Computer Learning

Going Forward

Statistics

- Toronto Intro Stats
- PSTAT 120A/B

Linear Algebra / Multivariable Calculus

- MIT Linear Algebra
- Math 108A/B
- CCS Linear Algebra
- MIT Multivariable Calculus
- CCS Multidimensional Analysis

Convex Optimization

- Convex 1
- Convex 2
- Stanford Lagunita
- ECE 271 Principles of Opt.

More Optimization

- Discrete Optimization
- Berkeley Optimization

Information Retrieval

Urbana Coursera

Stanford IR Course

Machine Learning

- Oxford ML
- UW Coursera ML
- Stanford Grad ML
- CMU Grad ML
- Caltech ML

Statistical Learning

- Stanford Statistical Learning
- MIT Statistical Learning
- Elements of Statistical Learning (Textbook)

Probabilistic Graphical Models

- Stanford PGM's
- CMU PGM's

Clustering

- Urbana Clustering
- CMU Grad ML

Deep Learning I

- Hinton's Class
- Deep Learning Summer School Slides
- DLLS Videos
- CS 290D (Xifeng)

Deep Learning

- Deep Learning for NLP
- Manning Tutorial Deep Learning for NLP
- CS231n: Convolutional Neural Networks for Visual Recognition

Deep Learning Extras

- UFLDL Stanford
- Github Deep Learning Reading List
- Theano Python

NLP

- Columbia NLP
- Stanford NLP
  - Parsers, Grammar etc.
- Stanford NLU

Games

- General Game Playing
- Game Theory Intro
- Game Theory Advanced

Conferences NLP

- ACL
  - CoNLL
- NAACL (NAACL 2015)
- EMNLP

Conferences ML/ Deep Learning

- NIPS
- IJCAI
- AAAI
- Sub-conferences
  - COLT

Questions

CNN: What other applications can this be used for?

Independent of the data sets themselves, how much storage space do the bigger neural networks take up (e.g. storing all of the weights/parameters)? If any of you have used or are using neural networks, can you explain your project and why you chose to use a NN instead of some other method?

Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264– 4096–4096–1000.

Our Faculty

Faculty

Ece

Grad Colloquium

Class Stats

Easiest Class:

2.375 = Probability Review

2.5 Basic Text Processing = Edit Distance

Hardest Class:

4 = Optimization = CNN

3.625 Statistical Decision Theory = Word Embedding = Deep Learning I

Most Boring:

3.125 = Linear Algebra Review

3.25 = Linear Algebra Advanced

Most Fascinating:

4.5 = Deep Learning II

4.375 = Deep Learning I

Overall Score: 4.625

Thanks!!!