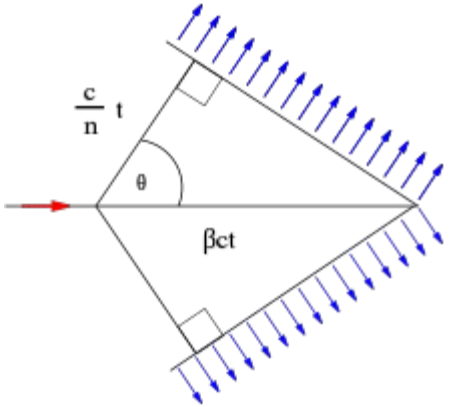
Neutrino Direction Detection

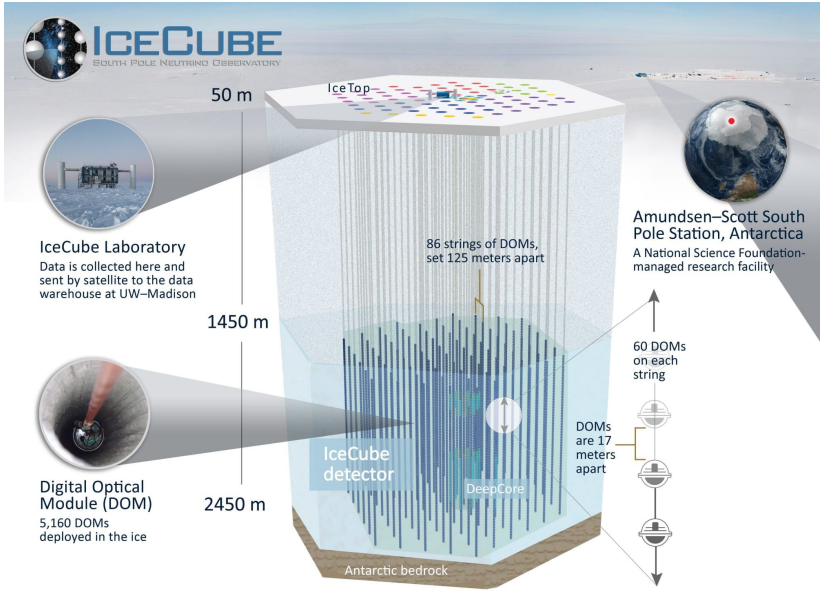
Team μ'n I: Direction Detection

Team Members: Chinmaya Kausik, Ben Riley, Lukas Scheiwiller, Chris Stith, Katja Vassilev
Mentor: Patrick Vallely

Ice Cube Background

- Doesn’t detect neutrinos directly, but instead Cherenkov radiation produced by neutrino passing through ice.
- The light travels in a cone
- Particle Detector in the South Pole consisting of 5160 optical sensors, with the goal of detecting Cherenkov Radiation





Goal

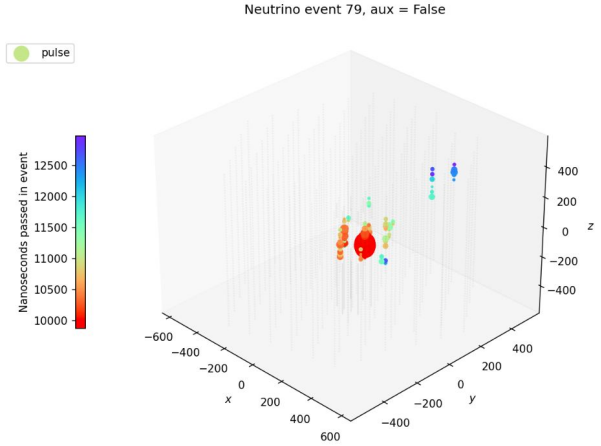
- Predict the direction of the neutrino by specifying its azimuth and zenith (angle in spherical coordinates)
- Minimize the Mean **Angular** Error (MAE): the angle between the actual and predicted direction
  - Takes values in the interval $[0, \pi]$
- IceCube has their own software to get the actual direction based on initial guess.
- Better guesses make running the software feasible. IceCube literature indicates an angle within 5 degrees is desirable.

