Neutrino interaction

*“Based on NOvA experiment”*

*Juan López-Ríos SN:20121702*

Department of Physics and Astronomy, University College London

INDEX:

[1. Introduction & Background 1](#_Toc131778896)

[1.1Neutrino Interaction 2](#_Toc131778897)

[1.2 Neural Network 3](#_Toc131778898)

[1.3 Convolutional Neural Network 3](#_Toc131778899)

[1.4 Implementation on NOvA like dataset 4](#_Toc131778900)

[2. Methods 5](#_Toc131778901)

[2.1 Binary Classifier (TASK I) 5](#_Toc131778902)

[A) PREPROCESSING 5](#_Toc131778903)

[B) MODEL ARCHITECTURE (BI-CLASSIFIER) 6](#_Toc131778904)

[C) TRAINING AND VALIDATION 8](#_Toc131778905)

[2.2 Testing Performance (TASK I & II) 8](#_Toc131778906)

[2.3 Testing Meta-Data (Task II) 10](#_Toc131778907)

[A) INTERACTIONS 10](#_Toc131778908)

[B) ENERGIES 10](#_Toc131778909)

[C) META-DATA 11](#_Toc131778910)

[2.4 Extensions 12](#_Toc131778911)

[2.4.1 Regression CNN for neutrino energies 12](#_Toc131778912)

[2.4.2 Neutrino Flavour Classifier 13](#_Toc131778913)

[2.4.3 Algorithm for lepton energy over neutrino energy (y) 14](#_Toc131778914)

[2.4.4 Interaction mode algorithm 14](#_Toc131778915)

[3. Results 15](#_Toc131778916)

[4. References 16](#_Toc131778917)

# 1. Introduction & Background

*This section is a first approach of deep learning implementation to particle interaction and all necessary theory.*

Classification of particle interactions recorded by detectors is a core problem to particle physicists. Before machine learning various techniques were employed such as event reconstruction which involved inferring particles properties from the signals captured by the device and further statistical analysis.

Using deep learning algorithms on the signals reconstructed by the device including jets, showers and tracks associated with different neutrino interactions. The successful implementation of machine learning for classification of reconstructed signals in particle physics is impeded by several challenges. Specifically, the reconstruction of high-level features is a complex process that can potentially lead to errors, leading to inaccuracies in the categorization of physics events. Moreover, the selection of features is restricted to those that are already known and characterized, which can limit the effectiveness of machine learning models.

## Neutrino Interaction

*This section discusses neutrinos interactions and relevant theory.*

Neutrinos are electrically neutral subatomic particles with a small however non-zero mass. As part of the lepton family, neutrinos have a specific ‘flavour’ that can be electron (), mu (), or tau (). Each of which associated with a heavier charged lepton.

Since neutrinos oscillate, they can change their flavour as they travel, in fact this phenomenon has been observed from muon to tau, however not from muon to electron. Exploring these oscillations is crucial for the field of particle physics and could potentially change the Standard Model as we know it.

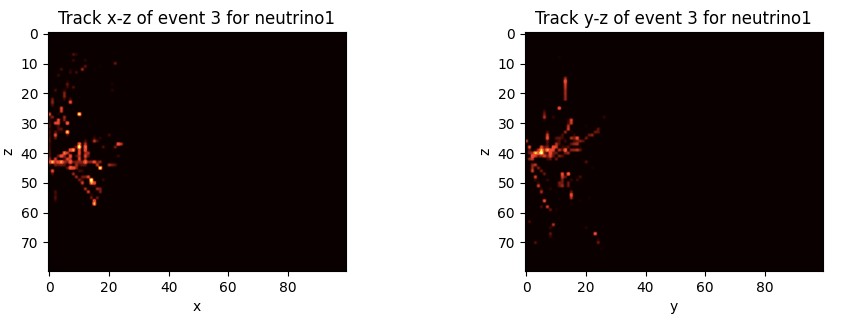
The NOvA experiment is a particle physics experiment designed to study neutrinos such as the one in *Fig.1*. and their oscillations. Involves the use of a beam of neutrinos produced at Fermi National Accelerator Laboratory (Fermilab) in Chicago, USA. This beam is sent to two detector, one near detector at Fermilab and a far detector in Minnesota, separated by 810 km.

The near detector measures the beam of the neutrino before any possible oscillation, while the far detector measures it after the possible oscillation. Comparing these measurements, the change on neutrino flavour can be studied.

NOvA measures the 750-kW beam of mainly neutrinos produced by NUMI beam line, whereas the far detector measures both electron neutrinos and muon neutrinos [1].

Neutrinos are notoriously challenging to detect due to their weak interaction with matter and the absence of any electromagnetic field interaction caused by their electrically neutral nature. In fact, neutrinos are considered "invisible" to traditional particle detectors. However, by exploiting the rare interactions that neutrinos do have with matter, detection is possible.

The NOvA detectors utilize a sophisticated detection system consisting of tubes filled with liquid scintillator to enable interactions between neutrinos and carbon nuclei to occur. Resulting in the release of a burst of charged particles that can be visualized. This system is coupled with wavelength-shifting fibbers connected to photodetectors are then used to calculate the energy of the particles they come to rest. By analyzing the pattern of light detected, it is possible to determine the energy and type of neutrino responsible for the interaction.



