Machine Learning Coding Tutorial

This tutorial focuses looks at building convolutional neural networks for supervised classification problems.

This url: https://confluence.slac.stanford.edu/display/AI/Machine+Learning+Coding+Tutorial

Computer Setup/Graphics

1. Have a laptop
2. get a terminal on a psana machine, ideally with graphics
    a. use nomachine - probably best Remote Visualization
    b. mac - can also install XQuartz, and use the terminal program
    c. windows - possible Xming , other heavyweight options: cygwin, setting up a virtual box running linux

Software

In bash, source ~davidsch/scripts/mlearntut-setup.sh for environment.

If you use cshell, start a bash shell first, or look at the script to modify for cshell.

The script clears environment variables like PYTHONPATH, LD_LIBRARY_PATH, and sets PATH to the rhel7 default, and puts a miniconda bin dir at the frount. Then it activates the 'mlearntut' environment in the conda install.

We will use Keras and tensorflow

There appears to be a new Keras like interface just for tensorflow: http://tflearn.org, don't have any experience with this yet

Data

Presently the data files are at /reg/d/ana01/temp/davidsch/ImgMLearnSmall
you can't see the data from pslogin, you have to be on the psana nodes.
There are 701 files there, each has 500 rows of data.

Some notes on the hdf5 files: final-h5-output

Code

1. from psnxserv if you used nomachine, or pslogin (a machine with outside internet), do git clone https://github.com/davidslac/mlearntut.git
2. For ex09, we will use the pslogin shell, so best to leave it up
3. start new terminal
    a. you can run gnome-terminal or xterm from your pslogin shell
    b. if not already at ssh -Y to pslogin, then ssh -Y to psana (ssh -X should work also)
4. cd to the mlearntut directory you made, from the pslogin terminal

Running

If resources are getting tight on the interactive nodes, you can launch jobs on the batch. The model in the last exercises is to pretty big to run on the interactive nodes.

You can launch the jobs in 'interactive' mode so you can see output, however you can't do any graphics or plotting from jobs running on batch. We will be able to use tensorboard from batch. Here is an example:

```
bsub -q psanaq -I python ex01_keras_train.py
```

I'm getting a big MPI warning when I run these jobs. I think it is safe to ignore it. To do so, do

export OMPI_MCA_mpi_warn_on_fork=0

I am not sure why we need to set mpi_warn_on_fork=0. Although we aren't using MPI, our hdf5 library is built with it, so some MPI aware code is running and complaining.

The examples in the tutorial do

```
sys.stdout.flush()
```

calls so that we can see print output more immediately while running on batch interactively, without the flush calls, the batch system queues up program output.