energy in each particle, a collision very briefly (10^{-22} s) form a quark-gluon plasma and expand outwards. As the plasma expands, it cools and new particles form from it which fly outwards towards the detectors, where they can then be detected.

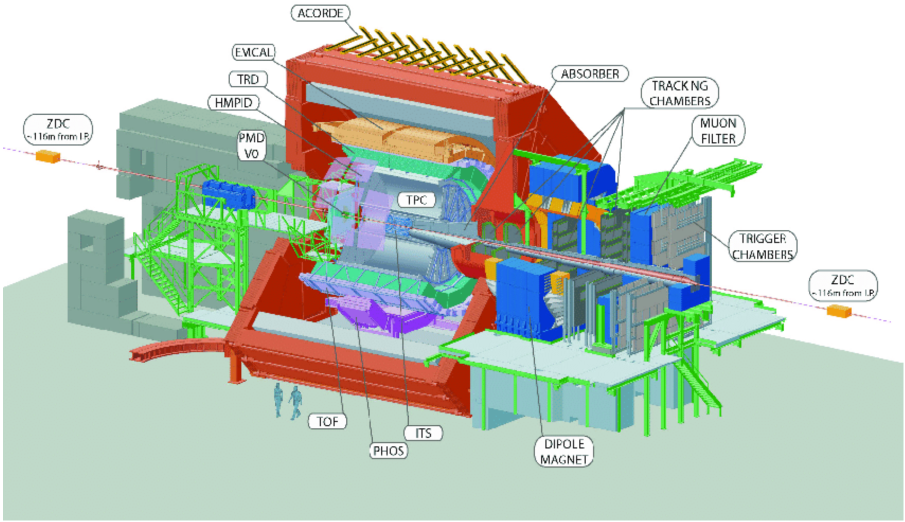


Figure 1.1.2: This figure shows an abstracted image of the detector used in ALICE, with the different components of the detector marked. Each labeled part is a different detector tasked with detecting different particles using many different methods, for more information on each detector visit [2]. The image was obtained from this site [3].

The first method of identifying particles involves many different detectors and is seen once the raw data is reconstructed; it involves tracing out the path of a charge particle. This method relies heavily on the Transition Radiation Detector (TRD) for electrons and pions, it works by having the charged particles enter a radiator, where it emits Transition radiation (TR). It then travels through a gas chamber, where this radiation can be detected. [4]

The second method of identifying particles uses the Time projection chambers (TPC) works by having layers of gas chambers, about 6 cm thick stacked on top of each other. Between each layer are some conducting wires which act as the detectors. The idea behind this detector is that as a charged particle travels through the gas chamber, ionizes some of the gas molecules. The ionized gas then travel to the nearest cathode/anode where it can then be detected. The amount of gas ionized depends on both the particle and its own energy, with higher energy resulting in more ionized gas. Since the amount of energy the particle has is measured, then a reconstruction its path can reveal the average amount of energy it loses per distance. If its momentum is known, then it may be placed on a Bathe-Bloch curve.

1.2 Decay channels

An alternative way in which particles can identified is through the decays of their parent particles ie,

$$K_s^0 \rightarrow \pi^+ + \pi^-$$
$$\Lambda \rightarrow p + \pi^-$$
$$\gamma \rightarrow e^- + e^+$$

There are more examples of decays but these are the ones we are interested in. This allows for a relatively pure sample for each particles since if for example if a gamma ray travels through the detector and then a it splits into two particles of opposite charge, then it would probably indicate a electron-positron pair production had occurred.

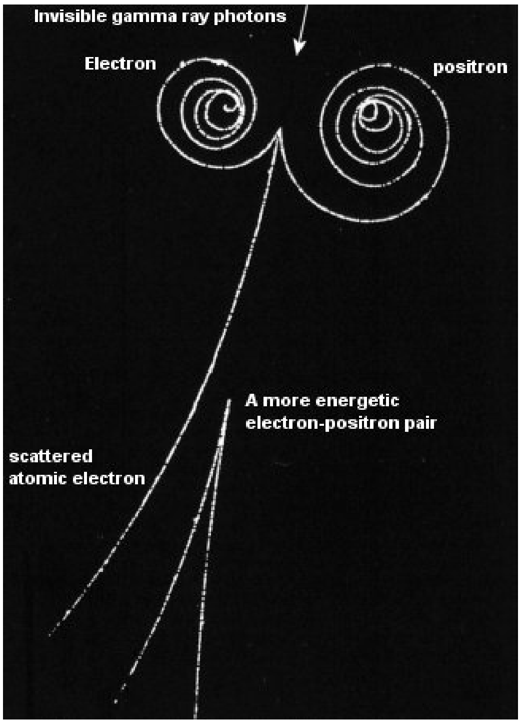


Figure 1.2.1: This figure shows what occurs in the detector when a γ-ray coming from above, collides with an atom in the detector and thus forms an electron-positron pair. In the figure, there are two examples of the pair production, the top pair has less translational energy resulting in them quickly spiraling inwards. The pair production which occurs further down the image has more momentum and thus have doesn't spiral inward as fast. This is relevant because if the pair production occurs with lower energy γ-rays, then the particles will not reach the TPC detector for further analysis. Image from [5].

