Dual-input convolutional neural network image classifier for NoVA-like simulations

I. Sanai, 27th of December 2020.

***Abstract*— Dual-input convolutional neural network classifiers were created to analyse images of interactions from a simulated detector similar to NoVA. The first aim was to classify charged current events, where the accuracy of the classifier reached 77 ± 0.01%. It was found that testing the model on QE events produced much higher accuracies, as opposed to DIS events. High lepton energy events had accuracies that were at least 10% higher than their low energy counterparts. The model performance was seemingly unaffected by neutrino energy, plausibly due to its low mass. Similar models were also created to determine the neutrino energy, the interaction mode, the flavour of the neutrino and the ratio of the lepton energy and the neutrino energy. All of these models were moderately successful in learning with the exception of the interaction mode classifier.**

***Index Terms*—Feature Maps, Convolution, Neutrino Oscillation,**

I. Introduction

IMAGE classification is an ever-advancing area in High-Energy Physics due to the need for algorithms that can separate signal from background and identify interactions within large datasets efficiently.

Convolutional neural networks (CNNs) are machine learning algorithms which extract features from data using two main operations: convolution and pooling. Convolution allows for features to be extracted using filters while preserving the mapping of the data, and the output of a filter applied to a layer is called a feature map. As these maps often have high dimensionality, the second operation, pooling, serves to reduce the number of parameters within the network. This lowers computation time and the risk of overtraining that is a common issue in traditional multilayer neural networks [1].

The process of extracting features is an element-wise multiplication of the filter (generally a square matrix) with an equivalent-sized matrix of pixel values from the image, adding the resulting terms, then moving this filter to the neighbouring pixel and repeating this process over the entire image to produce a feature map. This can be seen below in a simplified illustration.

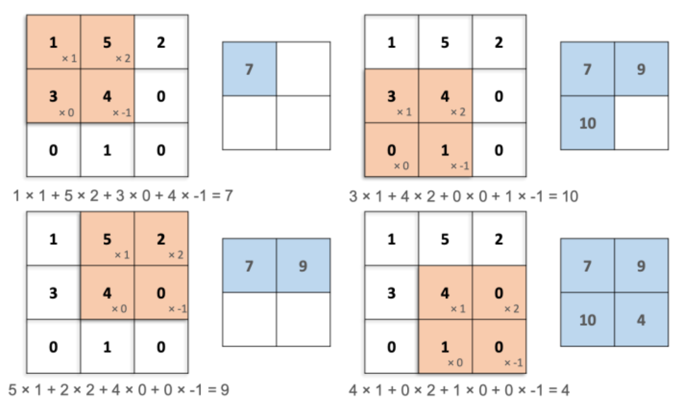


Fig. 1. Convolution of a 3×3 pixel image using a 2×2 filter with the following values: $\begin{bmatrix} 1 & 2 \\ 0 & -1 \end{bmatrix}$. The final feature map from this convolution is the blue map in the bottom right of the image.

Image classifiers are two-dimensional CNNs, as the filters move in two dimensions over the images (Fig. 1.). The inputs for a 2D CNN are three-dimensional arrays, where the first two dimensions correspond to the image pixels and the third signifies the number of channels (1 for grayscale, and 3 for RGB). Binary image classifiers are used when there are two possible labels, and categorical classifiers are used when the element belongs to one of multiple classes. CNNs are also used in regression models to predict a continuous outcome variable, such as a probability or energy value.

This project aimed to create a CNN image classifier for neutrino interactions within a simulated detector, similar to the NoVA detector in Fermilab. The variation in the probability of measuring a neutrino's flavour as it propagates through space is called neutrino oscillation, and NoVA's primary goal was to detect ν_μ to ν_e oscillation, because ν_μ to ν_τ oscillations had already been observed experimentally [2][3].

