A2
Deep Learning 2015, Spring

Throughout this assignment we introduce STL-10, here the short description from their website `http://cs.stanford.edu/~acoates/stl10/`.

"The STL-10 dataset is an image recognition dataset for developing unsupervised feature learning, deep learning, self-taught learning algorithms. It is inspired by the CIFAR-10 dataset but with some modifications. In particular, each class has fewer labeled training examples than in CIFAR-10, but a very large set of unlabeled examples is provided to learn image models prior to supervised training. The primary challenge is to make use of the unlabeled data (which comes from a similar but different distribution from the labeled data) to build a useful prior. We also expect that the higher resolution of this dataset (96x96) will make it a challenging benchmark for developing more scalable unsupervised learning methods."

Task
Your task is to use the data and train a model on Mercer. As with the first assignment you are required to hand in a deployment of your model, which includes the weights and the ability to produce predictions on top of the raw test set. This time the deployment is a bit more complex and the overall standards are a little higher.

2015-02-26 ADDITION: Further, feel free to ignore the training schedule mentioned on the website. Use all the training examples for training (for example 4500 for training, 500 for validation) and don't follow the folds.

Evaluation

- 30% - Kaggle performance - at least beat my score! (0.52650 on the public leaderboard based on the posted architecture)
- 40% - Three page paper plus one extra page for references (brevity preferred) - do mention your failed experiments and use a NIPS/CVPR template (LaTeX).
- 30% - Simple, readable, commented code of final, working algorithm able to execute on test data as found on Mercer

The paper should consist of a

- description of the architecture (number and type of layers, number of neurons, size of input)
- description of the learning techniques applied (which data augmentations?, used dropout?, etc.)
- description of the training procedure (learning rate, momentum, error metrics used, train/validation split, training/validation/test error)

Here is a link to a good paper (you don't need an abstract/related work/introduction/conclusion).
`http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf`

Your final submission should be a link to a single HPC Mercer qsub bash script called "TEAM_NAME.sh" (call caps, team name spelled as on the homepage's team roster) exposed through CIMS web hosting and sent to cpuhrsch@nyu.edu before the deadline.
I will then execute a single script, that calls your (and all other team's) scripts. Your qsub script will create a folder in the current user's home directory (found under: ~) with your team's name as the folder name. It will then fetch all of your code and assignment pdf and execute some program/torch script, that produces your best predictions and saves them out to "predictions.csv". You may use your own binaries or the cluster's binaries. Make sure you setup the right user permissions, when you make use of your binaries (you can test this by executing a program, that your buddy installed in his home directory). The data can be found under /scratch/courses/DSGA1008/A2 and the binaries under /scratch/courses/DSGA1008/bin. This is supposed to make you more familiar with UNIX and a high performance computing environment, plus also able to deploy code in a more concise fashion.

Happy training!

Additional resources
`https://github.com/clementfarabet/lua---mattorch`