The Ohio State University – Department of Physics

Machine Learning in Ultra-High Energy (UHE) Neutrino Analysis

Abdullah Alhag, with Prof. Amy Connolly

Introduction

What are UHE Neutrinos?

- Neutrinos are fermions with no charge and negligible mass. UHE neutrinos reside on (>10^{18} eV) of the energy spectrum.
- UHE neutrinos are theorized to be produced when cosmic rays interact with the cosmic microwave background.

Askaryan Effect:

- When neutrinos interact in a medium, they produce a particle bunch that propagates through ice. The particle bunch gives off EM radiation that is coherent for wavelengths greater than the 10-cm bunch size, equivalent to light frequency of around 1 GHz.

Radio Detectors

The Antarctica Impulsive Transient Antenna (ANITA)

- Radio detector suspended by a balloon in Antarctica.
- Surveys a large area of ice and records potential neutrino events.
- 4 flights each about one month long

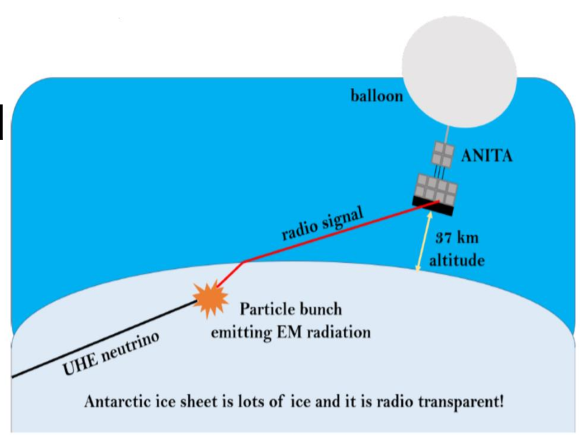

Fig. 1. The setup of ANITA

Askaryan Radio Array (ARA)

- 37 Radio Array Stations
- Each consists of an array of radio antennas drilled into the ice.
- Each station is separated by approximately 2 Km
- 3 of the planned 37 stations have already been built

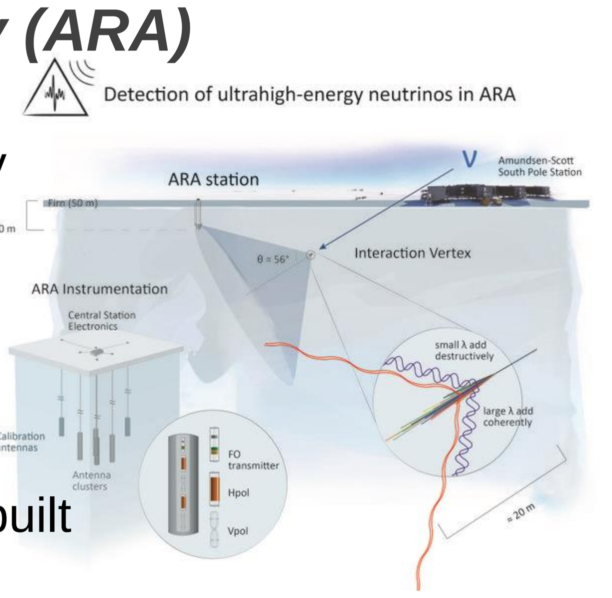

Fig. 2. The setup of ARA

Payload Blasts

- Payload blasts are background noises caused by the equipmentmen of ANITA flight.
- They are problematic as they pass the final analysis cuts that only UHE Neutrinos and cosmic rays are meant to.
- Payload blasts have a very unique waveform and are easy to distinguish.

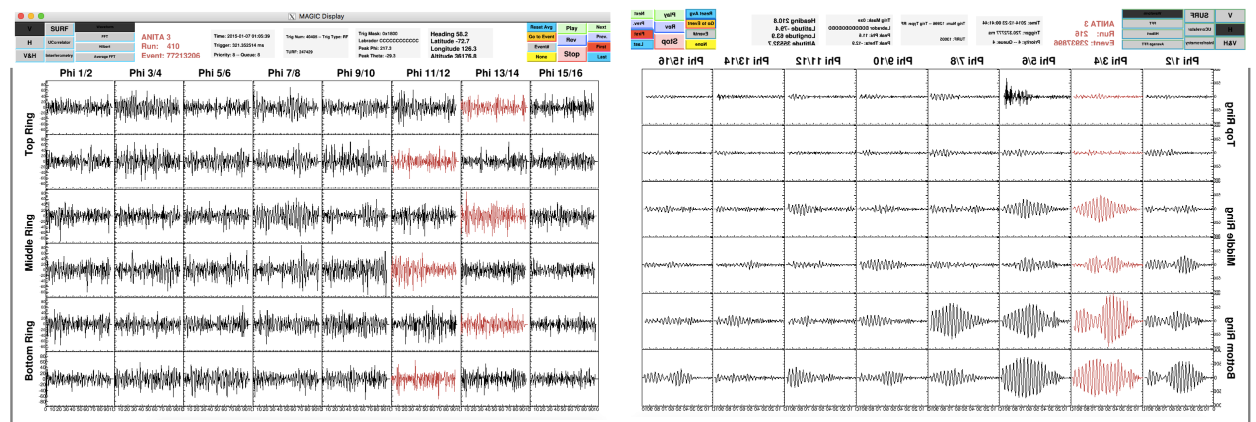

Fig. 3. Payload blast event (right) Vs. other event (left)

Why Machine Learning?