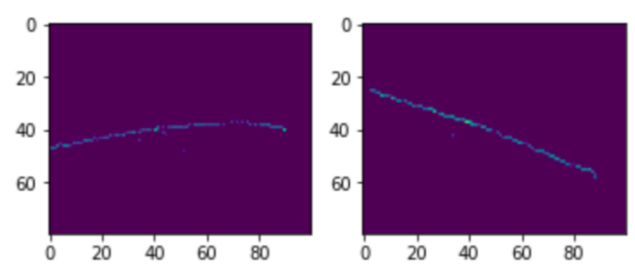
The neutrinos are indirectly observed via their weak interactions with particles in the detector. Charged-current (CC) events occur when the neutrino scatters off a nucleon and turns into its corresponding charged hadron, and a $W^\pm$ boson is exchanged in the process. A neutron also converts to a proton to conserve charge, so a ν_τ charged-current event might look like the following:

$$\nu_\tau + n \rightarrow \tau^- + p$$

Within CC-events, neutrinos can scatter off target hadrons in three different ways, and these differences are reflected in the images from the detector. Deep Inelastic Scattering (DIS) is when the target hadron is broken up, and the neutrino scatters off of a constituent quark. It is the highest out of all three in energy due to the target breaking up, and its images usually contain multiple showers. In resonant events (RES), the nucleon is excited to an unstable resonant state. Quasi-Electric (QE) events occur when the target nucleon remains unchanged. This interaction has the lowest energy, and in the detector images usually present as two long tracks.

II. METHOD

The machine learning algorithms were written in Python using the Keras Functional API, as it can create more flexible models with multiple inputs or outputs. The data used for this investigation was in the form of HDF5 files, each of which contained information from roughly 7000 simulated interactions. Each event had two images ($x \times z$ and $y \times z$ projections of the tracks of particles in the detector) as well as the meta information, namely the energy of the neutrino (in GeV), the energy of the lepton (in GeV), the final state of the interaction, and the interaction type. An example of the information stored for one event can be seen below.



(a) (b)

Fig. 2. A set of 100×80 images for a DIS ν_μ event. The $x \times z$ view is seen in (a), and the $y \times z$ view in (b).

Firstly, the data was extracted from the desired number of files using a function which looped over each file from which the images and metadata was appended to corresponding arrays. Generally, it proved difficult to extract more than 70,000 events without using all available RAM and crashing the program.

Each image event array was of the shape (100, 80, 2) where 100×80 is the image's pixel size, and 2 corresponds to the two images in each interaction. CNNs do not conventionally accept dual-image inputs, and therefore a brief function was written to separate the $x \times z$ and $y \times z$ images, thus creating two separate arrays with shapes (100, 80, 1) to feed into the CNN as grayscale images.

The first task was to create a classifier that could differentiate ν_μ-CC from non-ν_μ CC events. As this task studies only two classes, a binary classifier was created. This type of classifier is characterised by the sigmoid activation function and two output neurons in the final layer (which correspond to a 0 or 1 output). Thus, the interactions which corresponded to ν_μ CC events were assigned a label of 1, while other interactions were assigned a 0. Upon inspection of the resulting label array, it was immediately obvious that the data was very imbalanced, with ν_μ-CC events making up approximately 89% of the events in each data file. As skewed data is known to have a negative impact on machine learning algorithms, this was accounted for by the use of a data preprocessing method called under-sampling. This entails eliminating excess data from the majority class, thus creating a training set with roughly equal elements from each class and balancing the data [4].

The CNN architecture was two layers of convolution and pooling performed in succession, and at least one dropout layer. Bias and kernel regularisers were used in the convolution layers, and ReLU activation functions were used throughout as they are non-saturating over several layers, thus do not halt learning like sigmoid functions do in hidden layers [5]. The general structuer of the model can be seen in Fig. 3. This model was then tested on a random file from the simulator data, which was not preprocessed in any way.

