

LITERATURE REVIEW

THEORETICAL PHYSICS BSC PROJECT

UNIVERSITY COLLEGE LONDON

Deep-Learning Classifier Robustness in Neutrino Experiments

Author

I.J.S. SHOKAR - 17066988

Supervisor

Dr. C BACKHOUSE

October 21, 2019

Abstract. The project being undertaken is looking at using Convolutional Neural Networks (CNNs) to classify neutrino interaction types from the NOvA experiment, and ensuring robustness of the classification algorithms to be able to generalise to incoming data by introducing the addition of a Domain Adversarial Neural Network (DANN).

Introduction

Neutrinos are the most abundant non-massless particle, ‘Vitagliano et al’ (2017), yet due to their nature very little is known about them. Neutrinos are charge neutral particles and only interact through the weak interaction (and the force of gravity), meaning detecting them can only be done through the detecting the resultant particles after an interaction involving a lepton, ‘Maki et al’ (1962). To mitigate the effect of solar neutrinos and high energy cosmic rays, ‘Szadkowski et al’ (2016), from interfering with neutrino experiments, many experiments and detectors are placed deep underground, with accelerators producing the neutrinos. Many neutrinos do still penetrate through the ground and to the detectors, however they are able to be accounted for and excluded from the analysis. There are a number of underground neutrino experiments, notably MINERvA ‘Perdue et al’ (2018), IceCube, ‘Brenzke’ (2017), and DUNE, ‘Acciarri et al’ (2016), however the experiment that this project is working on is the NOvA experiment at Fermilab, ‘Aurisano et al’ (2016).

It was observed that two thirds of neutrinos that were expected to reach earth in the form of cosmic rays were not present, which led to the observation that neutrinos have a mass and the ability to oscillate between the different flavours, ‘Fukuda et al’ (1998). This explained how the neutrinos expected to be observed changed to a different flavour, a discovery that was awarded the 2015 Nobel prize, however, the exact masses of the the flavours as well as the precise values of the oscillation parameters are still unknown, ‘Capozzi et al’ (2016). The three neutrino flavours, each named due to the associated lepton that is produced or absorbed with the neutrino, do not correlate to the mass eigenstates, but are a superposition of them, where $|\nu_\alpha\rangle$ is a neutrino with flavour $\alpha = e, \mu, \tau$ for electron, muon and tau, and $|\nu_i\rangle$ a neutrino with definite mass m_i for $i = 1, 2, 3$.

$$|\nu_\alpha\rangle = \sum_i U_{\alpha i}^* |\nu_i\rangle$$

Neutrinos interact with other particles in their flavour eigenstates, but travel as mass eigenstates. As a neutrino travels, the quantum mechanical phases of the mass eigenstates advance at different rates due to mass differences, which results in a changing superposition of mass eigenstates corresponding to a different flavour. This violated the Standard Model which predicted neutrinos as massless, while this mass state superposition is seemingly unique to neutrinos, requiring exploration beyond the Standard Model which could produce new unknown properties of fundamental particles, ‘Sonneveld’ (2019).

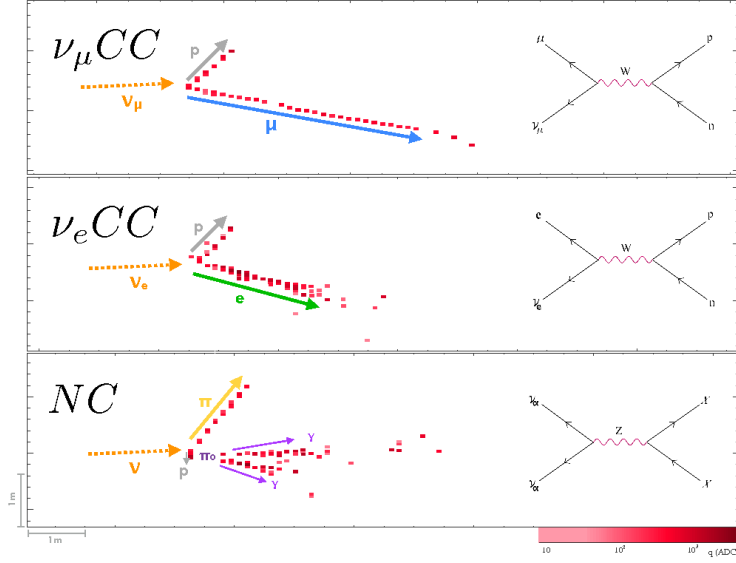


Figure 1. Plot showing three of the event topologies, for charged-current muon and electron neutrinos as well as a charged-neutral interaction, with their associated Feynman diagrams, 'Singh' (2017).

