ML Tools @ NERSC
(Plus A Science Example!)

Evan Racah
NERSC
8/22/16
Available Tools

Deep Learning Frameworks

- **Theano** - flexibility, not for beginners (good for research)
- **Keras / Lasagne** - Theano-based but higher-level for ease of use
- **TensorFlow** - ease of use and flexibility, large, growing community, some *multi-node* support
- **Caffe** - high performance (IntelCaffe with performance highly optimised for KNL), *multinode* (*no programming necessary*)

General Machine Learning:

- **Scikit-Learn** - great for non-image based machine learning, easy to use, support for wide range of algorithms
- **Spark** - *multinode*, great for data parallel, relatively easy to use, support for only a subset of ML algorithms
How Do I Use These Tools at NERSC?

Deep Learning Module

- Python deep learning tools available under the deep learning module
- Just one module load call and then they are ready to be imported in your python script!

Scikit-Learn

- available in standard python module and deep learning

Caffe and Spark

- available as separate modules

Interactive computing:

Q: You a big Jupyter notebook fan?!

A: No problem. The iPython deeplearning kernel allows for interactively using the deep learning module python tools

For more information visit http://www.nersc.gov/users/data-analytics/data-analytics/deep-learning/
Science Problem!
Daya Bay Antineutrino Detector Analysis
Daya Bay Reactor Neutrino Experiment

Goal:
- Determine neutrino parameter, $\theta_{13}$
- Will provide clues to extend Standard Model

Experiment
- Inverse Beta Decay (IBD)
  - Antrineutrino reacts with a proton, decays to a positron and neutron
  - Reaction measured by antineutrino detector
Experiment

- Detectors
  - (192 PMT sensors in cylinder) measuring charge.
  - Events above a certain energy “trigger” a snapshot
- Snapshots (8x24 arrays of floats) analyzed in order to infer $\theta_{13}$
- Signal events
  - IBD prompt (positron)
  - IBD delay (neutron)
    - Occurs several microseconds after
- Non-neutrino events = backgrounds:
  - Flasher (detector malfunction)
  - Muon
  - Other
    - Everything else
    - (contains false negatives?)
Why Deep Learning?

Deep learning could help:
- More powerfully discriminate between signal and noise
- Identify new unexpected sources of noise
- Determine structure in the signal as well as in the different types of noise
- Would be interesting to see if deep learning could group together different physical phenomena

Our Approach
- Learn an *unsupervised* feature vector using a *convolutional autoencoder*
- This can help cluster related events, revealing patterns
Convolutional Autoencoder

What is a Convolutional AE?
- neural network trained to reconstruct its input
- Encoder
  - Transforms input image into "feature" vector
- Decoder
  - Attempts to reconstruct input image from this vector
- When certain restrictions applied to network
  - forced to learn only the most important features of data

Why an autoencoder?
- Manifold Assumption
  - We assume most of the data sample images come from small number of physical events

Why convolutional networks?
- We treat our data as images
  - Parameter sharing/Translation equivariance
    - Fewer parameters to learn
    - Translating a feature results in an identical but translated representation
  - Translation Invariance
    - whether feature is present important, not so much where it is
  - Both important