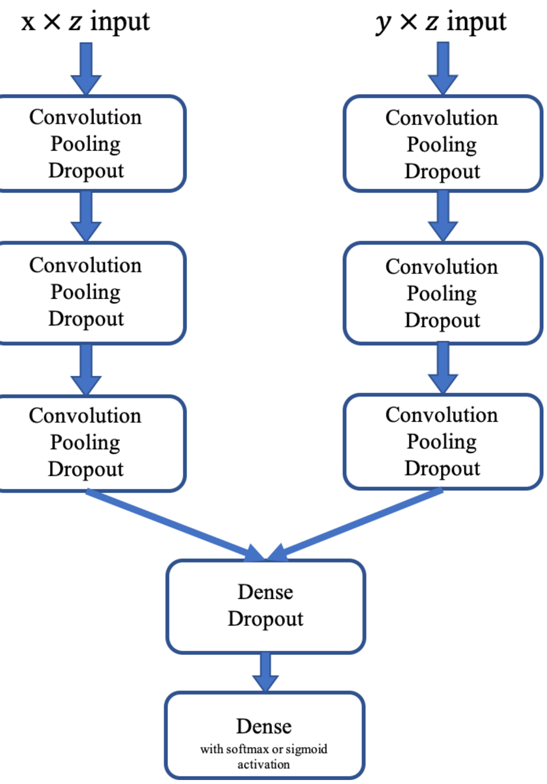


Fig. 3. An illustration of the general CNN architecture for this project. The activation functions for the convolution and dense layers were ReLU, except for the last dense layer.

To determine how the classifier's efficiency depended on different metavariables, three categories were analysed: neutrino energy, lepton energy, and interaction types. Six smaller test sets were created using test file mentioned above; the CNN's success in classifying ν_μ CC events on a dataset composed solely of high neutrino energy events was compared to its success on a low ν energy test set, and the same was repeated for high vs. low lepton energy sets. In a similar fashion, the classifier's efficiency was compared for a QE-only test set and a DIS-only set. Determining what qualifies as "high" or "low" energy for leptons and neutrinos proved difficult to assess, as these definitions vary vastly within literature [6][7].

Instead of trying to determine what is objectively considered high and low energy for these particles, I decided to determine this based on the incidence of energy ranges within the datasets. Histograms were plotted to determine where to set the threshold for high/low energies based on the energy distributions.

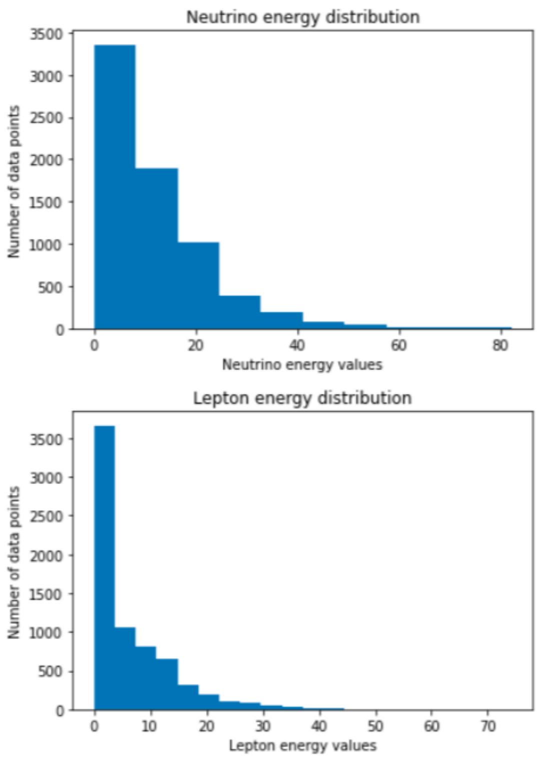


Fig. 4. Histograms that show the energy distribution for neutrinos (above) and for leptons (below) which were used to determine what is considered a "high" energy particle.

As for comparing E_ν values, a roughly fair split of "high" and "low" energy values was created. As 50% of the data had energy values of 16 these were considered "low" energy while the remaining half (energies of > 16 GeV) were in the "high" category. This process was repeated for high and low E_ℓ categories, where the threshold for high values was 3.8 GeV.

The

Next, a machine learning algorithm was written to identify the neutrino flavour in each event. It was noted that there were no ν_τ events in the entire dataset; this is most likely due to the nature of the NoVA data which focused specifically on $\nu_\mu \rightarrow \nu_e$ oscillations. Thus, again, a binary classifier was created. Similarly to before, this data was also heavily skewed with ν_e events making up no more than 2% of the set; under-sampling was again employed to remedy this and a larger number of data files were used to obtain sufficient training data.

The final classification model was a categorical classifier which employed a categorical cross entropy loss function. The labels had to be processed to change them from binary to categorical, and the activation function in the final dense layer was replaced with a softmax function. This layers also had 5 neurons to represent the five categories (CC DIS, CC RES, CC QE, other and NC).

