

CS 636/838: BUILDING Deep Neural Networks

Jude Shavlik
Yuting Liu (TA)

Deep Learning (DL)



- Deep Neural Networks arguably the most exciting current topic in all of CS
- Huge industrial and academic impact
- Great intellectual challenges
- Lots of cpu cycles and data needed
- Currently more algorithmic/experimental than theoretical (good opportunities for all types of CS'ers)
- CS Truism: Impact of algorithmic ideas can 'dominate' impact of more cycles and more data

Waiting List



- Bug in Waiting List fixed
- So add yourself ASAP
- If you decide to DROP, do that ASAP
- We'll discuss more THIS THURS



Class Style

- This is a 'laboratory' class for those who already have had a 'textbook' introduction to artificial neural networks (ANN)
- You'll mainly be LEARNING BY DOING rather than from traditional lectures
- Will be implementing ANN algorithms,
not just using existing ANN s/w
- You need strong self motivation!

Logistics



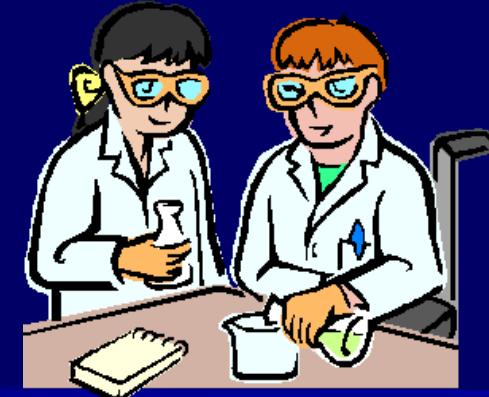
- We'll aim to use Rm 1240, since large desk space for coding and good A/V
- Room needs a key, so be patient
- Can 'break out' into nearby rooms if needed (eg, Rm 1221)
 - might need to use RM 1221 when I'm out of town
- Building gets locked at some point, so be careful about leaving temporarily

More Details



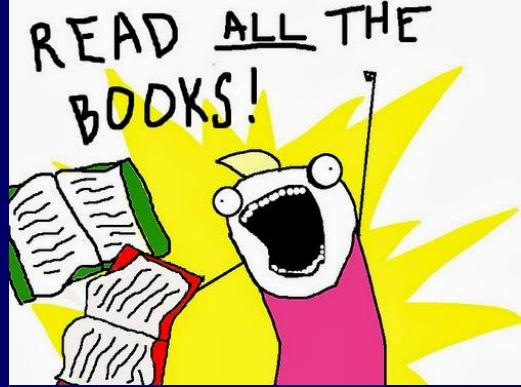
- Prereq: CS 540 or CS 760
- Meant to be a 'beyond the min' class
 - Doesn't count as elective toward CS BS/BA degree
 - Also not a CS MS 'core' class
- I expect a higher GPA for awarded grades than typical
 - Maybe 3.5-3.7 for cs 638 (ugrads)
 - Maybe 3.7-3.9 for cs 838 (grads)
- Grading more about effort/creativity than 'book learning'
- **Attendance important** (eg, listen to others' project reports)
- Likely to be 1-2 quizzes on ANN and Deep ANN basics

More Details (2)



- Lots of coding and experimenting expected
 - No waiting until exam week to start!
 - Generate and *test* hypotheses
- After initial 'get experience with the basics' labs, you'll work on four-person teams
 - Each group chooses its own project (but my approval will be needed)
- We'll use Moodle (for turning in code, reports, etc) and Piazza (for discussions)
 - SIGN UP AT piazza.com/wisc/spring2017/cs638cs838
- Fill out and turn-in surveys now

Read



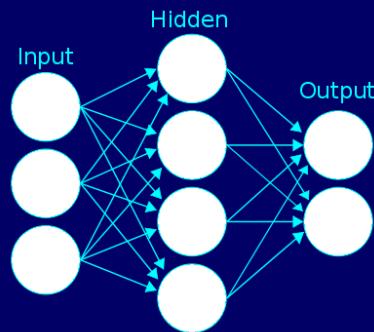
- Read Intro to Deep Learning textbook (free on-line) – overall the book is advanced, written by leaders in DL

"As of 2016, a rough rule of thumb is that a supervised deep learning algorithm will generally achieve acceptable performance with around 5,000 labeled examples per category, and will match or exceed human performance when trained with a dataset containing at least 10 million labeled examples.

