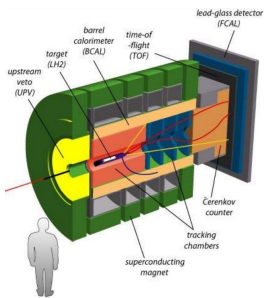


# Applying Machine Learning to Particle Detectors

Alexander Ercolani

## The Problem

Particle detectors are instruments used to collect data on high energy particle interactions. These interactions, called events, produce large numbers of particles, and in turn, a large amount of data. The raw data produced by the detectors must be processed so we can determine what to save for analysis and what to discard. However, the calculations used when recreating these events take time and a large amount of computing power. To combat this, machine learning can learn patterns to perform the same calculations more efficiently. This project explores the applications of machine learning on data from the GlueX Detector.



## Scope

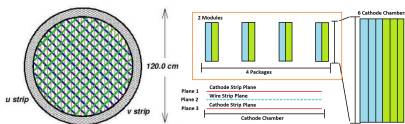
This project looks to develop a prototype neural network that takes raw detector data and outputs something meaningful to the reconstruction of particle paths. This requires learning about both neural networks and the GlueX detector in order to determine how to best approach the problem. The insights learned can be used to develop more sophisticated neural networks down the line.

## Data

For the purpose of development, rather than using real data, the project has used simulated data from monte-carlo experiments. The simulated data allows for more control over the type of particle events as well as what information is being reconstructed from each event. This helps us label our training data and provides clean data for prototyping. As well, to limit complexity, only data from one part of the detector, the FDC, was used in training.

## FDC

The forward drift chamber (FDC) is a cylindrical section of the detector that is used to trace particle trails in a three dimensional space. The FDC contains 24 circular chambers aligned along the particle beam axis. Each chamber consists three layers. The two outer layers are made of 192 cathode strips each. The middle layer is made of 96 anode wires. These chambers are capable of detecting the movement of charged particles due to current and voltage changes as the particle passes. The raw data from these wires is sent to a computer which reconstructs the particle path. This data can ultimately be used to determine what type of particle passed through the detector.

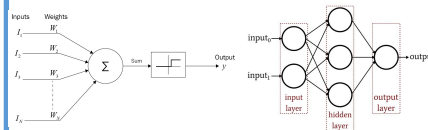


## Keras and Tensorflow

Keras is a high level API for python that allows easy implementation of neural networks using Google's Tensorflow library. By using Keras we are capable of building several neural network models while easily changing the input, output, and hidden parameters.

## Neural Network

Neural networks are a type of machine learning algorithm that imitates the structure of the brain. This project focuses specifically on feed forward neural networks, also known as multilayer perceptrons. These types of networks work by using layers of artificial neurons. Each neuron in a given layer is connected to all the neurons in the next layer. Each of these connections carries a weight. These weights are multiplied by the input from previous layers' neurons and fed into an activation function in the next layer which determines the output of each neuron. The network adjusts these weights based on a cost function which compares the expected output and the current output.

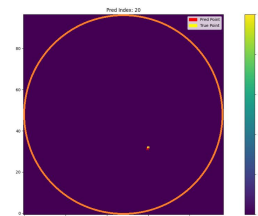


## Training

For the purposes of this project we looked at single particles passing through the FDC. Since we used simulated data, for each particle interaction we are also given the coordinates that the particle hit as it passed through each chamber. These coordinates act as the training labels. Therefore, the network is expected to be given raw data from a single chamber and determine where in the detector a particle hit. In order for the network to be able to pick up on meaningful patterns, it is necessary that we train it with statistically significant data. This is accomplished by using large quantities of data to ensure the data contains hits which represent the expected uniform distribution. Our first successful network used 50,000 training hits, ensuring that in a 1mm by 1mm section of the detector the data should have approximately 5 hits for that given section.

## Results

Below is an image generated by our network, in it we can see a yellow square representing the actual location a particle hit and a red 'x' representing the networks predicted location. On unseen data it is expected that the network will be within 2mm of the actual hit. This discrepancy can be minimized by using larger sets of training data.



## Future

The scope of this project was to study applications of neural networks in particle detectors and has lead to the working prototype network described previously. It is now possible to scale up and implement networks which can detect and differentiate multiple particle hits at the same time, eventually working towards the total reconstruction of particle paths.

## Acknowledgements

This project was funded by the Summer Undergraduate Research Funding (SURF) award. The project supervisor is Professor Richard Jones.