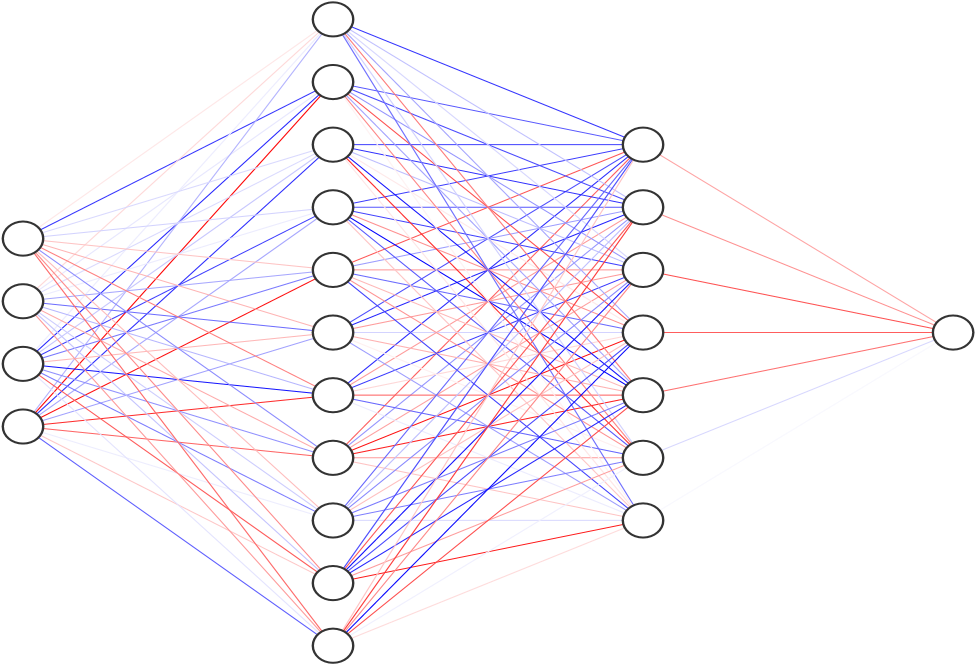
*Fig.1. Neutrino event from dataset*

## 1.2 Neural Network

*This section discusses neural networks theory.*

A neural network is composed of multiple mathematical operations and linear algebra [2].

Each layer contains 𝑛 neurons, two hidden layers in *Fig.2.* with 11 and 7 neurons respectively. Each neuron of one layer is connected to the following layer’s neurons. These connections are governed by weights*.* These are relevant since it determines the strength of connections. Each layer also has a bias (an extra neuron with no connections and holds a single numerical value).



*Fig. 2. Neural network architecture with 2 hidden layers.*

Data from the input layer passes through the hidden layers until it reaches the output layer.

Equation (**1**) defines the weighted sum that is passed along the network.

**𝜔** is the weights of each enlace to the neuron, **𝑥** is the value of the connected neuron from the previous value. 𝛽 is the added bias. **𝛿** is the output, **A** is the activation function that brings non-linearity to the system and **n** the number of neurons in a layer.

The network starts with random weights and biases. During training it ‘learns’ to correct the weights and biases and adjusts the network accordingly.

## 1.3 Convolutional Neural Network

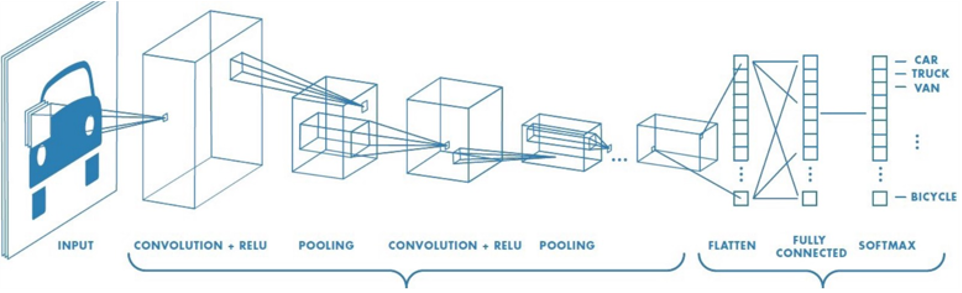
*This section discusses convolutional neural networks theory.*

Convolutional neural networks are best algorithm for computer vision problems. Feature extraction allows to detect complex patterns. These are detected locally, using only a part of the data (image) as input on a layer.

The main layers are convolution, pooling, and fully connected, as in *Fig. 3*.

The convolution layer computes the dot product of a matrix filter and a part of the data. Sliding through the whole input and producing a feature map that detects patterns from the image.

The main advantage that comes from this architecture is that feature detection is spatially independent, this improvement is because the filters are applied to all the parts of the image rather than to the whole image data.



*Fig. 3.*  *Convolutional Neural Network visualization [3].*

The dimensionality of the feature matrix is reduced by the pooling layers. This reduces the number of parameters and therefore the risk of overfitting by aggregating the neighbour pixels into a smaller feature map as seen in *Fig. 3*.

The final Dense layers on CNN’s are preceded by a flatten layer that flattens all the data into one dimension. Then passed to the fully connected layers that sum the weights of the input and applies an activation. These layers combine the high-level features learned in the convolutional and pooling layers to make a prediction on one of the possible final outputs.

## 1.4 Implementation on NOvA like dataset

*This section discusses the dataset used (NOvA-like data)*

The main task of the project was to classify neutrino interaction images from the NOvA-like dataset [4] using a complex CNN architecture into charged current (CC) muon neutrino and other interactions.

The NOvA experiment discussed above makes reconstruction of these interactions using a sophisticated detection mechanism for the oscillation of neutrinos (flavours of neutrinos changes spontaneously). The CNN extracts feature that could distinguish in between the different interaction types, including:

* 𝝂𝝁 **CC** Interaction between a muon neutrino and a nucleus.

Characterized by the production of a muon, which typically produces a long, low dE/dx track as it passes through the detector.

* 𝝂𝝉 **CC** Interactions are characterized by the production of a tau lepton, which can travel a short distance before decaying into other particles.

These interactions are characterized by the presence of a shorter, more diffuse track that terminates in a shower of particles.

* 𝝂𝒆 **CC** Production of an electron, which may produce a more compact, high-energy shower of particles in the detector. In some cases, multiple electrons may be produced, resulting in multiple showers.

