Neutrino Event Classification

Louis S Sharma

University College London

January 2022

Abstract

Neutrinos are elusive particles which only rarely interact with matter. The NOvA experiment at Fermilab studies neutrino oscillations, which occur when a neutrino changes flavour in flight. This phenomenon is not accounted for by the standard model of particle physics. In this paper, an attempt is made to classify images from a hypothetical neutrino detector to predict whether the interaction represented by the image is a ν_μ charged-current interaction. A two-branch convolutional neural network followed by a series of dense layers is used for this task. In the first section, the methods and findings for gathering, pre-processing, building, training and evaluating the model are discussed. Then, the classifier's performance is investigated for some of the image metadata.

Contents

1 Introduction

Neutrinos are particles that are electrically-neutral and almost mass-less. They fall into the lepton family of particles. As of now, there are three different known neutrino 'flavors', where neutrinos are paired with a charged lepton: the electron $e^- : \nu_e$, the muon $\mu^- : \nu_\mu$ and the tau $\tau^- : \nu_\tau$. Neutrinos are invisible to particle detectors and can only be detected when they interact (very rarely) with material in the detector through *weak* processes. These fall into two categories: *charged-current* events, where the neutrino is turned into its charged partner (EG: $\mu_e + \rightarrow e^- + p^+$) and *neutral-current* events in which the neutrino is scattered but not converted [1].

Fermilab's NOvA experiment [2] aims to study neutrinos, which are known to exhibit a quantum mechanical behavior called neutrino oscillations (neutrinos changing flavor in flight). This research is highly important because neutrino oscillations are the only process known to *break* the standard model of particle physics [3]. In the NOvA experiment, a beam of muon neutrinos is directed to a 'far' detector located in Minnesota. The detector is made with tubes filled with liquid scintillator, an organic compound mainly comprised of carbon atoms. The images thus represent the interaction between neutrinos and carbon atoms. Each doublet of images corresponds to a different view of the same event.

2 Data Gathering and Pre-processing

In this section, the methods used for gathering and pre-processing the data are discussed, such as image normalisation, splitting the data into training, validation and testing sets and under-sampling to balance the distribution of classes in the training data. Though it may not be the most exciting part, data cleaning is a key step in any machine learning/data analysis task and must not be overlooked.

2.1 Introduction to the Data

The data consists of doublets of images of simulated neutrino interactions in a hypothetical detector which resembles the detectors of the NOvA experiment. For each interaction, there are two 100×80 images, representing a view of the same event from the $x \times z$ and $y \times z$ planes.

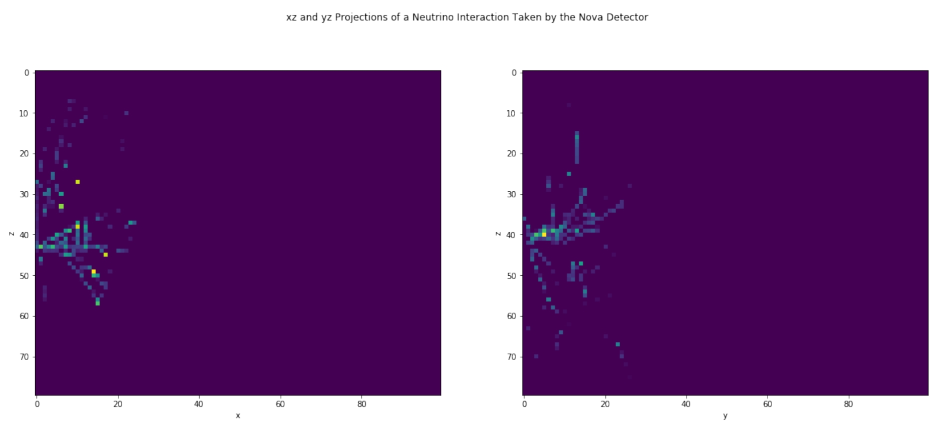

Figure 1: $x \times z$ (left) and $y \times z$ (right) Image Projections of a Neutrino Interaction Taken by the NOvA Detector

For each interaction, there are 16 labels representing various features, namely, the type of neutrino interaction each doublet of images represents, the lepton and neutrino energies and the velocities along different planes.