- Re-read your cs540 and/or cs760 chapters on ANNs

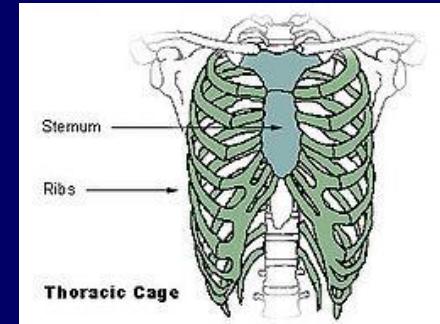
Initial Labs

– work on next several weeks



- Lab 1: Train a perceptron (see next page)
- Use 'early stopping' to reduce overfitting (Labs 1 and 2)
- Lab 2: Train a 'one layer of HUs' ANN (more ahead)
 - Try with 10, 100, and 1000 HUs
 - Implement and experiment with at least two of
 - (i) **Drop out**
 - (ii) **Weight decay**
 - (iii) **Momentum term**
- OK to work with a CODING PARTNER for Lab 2 (recommended, in fact)
- **You can only share s/w with lab partners, unless I explicitly give permission (eg, useful utility code)**

Validating Your Perceptron Code



- Use my CS 540 'overly simplified' testbeds
 - Wine (good/bad); about 75% testset accuracy for perceptron
 - Thoracic Surgery (lived/died); about 85% testset accuracy
 - Use these for 'qualifying' your perceptron code (ie, once you get within 3 percentage points on these, turn in your perceptron code for final checking, on new data, by TA)
 - Ok to use an ENSEMBLE of perceptrons
- See my cs540 HW0 for instructions on the file format and some 'starter' code



Calling Convention

- We need to follow a standard convention for your code in order to simplify testing it
- For Lab 1 (code in Lab1.java):
`Lab1 fileNameOfTrain fileNameOfTune fileNameOfTest`

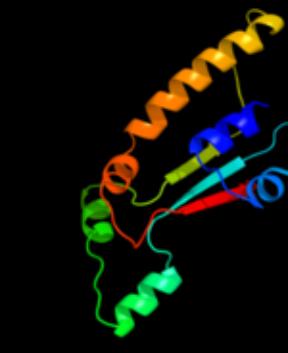
UC-Irvine Testbeds



You might want to handle multiple UC-Irvine testbeds

- Ignore examples with missing feature values
(or use EM with, say, NB to fill in)
- Map numeric features to have
'mean = 0, std dev = 1'
- For discrete features, use '1 of N' encoding
(aka, 'one hot' encoding)
- Use '1 of N' encoding for output (even if N=2)

Validating Your One-HU ANN



- Let's go with Protein-Secondary Structure
 - has THREE outputs (alpha, beta, coil)
 - features are discrete (1 of 20 amino acids)
 - need to use a 'sliding window' (similar in spirit to 'convolutional deep networks')
 - approximately follow methodology here (Sec 3 & 4)
- Testset accuracy
 - 61.8% (our train/test folds; some methodology flaws)
 - 64.3% (original paper, but worse methodology)
 - turn in when you reach 60% (using early stopping)