Further division of interactions, CC interactions can be divided in Quasi-Elastic (QE), Resonant (RES), and Deep-Inelastic Scattering (DIS) categories which vary in visual features, complexity, and presence on the dataset.

In QE events, neutrino interacts with a nucleon in the nucleus and produces a charged lepton (muon or electron) and a recoiling nucleon. Which remains intact after the interaction and recoils away from the lepton.

For RES events, the incoming neutrino interacts with a nucleon in the nucleus and excites it into a resonance state (excited state of the nucleon). Then decays into other particles, for example a charged lepton and/or additional particles.

Whereas in DIS events, the incoming neutrino has enough energy to interact with quarks inside the nucleons mediated by the weak force in the nucleus, that breaks apart in the process. Potentially producing a wide range of final state particles, including charged leptons, hadrons, and other particles.

As explained in *FAQ for PHAS0056 Neutrino Event Classification mini project* [5]. A simpler description could be such as this:

* QE: Clean event, normally just two tracks
* RES: Something in the middle
* DIS: Messy event potentially many tracks and showers

# 2. Methods

The primary objective is to develop a machine learning classifier capable of accurately identifying 𝜈𝜇 charged-current events. In addition, the efficiency of the classifier will be evaluated by analyzing how it depends on the various meta data variables provided. The project also includes several potential extensions, including the development of algorithms for determining the energy and flavour of the neutrino, as well as the lepton energy over neutrino energy and interaction mode. The data used for this project will primarily consist of images of neutrino interactions from NOvA.

## 2.1 Binary Classifier (TASK I)

*This section discussed the first network a binary classifier using an inception module.*

The first task was to build a binary classifier that accurately determine whether a neutrino interaction image from the NOvA dataset corresponds to a CC or to another type.

### 

### A) PREPROCESSING

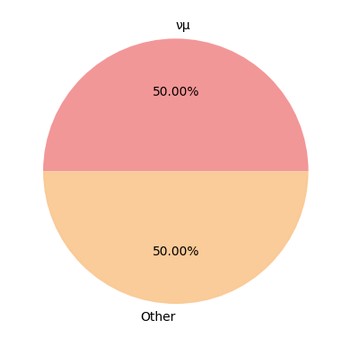
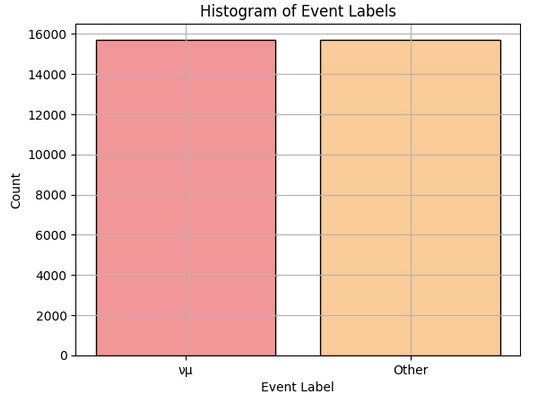
For that purpose, after extracting the images and interaction labels from the dataset. A simplification was performed to ensure that the classifier was binary and distinguish in between CC and others.

Any interaction label from the data smaller or equal to 3 would be converted to a Boolean ‘True’ and any other label to ‘False’. Ensuring correct target values for classifier.

After inspecting the data classes, it was inferred that there was class imbalanced on the data extracted from the NOvA experiment and to avoid any biased problems, since in training this would be perceived as an improvement of accuracy, when making prediction it would lead to many incorrect ones and accuracy would not eb a proper measure of the model performance.

To address the issue under sample was performed to reduce the dominant class, sacrificing some images from the data, but since the dataset was large enough this would not raise any complications in overfitting.

The result of under sample can be appreciated in *Fig.4* where both the percentages and the number of items in the data are illustrated.



*Fig.4. Under sample results on data*

Continuing with the methodology, to ensure that the data fed into the machine learning algorithm is of high quality, normalization techniques were employed to ensure a better data was feed to the network. Specifically, contrast normalization was applied to the images to improve the signal-to-noise ratio and enhance the features relevant for identifying the CC events. This normalization technique adjusts the contrast of the images so that the minimum pixel value becomes zero and the maximum pixel value becomes one, while keeping the mean and standard deviation of the pixel values unchanged. This allows for a more efficient and accurate training of the machine learning model.

With the aim of flagging any possible overfitting and evaluate the model the data was split into training, validating, and testing sets.

The training set was used to train the machine learning classifier on how to distinguish between CC events and other types of neutrino interactions. The validation set was used to optimize the classifier's hyperparameters and prevent overfitting. Finally, the testing set was used to evaluate the performance of the classifier on new, unseen data. Finally, the ratio (0.64, 0.16, 0.2) of these three sets was chosen carefully to ensure a sufficiently large training set while still allowing for reliable evaluation of the classifier's performance.

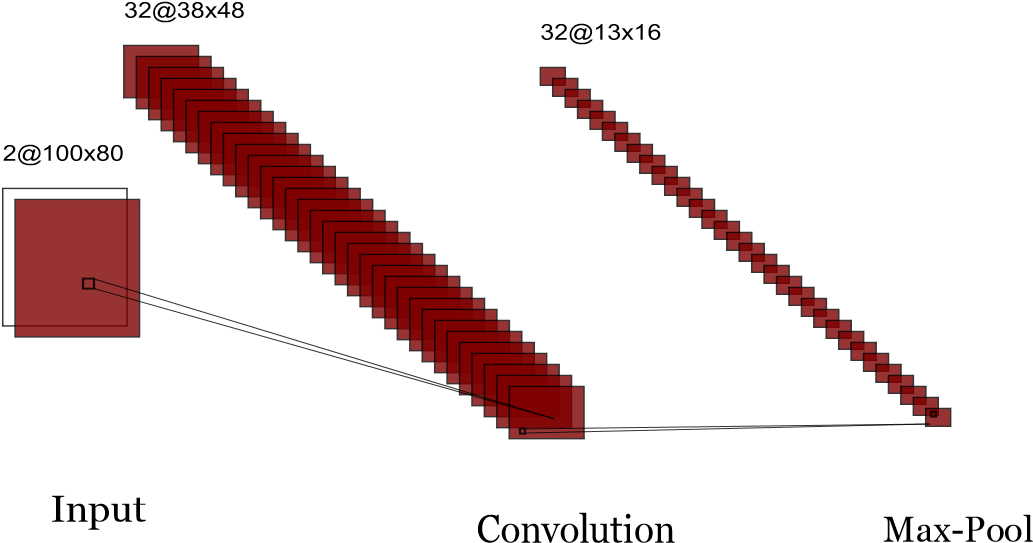
### B) MODEL ARCHITECTURE (BI-CLASSIFIER)