NOvA Experiment

To measure these oscillations, neutrino energy and flavours need to be reconstructed; as neutrinos are invariant to charge and magnetic fields they travel without deflection they indicate their sources of production, this means they can easily be traced back to an inception vortex, examples of these images are shown in Figure 1. The flavour can be determined from charged-current interactions (CC) from the associated lepton that is produced. Neutral-current interactions don't indicate an associated lepton, and thus aren't included in analysis. The dominant CC interactions are: quasi-elastic - where the nucleon recoils from the scattering lepton; resonant - where the nucleon is deflected into baryonic resonance; deep-inelastic-scattering - where the nucleon breaks up in the process of hadronisation, and meson exchange currents - where two-nucleon emission takes place, 'Aurisano et al' (2016).

At NOvA, a near detector and a far detector are used to determine the flavour of neutrino passing at each, and any changes to the neutrinos over the distance can be compared. Traditionally classification required reconstructing high-level components associated with particle interactions such as clusters, tracks, showers, jets, and rings and summing the directions, and shapes of these objects, 'Aurisano et al' (2016). However, the method now implemented at NOvA is machine learning algorithm, where the model learns to be able to classify the interaction type from examples. Currently a CNN is being used, which have been found to be capable of pattern detection and particularly effective at image classification with real world photographs. NOvA's classifier is based on the GoogLeNet algorithm which achieves an error rate of only 6.7% on images, 'Szegedy et al' (2014). The network at NOvA outperforms the previous track-based modestly with a measurement-optimised efficiency of 58% over the of

57% from ‘Adamson et al’ (2016)[I] for ν_μ CC interactions and for ν_e CC interactions 49% over the 35% when compared to results from ‘Adamson et al’ (2016)[II], ‘Aurisano et al’ (2016), displaying that there is a large room for improvement.

Artificial Neural networks (ANNs), ‘McCulloch and Pitts’ (1942), are not a new phenomenon, however the recent increase in their popularity has come about due to the advances in hardware that allows the computationally expensive training of networks. Neural networks increase in performance when given more data to learn from, and the volumes of data that are produced in this digital age has led to increased research into the development of these algorithms.

As a CNN is a supervised learning algorithm, the network must be trained examples of labeled data, where the network will determine which patterns, or features, correspond to which event. Data from the detector will not be labeled, thus simulated Monte Carlo event generators will provide the training examples, the two simulators that will be used are GENIE, ‘Andreopoulos et al’ (2009) and GiBUU, ‘Lalakulich et al’ (2011), however there are many different models that all simulate events uniquely. The GENIE Monte Carlo Simulation simulates the initial iteration of a neutrino with nuclei in the detector and the resulting scattered products to the surface of the nucleus. ‘Andreopoulos et al’ (2015). It mainly reproduces neutrino-nucleon scattering data but limited neutrino-nucleus scattering data means there are deficiencies in the primary physics model. ‘Perdue et al’ (2018). While the GiBUU simulation based on the Giessen–Boltzmann–Uehling–Uhlenbeck model which is a semiclassical transport model, which describes the evolution of a many-body system in the presence of potentials and a collision term, with the addition of neutrino-induced interactions. The two simulations differ on which models are used to simulate different interactions, leading to discrepancies between different models and how they model events.