The spiral of the electron and the positron shown in $figure$ 1.2.1 is caused by the magnetic field in the detector, which perpendicular to the particles motion. The particles motion actually follows a helical motion [6] but we will be considering a velocity which is perpendicular to the magnetic field. The momentum of a particle is found in this stage by considering the following formula:

$\frac{mv^2}{r} = F = Bvq,$

which gives us the centrifugal force of a moving charged particle in an always perpendicular magnetic field. Thus, can rearrange this equation to:

$\therefore p = mv = Bqr$

Since the B-field is assumed to be constant on a this small scale in the detector and these are elementary particles; q is constant. Thus we have momentum of a charged particle as a function of the radius of its spiral.

These particles formed through decay channels are identified by first finding two tracks which originate from the same (or sufficiently close) point. Once these tracks are identified their momentums are calculated using the method described above and their charges can be deduced by how they move through the magnetic field. The detectors within ALICE which are responsible for the tracking are the ITS, TRD and TPC detectors.

It should be noted that as in $Figure$ 1.2.1, the radius changes as the particle moves within the detector. This changed in radius can be used to find the force experienced by the particle. With this we can find the energy lost per distance through the equation:

$\frac{dE}{dx} = \frac{d}{dx}\int Fdl = \frac{d}{dx}\int Bq\frac{dr}{dt}dl$, where dl is the path traveled.

1.3 Bethe-Bloch graph

The Bethe-Bloch formula is which describes how a particle moving through a material loses energy as it travels through it. The general formula for the Bethe-Bloch is given to be:

$$-\frac{dE}{dx} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z}{A}\frac{z^2}{\beta^2}[\log(\frac{2m_e\gamma^2 v^2 W_{max}}{I^2}) - 2\beta^2]$$

Where r_e = classical electron radius m_e = electron mass
N_a= Avogadro's number I = mean excitation energy
Z = atomic number of absorbing material A = Atomic weight of absorbing material
ρ = density of absorbing material z = charge of incident particle (in units of e)
β = v/c of incident particle $\gamma = 1/\sqrt[2]{1-\beta^2}$
W_{max} = maximum energy transfer in single collision.

For more information on this formula visit [7].
But an easier way to visualize this formula is through a graph as a function of momentum, as given below:

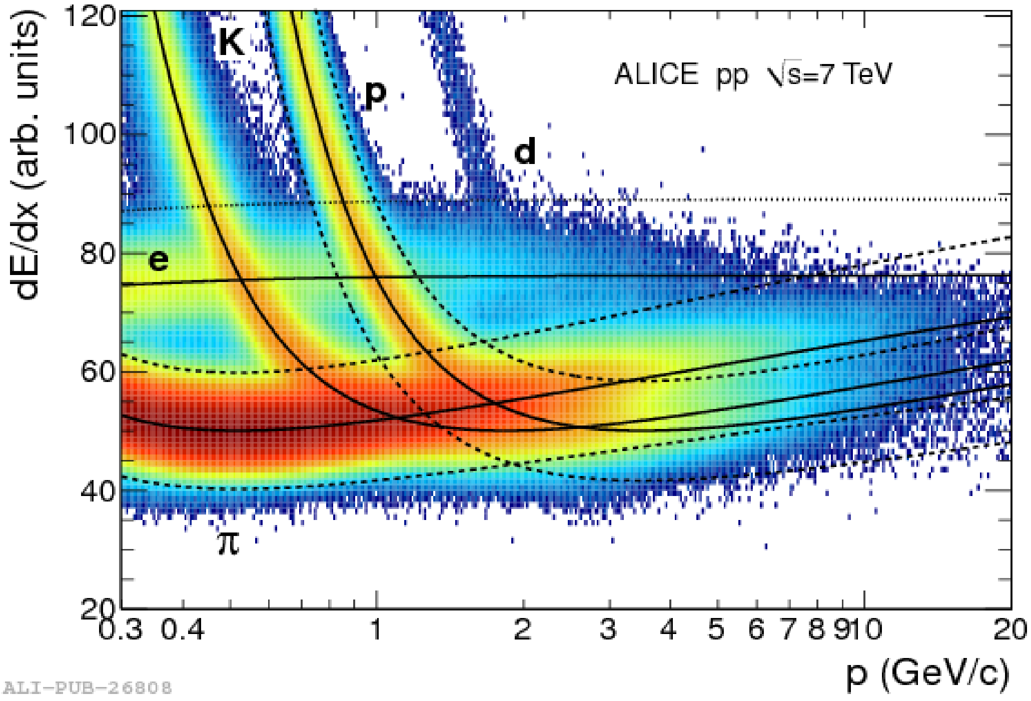


Figure 1.3.1: This graph shows the the energy loss per distance, dE/dx as a function of momentum for each species of particles, superimposed onto the same graph. The solid lines show average energy loss for each species as it travels through the material. While the dashed lines of the pions and protons represent each particles exclusion zone and the dotted line is the 3σ away from the electron line. The different colours show the intensity of particles in the region with red meaning a higher concentration of particles. This image was obtained in [8].