The development of an effective machine learning classifier for identifying neutrino events relies not only on a high-quality dataset but also on the appropriate model architecture. An inception module was employed in this project, inspired in the network design by Fermilab [6], to construct a machine learning model for identifying CC events. The inception module can be extremely useful and efficient, this kind of network is characterized by exceptional performance in computer vision tasks.

The objective is to utilize the unique capabilities of the Inception module to enhance the identification efficiency of CC events and gain insights into the properties of neutrinos and their interactions with nuclei.

The input to the model were 100x80 pixels image, which passes to a Separable

Convolution of 32, with filter size of 7x7, to then be followed by a Max-Pooling layers (2x2 filter) with a stride of 2 to reduce dimensionality and a 0.4 Spatial Dropout to avoid overfitting, similar to simple Dropout layer however, entire 2d feature maps are dropped instead of only neurons to promote independence between feature maps. These first layers oversee extracting the most general features of the image. The separable convolutions as illustrated in *Fig.5* perform a separate spatial convolution on individual channels followed by a pointwise convolution mixing the resulting output channels.



#### Fig.5. Separable Conv block

Continuing with another convolution of kernel size 3 and Max-pooling like before and another Spatial Dropout layer the data is feed to the inception module, this is the most important section of the architecture model.

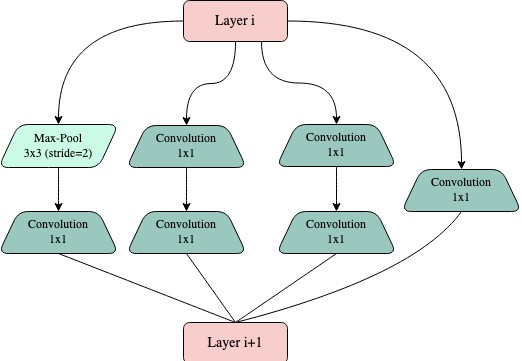
**Inception Module:**

The inception module is a frequently used building block in neural networks that enhances both efficiency and performance by incorporating multiple filter sizes in a single layer.

The feature map obtained from the output of the previous convolutions and pooling layers is then passed to three different Convolutional layers with ReLU activation function, along with a Max-Pooling layer having a 3x3 filter.

Following the first Convolution layer, no additional layer is added. However, after the second Convolution layer, another Convolution layer with a larger 3x3 filter is added. Before the third Convolution layer, a preceding Convolution layer with a kernel size of 5 is introduced. After the Max-Pooling layer, a Convolution layer with a kernel size of 1 is added.

Subsequently, the output of each of the four channels is concatenated to form a single output layer as seen in *Fig.6.*



*Fig.6. Inception block*

Two inception modules are utilized for the model, with same number of filters for the layers specified above. The inception blocks are constructed with 64, 96, 16 and 32 for the three convolutions and the pooling layer respectively.

The feature map resulting from the second inception block is continued by a third Convolution with 64 filter of size 3x3 and reLU and a normal dropout layer of 0.7. Then the output is flattened and fed to multiple fully connected layers of 128, 64 neurons intercalated by more dropout layers. These last dense layers use also reLU and are followed by an output layer with sigmoid activation of 1 neuron since the task is binary.

### C) TRAINING AND VALIDATION

The model was trained on a part of a NOvA like dataset.

Using a consider able number of images for training 20116, and 5029 for validation. The model was trained using Adam as optimizer with a learning rate of 0.001 to ensure convergence and measuring the Binary cross entropy loss and the Binary accuracy after every epoch for training and validating. The architecture was trained in batches of 100 during 35 epochs.

After training the accuracy and loss achieved are illustrated in *Fig.7*. However, since many other networks were tested to determine the most optimal architecture a confidence matrix was used to find the number of true positives, false positives and so on as illustrated in *Fig.8*. The best result was achieved with the architecture described above.

#### 

Gráfico

Descripción generada automáticamenteGráfico, Gráfico de rectángulos

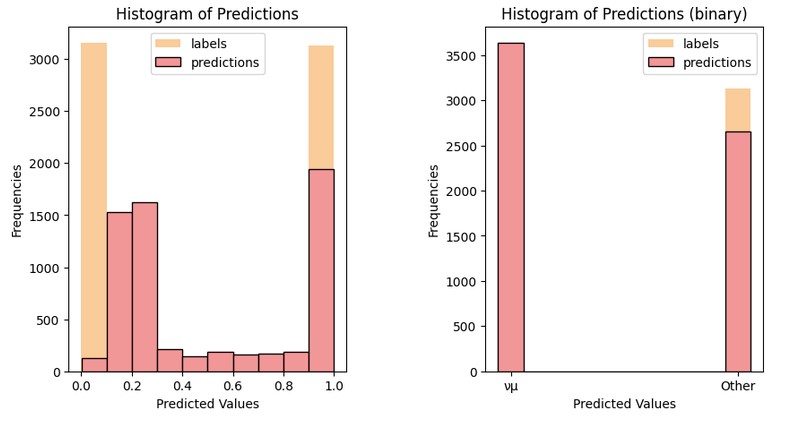
Descripción generada automáticamente

*Fig.7. Training loss and accuracy Fig.8. Confidence matrix*

## 2.2 Testing Performance (TASK I & II)

*This section discusses the performance evaluation done to the model.*

The target labels were predicted using the testing data and the confidence scores were stored for further comparison with the actual labels (binary). First, two histograms were generated by comparing the predicted labels (along with their associated certainties) with the actual certainties of the test data, as shown in *Fig*. 9. One histogram provided detailed information, while the other was truncated, excluding predictions with a certainty greater than or equal to 0.5 and any probability less than 0.