This highlights the importance of using two different event generators, as it is required that the algorithm is trained to classify regardless of bias from the statistics of the data. The robustness of the classifier will be determined by its ability to not show a domain-bias from the generators it was trained on, with the simulations based on models that are not identical to nature. ‘Perdue et al’ (2018), and that the model is able to generalise to incoming detector data. By only showing the model data from one generator, it may classify real data incorrectly if the statistics vary from that of the simulation model. Neural networks are still subject to overfitting to their training data like this and overcoming this domain bias is the main challenge of the project.

Model

In a traditional Forward Feed Neural Network (FFNN), the most basic ANN, for each neuron in each layer the output will be a weighted sum of the inputs from every neuron in the preceding layer, with this sum being passed to an activation function, which brings non-linearity allowing the modelling of more complex patterns, ‘Acciarri et al’ (2017). The output, y_k , for the k^{th} neuron in a layer where $i = 1, 2, \dots, n$ where n is the number of neurons in the previous layer, and x_i their corresponding outputs, with w , the weights, b , the bias, and σ the

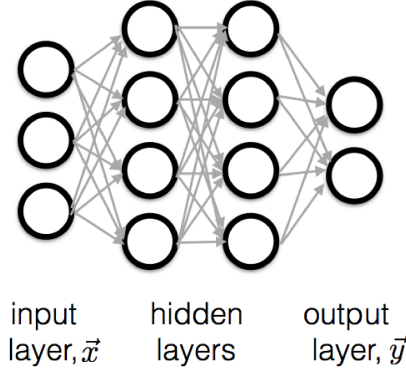


Figure 2. Plot of the architecture of a basic FFNN, 'Acciarri et al' (2017).

activation function.

$$y_k = \sigma\left(\sum_i w_{ki}x_i + b\right)$$

The algorithm becomes connected network of these weighted sums, where the parameter weights, which are initially random, are learned by giving examples, where the algorithm tunes the values to produce the desired output through backpropagation.

Much like a traditional FFNN, CNNs take in an input, have a number of hidden layers, and return an output; but work differently in that rather than every neuron being connected to each input value, a kernel, a small matrix, is passed over the image, pixel by pixel. This is due to the size of the input which can be several million pixels, where the process would become too computationally expensive to have fully connected layers. This kernel will apply a convolution function, using the pixel and the surrounding pixels as inputs to produce a weighted sum, which it will produce another image from, with these convolutions detecting local structures,' Brenzke et al' (2017), such as edges or other patterns, building several feature maps and through training determining which combinations of these are most important to producing the correct output class, see Figures 3 and 4. Each layer passes multiple kernels over the same input, to produce a number of output matrices, all applying a different convolution, to form a tensor as the output.

In order for the network to be able to complete the process in a reasonable amount of time, each convolution layer is followed by a pooling layer, which reduces the size of output feature map by partitioning each matrix into subsections, where either the maximum value of the set (max pooling) or the average of the values (average pooling) is taken, reducing the number of parameters for each subsection to a single value. With these steps of convolutions and pools making up the majority of the layers in the network, producing a final vector, that feeds into a traditional FNN to produce scores for the confidence level for each label it thinks it may be classifying. It is this output value that cuts are taken on when analysing data, using the simulation results where the score from the network is over a threshold value.

A large problem that faces neural networks is the problem of overfitting. This is whereby