The idea with this is that the when a particle is given after being identified through its parent particles decay (section 1.2); its dE/dx is measured by the TPC detector and compared to its momentum. With this information, you can place it on the Bethe-Bloch graph and see how far it deviates from the mean line for each particle species. The two particles of interest are electrons and pions, which have their respective lines in the graph not intersecting, except when momentum is 100 MeV; allowing us to compare each electron and pion from the method used in section 1.2 with each others line to see how well they agree to their line. This would not be the case if we were to compare the Kaons with the Pions, since their lines intersect around 1.2 GeV/c, as shown in Figure 1.3.1; making it more difficult to draw a conclusion from their position since they would both fit each other's dE/dx lines.

1.4 Neural network

A neural network is a program which is able to 'learn' through pattern recognition of a large data sample (larger data samples allow for a strong pattern to be established), with this it can then draw conclusions one the properties of the said data set and make predications or statements on future (similar) data. A neural network was used in this project to test its potential in being able to quickly (after being trained) identify particles from a given data set. The neural network used was not designed/written by me, but was written by Chris Finlay, an honours student from UCT.

2 Reference Data

In this section, we will be building the reference data through section 1.2 and verifying by using the TPC data to compare the particles Bethe-Bloch lines, section 1.3.

2.1 AliRoot and AliPhysics

AliRoot and AliPhsics are two c++ packages which provide a vast number of functional it for writing programs to be used in the grid in CERN. AliRoot provides core functionality such as data reconstruction and undergoes updates every few weeks. While AliPhysics is focused on more the analyzing the reconstructed data from AliRoot. This package undergoes daily updates, which is updated at 4 pm Geneva time.

The grid is the name which will be used to describe the supercomputer at CERN which is used in ALICE and the database which holds all of the raw and reconstructed data from each interesting collision (called events).

2.2 Running the code

This code is available freely in my git repository [9]. To run it, (assuming you have a CERN certificate and in AliPhysics), you would have to run the command;

- aliroot ana.C in the main directory to submit the program to the grid for analysis.

- after the program has completed its run, modify ana.C the line; 'alienHandler->SetRunMode("full");' to 'alienHandler->SetRunMode("terminate");'

- run aliroot ana.C again, this step collects the data from the different clusters and merges them to one DigitsExtractQA.root file.

- Once the merging has occurred, copy the output file by changing in ana.C from 'alienHandler->SetMergeViaJDL(kTRUE);' to 'alienHandler->SetMergeViaJDL(kFALSE);'.

- Now the data in DIgitsExtractQA.root should be in the main directory and is accessable by typing 'new TBrowser' in aliroot or by running 'aliroot histMod.C', although this file may have to be modified to extract the right histogram.

2.3 Data extraction

The data used for this project was a data from /alice/data/2016/LHC16q, which is a reconstruction of varies p-Pb collisions. The specific runs which were used are given in the appendix. Two different files were used, for the particle identification part of this experiment, the files used are located in pass1_CENT_sDD/*/*ESDs.root. But the TRD data for the events are located in the directory pass1_CENT_sDD/*/TRD.FltDigits.root.

2.4 TRD graphs

This section will provide a comparison of all of the electron and pions (in separate graphs) with and without TRD data. This means that the particle has not been tagged as having data from the TRD detector.

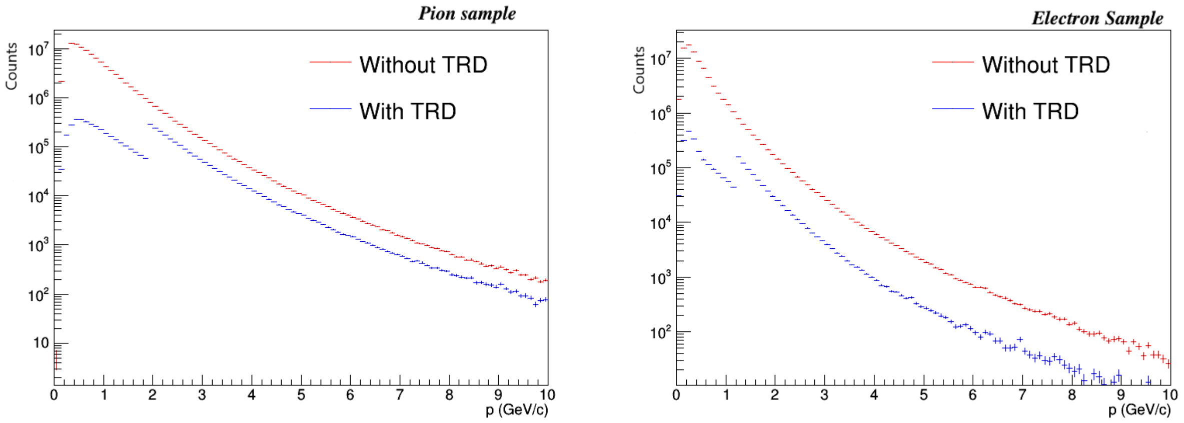


Figure 2.4.1: This figures show the ratio of particles on a log scale with and without TRD data as a function of momentum (GeV/c). The number of pions according to this figure is always greater than the number of electrons. The figure on the left concerns the pions while the right figure shows the results for the result for an electron.