Two regression models were also created to evaluate the energy of the neutrino as well as the ratio of lepton energy to neutrino energy. These models differ vastly from classifiers, as the success of a regression model is measured by finding the difference between the predicted and true energy values. A root mean squared optimiser was used, and a mean square error loss function, and the metric was mean squared error.

The energy values used in the regression models were normalised by dividing them through by the highest value in each dataset. For the model that predicted $y = {}^{E_\ell}/_{E_\upsilon}$, the data was preprocessed by eliminating data points with $E_\upsilon = 0$ to avoid dividing by zero.

III. Results and Data Analysis

Most of the results for this project are in the form of accuracy and loss graphs. The results and analysis were split into five sections for the five models created in this investigation.

I. ν_μ-CC Event Classifier

The model was run on 39 preprocessed data files with ~62,000 data points. Usually, in Fig. 4. the validation accuracy would follow the trend of the training accuracy while remaining slightly lower throughout as it is unseen data, so there is higher error. In this model, however, the validation accuracy was consistently higher than that of the training data, and the opposite was true for the loss. This was not an error in the model, but rather due to the fact that the dropout is not applied to the validation data, which means its accuracy is always a little higher.

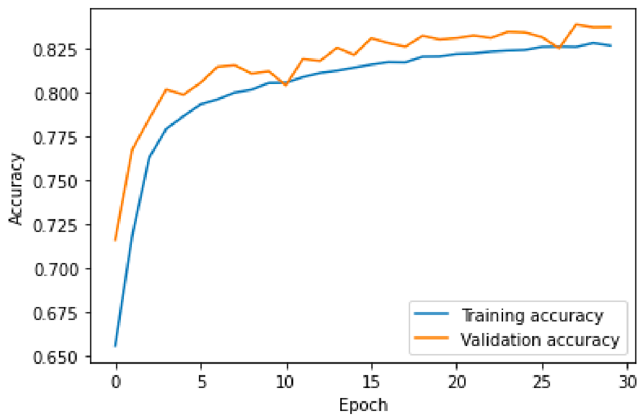


Fig. 4. The accuracies of the muon neutrino event classifier against epoch for validation and training data. The final test accuracy was 77%.

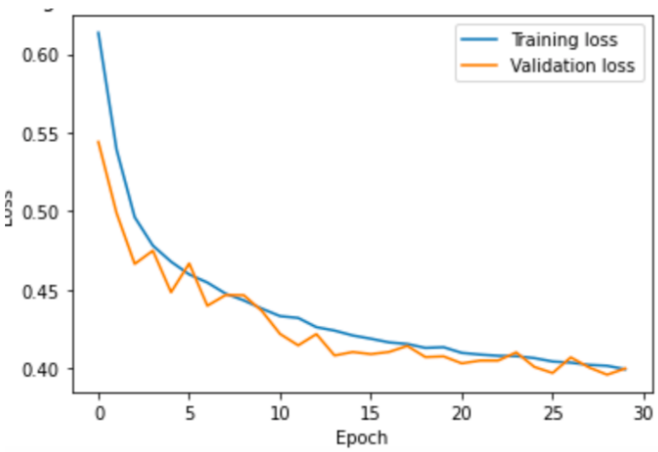


Fig. 5. The losses of the ν_μ-CC event classifier against epoch for validation and training data. It can be seen from the shape of the loss curve that the model was learning successfully on the balanced dataset.

The success of the final model was difficult to assess, as the same exact model on the same data could have a test accuracy of anywhere between 72% and 79% depending on the run. Thus, an average final accuracy was calculated by compiling the model ten times and computing the final accuracy as the average of the ten values, with the uncertainty calculated as the standard error.

TABLE 1
MUON NEUTRINO CC EVENT CLASSIFIER TEST ACCURACY

Run	**Accuracy**	**Δ Accuracy**
#	%	
1	72.39	
2	78.04	
3	75.97	
4	79.09	
5	78.84	
6	74.30	
7	77.21	
8	74.06	
9	77.04	
10	79.30	
Average	77.08	0.01

Table. 1. The accuracies obtained from running the muon neutrino CC classifier ten times, with the uncertainty given as the standard error.

The final accuracy was 77.08 ± 0.01%, which is substandard given that other CNNs in the literature with similar structures reach accuracies of 90%+ [9]. This sub-optimal accuracy may be explained by the influence of metavariables on the model performance, and is discussed below.