2.2 Data Gathering

For this project, 15 '.h5' files were gathered from the UCL High Energy Physics page. This corresponds to $\approx 10^5$ neutrino interaction events. On one hand all the image data was gathered and stored into a tensor $\mathbf{X}$ of shape $(N, 100, 80, 2)$ with N being the number of observations. All the label data was stored in a Pandas DataFrame object.

2.3 Data Pre-processing

2.3.1 Image Normalisation

The image pixels of $\mathbf{X}$ were all re-scaled to $[0, 1]$ by applying the following transformation:

$$X_{ijkl} = \frac{X_{ijkl} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})} \tag{1}$$

It is considered standard practice to apply some sort of normalisation to deep learning input data because it is known to speed up and stabilise the learning process [4].

2.3.2 Train Validate Test Split

The images $\mathbf{X}$ and the label metadata were then split into training, validation and testing sets. Using a validation set gives room to fine tune certain parameters of the model after it has been trained. It was decided to opt for a 80% 10% 10% training, validation and test size respectively. This is because we would like to use the maximum amount of data for training.

2.3.3 Interaction Label Data

A label **y** array for the binary classification task discussed was then created. In the Pandas DataFrame containing the image metadata, the ν_μ charged-current events are represented by values ≤ 3 in the 'interaction' column. Therefore, to create **y**, the following was used: suppose I is the set representing the interaction column.

$$\forall x \in I, y : I \to \{0, 1\} : \begin{cases} y(x) = 1, & \text{if } x \leq 3 \\ y(x) = 0, & \text{if } x > 3 \end{cases} \tag{2}$$

The elements of this array are thus 1 if the event is ν_μ charged-current and 0 otherwise. This function was then applied on the interaction data to create label arrays for the training, validation and testing sets.

When analysing the distribution of ν_μ charged-current events in the training data, it became apparent that the classes were heavily imbalanced, with about 88% of observations representing ν_μ charged-current events.

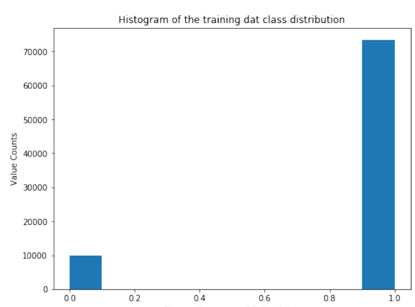

Figure 2: Histogram of the Training Data Class Distribution

Since most binary classification algorithms assume that the data is evenly distributed [5], this issue could cause potential models to ignore the minority class and only predict the majority class in order to maximise the accuracy. To circumvent this, the training data was under-sampled to have a $50-50$ class distribution for training. Under-sampling is a simple method which consists in getting rid of excess majority class data. However, it is not adapted to cases where data is not widely available. Since there are copious amounts of data available for this task, this method was considered appropriate.

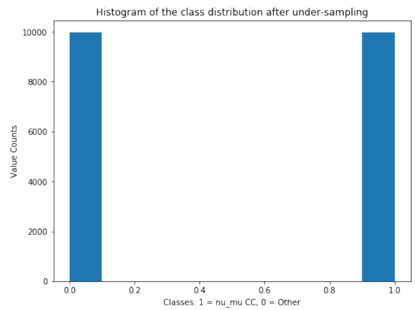

Figure 3: Histogram of the Under-Sampled Training Data Class Distribution

3 Classifying ν_μ Charged-Current Events

3.1 Model Architecture

For this task, a multi-input neural network was chosen. This is because the input data consists of two images. The model has two branches, one for each projection ($x \times z$ axis and $y \times z$ axis) of the event. Each image goes through a 2D convolution layer with a ReLU activation and a 2D max pooling. Both images are then flattened and concatenated into a vector connected to seven dense layers. Since the model is quite complex, with about 4×10^6 trainable parameters, it is prone to over-fitting. To combat this, dropouts were added with values ranging from 0.1 to 0.4. This randomly deactivates certain inputs, preventing the network from becoming too reliant on certain neurons. The final layer contains one output with a sigmoid activation function:

$$f : \mathbb{R} \to [0,1] : f(z) = \frac{1}{1+e^{-z}} \tag{3}$$

Which guarantees that every doublet of input images is mapped to a number between 0 and 1, representing the probability of the interaction being a ν_μ charged-current event.