These graphs figures show that the amount of particles with TRD data increases as the particles momentum increases and that the amount of electrons are near to zero when p > 9 GeV/c, while pions still have 2 orders of magnitude more. There exists a noticeable jump in the number of particles detected at 2 GeV/c for the pions and 1.2 GeV/c for the electrons. This is jump exists because around 2 years before this project began, my supervisor managed to produce a filtered version of the raw data from these runs; with a higher ratio of electron and pion in the sample. This filtered data had a higher purity of electron and pions by reducing the number of all other particles as well as not including events with a momentum less than 1.2 GeV/c and 2 GeV/c for the electron and pion respectively. The main reason for there still existing some electrons and pions below this threshold is mainly due to how the filtering chooses which events to consider. An event is placed in to the TRD.FltDigits.root folder only when the event has at least one electron/pion above the threshold, which means if one electron/pion has an energy above this threshold, then the event is considered and any other electrons/pions (even ones below the threshold) from this event are saved as well.

2.5 n-σ values of π and e^-

In this section, we will be seeing how well the method in (section 1.2) of using the decay channels of different particles and how well these particles agree with their Bathe-Bloch curves.

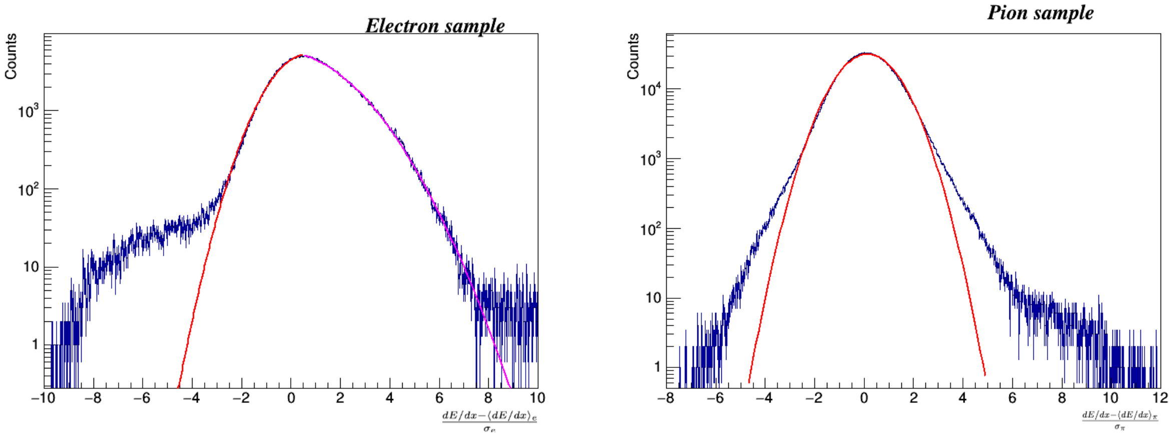


Figure 2.4.2: Here are two graphs which show how well the first method manages to capture the number of electrons (on the left) and the number of pions (on the right) fit their respective Bethe-Bloch curves. The red lines in the graphs indicate a best fitted Gaussian curve on each peak. All of the particles here were chosen to have a momentum value of $2 < \text{p} \leq 3$ GeV/c. A curious consideration is that the electron sample has two halves of a Gaussian fitted to it, the magenta line to the right of its centre seems to fit well except for the tail which starts to form around 7.5 $\frac{dE/dx-\langle dE/dx\rangle}{\sigma_e}$. The magenta gaussian has a σ_e value of 2.09 ± 0.03. This contrasts with the red line which has a σ-value of 1.17 ± 0.01. This is interesting but unfortunately indicates that the distribution does not follow a Gaussian curve. The same can be said for the graph on the right, which fits a Gaussian around its peak.

In both graphs there seems to be a higher tail on one side. It appears on the left for the electron sample and on the right for the pion sample.

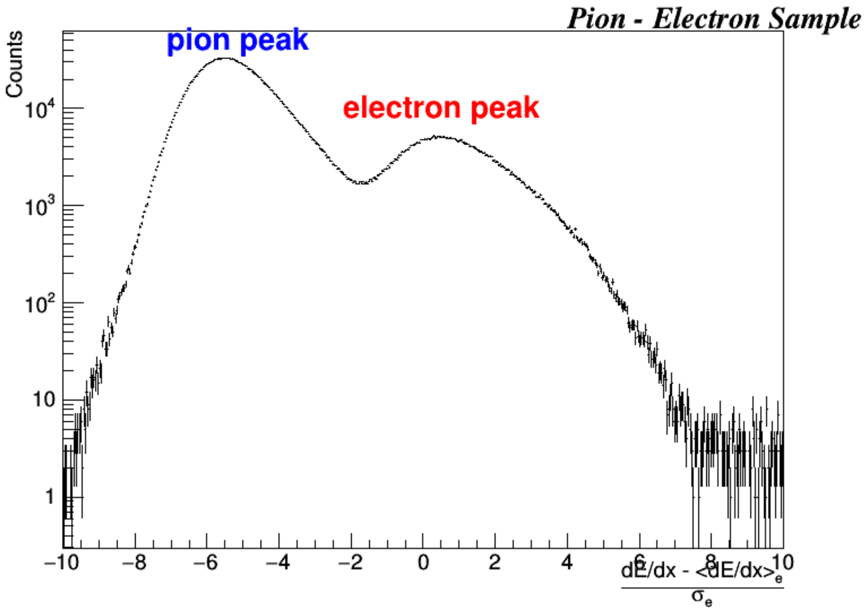


Figure 2.5.2: This figure shows the superposition of the electron and the pion sample compared to the electron Bethe-Bloch curve for when p ∈ (2, 3] GeV/c. It is clear from this figure that there is most probably some pion contamination in the electron sample.

