STUDENT JOURNAL OF PHYSICS

Fitting IceCube Neutrino Path models using Neural Networks

Mohammed Mohammed Salih [1], Jessie Micallef [2,3], Dirk Colbry [2]

[1] Department of Computer Science and Engineering, Junior, University of Michigan, Ann Arbor, USA
[2] Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, USA
[3] Department of Physics, Michigan State University, East Lansing, USA

ABSTRACT

To check if machine learning is an appropriate method to attack the problem of neutron detection in IceCube experiment, a virtual IceCube simulator (VIceCube) was created which generates phantom data to simplify and control the data set. Machine learning techniques (neural networks) has been tested on the phantom data to determine the limitations of the approach such as sensitivity to noise and measurement precision/error.

Keywords: *Neutrino, IceCube, Machine Learning*

1. INTRODUCTION

Neutrinos are small, fundamental, nearly massless particles. Their rare interaction with matter makes it difficult to determine their precise mass along with many of their other properties. Researchers on the IceCube experiment are working to constrain the values of the neutrino masses, along with searching for sources of astrophysical neutrinos and using them to search for dark matter. The IceCube Neutrino Observatory [1] is a particle detector located inside the ice in the South Pole that records the interactions of neutrinos with ice (Figure 1). The instrumented ice spans about a 1km cubed grid, with the detectors arranged in a hexagonal shape buried about 1.5 km under the ice. Each hole in the grid has a 2.5 km string, with 60 light sensors called photomultiplier tubes spaced out along the last kilometer. The interaction of a neutrino with the ice creates a charged particle that travels faster than the phase velocity of light in ice, making it emit light inside of the ice along its path (Cherenkov radiation). The emitted photons are detected by the photomultiplier tubes. Using the array of detectors' measurements, analytics and machine learning can be used to determine the angle, origin of interaction, speed, and energy of the incoming neutrino.

The focus of my current project is to reconstruct many neutrino interactions to gain statistics to better understand neutrino properties and sources. To check if machine learning is an appropriate method to attack this problem, a virtual IceCube simulator (VIceCube) was created which generates phantom data to simplify and control the data set. Machine learning techniques (neural networks) are tested on the phantom data set to determine the limitations of the approach such as sensitivity to noise and measurement precision/error.

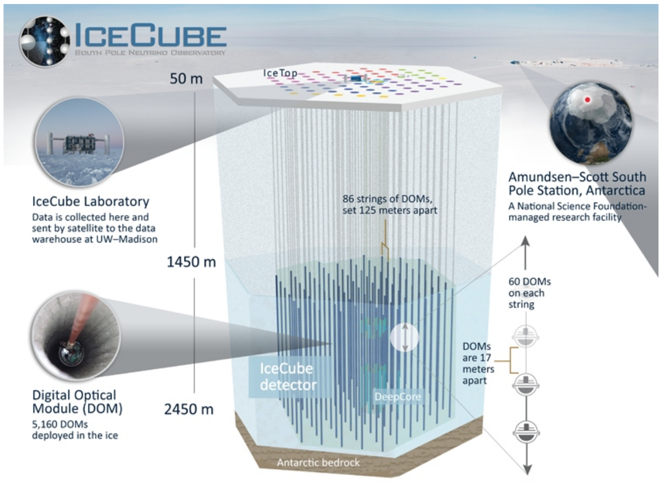

Figure (1): Full IceCube detector with 5160 detectors arranged in a 3D grid. [2]