The model was then compiled. The loss function chosen for this particular task is the binary cross entropy defined as:

$$L(N, \hat{y}_i, y_i) = -\frac{1}{N}\sum_{i=1}^{N} y_i \ln \hat{y}_i + (1-y_i)\ln(1-\hat{y}_i) \tag{4}$$

Where $\hat{y}_i$ and y_i represent the predicted and actual class of the ith pair of images out of N total doublets respectively. Since its gradient scales quickly, The binary cross-entropy loss function is better at penalising the model for incorrect predictions during the backpropagation [6] phase than more commonly used functions such as the mean square error. Larger gradients also allow gradient-descent based algorithms to progress quicker in their search for minima.

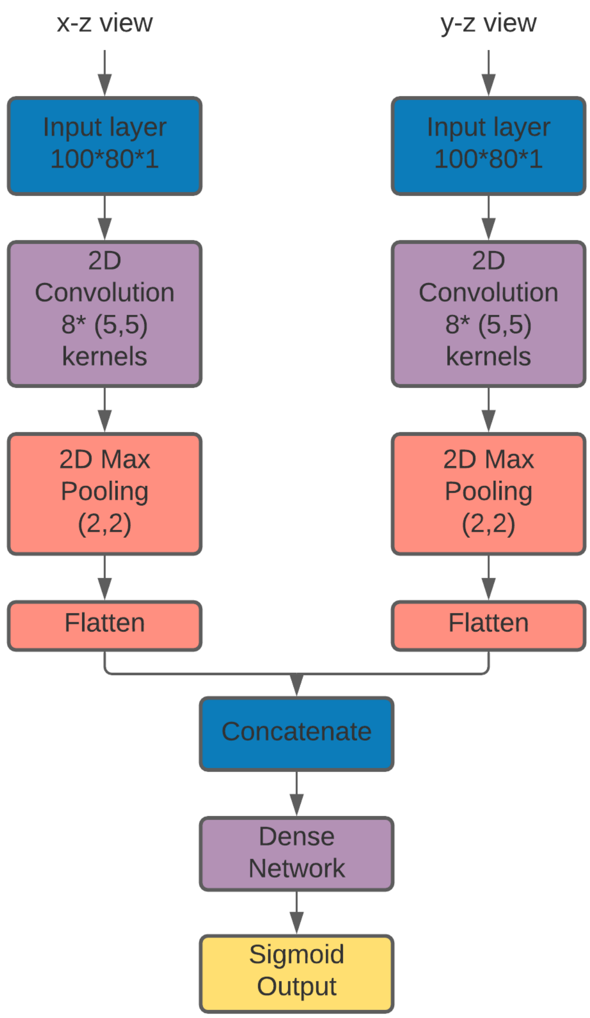

Figure 4: Outline of the Model's Architecture

The model was then trained over 15 epochs with a batch size of 500. This task was rather computationally expensive with run time of about 90s per epoch (without any GPU). Previous attempts to build simpler models with less dense layers resulted in the classifier failing to identify the non ν_μ charged-current events and getting stuck at around 50% training accuracy. A checkpoint was created to save the version of the model with the minimum validation loss.

3.2 Testing and Evaluation

3.2.1 Accuracy and Loss

Overall, the training loss steadily decayed whilst the training binary accuracy increased to $\approx 85\%$. The validation loss decreased during the early epochs of training but began to stagnate after ≈ 10 epochs. Similarly, the validation binary accuracy converged to $\approx 75\%$. On the testing data, the classifier yielded similar results (73.6% accuracy).

Figure 5: Training and Validation Loss and Accuracy as a Function of Epochs

3.2.2 Confusion Matrix

The confusion matrix is metric for assessing the efficiency of a classifier. Its diagonal entries represent the correct predictions (true positives and true negatives) whilst the off-diagonal entries represent incorrect predictions made by the model (type I and type II errors).

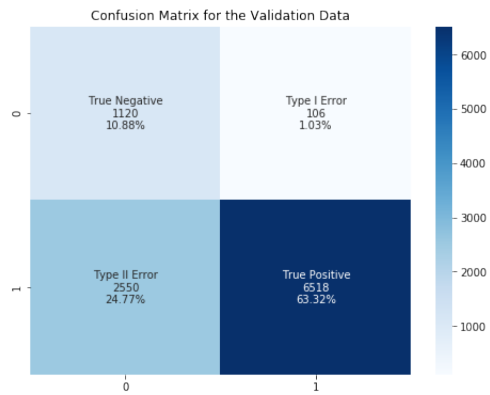

Figure 6: Confusion Matrix for the Binary Classifier on the Validation Data

The confusion matrix indicates that the model struggles to recognise some of the ν_μ charged-current events from the images. Indeed, the classifier has made at lot of type II errors, with $\approx 25\%$ predictions on the validation data being false negatives. However, it is successful in identifying ν_μ charged-current events and outputs a low type I error rate ($\approx 1\%$).