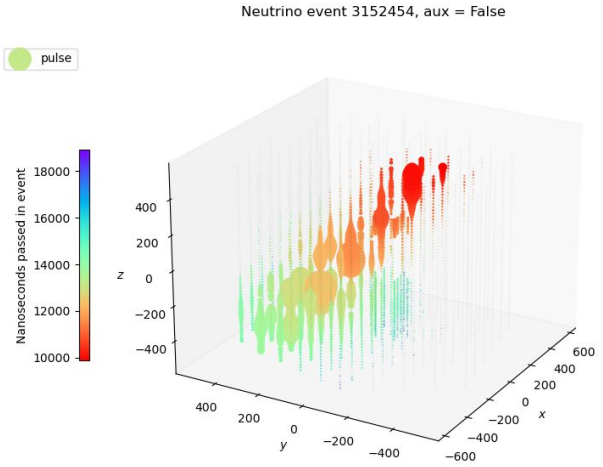
Description of Data

- Batches of neutrino events, each event has multiple pulses (often thousands).
- Each pulse detected on exactly one sensor.
- Batch files (Input data): Each row represents a pulse and contains 5 columns:
  - Event ID
  - Time of the pulse in nanoseconds
  - Sensor ID for the sensor that detected the pulse
  - Charge of the pulse
  - An auxiliary label (true/false: whether the pulse was of low quality)
- Meta file (Label data): Each row represents a neutrino event and contains 6 columns:
  - Batch ID for the batch containing the event
  - Event ID
  - Indices of the first and the last pulses in the event
  - Azimuth and zenith of the neutrino

Data Visualization

- Each data point represents a sensor that registered a pulse during that event
- The size of each point is the total sum of charges registered across all pulses
- The color of each point corresponds to the relative time of the first pulse registered at that sensor
- Can tune various parameters of the visualization





Feature Extraction

- The physics literature, gave a baseline model:
  - Compute estimated velocity of the particle by minimizing the distance of a sensor detection to the estimated position of the particle at the time.

- Focus on features which help us identify how the data is skewed on the cone:
  - MSE of our baseline model
  - Number of clusters in the event (values: 1 - 11)
  - Were the detections skewed to one side of the detector?
  - A comparison of PCA best fit line to time-best-fit

Model	MAE
Pre-Extraction LR	1.564
Baseline Physics	1.213

Linear Regression

- Tuned which parameters to use as well as tuning the cutoffs for categorical features(cutoffs: mse = 721, clusters = 2,9, skew = 90%)
- Best LinearRegression: uses predicted azimuth and zenith as well as categorical variables for skew, mse, clusters (on k-fold was 1.506)
- Best SGDRegression used loss function of huber

Since the regressions were strictly worse, we used tensorflow to minimize the loss that we care about

- Used optimizer ADAM and looked at various epochs

Model	MAE
Baseline	1.213
LinearRegression	1.506
SGDRegression (huber)	1.442
TensorFlow (Adam)	1.210

Fully-Connected Neural Network

Main motivation: Linear regression doesn't capture complex dependencies between pulses at different sensors, use universal approximators like neural networks.

Tried various architectures:

- 3-hidden-layer neural network with 5160*3 -> 100 -> 50 -> 10 -> 2 parameters
- 8-hidden-layer neural network with 5160*7+3 -> 4000 -> 2000 -> 1000 -> 500 -> 100 -> 50 -> 10 -> 2 parameters
- Used 3 hand-crafted event-specific features in the input to the 8-layer network.
- Tried both ReLU and tanh activation
- Optimization using both SGD (for 3 layer) and Adam (for both)