Unfortunately, I was not able to obtain how a pion sample compares to the electron Bethe-Bloch curve, but we can get around this by using subtracting the electron peak from *Figure* 2.5.1 from *Figure* 2.5.2. This would produce larger uncertainties than if a pion sample was used, but it will allow us to estimate the amount and shape of the pions in the electron sample.

It should be noted that we are mainly interested in the number of pions in the electron sample and not so much how many electrons are in the pion sample. For this reason, the next part of this section will be dedicated to finding the amount of pions are probably present in the electron sample.

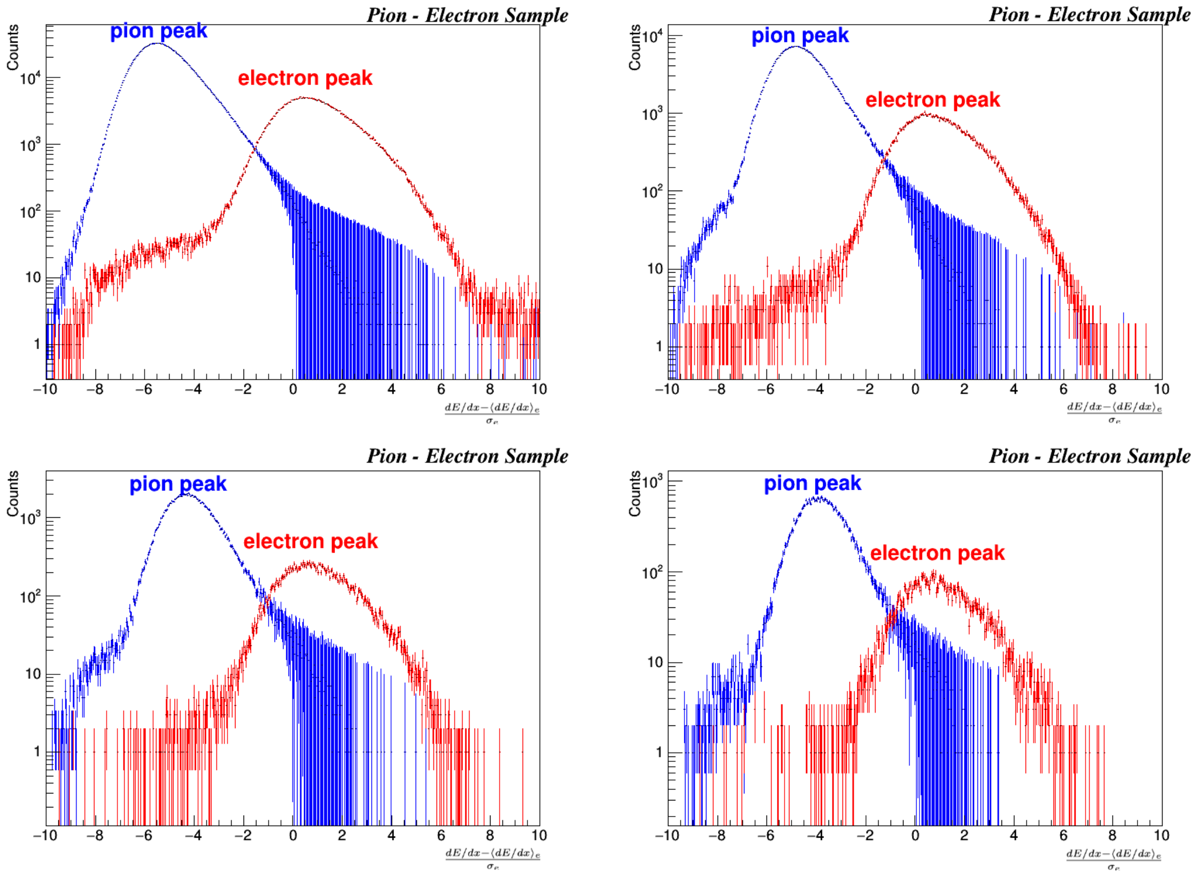
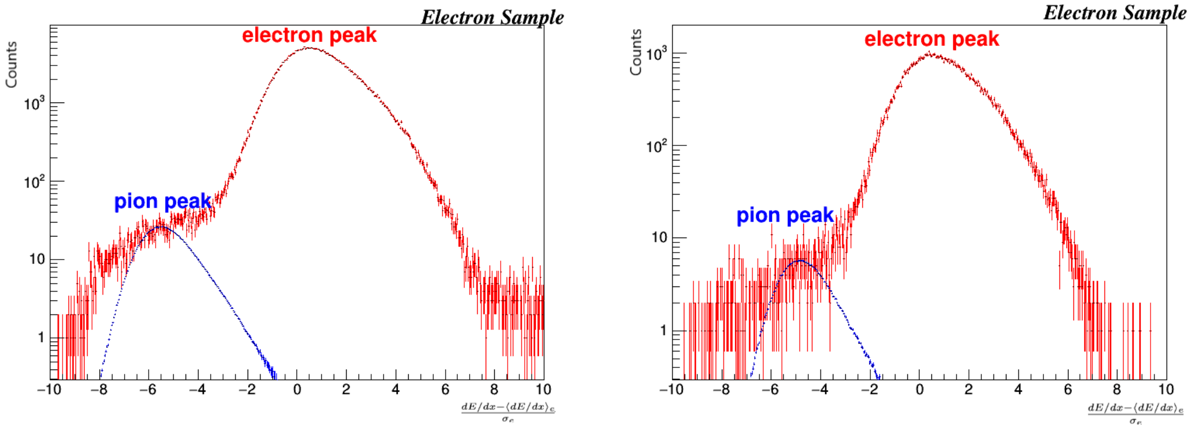