Predicted Secondary Structure

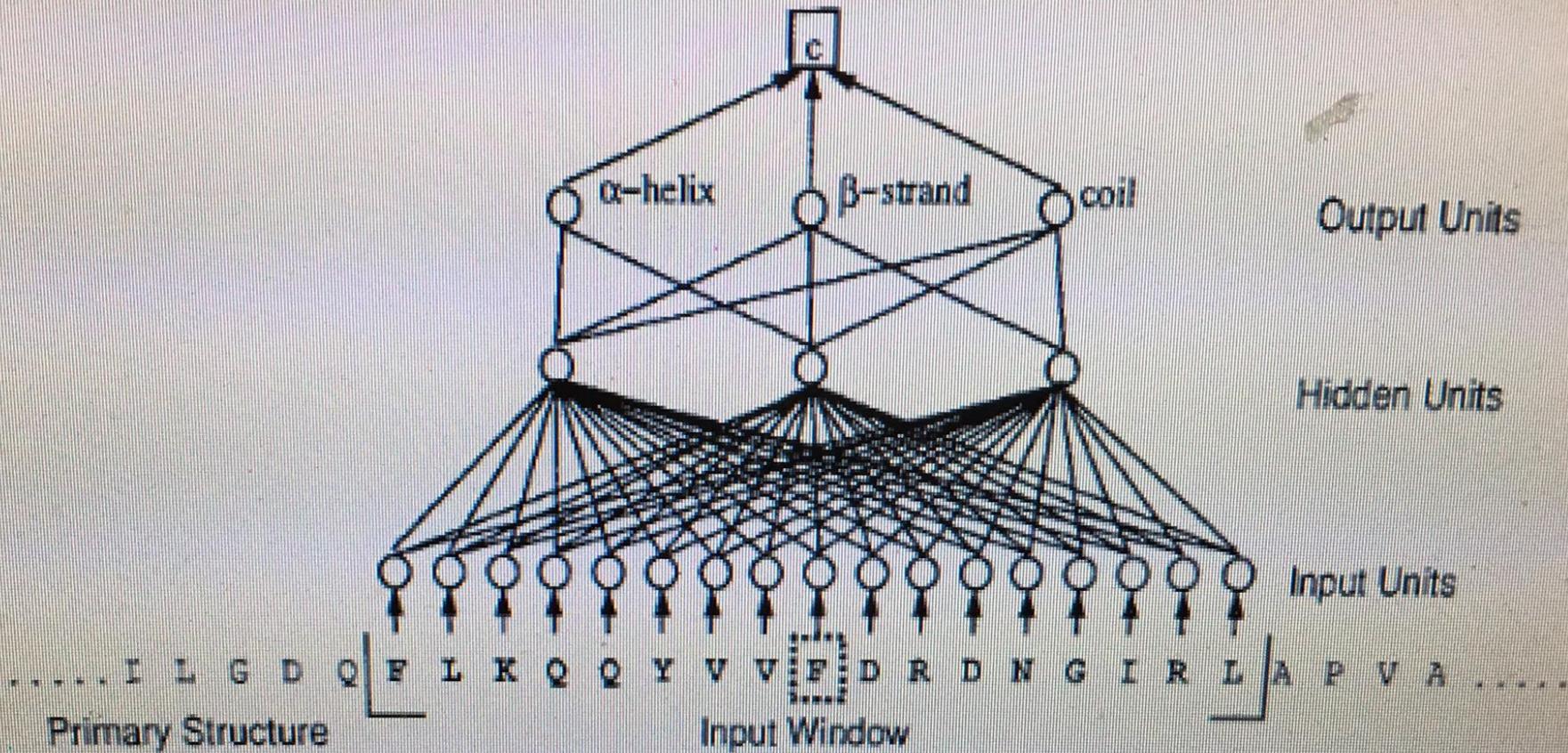
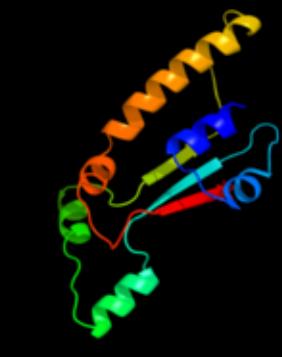


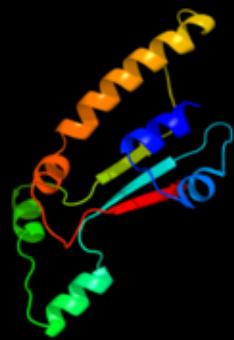
Figure 4. Neural network architecture used by Qian and Sejnowski (1988).