3.2.3 Matthews Correlation Coefficient

The Matthews Correlation Coefficient (MCC) or Phi Coefficient is another metric which can be used to evaluate a binary classifier. Whilst the accuracy gives us information on the amount of correct positive **and** negative predictions made by the classifier, its results can be heavily skewed if one class dominates the validation set. Another popular metric for assessing the efficiency of a binary classifier is the F1 Score which accounts for class imbalance but does not penalise the model for incorrect negative predictions.

Let C_{ij} be the entries of the confusion matrix, with the rows representing the true classes and the columns the predicted classes. We can define the accuracy A, F1 score F_1 and MCC, ϕ as follows:

$$A = \frac{\sum_{i=j} C_{ij}}{\sum_{i,j} C_{ij}} \tag{5}$$

$$F_1 = \frac{C_{11}}{C_{11} + \frac{1}{2}(C_{01} + C_{10})} \tag{6}$$

$$\phi = \frac{C_{11}C_{00} - C_{01}C_{00}}{\sqrt{(C_{11}+C_{01})(C_{11}+C_{10})(C_{00}+C_{10})(C_{00}+C_{10})}} \tag{7}$$

The formula suggests that the MCC condenses information on the classifier's performance into one statistic. Therefore, it will only produce a high score if the model correctly predicted a high amount of positive **and** negative images, regardless of how balanced the data is, making it a more meaningful metric since the validation data presents an 88% - 12% imbalance see (2) [7]. The MCC is such that $|\phi| \leq 1$ and a value of 0 indicates that the model is no better than flipping a coin. We see that if $C_{01} = C_{10} = 0$, $\phi = 1$ (i.e the model making no Type I or Type II errors), so a value of 1 implies a perfect positive correlation, meaning that all the data falls along the confusion matrix's diagonal. In fact, the MCC is a special case of the Pearson Correlation Coefficient and can be interpreted in the same manner.

Computing this statistic for the confusion matrix above resulted in a value of 0.43, indicating a *moderate* correlation between the predicted values and the truth. This reflects the classifier's high false negative rate.