*Fig.9. Histogram of predictions*

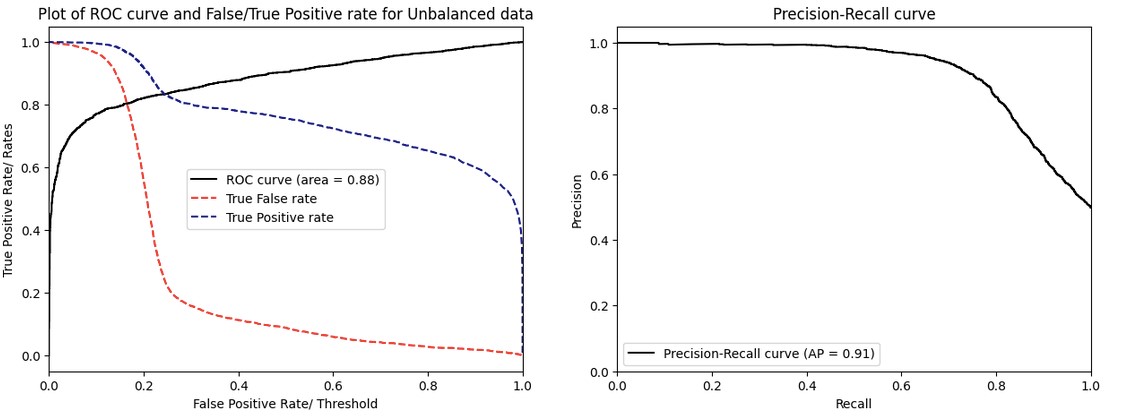
To further investigate the performance and evaluate the model's ability to accurately classify positive and negative samples. Receiver Operating Characteristic (ROC) and precision-recall curve plots were made using the predictions.

The ROC curve plots the trade-off between the false positives and the true positives rate for different probability thresholds, the value that truncates the probabilities to one class or the other. Whereas the precision-recall curve shows the precision (positive predictive value) and recall (sensitivity) for different thresholds.

Employing these plots, it was possible to illustrate the performance of the predictions of the model.

The ROC curve as seen in Fig.10. illustrate a satisfactory classification performance as it was close to the top-left corner. However, the non-symmetric shape of the curve showed a faster increase in the false positive rate than the true positive rate, indicating an overall bias in the classifier towards predicting false positives.

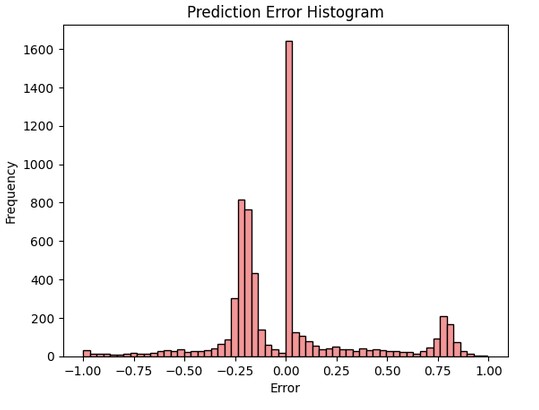
The precision-recall curve shows a similar trend as expected, a high precision at the expense of a lower recall, indicating that the model performed better at predicting the true positives than avoiding false negatives.



*Fig.10. ROC curve and PRC*

Finally, the last plot that was made for performance reasons was a histogram of the prediction errors, which shows that the most frequent error is of the order of magnitude of

0.2 which is a positive result as shown in *Fig.11*



*Fig.11. Error magnitude count*

## 2.3 Testing Meta-Data (Task II)

*This section discusses the testing of the model for the other meta-data variables.*

### A) INTERACTIONS

Next to test our model performance on different meta-data. First the analysis of the performance on the energies was done. Evaluating the model on the different interactions, QE, DIS, RES and Other, the predictions were made using a threshold of 0.5 as before, the corresponding accuracies for each interaction are represented in the histogram below *Fig.12*. Where for QE the model achieves an accuracy of 0.92, in DIS interactions 0.76, whereas for RES an accuracy of 0.83 and finally for other interactions a value of 0.89 is achieved.

This was expected since QE events are a clearer two track interaction as stated above, compared to DIS which are messier with disperse tracks (showers). RES events are clearer and less dispersive than DIS but not as clear as QE events, therefore the accuracy for these events makes sense to be in between them.

Evaluating in balanced data the result was quite optimal as illustrated in *Fig.13* where the accuracies are similar.