2. METHODS

Phantom data generation uses random inputs and a model of the neutrino/ice interactions to generate artificial data (forward problem). This artificial data is then used as training data for a machine learning model to see if the inputs can be learned (inverse problem). Since everything is simulated, the true inputs are known and full evaluation of the approach can be conducted. The outputs of the simulator are the origin of interaction of the neutrino, which are x and y coordinates, and the angle as the neutrino crosses into the detector. The origin of interaction and the angle are randomly generated using a uniform distribution. The input generated by the simulator is the charge that each sensor measures as a “virtual” neutrino goes past each sensor. The charge is correlated to how much energy each sensor detects as a neutrino passes through the detector. For this simulator there are 121 “virtual” sensors. The current simulator uses a 2 dimensional grid (Figure 2) which will be upgraded to 3 dimensions in our future work. The neutrino follows a path given the angle that the particle hits the virtual IceCube. Given the origin of interaction of a neutrino, the following method was used to generate a charge for each sensor:

$$Charge = \frac{(MaxDistance - DistanceToEachNode)^2}{MaxDistance^2}.$$

Max Distance is the maximum distance a sensor can be from the path, which is a constant. Distance to each sensor is the distance of each sensor to the origin of interaction of the neutrino. An example of how the simulator works is shown in Figure 3.

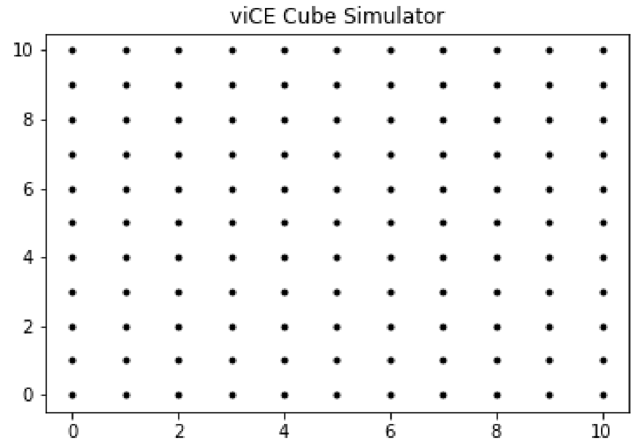

Figure (2): Example of a 2 dimensional grid with each Sensor marked evenly from 0 - 10.

To simplify the problem for machine learning, the initial simulator was tested without the path detection inside of Tensorflow [3]. Tensorflow is an open-source machine learning framework by Google. Given the simple nature of the phantom data, the neural network model was straightforward to train and was able to accurately predict the origin of interaction given only the charges of the sensors. To further test the approach, a Gaussian noise was added to the charges.

Origin_x	Origin_y
5.21925591133953	0.172719507669898

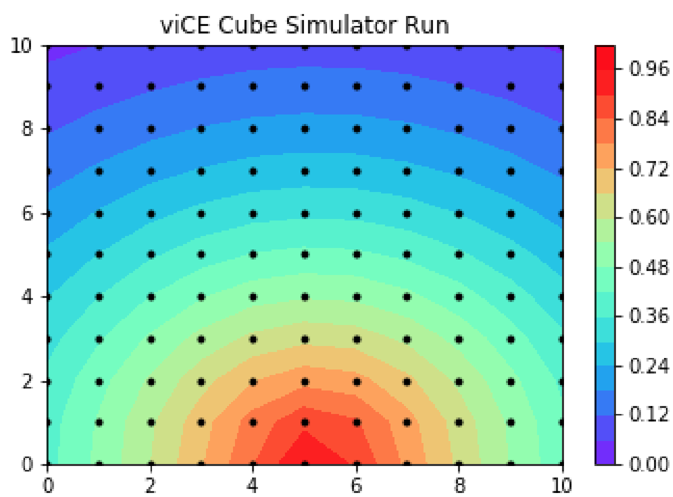

Figure (3): An example of a run of our simulator only with the real origin of interaction detected without a neutrino path

3. RESULTS

Using phantom data created by the VIceCube simulator (Figure 3), the hyper parameters for our neural network were determined. We have used a convolutional neural network (CNN), which consists of an input, output layer and multiple hidden layers. DCross validation was used to split our testing and training data. The Mean Squared Error (MSE), which measures the average difference between the estimated values and the real values, was used to determine the precision of the algorithm. As the initial simulator was rather simple, it had a very low error, close to 0. There was very little difference between the estimated and real values, and therefore the simulator is a good Machine Learning Predictor.