Model dependence on metadata

As it was not possible to obtain average test accuracies for these preprocessed test sets by compiling the model multiple times, the results were compared qualitatively based on the percentage ranges of the accuracies, as opposed to quantitatively.

TABLE 2
METADATA ACCURACIES FOR MUON NEUTRINO CLASSIFIER

Test Data	**Accuracy**
Metavariable	%
QE	72.39
DIS	78.04
High E_ν	79.09
Low E_ν	78.84
High E_ℓ	77.21
Low E_ℓ	74.06

Table. 2. The accuracies obtained from running the muon neutrino CC classifier once on test sets with specific metadata categories.

The model performed much better on QE events (accuracies between 87% and 91%) and DIS events (64-

70%), possibly due to the fact that leptons are some of the heavier particles within these collisions (relative to neutrinos) so they can have a stronger effect on the interactions within the simulator. As with the QE vs. DIS events there was a much lower incidence of high lepton energy events in the dataset, hence the lower overall accuracy.

There was a maximum accuracy difference of 2% between the high and low neutrino energy groups. This is likely due to the neutrino's characteristic low mass and weak interaction, meaning the images would remain largely unaffected by the neutrino energy and dominated instead by the hadron and lepton energies.

II. Neutrino Flavour Classifier

This classifier had similar training and validation accuracy and loss graphs to the muon neutrino classifier (both reaching similar accuracies at the end of training, as can be seen in Fig. 6.), however the test accuracy for the flavour classifier was much higher. As in the previous task, the model was compiled 10 times and the final test accuracy was calculated as the average of the accuracies from the ten runs, with the uncertainty as the standard error. The average accuracy was 89.49 ± 0.01, which is very successful. The accuracies were in a range of 86% to 93%. The high performance of this model may be due to the lack of neutral current events which might be messy and make classification tasks more difficult for the computer to perform.

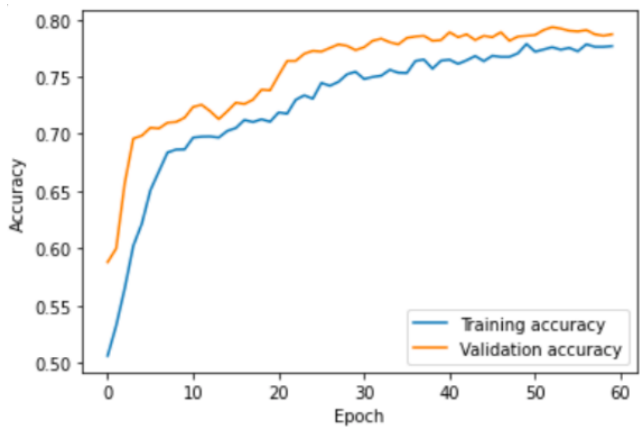


Fig. 6. The accuracies of the ν flavour classifier against epoch for validation and training data..

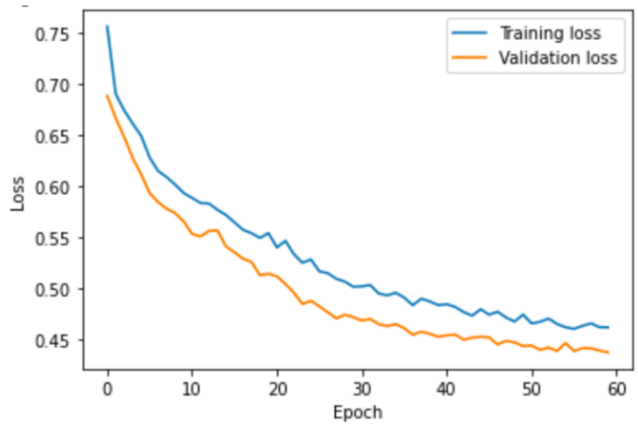
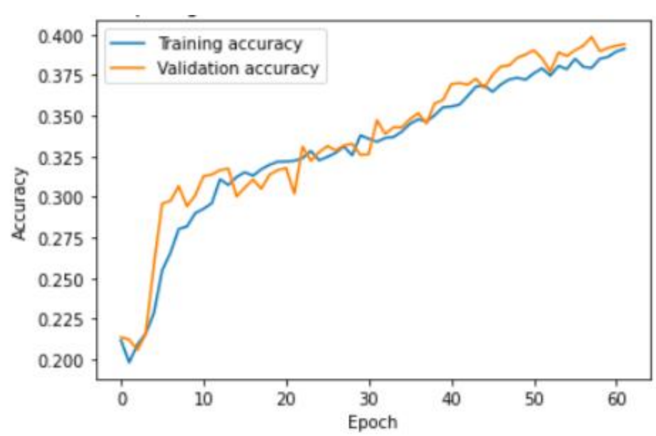


Fig. 7. The losses of the ν flavour classifier against epoch for validation and training data.