Gráfico, Gráfico de barras

Descripción generada automáticamente Gráfico, Gráfico de barras

Descripción generada automáticamente

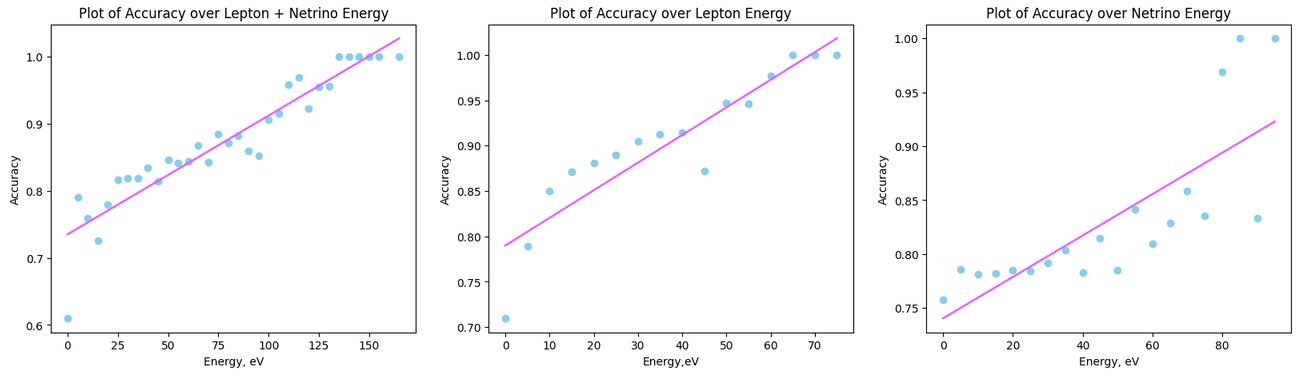
*Fig.12. Different interactions accuracies* *Fig.13 Accuracy on even data*

### B) ENERGIES

For the purpose of further studying and testing other meta data, the different energy data was tested, lepton and neutrino energy.

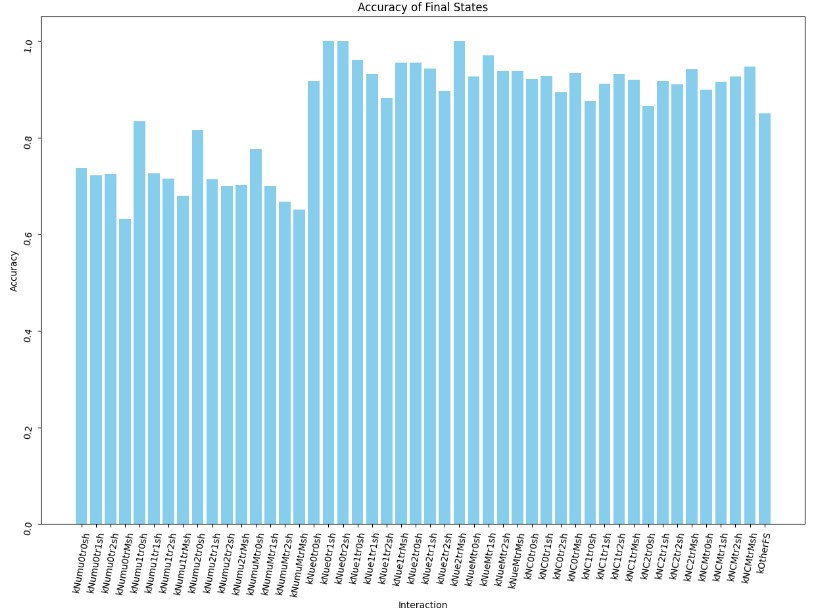
With a simple calculation it was decided to group the energy values in bins of size 5 eV since it would make it easier to plot and wouldn’t sacrifice much the visualization accuracy.

Using a threshold 0f 0.5 on the predicted values the accuracy was calculated. The main goal was to visualize how accuracy of the model changes for low energy neutrinos compared to high energy ones and the same with lepton energy. The plots from *Fig.14*. illustrated that accuracy improved directly with energy, the higher the energy the higher the accuracy. The reasons behind this correlation could be related that the higher the energy the clearer (brighter) the tracks are, and higher energy would make the interaction last longer and the tracks would be larger and easier to identify. It can be concluded that energy is directly correlated to the visual topology of the images and therefore high energy events are easier to classify.

 *Fig.14 Energies accuracy*

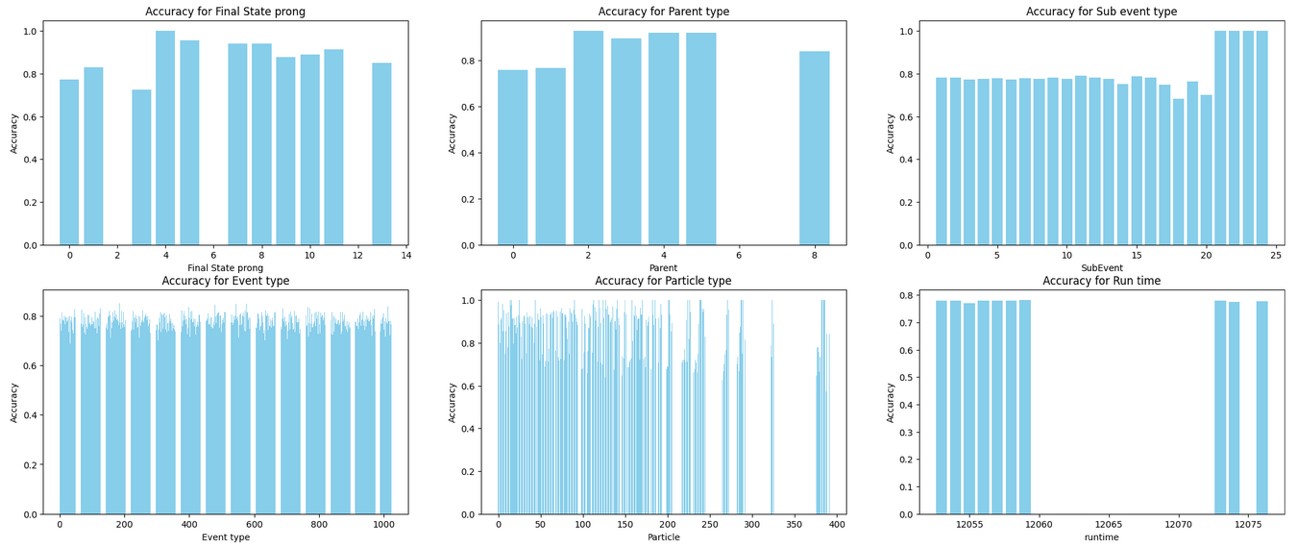
### C) META-DATA

Trying to find if the model was correlated to any other meta data variable, predictions using accuracy were performed on almost every meta-data key of the original data, excluding the “vtxx”, “vtxy” and “vtxz”. The final states could potentially be an important metadata variable but as illustrated *in Fig.15*. muon final states have an average accuracy of 74% compared to the other final states which have average accuracy of 88%. This could be because there are many muon interactions recorded, resulting in more overlaps with other interaction types that lead to misclassification. Additionally, muons tend to make long, straight tracks, which can also lead to misclassification with other states. Therefore, the model's performance can be affected by the type of interaction being classified.