To understand how the machine learning responds to noise in the system, a Gaussian noise was added to the phantom data generator (Figure 4). The noisy data was trained and predicted with multiple intervals of Gaussian Noise (Figure 5), which shows an expected linear reduction in the performance of the predictor, making it more challenging to determine the hyper parameters.

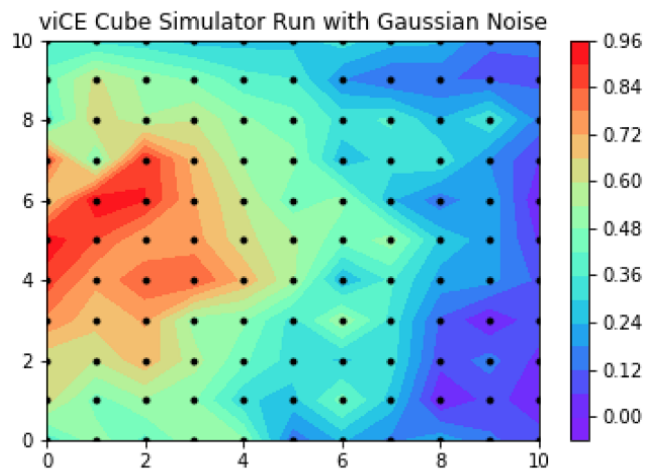

Figure 4: VIceCube simulator with Gaussian Noise. Sigma of noise = 0.1.

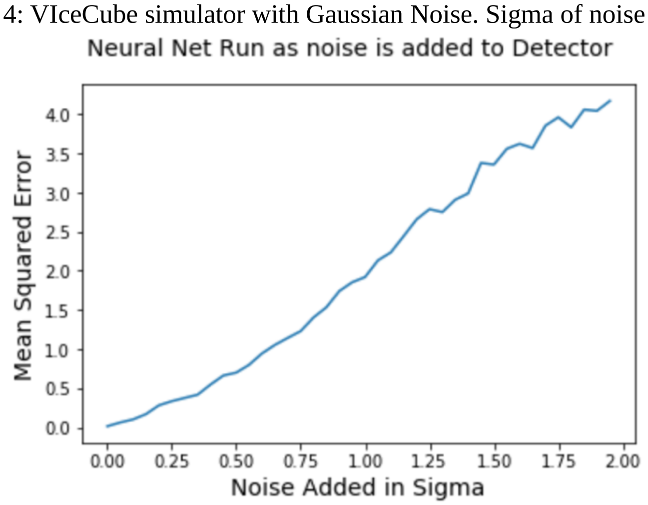

Figure 5: Comparison of Noise Added vs Mean Squared Error

4. FUTURE WORK

To begin this project, we started simple so we could build this study step by step to explore the robustness of using machine learning for a more complex data set. Future work for this project includes using a 3D simulator rather than the 2D simulator. Using 3D would be more accurate; however it would only simulate a cube and not a hexagonal ring. Predicting in 3D would likely be much more difficult, thus our predicting accuracy will be lower than the accuracy of the 2D detector. Other factors that have not been taken into account in this simulator are the energy and time components. We plan to add those in the future. Also, since we did not have access to the IceCube data during this project, we plan to test our neural network approach on the real IceCube data and measure its performance. The ultimate goal of the IceCube project is to determine the mass of the neutrino, and we believe that machine learning can be a useful tool to achieve this goal.

References:

[1]: Goldschmidt, A. (2002). Scientific Goals of the IceCube Neutrino Detector at the South Pole. *Nuclear Physics B - Proceedings Supplements,110*, 516-518. doi:10.1016/s0920-5632(02)01552-9

[2]: Icecube.wisc.edu. (n.d.). *Detector*. [online] Available at https://icecube.wisc.edu/science/icecube/detector

[3]: TensorFlow. (2018). *TensorFlow*. [online] Available at: https://www.tensorflow.org/