Figure 2.5.3: These graphs show the result of the subtraction of the electron sample in *Figure* 2.5.2 from the pion-electron sample in *Figure* 2.5.2 with its uncertainties. This results in an estimate for the pion peak and its shape within the electron sample. Each one of these graphs are from a different momentum bracket. Moving from left to right and top to bottom, the momentum intervals are $2 < p \leq 3$, $3 < p \leq 4$, $4 < p \leq 5$ and $5 < p \leq 6$ GeV/c, respectively.

With the information from *Figure* 2.5.3, we have an estimated shape for the pion peak. If we assume that the pion sample retains its shape, we may be able to obtain an estimate for the amount of pions in the electron sample in *Figure* 2.5.1. This would be done by scaling the pion peak in *Figure* 2.5.3, down to the size of the left tail of the electron peak.



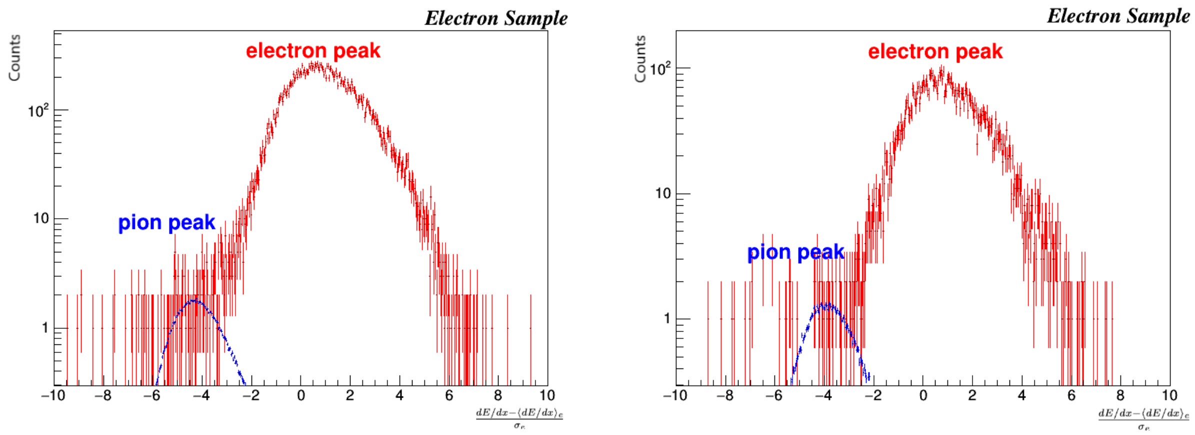


Figure 2.5.4: Here is the same set of electrons and pions as in figure 2.5.3, with the pion peak scaled down till its about the size of the left tail of the electron peak. The momentum for these graphs are the same as in $Figure\ 2.5.3$. The scaling of the pion peak was chosen to be around the same height as the left tail, which means it provides an upper bound for how much it can effect the electron sample. An interesting thing to note is that to the left of the pion peak, there appears to be some other contamination or effect which causes there to still be a few unaccounted for counts. A possible explanation for these counts would be a few protons and kaons are contaminating the electron sample. This is a possibility because the proton and kaon Bethe-Bloch curves (see in $Figure\ 1.3.1$) are below the pion Bethe-Bloch line, but within its 3σ line. Interestingly enough the scaling factor for the pion peak is constant throughout the histograms, maintained at about 0.001 times smaller than the pion peaks in $Figure\ 2.5.3$. This equal ratio is expected since the pion Bethe-Bloch line and the electron Bethe-Bloch line are approximately parallel around in the p $\in$ (2, 6] GeV/c range. Although it should be noted that the peak was more difficult to discern with the bottom two histograms and was more akin to a guess .

With the top-left histogram in $Figure\ 2.5.4$, we can calculate the over all purity of the electron sample by summing all of the entries in each bin. Only the one histogram will be considered because it contains far more particles than the other graphs and due to the pion peak being scaled down by same amount for all 4 histograms, the assumption is that the ratio of electrons and pions will remain the same for all p $>$ 2 GeV/c:

Total counts $= \sum_{i=0}^{n} e_i$, where e_i is the entry of bin i and n is the total number of bins.

The uncertainty in this case would follow the formula:

uncertainty $= \sqrt[2]{u(e_0)^2\ +\ u(e_1)^2\ +\ u(e_2)^2\ +\ ...\ +\ u(e_n)^2}$, where $u(e_i)$ is the uncertainty of e_i.