*Fig.15. Interaction meta-data accuracy*

Finally, to finalise this part other meta data was tested as seen on *Fig.16*. Maybe the most illustrative of a possible relation is ‘subevt’ key. Where for sub-events from 20, accuracy peaks and there is a step up in an accuracy for those values.



*Fig.16. Meta-data accuracy*

## 2.4 Extensions

*This section discusses all the extension of the project.*

### 2.4.1 Regression CNN for neutrino energies

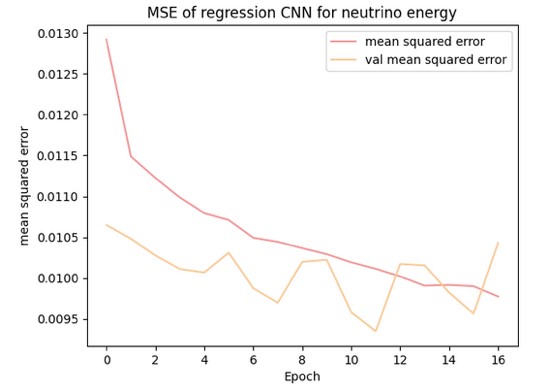
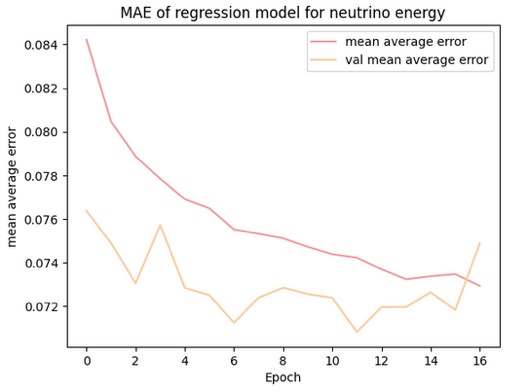
In this first extension the goal was to determine the neutrino energy of neutrino after training over the images of the NOvA data and with the energies as target.

The first step was to download new data to avoid any possible issue since the previous data was altered and prepared. Using a new function for downloading efficiency. Since a new network would be implemented because we are dealing with another type of task, regression rather than classification. The images input was split into x-z and y-z view to feed the network with to input channels. The target labels were the energies.

Data is split into evaluation/training/testing and further normalized using scaling technique, rescaled to a range of 0-1 by dividing each pixel by the maximum value in this case 255. Continuing with preparation of data, the inputs are expanded, adding an extra dimension of size 1 to the input tensors to match the input expected by the regression CNN model.

The model used for this task was a CNN architecture. Two branches of data are feed to the model, staring with a combination of convolutions with 16,32,32 filters and kernel size of 3x3 employing reLU as the activation function. These layers are intercalated by Max-Pooling layers of 2x2 filter. Flatten and passed through a Dropout layer of 0.4 parameter to avoid overfitting. Concatenating the channels to finalize the regression with two Dense layers of 32, 16 and 1 for the output layer, since only one energy is calculated.

During training utilizing Adam optimizer, the performance is evaluated by measuring the difference between predicted and actual values using two commonly used metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE calculates the average squared difference between predicted and actual values, while MAE calculates the average absolute difference between predicted and actual values. Results can be appreciated in *Fig.17*.

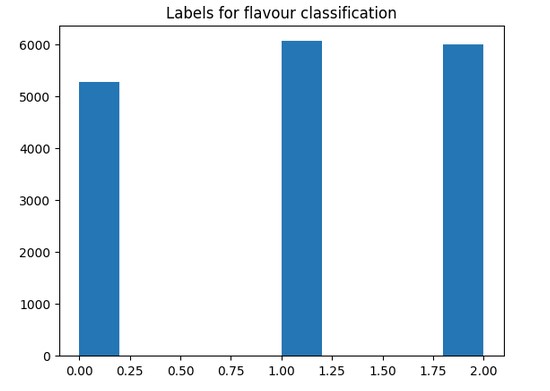


*Fig.17. Training metrics for first extension (energy regression)*

### 2.4.2 Neutrino Flavour Classifier

Second extension goal was to build an algorithm that can potentially determine the flavour of the neutrino given the same type of images.

To obtain even distributed data, a large set of data was analyzed and further divided by class distribution, creating new data while maintaining balance. Instead of under sampling, iterating different parts of the data was preferred due to computational efficiency. The resulting data was classified into muon neutrinos, electron neutrinos, and other, with even distribution as shown in *Fig 18*.



*Fig.18. Labels count flavour*

Data was shuffled to avoid biased, expanded, and split. Then feed to the network, which in this case was a CNN with over 579000 trainable parameters, with two convolution blocks as shown in *Fig.19*.

Diagrama

Descripción generada automáticamente

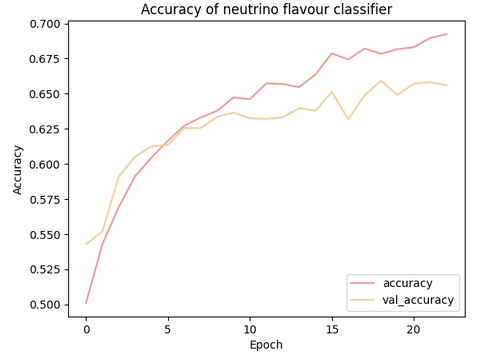
*Fig.19. Convolution block*

The architecture finishes with a 3 neurons dense layer to match the required classes.

