Team: Mu n' I: Neutrino Direction Detection
Members: Chinmaya Kausik, Benjamin Riley, Lukas Scheiwiller, Christopher Stith, Katja Vassilev

Description: The goal of our project was to determine an effective and efficient method for computing the direction of incoming neutrinos, using data collected at the IceCube Neutrino detector and provided via the associated Kaggle competition. The direction consists of two angles: an azimuth and zenith. Our goal was to minimize the mean angular error (a value between 0 and pi) between the true direction and predicted direction.

Background: The difficulty in this problem comes from the fact that a neutrino is never itself detected. Rather, when the neutrino hits a nucleus in the IceCube detector, the collision produces other particles which themselves produce a burst of light. These bursts are called Cherenkov Radiation and travel perpendicular to a "Cherenkov cone".

Data Overview: The data provided by Kaggle consisted of 660 parquet files, each consisting of approximately two hundred thousand individual neutrino events. An event consists of an arbitrary number of individual sensor activations, consisting of the time, charge and sensor ID number. It is possible for a single sensor to trigger multiple times for the same event. Additional metadata, including sensor geometry and true azimuth and zenith, along with the mean angular error (MAE) scoring metric were provided.

Approaches and Challenges: Our team decided to split into two primary working groups. One group focused on utilizing regression analysis combined with additional derived features to produce an estimate, while the second group focused on implementing a machine learning approach. Additionally, prior to these two approaches, we developed a data visualization module to visualize each neutrino event. This aided our preliminary data exploration by allowing us to see what was happening in the detector when a neutrino passed through.

The regression analysis consisted of extracting several physically meaningful features that were quite intensive to compute given the large scale of each event and the number of events. Since each of the detections in an event represent light traveling off of the Cherenkov cone, our models should find features relating to this cone. Upon reading the IceCube physics literature, we discovered IceCube's baseline model which extracts a time-best-fit line of the points in an event and then calculates an azimuth and zenith. Although this gives a preliminary guess, it does not take into account that we are potentially not getting points evenly distributed off of the Chernkov cone. Therefore, we attempted to find some features which would take this into account. Several features that we considered, but that turned out to not be relevant, included a more classical best fit line as well as the dot product between our two lines. Some features that we believed to be indicative of a more even distribution of points in the Cherenkov cone were the number of clusters in a single event as well as a higher MSE of our time-best-fit line. For points without an even distribution, we wanted to categorize them further to determine which side of the Cherenkov cone was more likely. We did this by finding the distribution of detections leaning to one side of the detector in each of its 3 coordinates. We ran into several difficulties trying to use all these features in a linear regression given that minimizing low MSE was not indicative of MAE. In fact our linear regression models performed worse than the baseline guess. So we performed a few regressions with various loss functions. Finally, we used tensorflow to be able to test with a custom loss function.This was able to improve upon our baseline.

The machine learning approach initially attempted to implement a simple fully connected neural network, based on associating to each sensor a list of characteristic information meant to consolidate the total amount of data passed to the machine. However, after initially promising results were invalidated after locating a bug in the DataLoader, it was clear a more sophisticated approach was necessary. Therefore, we decided to implement a convolutional neural network. We chose a three dimensional network with four channels as it seemed to allow us to encode the spatial position of the detectors more effectively than our previous approach. Additionally, we felt that our problem was sufficiently similar to image processing that a convolutional network seemed appropriate. After the convolutional step, we reintroduced three additional features derived from the computations of the regression group: a regression based initial guess of both azimuth and position, along with an estimate of the number of spatial clusters.

<u>Conclusion:</u> Our final results are ultimately unsatisfying, in the sense that we have not improved meaningfully on deterministic methods. However, we were able to match the initial baseline, and are confident that given additional training time and resources our convolutional model would be able to outperform the baseline.