Convolutional Neural Network (CNN)

- CNN is a proven method in classifying images of different classes and works extremely well given enough training data.
- Goal: to able to classify payload blasts and remove them from data

Regression

- Regression analysis is used to describe a data set by a simple function.
- Goal: to model all background noise data and hence filter these events from the data set.

Regression explained

- The process starts with the selection of random functions.
- These functions are tested for their fitness score against the data and functions with the best fitness score are selected to parent the next generation.
- The functions are then evolved by either of the following:
  - Reproduction: the tree is copied without mutation to the next generation
  - Point/Branch mutation: a single node/branch mutates before added to next generation
  - Crossover: 2 tree parents are selected to produce 2 offspring by mutation as summarized in Fig. 4.

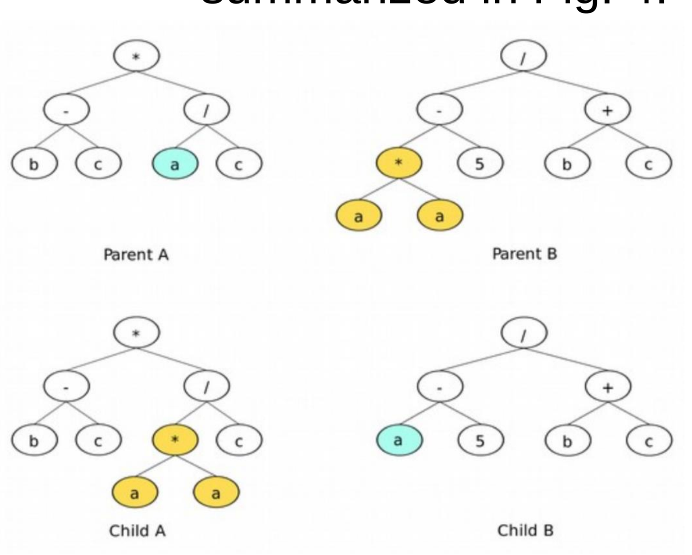

Fig. 4. An example of a crossover mutation process

Results

- Various implementation of regression analysis were used. Some of these were Karoo-GP, Eureqa, and HeuristicLab.
- Eureqa yield the best result in the shortest time.
- A plot of such model is shown below.