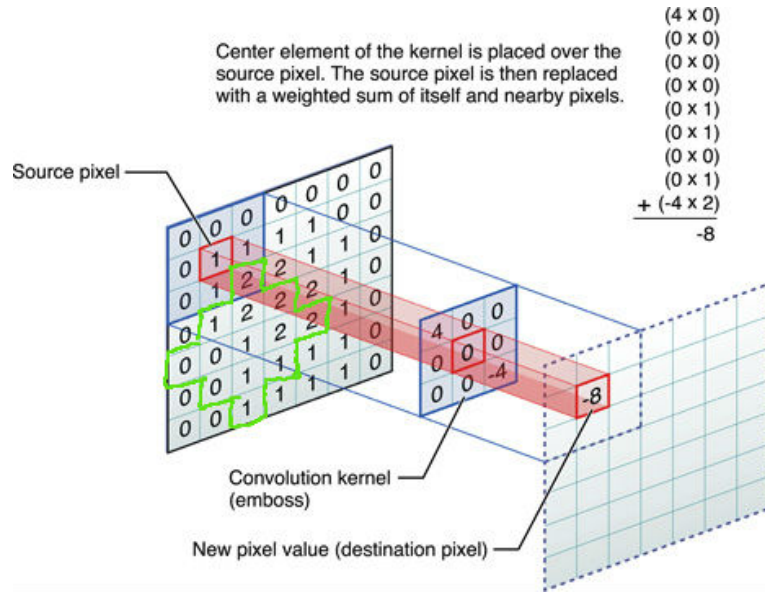


Figure 3. Diagram of a Convolutional Kernel acting on a source matrix. 'Brenzke' (2017)

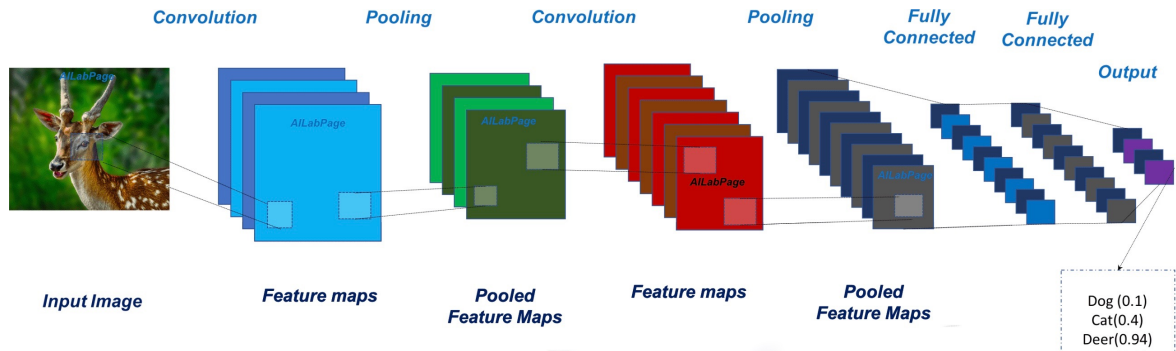


Figure 4. Architecture of a basic CNN, displaying the hidden convolution and pooling layers.

the model becomes very good at picking up details from the dataset and essentially learns the dataset so much so that it struggles to then generalise on new data. In a CNN as there are fewer free parameters than with a fully connected network then the chance of overfitting is reduced, 'Brenzke et al (2017), and methods such as Dropout, which is dropping random neurons in training so that that the network doesn't rely on a small number of neurons can help reduce this. However, they are still subject to becoming too specialised to the data they have been trained on, and the CNN being used to identify particle events, the model will be trained on the Monte Carlo simulations, neither of which perfectly model real events. This means that the model is susceptible to learning on these unrealistic data sets and becomes inaccurate when fitting for real data.

An alternative network architecture that will be explored in this project is that of a DANN, which will be joined to the CNN classifier in order to reduce the domain dependence of the