III. Interaction Mode Classifier



The performance of this model was very poor, with the accuracy reaching a final value of ~40%. This could not be considered as a successful model, and while it was clear from the decreasing loss that the model learned, it is unlikely that the model could produce any meaningful predictions on the test dataset. This may be due to the fact that a fifth of the events in the dataset were neutral current events, in which the neutrino does not change into its charged hadron partner. The absence of these hadrons may have a significant impact on the clarity of the NC events.

IV. Neutrino Energy Regression Model

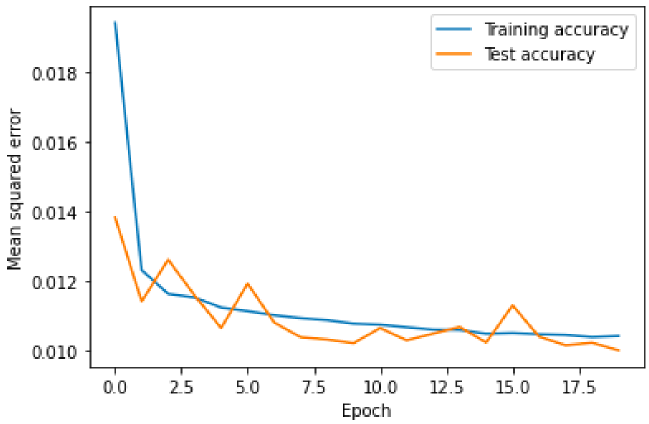


Fig. 7. The mean squared error for the energy regression model. These values are 10x smaller than the loss values.

The mean squared error is a value which was expected to decrease over multiple iterations if the network were to learn successfully, as opposed to the accuracy metric for a classifier which would be expected to increase.

V. $y = {}^{E_\ell}/_{E_v}$ Regression Model

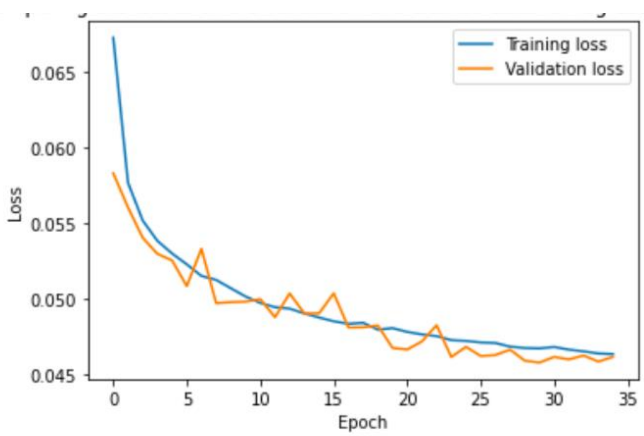


Fig. 8. Losses for the second regression model.

This network had similar performance to the previous model and was successful in learning.

Network Optimisation

A kernel stride of (2,2) produced lower accuracy on both training/validation and on the test data than the default stride of (1,1). There are two possible reasons for this: the first is that important information lies in each pixel, and thus skipping over one pixel when passing the filter over the image results in loss of valuable information. However, that seemed unlikely as the images contain tracks and showers where a gradient of each line should be sufficient to extract enough information. Another possible reason is that a smaller stride increased the number of parameters to ~60,000 which is slightly lower than the number of data points (65,986). It was predicted that this would result in overtraining, as when parameters are equal to or larger than the number of data points, the model has the capacity to perfectly memorise the training data, and fails on the test data. However the testing accuracy was generally high with this number of parameters.