Employing categorical cross entropy as the loss function to measure the performance with the accuracy as the main metric of the model and making use of Adam for optimization the model is trained on batch sizes of 64.

The results for training can be appreciated in *Fig.20*

Gráfico, Gráfico de líneas

Descripción generada automáticamente 

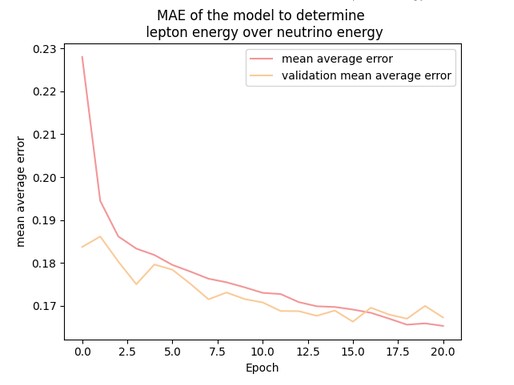
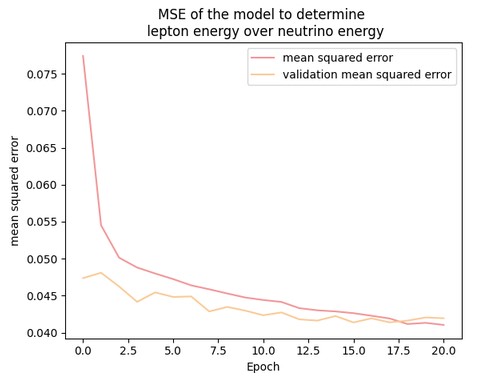
*Fig.20. Flavour classifier training*

### 2.4.3 Algorithm for lepton energy over neutrino energy (y)

To fulfil this extension and determine **y**, lepton energy over neutrino energy with an algorithm. The first step required to find any interaction that could potentially raise mathematical problems or non-valuable data. Therefore, to find any interaction with 0 energy, a large set of data was first investigated.

It was found that interaction 15 yielded 0 energy and cannot be used to find **y** and should be excluded from this extension.

The actual data prepared without interaction 15 was expanded and the energy labels normalized. This was split and feed to the new network architecture. In fact, for this regression task the same regression model from **3.4.1** was employed. After training the results for the mean square error and mean absolute error were plotted and are illustrated in *Fig.21*. Giving very good first results for training.



*Fig.21. Training energy ratio y*

### 2.4.4 Interaction mode algorithm

For this multi class problem of neutrino interaction modes. The classes were, 0 for charged-current (CC) quasi-elastic events, 1 for CC resonant events, 2 in case of CC deep inelastic scattering, 3 for CC other events and finally 4 for neutral current events.

As for the other extensions data was first downloaded and divided into classes in a new array. The same technique as in **3.4.2** was used to even the data since very large sets were being used. Once the classes were corrected the data was almost prepared for the network, splitting the data and further converting labels to categorical to utilize categorical cross entropy.

The data was feed to the network in this form. Such network was composed of two convolution blocks as *Fig.16* and continuing with multiple convolutions and pooling layers to finalize with multiple fully connected layers of decreasing neurons until the output layer with only 5 of these to match the number of classes.

Employing Adam, the model was trained on batch sizes of 64. After training accuracy that was the metric being used to measure its performance gave a not so satisfactory insight with 39% in accuracy and lower in validation.

# 3. Results

To summarize the results for the code were quite satisfactory.

The performance of the bi-classifier was evaluated on the test dataset, and after evaluation an accuracy of 83% was achieved that along with several metrics including the receiver operating characteristic (ROC) curve, the precision-recall curve (PRC), and the error count. The ROC curve showed a high true positive rate (TPR) with a low false positive rate (FPR) indicating that the model can accurately identify positive samples while minimizing false positives. The PRC also showed high precision and recall.

Further analysis on meta data variables showed some possible future implementations to improve the model based on new correlations.

The extension of the project also brought some valuable information and could potentially be used in analysis of these interactions.

The energy model was evaluated and achieved a loss of 0.009.

For the neutrino flavour classifier, the final accuracy was 66% which is a positive result and is enough to give valuable insight.

The energy ratio algorithm performed well also, and the mean square error was 0.05, which is an excellent result.

Finally, the last extension was not as efficient, and the performance was considerably poor compared to the other with an accuracy of 39% during training. The reason for this could be gradient exploding during backpropagation but further study would be needed.

Overall, the results of the project were satisfactory, with the bi-classifier achieving an accuracy of 83% on the test dataset, and other extensions also showing promising results. However, further analysis and future improvements can be made based on correlations with meta data variables.

# 

# 4. References

[1] A. Albert et al., "Search for steady point-like sources in the astrophysical muon neutrino flux with 8 years of IceCube data," New Journal of Physics, vol. 18, no. 1, p. 015009, 2016.

[2] "TensorFlow," TensorFlow. [Online]. Available: <https://www.tensorflow.org/guide>. [Accessed: Mar.19, 2023].

[3] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," Medium, Nov. 16, 2022. [Online]. Available: <https://towardsdatascience.com/acomprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed: Apr. 07, 2023].

[4] "Neutrino Detection with the NOvA Experiment," UCL. [Online]. Available: <https://www.hep.ucl.ac.uk/undergrad/0056/other/projects/nova/neutrino>. [Accessed: Mar. 03, 2023].

[5] "Mini-project-neutrino," Moodle. [Online]. Available: <https://moodle.ucl.ac.uk/login/index.php>. [Accessed: Feb. 07, 2023].

[6] C. Szegedy et al., "Going deeper with convolutions," [Online]. Available: <https://arxiv.org/pdf/1409.4842.pdf>. [Accessed: Feb. 18, 2023].