Some More Details



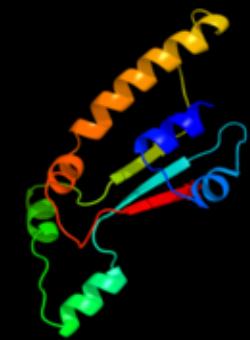
- Ignore/skim the parts of the paper about 'fsKBANN,' finite automata, etc – focus on the paper's *experimental control*
- The 'sliding window' should be as in Fig 4 (prev slide) of the Maclin & Shavlik paper (ie, 17 amino-acids wide)
- When the sliding window is 'off the edge,' assume the protein is padded with an imaginary 21st amino acid (ie, there are 21 possible feature values; 20 amino acids plus 'solvent')
- There are 128 proteins in the UC-Irvine archive (combine TRAIN AND TEST into one file, with TRAIN in front)
 - put the 5th, 10th, 15th, ..., 125th in your TUNE SET (counting from 1)
 - put the 6th, 11th, 16th, ..., 126th in your TEST SET (counting from 1)
 - put the rest in your TRAIN set
 - use early stopping (run to 1000 epochs if feasible)

Some More Details (2)



- Use *rectified linear* for the HU activation function
- Use *sigmoidal* for the output units
- We might test your code using some fake proteins (ie, we'll match the data-file format and feature names for secondary-structure prediction)
- Aim to cleanly separate your 'general BP code' from your 'sliding window' code – maybe write code that 'dumps' fixed-length feature vectors into a file formatted like used for Lab 1
- Turn into Moodle a lab report on Lab 2
 - BOTH lab partners turn in SAME report, with both names on it
 - no need to explain backprop algo nor the required extensions
 - focus on EXPERIMENTAL RESULTS and DISCUSSION
- Other suggestions or questions?

Some Data Structures



- Input VECTOR (of doubles; ditto for those below)
- Hidden Unit VECTOR
- Output VECTOR
- HU and Output VECTORS of 'deviations' for backprop (more later)
- Ditto for 'momentum term' if using that
- 2D ARRAY of weights between INPUTS and HUs
- 2D ARRAY of weights between HUs and OUTPUTS
- Copy of weight arrays holding BEST_NETWORK_ON_TUNE_SET
- Plus 2D ARRAY for each of TRAIN/TUNE/TEST sets

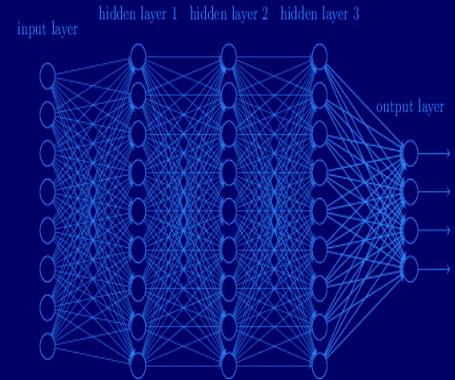


Calling Convention

- We need to follow a standard convention for your code in order to simplify testing it
- For Lab 2 (code in Lab2.java):
`Lab1 filename // Your code will create train, tune, and test sets`

Third Lab

- start when done with Labs 1 & 2
(aim to be done with all 3 in 6 wks)



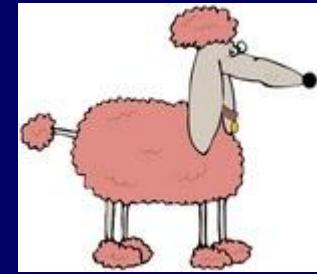
- We'll next build a simple deep ANN
- The TA and I are creating a simple image testbed (eg, 128x128 pixels), details TBA
- Implement
 - Convolution-Pooling-Convolution-Pooling Layers
 - Plus one final layer of HUs (ie, five HU layers)
- Ok to work in groups of up to four (should all be in 638 or all in 838)



Calling Convention

- We need to follow a standard convention for your code in order to simplify testing it
- For Lab 3 (code in Lab1.java):
`Lab3 fileNameOfTrain fileNameOfTune fileNameOfTest`

Moodle



Lab 1: due Feb 3 in Moodle

Lab 2: due Feb 17 (two people)

Lab 3: due March 3 (four people)

- Labs 1-3 must be done by March 17
- when nearly done with Lab 3,
propose your main project
(more details later)

Main Project



The majority of the term will involve an 'industrial size' Deep ANN

- Can use existing Deep Learning packages
- Can use Amazon, Google, Microsoft, etc cloud accounts
- In fact I expect you will do so!
- *Extending* open-source s/w would be great (especially for 838'ers), but wont be required

Some Project Directions



- Give 'advice' to deep ANNs
 - use 'domain knowledge' and not just data
 - Knowledge-based ANNs (KBANN algo)
- Explain what a Deep ANN learned
 - 'rule extraction' (Trepan algo)
- Generative Adversarial Networks (GANs)
- Given an image, generate a description (captioning)
- RL and Deep ANNs, LSTM, recurrent links
- Transfer Learning (TL)
- Chatbots (Amazon's challenge?)