The hyper-parameters were optimised both by using verified methods as well as through trial and error. The more methodical approaches included using an in-built function that halts training once validation loss no longer decreases significantly, thus reducing the number of unnecessary iterations. Additionally, dropout layers were interspersed throughout and hidden layers were minimised to prevent overfitting. The number of parameters in all the models was kept lower than the number of data points, as this is considered good practice [10]. Experimenting with different hyperparameters was a tedious method, however more complex methods were out of the scope of this course.

IV. Conclusion

This project aimed to create an image classifier to identify charged current muon neutrino events using images from a simulated detector. A convolutional neural network was built using the Keras functional API in Python. All the models contained three two-dimensional convolution layers and three max pooling layers, as well as dropout layers to reduce the risk of overtraining.

The model's average accuracy on a test dataset was measured by running the model ten times and computing the average accuracy over the ten runs, with the error as the standard error.

The model had a final test accuracy of 70.08 ± 0.01%, and while the loss function clearly showed that the model was learning, it cannot be said that its performance was optimal given that successful classifiers usually reach accuracies of 90% and above [5].

However, while the model structure is an important factor in successful learning, the quality of the data is equally, if not more, important, and thus the model's dependence of metavariables (such as interaction type, energy and interaction mode) were analysed. This was done by splitting the test data into three categories (neutrino energy, lepton energy and "cleanness" of the images), and evaluating the model on two datasets from each category. Neutrino energy was evaluated with a high energy (>16 GeV) and low energy dataset (<16 GeV), and the same was done for lepton energy, however the threshold for high lepton energy was 3.8 GeV. Two sets were also made to compare quasi-electric and deep inelastic scattering interactions, and it was expected that the former would have better performance. The model had a much higher performance on "clean" (quasi-electric) events as opposed to messy (deep inelastic scattering) events. Neutrino energy did not seem to have an effect on the testing accuracy, and it was postulated that this is because of its characteristic weak interactions and low mass. The classifier performed better on high lepton energy events, where its accuracy was always 10% higher or more than that of the low lepton energy set. This is likely due to the leptons' effect on image quality. Indeed, this reflected that certain metavariables greatly affected the model's performance.

The final classification model was a categorical classifier to identify the interaction mode. This model performed very poorly with accuracies no higher than 40%. This might be due to insufficient data or neutral current events being more difficult to classify.

Two regression models were made to determine the energy of the neutrino and the ratio of lepton energy to neutrino energy. Both of these models were generally successful in learning with both the mean squared error and loss decreasing significantly during training.

This project could be improved upon by using more sophisticated network optimisation methods such as cross validation and using other types of regularisers. Further research could entail researching neutral current events and investigating whether the lack of a charged hadron partner results in poor classification as was postulated in this report.

V. Works Cited

[1] R. Reed and R. Marks, *Neural smithing*. Cambridge, Mass.: The MIT Press, 1999.

[2]B. Barish, "Tau neutrino physics: an introduction", *Nuclear Physics B - Proceedings Supplements*, vol. 98, no. 1-3, pp. 12-25, 2001. Available: 10.1016/s0920-5632(01)01190-2.

[3]M. De Serio, "Tau neutrino appearance in the CNGS muon neutrino beam: the OPERA experiment", *Nuclear Physics B - Proceedings Supplements*, vol. 169, pp. 321-325, 2007. Available: 10.1016/j.nuclphysbps.2007.03.019.

[5] A. Aurisano et al., "A convolutional neural network neutrino event classifier", *Journal of Instrumentation*,

vol. 11, no. 09, p. 2, 2016. Available: 10.1088/1748-0221/11/09/p09001.

[6] H. Minakata and H. Nunokawa, "Measuring leptonic CP violation by low energy neutrino oscillation experiments", *Physics Letters B*, vol. 495, no. 3-4, pp. 369-377, 2000. Available: 10.1016/s0370-2693(00)01249-1.

[7] J. Sinatkas, V. Tsakstara and O. Kosmas, "Simulated Neutrino Signals of Low and Intermediate Energy Neutrinos on Cd Detectors", *Frontiers in Physics*, vol. 7, 2019. Available: 10.3389/fphy.2019.00042.

[8] E. Baum and D. Haussler, "What Size Net Gives Valid Generalization?", *Neural Computation*, vol. 1, no. 1, pp. 151-160, 1989. Available: 10.1162/neco.1989.1.1.151.

[6]