With this formula, we can calculate the amount of pions to be 2071 $\pm$ 2 and the number of electrons in the same sample being equal to (4.256 $\pm$ 0.007) $\times$ 10^5 counts. This means that:

$\frac{number\ of\ pions}{number\ of\ electrons}$ = (4.866 $\pm$ 0.008) $\times$ 10^{-3}
$\therefore$ (4.84 $\pm$ 0.01) $\times 10^{-1}$ % of the electron sample are pions.
From here we can calculate the ratio of pions and electron for each bin in $Figure\ 2.5.4$, this would provide us some information about the difference between the number of electrons and pion in the electron sample. As well as the region where the least amount of pion contamination is likely to be.

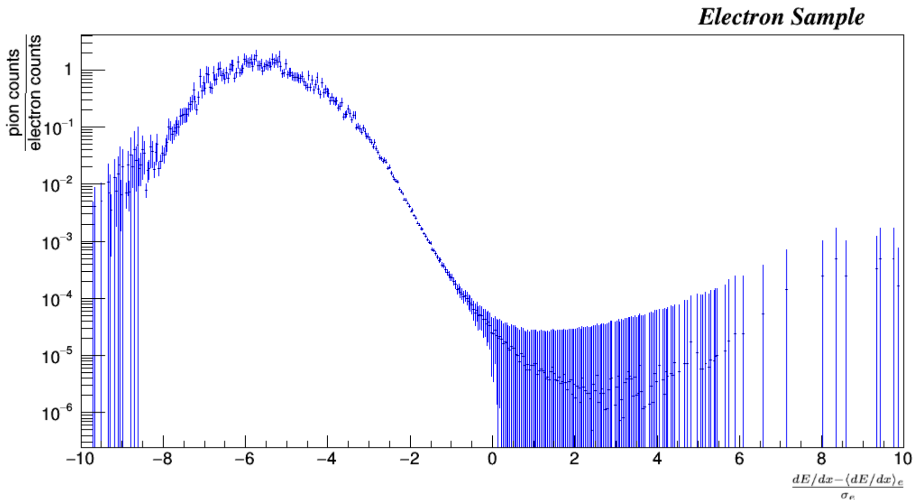


Figure 2.5.5: This figure shows the ratio between the number of pions see in $Figure\ 2.5.4$ when p $\in$ (2, 3], over the number of electrons, also from the same figure.

From $Figure\ 2.5.5$, we can see that if a very low pion contamination of the electron sample is required, then the range from 0 to 2 $\frac{dE/dx-\langle dE/dx\rangle}{\sigma_e}$ should be chosen, since it has the smallest amount of pions to electron ratio. Although the full data set was used when training the network.

3 Analysis

Here I will be describing how the neural network was used and how its input data was chosen. We have an idea of how pure this data is from section 2.5, and with it we will see how well the network preforms as well as discussing some limitations which arose and suggestions on how to solve them.

3.1 Machine Learning Input

The data which inputted in to the neural network for would have to exclusively be data with TRD data provided for it. This mean that the data with TRD shown in $Figure\ 2.4.1$ was used. In this case all of the TRD data from all of the momentum values above each particles respective threshold.

The data obtained from the grid with TRD detector data was placed in a text file, but unfortunately this data had some invalid data, for example, it would show that the particle lost no energy will it traveled through the detector or it had no information on how it interacted with the detector. These false positives had been removed, which provided a 'clean' data sample which could inputted into the network.

There are two components to the neural network provided: the data extraction class, which is responsible with turning a text file with the data from the TRD detector on a particle in to a *.npy file, which can then be feed to the network. The second component is the actual learning part, where the network is given the *.npy file and it goes through varies iterations while outputting the results it gets.

3.2 Machine Learning results

The results given by the network after running once for the 500 epochs (epoch in this case means iteration or generation in this context), a batch size of 100 and keeping the threshold at 0.6, the following results are given below:

