Pseudo code for algorithm

* Cython and MPI
* data is stock vs date
* Neural network

N = # of nodes

* Import packages: MPI, theano
* Read dataset
* Subset data for each data to mini-batches equivalent to 64 processors of memory? Will have N-1 mini-batches. Won’t this take to long and be inefficient?
* Pick W0 as initial vector of derivatives
* Start N nodes (MPI). Send W0 to node #1. Send mini-batches to nodes #2-N.
  + Node #1 (main node): will serve as parameter server. Reserve one core in this node
    - send W0 to all other nodes to commence algorithm
    - for each *j* in 1 to epoch\*(N-1) [will we have to code this for loop?]
      * receive W from node *x*
      * update Wj = Wj-1+W(step)
      * send Wj to node *x*
  + Node #2 — Node #N:
    - Receives a mini-batch using MPI
    - for each *i* in 0 to epoch:
      * Parallelize across 64 cores:
        + Run neural network with data in parallel (64 processors)
        + …
      * Result is vector of derivatives, Wi. Send to node #1
      * Fetch Wi+1 from node #1
* Save Wepoch and close all nodes