3.3 Optimising the Probability Threshold to Reduce the Type II Error Rate

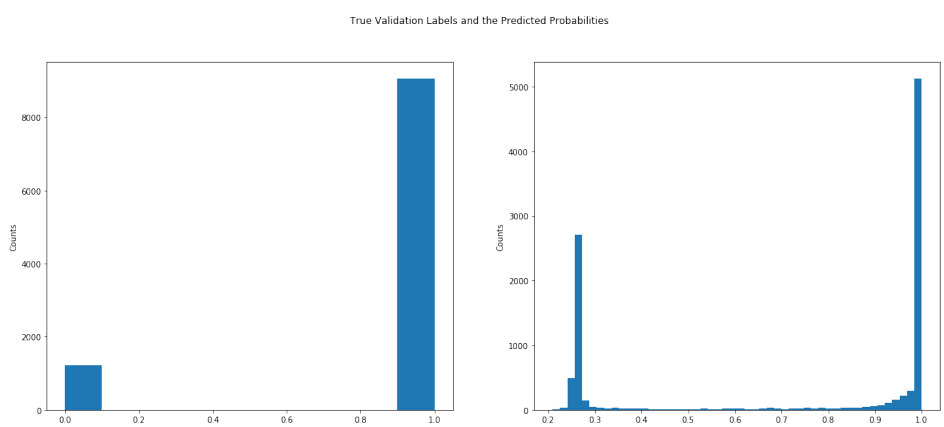

Figure 7: Output of the Classifier on the Validation Data (left) and True Values

The above plot of the distribution of the predicted probabilities output by the classifier indicates that it is able to distinguish between the two classes because there is a large peak of values close to 1 in probability, and smaller cluster of probabilities near 0 with very few values lying in between. It can also be inferred that the classifier is more confident in its predictions of ν_μ charged-current events, because of the low 'spread' or variance in its predictions close to 1 and less confident in its predictions of non ν_μ charged-current events as the distribution of the predictions close to 0 follow some sort of bell-shaped curve. Perhaps the default probability threshold of 0.5 is not suited for this task. Changing the probability threshold could lower the false negative rate of the classifier. To investigate this, the validation data was first under-sampled. Since the validation set is highly imbalanced (88/12%), searching for the optimal threshold on this data set could yield a value of 0, which would ensure an accuracy close to 90%. The effect of varying the probability threshold from 0 to 1 in steps of 0.01 was then tested on the validation accuracy (for the under-sampled validation set) and the validation MCC (without under-sampling). Both tests yielded the same *optimal threshold* value of ≈ 0.42. This makes sense as lowering the probability threshold would classify less images as 'negative' and could therefore help reduce the type II error rate.

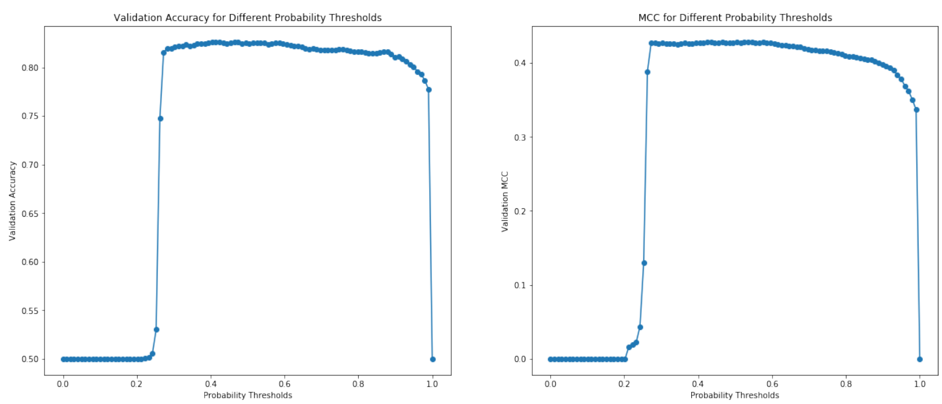

Figure 8: Threshold - Accuracy Curve (right) and Threshold - MCC Curve (left)

When the probabilities output by the classifier on the **testing** images were rounded using this optimal threshold value, no significant improvement in accuracy or MCC was detected.

4 Efficiency of the Classifier as a Function of the Meta Data

In this section, the performance of the classifier will be investigated for some of the image metadata, namely the different types of ν_μ charged-current interaction and the lepton and neutrino energies.

4.1 Types of ν_μ Charged-Current Events

4.1.1 QE, RES and DIS interactions

Since nucleons are not fundamental particles, neutrinos can interact with them in different ways. Therefore, ν_μ a charged current interactions can represent a *quasi-elastic* interaction, (QE), a *resonant* (RES) interaction or a *deep inelastic scattering*. A somewhat simple way of thinking of these events in terms of the images taken by the NOvA detector is the following: QE interactions tend to leave two clean tracks tracks, DIS interactions are often *messy* leaving potentially many tracks and showers and RES events lie somewhere in between [1].

4.1.2 Efficiency of the Classifier for Different Interactions

To examine the performance of the classifier on the different ν_μ charged-current interactions, four new testing sets were created from the original one. Each set included observations for one type of ν_μ charged-current interaction (QE, RES, DIS and Other) and the non ν_μ charged-current events. The reason for removing the observations corresponding to the other three ν_μ charged-current events is quite straightforward: the classifier was trained to identify all four types of ν_μ charged-current events. Inputting images of all four of these interactions into the model and 'telling it' that three of them should be negative would obviously lead to poor results during the testing phase.

The four sets were inputted into the model one-by-one in a for loop and the following metrics were calculated for each type of ν_μ charged-current interaction: accuracy and MCC.

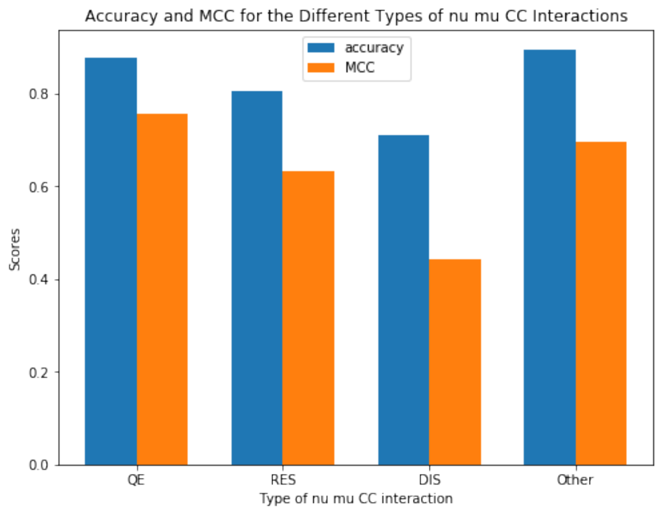

Figure 9: Double Bar Chart of the Classifier's Accuracy and MCC for Different Types of ν_μ Charged-Current Events