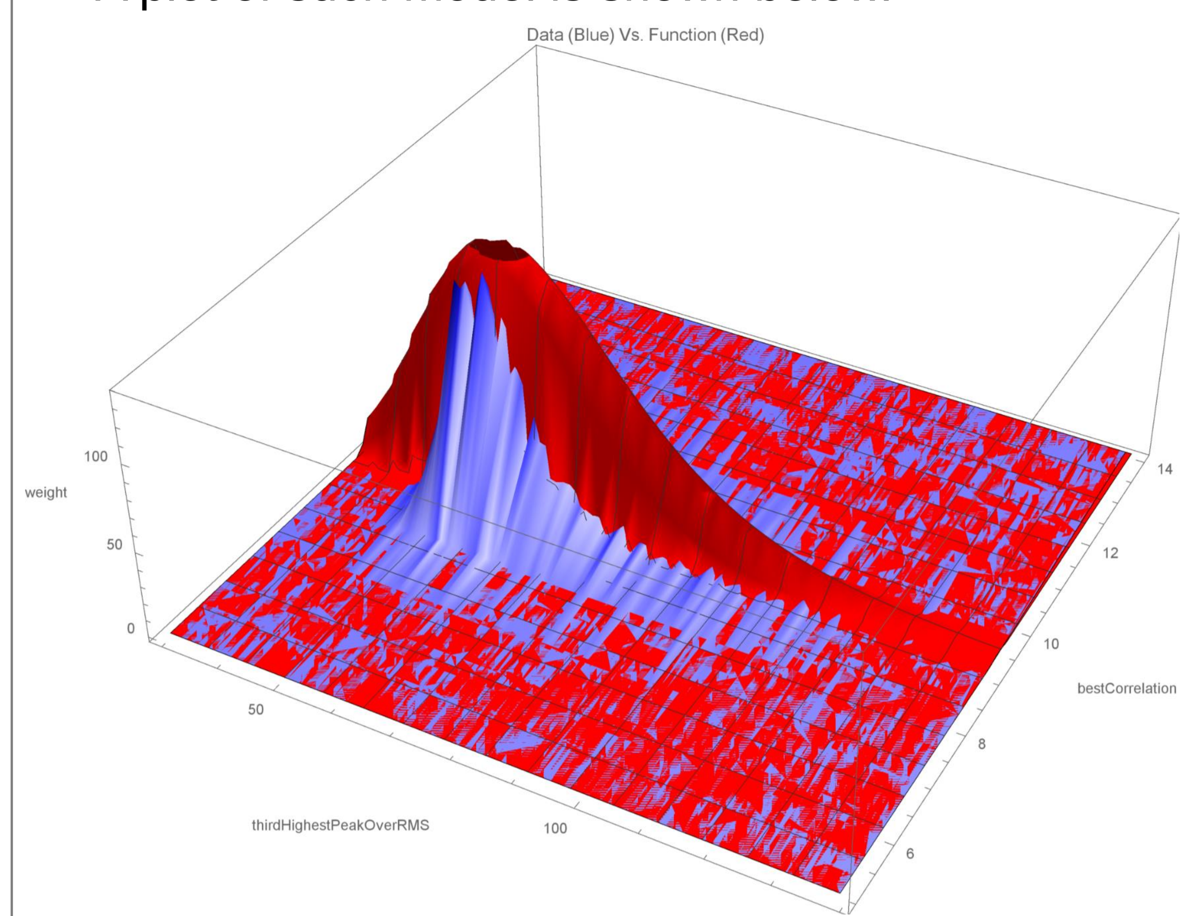

Fig. 5. A plot of the Background data (Blue) and the Eureqa model (Red)

Verifying the model

- It is very common for such models to be over trained. To verify that this was not the case, a pseudo experiment was conducted based on the model and compared with the data.
- This consisted of converting the Probability Density Function (PDF) from Eureqa to a Cumulative Density Function (CDF) by numerical integration.
- The data from the CDF were then used to generate PDF plots by random selection of points from the CDF data.
- If the resulting PDFs were similar to the original, we conclude that there was no over training.
- The last step was to compute the likelihood for each pseudo experiment and compare it to the expected.

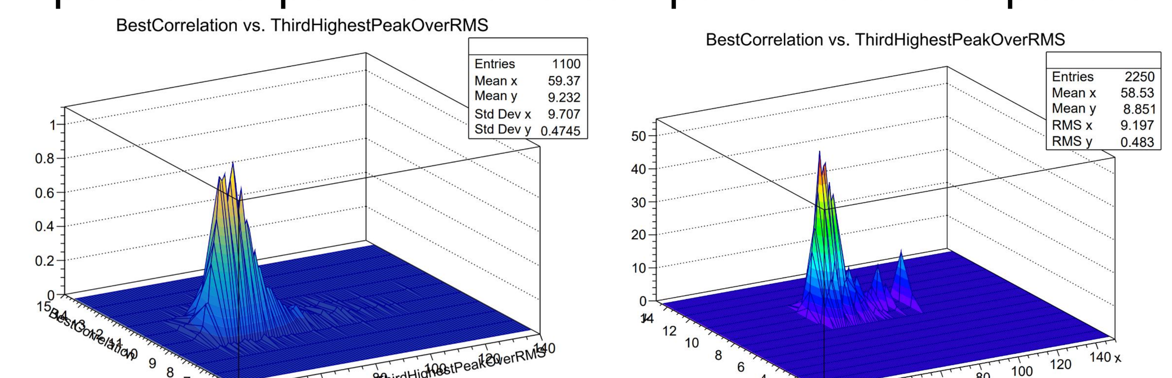

Fig. 6. A plot of the data normalized (left) Vs. pseudo (right).

CNN Explained

- The concept is best explained by 3*3 pixels model where only 4 characters are allowed; \, /, O, and X.
- We start by picking up filters, for this example we pick up / and \ filters.
- We pass the two filters over the characters and see how they fit (Convolution Layer) .
- Based on a threshold value we decide whether we found something or not, here either characters / or \ (Pooling Layer).
- We build filters by encoding our character as matrix in the / and \ vector space (Fully Connected Layer)
- The whole process of evolving filters is done by a gradient descent of an error function starting by a random guess.