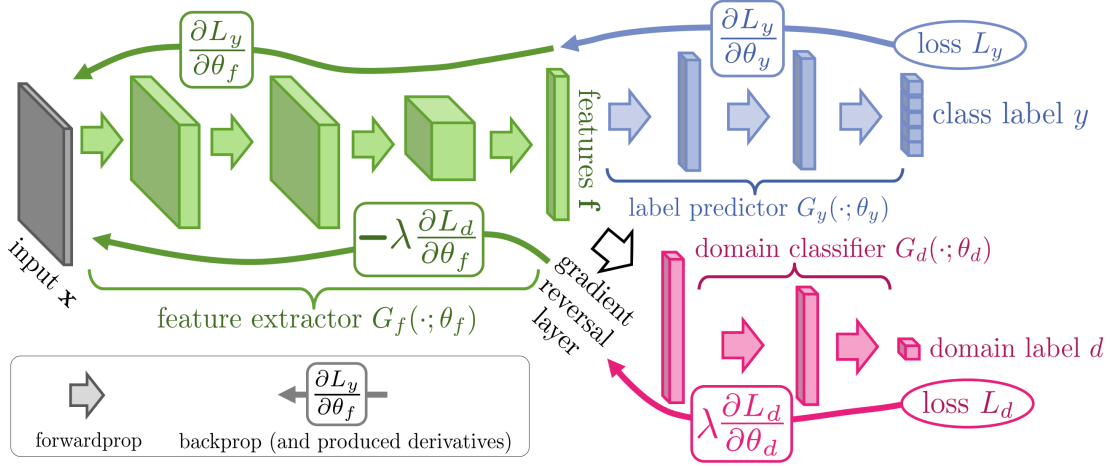


Figure 5. Plot of the architecture of a DANN joined to a network, where y is the outputs from the feature classifier and d the outputs of the domain classifier. The gradients show the error, or loss, functions that back-propagate to determine the weights and biases. 'Acciarri et al' (2017).

model. 'Bousmalis et al' (2016). Two sources of data, or domains, are introduced, a source domain where labels are available but not used so that the input is similar to unlabelled data from the target domain where labels are not available. The network produces two classified outputs, a features classifier, which predicts class labels in a way similar to a normal ANN, but also a domain classifier that discriminates between the source and target domains- determining which inputs are from simulations or detector data, 'Ganin et al' (2016).

'Perdue et al' (2018), explains how DANN can also be applied to compare multiple simulation only datasets, such as GENIE and GiBUU, to determine whether the network can tell which raw images belong to which. To have a robust model the network will need be invariant to the differences between these, as well as between the simulations and real data. Only data from the source domain is used to determine the parameters for the features classifier, with data from both domains determining those of the domain discriminator. By optimising for accuracy on the features classifier and maximising the error for the domain discriminator the classifier is trained to only use features in both domains. When an equilibrium is reached the domain discriminator is only able to distinguish samples from the sources by chance, Louppe et al' (2017). Figure 5 shows this additional part of the network, which in theory should produce a network trained to be invariant to the differences in the simulations as well as with real data, and results from 'Perdue et al' (2018) at MINERvA showed a small increase in accuracy at 96% to 94.5% for out of domain trained networks. The current accuracies of the No-DANN CNN in place at NOvA are much lower, and provides much greater scope for improvements, Aurisano et al' (2016).

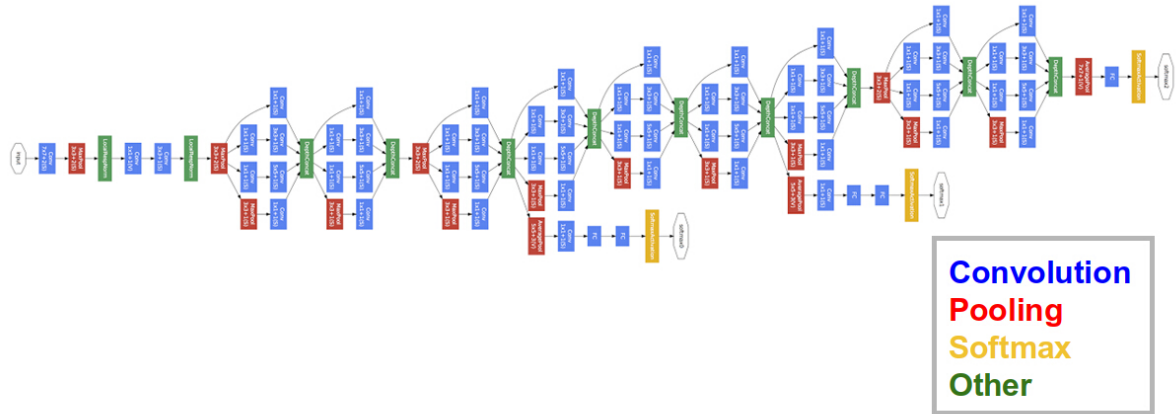


Figure 6. GoogLeNet Architecture schematic, 'Szegedy' (2014).