The double bar chart implies that the classifier performs best on QE and 'Other' interactions: 87/89% accuracy and 0.75/0.69 MCC respectively. This makes sense because overall, QE interactions are the easiest to identify, often leaving two distinct tracks on the detector. On the other hand, the classifier had the worst performance on the DIS interactions (71% accuracy and 0.44 MCC), which, as discussed above, tend to leave many tracks and showers on the detector.

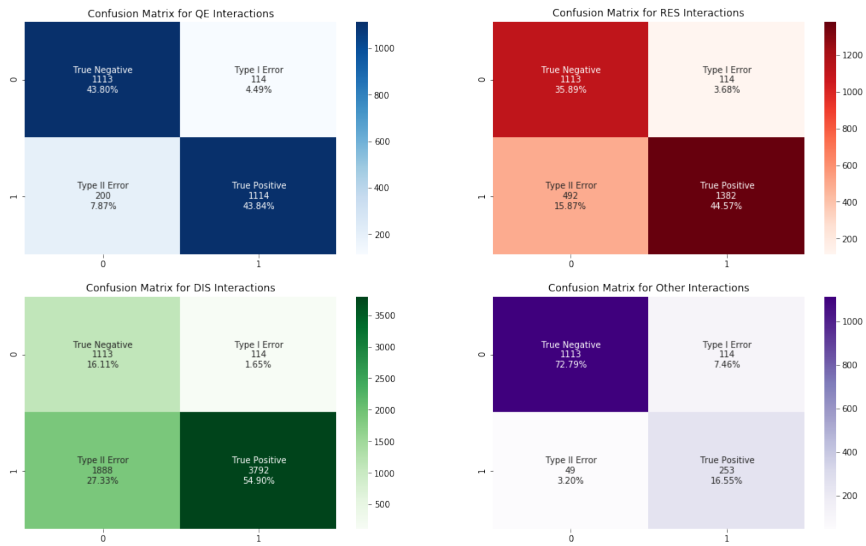

Figure 10: Classifier Confusion Matrices for the 4 Types of ν_μ Charged-Current Interactions

The confusion matrices for each type of interaction further support the double bar chart (see figure 9). For QE interactions, the classifier performed extremely well with a much lower false negative rate, close to 7%. For DIS events, the classifier performed worse than when shown all the ν_μ charged-current events combined.

4.2 Neutrino and Lepton Energies

The neutrino energy refers to the energy of the incoming neutrino. Upon interaction with the material in the detector, it will produce a final state lepton, which can be characterised by its own energy. In the case of neutral-current events, the lepton is the scattered neutrino [1].

The neutrino and lepton energy distribution in the validation data is heavily skewed to the right. Data pertaining to high energy events was very scarce so for this task, it was decided to concatenate the validation and testing sets to maximise the amount of data available.