Convolutional Neural Network

Underlying idea: encode the geometry of the sensor into network

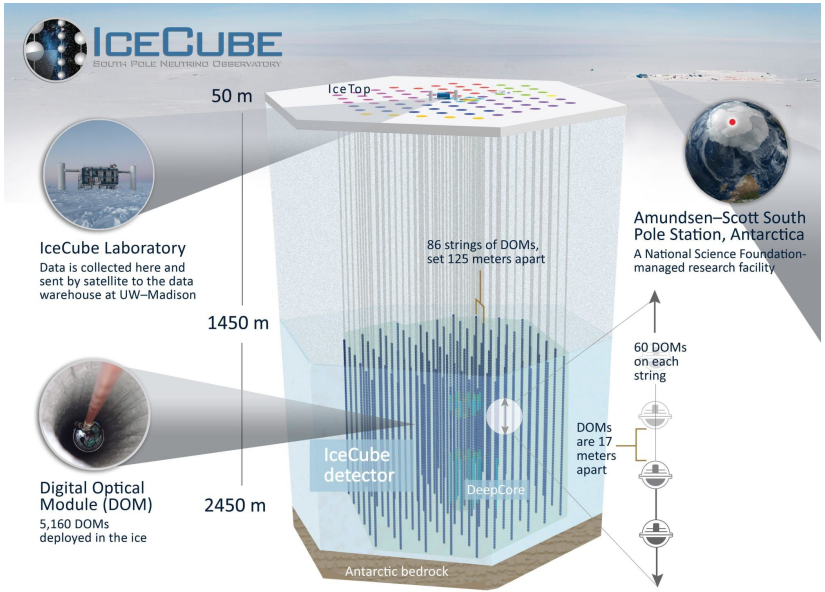
3D- Convolutional Neural Network

- 118 x-values
- 117 y-values
- 4974 z-values (divided up into 207 blocks)

Base tensor of dimension:

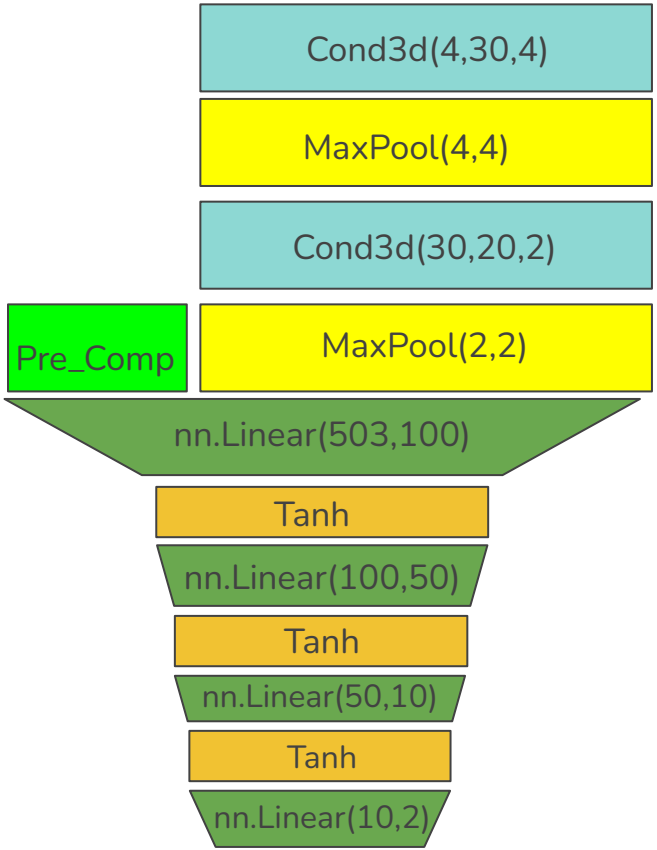
- 22 x-value blocks
- 20 y-value blocks
- 207 z-value blocks

Choose not to have 4D (otherwise base tensor dimension approx. 550 million)



4 channels

- First time
- Last Time
- Total Charge
- Scattering Metric



20 Epochs

Training accelerated using CUDA cores (Utilized an Nvidia 4070ti for training)

Reduced to smaller set with 75/25 split and validation set for cut-off

Validation MAE: 1.219

Test MAE: Currently running, optimistic given validation MAE.

Conclusion and Future Direction

- Best Tensorflow model performed at an MAE of 1.203
- Consider Graph Neural Networks as alternative spatial encoding method
- Add Transformers (as implemented on Kaggle winning solution)
- Fully utilize UofM GreatLakes cluster to accelerate training
  - Allow training on more data batches
  - Allows for faster tuning of input parameters
- Derive more meaningful initial features to enhance training

Acknowledgements

We would like to thank the following individuals:

- Oliver Knitter and Deniz Keles for consultation on architecture.
- Our mentor Patrick Vallely for his advice and support.
- Erdos Institute for providing this excellent learning opportunity.