Your Existing Project Ideas?

- 1)
- 2)
- 3)
- 4)
- 5)



Preferences for Cloud S/W?

Needs to offer free student accounts

Amazon?

Google?

IBM?

Microsoft?

Other?



Preferences for DL Packages?

MXNet (Amazon)?

Paddle (Baidu)?

Torch (Facebook)?

Tensor Flow (Google)?

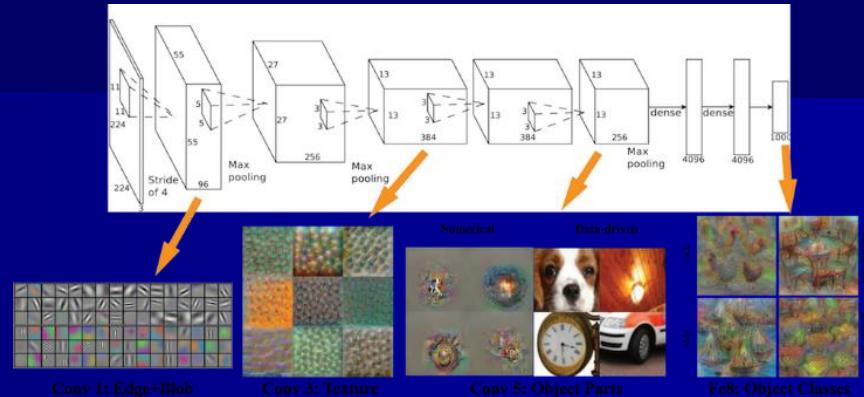
Power AI (IBM)?

CNTK (Microsoft)?

Caffe (U-Berkeley)?

Theano (U-Montreal)?

Other? (Keras, python lib for Theano & TensorFlow?)



My Major Challenge



Ensuring that everyone on a TEAM project contributes a significant amount and learns enough to merit a passing grade

During project progress reports, poster sessions, etc, all project members need to be able to be the oral presenter, even for parts they didn't implement

- I might randomly draw name of the presenter



Schedule

- Rest of Today
 - Quick review of material (from my cs540) for Labs 1 & 2
 - High-level intro to Deep ANNs if time permits (also from my cs540 lectures)
 - We'll take 10-min breaks every 50-60 mins
- THIS Thursday (same room)
 - Complete above if necessary
 - Intro to Reinforcement Learning (RL)
 - Hot, relatively new focus in deep learning
 - Even if your project not on RL, good to know in general and to understand reports on others' class projects

More about the Schedule

- I will be out of town next Tuesday
- BUT meet here to help each other with Labs 1 & 2 (ie, view it like working in the chemistry lab, all at the same time) – TA will be present
- I return to Madison Tuesday, Jan 31
 - But flight might be late (scheduled to arrive in Madison 3:42pm)
 - **We'll just have another 'code in the lab' meeting with TA, at least until I arrive**
- Email (after this Thursday's class) topics that you'd like me to re-review that week



What Else in Class?



- Some 'we all work on code in lab' sessions
 - Give me demos, ask questions, etc
 - Help one another, meet with partners
- Oral design and progress reports
(final reports will be poster sessions)
- Intro's to Cloud and Deep ANN s/w
- Lectures on advanced deep learning topics
(attendance optional for 638'ers)
 - we'll collect possible papers in Piazza

Some Presentation Plans

- Initializing ANN weights (Feb 7)
- S/W design for Lab 3
- Intro to tensors and their DL role
- Using Keras, Tensorflow, and ???
- Tutorial on LSTM
- Explanation of some major datasets





Quizzes

- I might see a need for quizzes on basics of ANNs, DL, etc (harder than one above)
- Might just be for cs638 students
- Show me they aren't needed :-)
- I'd announce at least a week in advance

Additional Questions?