Project Timeline

A rough timeline of how the project will run looks like this: Between September and November, the main priority is become familiar with the framework that the data is stored in. NOvA uses software called NOvASoft which is built on scientific toolkit ROOT. Learning how to use ROOT will be essential to be able to perform analysis on the simulation data. The framework that will be used is NOvA's CAFAna, and learning the functionality as well as capability of CAFAna will be the main focus of the first few months, in order to make cuts of the simulation data. ROOT is written in C++, which is a language I will need to become more familiar with.

Between December and January, the focus will shift towards the deep-learning and understanding the CNN that is already been implemented by NOvA- determining its architecture, which is adapted from the GoogLeNet Architecture 'Szegedy et al' (2014), and its shortcomings. The final, yet most demanding task would be developing a classifier that can surpass that already in use. Using a DANN we would look to build a network, using the Keras deep-learning library running on top of Tensorflow- Google's software library, that can tell between the GENIE and GIBUU simulation statistics, before training a network to be used on detector data that is invariant to their differences and more accurate than the current model.

References

- E. Vitagliano, J. Redondo, G. Raffelta et al. *Solar neutrino flux at keV energies*, 2017.
- Z. Maki, M. Nakagawa, S. Sakata et al. *Remarks on the Unified Model of Elementary Particles*, 1962.
- Z. Szadowski, D. Glas, K. Pytel et al. *Artificial Neural Networks as a FPGA Trigger for a Detection of Neutrino-Induced Air Showers*, 2016.
- G.N Perdue, A. Ghosh, M. Wospakrik et al. *Reducing model bias in a deep learning classifier using domain adversarial neural networks in MINERvA experiment*, 2018.
- M. Brenzke. *Development and Optimization of Deep Neural Networks for Energy Reconstruction of Muon Events in IceCube*, 2017.
- R. Acciarri, M. A. Acero, M. Adamowski et al. *Long-Baseline Neutrino Facility (LBNF) and Deep Underground Neutrino Experiment (DUNE) Conceptual Design Report Volume 1: The LBNF and DUNE Projects*, 2016
- A. Aurisano, A. Radovic, D. Rocco et al. *A Convolutional Neural Network Neutrino Event Classifier*, 2016.
- Y. Fukuda , T.Hayakawa , E.Ichihara et al. *Evidence for oscillation of atmospheric neutrinos*, 1998.
- F. Capozzi, E. Lisi and A. Marrone et al. *Neutrino masses and mixings: Status of known and unknown 3ν parameters*, 2016
- J. Sonneveld. *Searches for physics beyond the standard model at the LHC*, 2018.
- P. Singh. *Extracting neutrino oscillation parameters using a simultaneous fit of the ν_e appearance and ν_μ disappearance data in the NOvA experiment*, 2017
- C. Szegedy, C. Hill and Y. Jia et al. *Going deeper with convolutions*, 2014.
- P. Adamson, C. Ader and M. Andrews et al. *First measurement of muon-neutrino disappearance in NOvA*, 2016
- P. Adamson, C. Ader and M. Andrews et al. *First measurement of electron neutrino appearance in NOvA*, 2016
- W. McCulloch and W. Pitts. *A Logical Calculus of Ideas Immanent in Nervous Activity*, 1943.
- C. Andreopoulos , C. Barry , S. Dytman et al. *The GENIE Neutrino Monte Carlo Generator PHYSICS & USER MANUAL*, 2015.

C.Andreopoulos , A.Bellb, D.Bhattacharyab et al. *The GENIE Neutrino Monte Carlo Generator*, 2009.

O. Lalakulich, K. Gallmeister, U. Mosel et al. *Neutrino nucleus reactions within the GiBUU model*, 2011.

R. Acciarri, C. Adams, R. An et al. *Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber*, 2017.

Y. Ganin, E. Ustinova, H. Ajakan et al. *Domain-Adversarial Training of Neural Networks*, 2016.

Z. Bousmalis, G. Trigeorgis, N. Silberman et al. *Domain Separation Networks*, 2016.

G. Louppe, M. Kegan K. Crammer et al. *Learning to Pivot with Adversarial Neural Networks*, 2017.

R. Brun, F. Rademakers et al. *ROOT - An object oriented data analysis framework*, 1997.