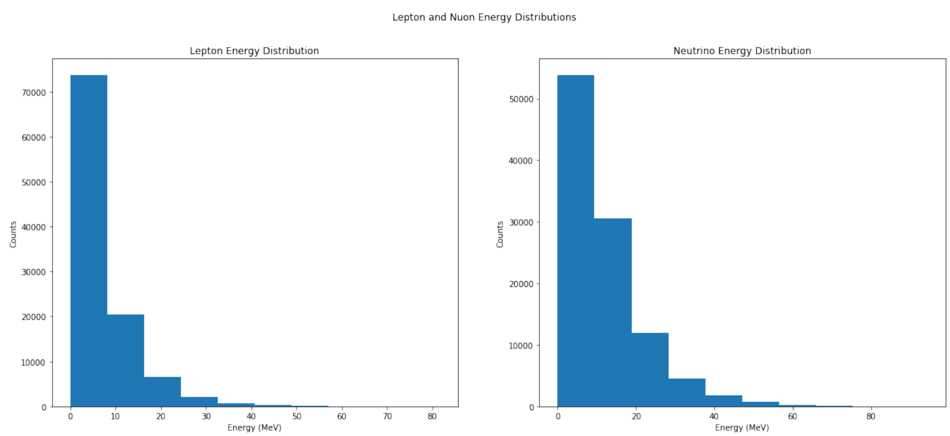

Figure 11: Distribution of Lepton (left) and Neutrino (right) Energies

Both the neutrino and lepton energy data were split into bins of 20 MeV. This resulted in four energy levels ranging from 0 to 80 MeV. The images corresponding to events falling into these discrete energy bins were then fed into the model and the model was evaluated for each each of neutrino and lepton the energy bins.

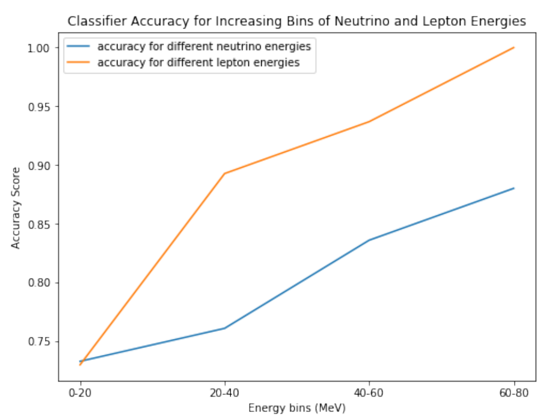

Figure 12: Accuracy of the Classifier for Different Neutrino and Lepton Energy Bins

This analysis implies that the classifier performs better on high energy neutrinos than it does on lower energy ones and high energy leptons than lower energy ones. However, since there is very little data available on high energy interactions, these results could be skewed. In fact, there are only two observations for images corresponding to the highest neutrino energy band but 95 for the second highest bin. Overall, the plot suggests that the accuracy of the classifier increases with the neutrino and lepton energies.

5 Potential Improvements

Increasing the model's complexity by adding more convolutional layers and max pooling layers as well as by adding more training epochs could result in a better performance, but this was omitted in order to keep the model simple and training runtimes reasonable. Perhaps using a couple hundreds of thousands of doublets of images would also have yielded better results but once again, for computational reasons, the data was limited to $\approx 10^5$ pairs of images and metadata.

More effort could've been placed in optimising the neural network's hyperparameters. Techniques include, tweaking the learning rate and setting an l2 regularisation on the loss function

Finally, gathering more images corresponding to high energy neutrinos and leptons would have allowed to better generalise the model's performance on these higher energy bands.

6 Conclusion

To conclude, using the architecture described in 4 the classifier achieved a validation and testing accuracy of $\approx 75\%$. When examining the confusion matrix, it was found that the classifier had a high type II error rate suggested that the model was quite inefficient at identifying some of the ν_μ charged-current events. Perhaps this is a sign that the classifier was over-fitting the balanced training data. Computing the Matthews Correlation Coefficient, a more meaningful metric for assessing the performance of a binary classifier in cases of class imbalance, indicated only a moderate positive correlation between the predictions and the the truth.

To reduce the false negative rate, it was investigated whether lowering the the probability threshold for rounding the predicted values would yield better performance metrics on the testing set. The threshold was set to optimise the validation data's accuracy and MCC but implementing it on the testing data did not lead to any noticeable changes.

The performance of the classifier was then investigated for certain metadata. First, the classifier was tested on different types of ν_μ charged current interactions: QE, RES, DIS and Other. The model performed worst on the DIS interactions and best on QE and 'Other' interactions, in terms of the confusion matrix, the MCC and the accuracy.

Finally, the efficiency of the model was investigated for different neutrino and lepton energy levels. Overall, classifier's accuracy was best when shown images pertaining to high energy neutrinos and leptons. However, these results could be skewed by the fact that there are very few observations of high energy interactions in the dataset.

References

[1] Ryan Nichol Cheryl Patrick. Additional information for this project, 2020.

[2] Fermilab is america's particle physics and accelerator laboratory.

[3] Giorgio Giacomelli. The standard model of particle physics. neutrino oscillations. *Radiation measurements*, 44(9-10):826–833, 2009.

[4] Jorge Sola and Joaquin Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, 44(3):1464–1468, 1997.

[5] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

[6] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[7] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.