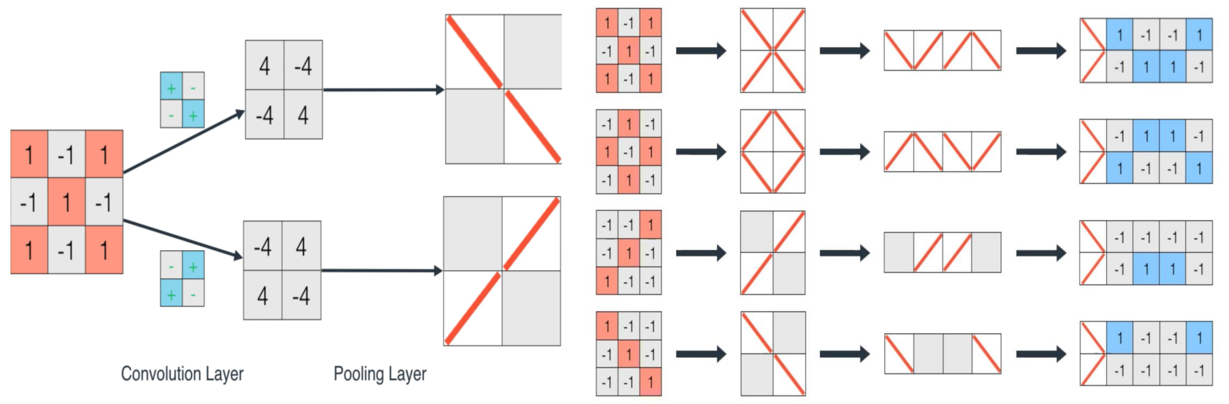

Fig. 7. Convolution/Pooling Layer (left) Fully Connected Layer (right)

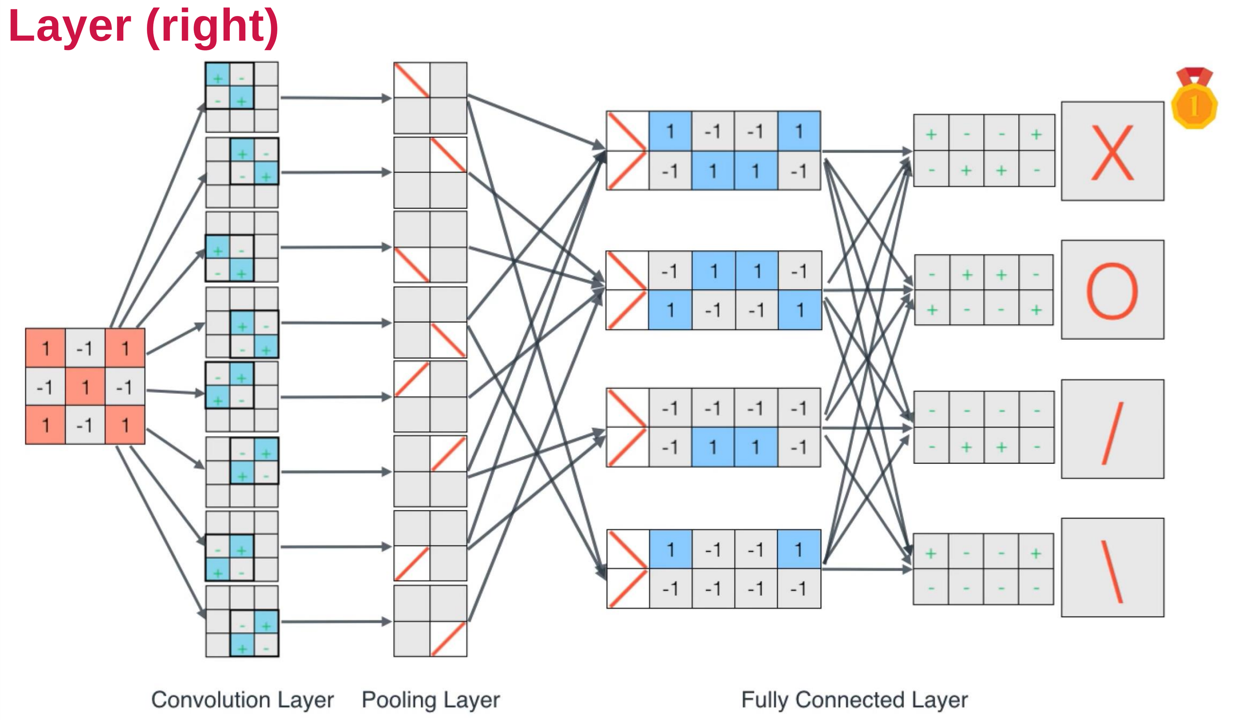

Fig. 8. The steps followed in classifying the X character

Results

- Building the CNN was met with success with the ability to classify over 1000 classes, some of which are special classes.
- To train the CNN to classify the payload blasts, data is being collected and it should be straight forward to classify thereafter.

Acknowledgements

Full acknowledgement to my advisor, Prof. Amy Connolly, who has guided me through almost every step of the way.