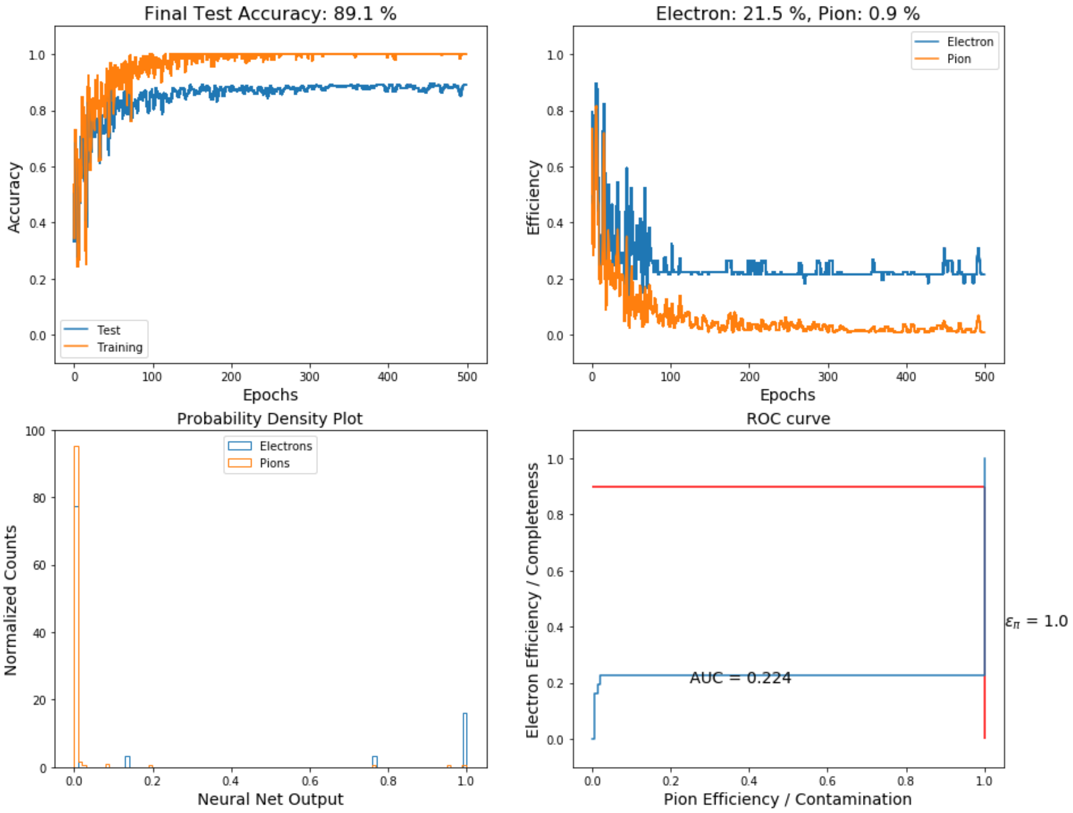


Figure 3.2.1: This are the graphs outputted by the neural network after being fed the data described in subsection 3.1. Discussing first the top-right graph, this shows the evolution of the how accurate the neural network can identify the different particles as it is allowed to correct itself until the end. The top-right graph describes the pion and electron efficiencies of the neural network evolve over the epochs. The bottom-left graph shows the output between 0 and 1 from the neural network, of how certain it is in the result. With a 0 meaning that the neural network is certain that it is a pion and 1 when it is certain that it is an electron. The bottom-right graph tells us how the contamination of the data set and how many electrons are obtained depending on which threshold is used as a cut off point.

3.3 Machine Learning Results

Judging from the figure above, it seems that the neural network with the data set used is able to separate electron from pion well. By well I mean that if a threshold of 0.5, then 21.5% of electrons will be captured while only 0.9% of pions will be miss identified as electrons. An unfortunate down side to using this method is that about 78.5% of electrons are miss identified as pions, with the majority of them having the same neural net output as the pions. This means that this method will inevitably remove most of the electrons from a sample, but on the flip side, it seems to remove 99.1% of pions from the sample.

4 Discussion and Conclusion

4.1 Discuss results

Starting first with the main particle identification, it seems to produce very pure electron samples, obtaining a pion/electron ratio of about (4.866 $\times 10^{-3}$). This means that a given sample of electrons obtained from the method described in section 1.2, 0.484 ± 0.001 % are pions pretending to be electrons. If this method is paired with the machine learning results shown in $Figure\ 3.2.1$, which is able to remove 99.1% of the pions in a sample, then perhaps it would be able to remove 99.1% of pions from the 0.484 ± 0.001 % still present after the particle identification.

4.2 Limitation

The only real limitation is from the input data to the neural network, more specifically the data extraction part of the neural network. If the input file was too large $>$ 15 MB, the data extraction program would crash and my computer would freeze for a second. As a result, I was not able to use the full extend of all of the data available (which is a text file of about $>$ 1.1 GB). This might be a hardware limitation, although I am not sure, computer specs given here [10]. As a result, the runs which were used in training the network were the first 10 directories in the runs 000 256 309, 000 256 334, 000 256 335, 000 256 336 and 000 256 337.

4.3 Conclusions

In conclusion, it seems that the neural network could be used in trying to minimize the amount the amount of pions present in a data sample of pions and electrons. The unfortunate downside to using this method is that the a large number of electrons (78.5%) will be miss identified as pions. Resulting in a much smaller electron sample to work with.

5 References

[1] - $https://spectrum.ieee.org/computing/software/analyzing-the-lhc-magnet-quenches$

[2] - $http://aliceinfo.cern.ch/Public/en/Chapter2/Chap2InsideAlice-en.html$

[3] - $The\ ALICE\ Detector$

[4] - $http://aliceinfo.cern.ch/Public/en/Chapter2/Chap2_TRD.html$

[5] - $http://www.alternativephysics.org/book/MatterEnergy2.htm$

[6] - $https://courses.lumenlearning.com/boundless-physics/chapter/motion-of-a-charged-particle-in-a-magnetic-field/$

[7] - $Techniques\ for\ Nuclear\ and\ Particle\ Physics\ Experiments,\ William.R.Leo,\ page\ 24$

[8] - $CERN\ Document\ Server$

[9] - $https://github.com/ForeverANoob/trdML$

[10] - $MSI\ GP62\ Leopard\ Pro\ specification$

6 Appendix

Here is a list of all of the runs used in this project:

- 000265 309
- 000265 332
- 000265 334
- 000265 335
- 000265 336
- 000265 338
- 000265 339
- 000265 342
- 000265 343
- 000265 344
- 000265 377
- 000265 378
- 000265 381
- 000265 383
- 000265 385
- 000265 388
- 000265 419
- 000265 420
- 000265 425
- 000265 426
- 000265 499