Deep Learning for Automatic Speech Recognition

CSCI/LING-5832: Natural Language Processing

Ryan Hartsfield Garrett Lewellen

April 23, 2015

Outline

Outline

Outline

Outline

Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition

George Dahl Dong Yi Li Deng Alex Acero

IEEE Transactions on Audio, Speech, and Language processing,
Vol. 20, No. 1, January 2012

Problem Formulation

Noisy channel model: maximize the likelihood of a decoded word sequence, $\hat{w}$, given our observed audio input, x:

$$\hat{w} = \underset{w \in \mathscr{L}}{\operatorname{argmax}} \underbrace{\mathbb{P}(x|w)}_{\text{Accoustic Model}} \underbrace{\mathbb{P}(w)}_{\text{Language Model}}$$

Use N-gram language model, CD-DNN-HMM for acoustic model

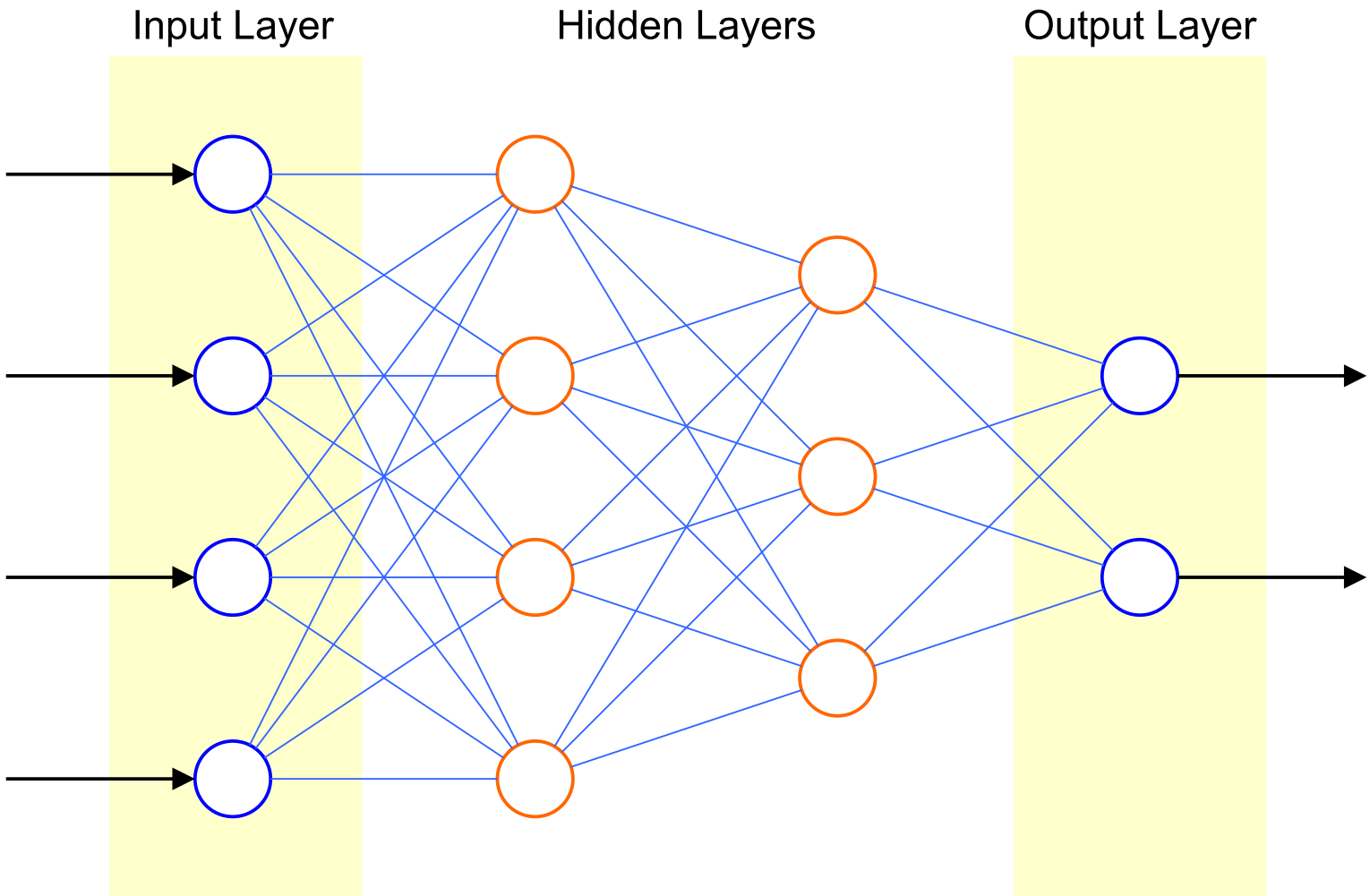
Problem Formulation (Cont)

Here the acoustic model is viewed as a sequence of transitions between states of tied-state triphones referred to as **senones**.

$$\underbrace{\mathbb{P}(x|w)}_{\text{Accoustic Model}} \approxeq \max \underbrace{\pi(q_0)}_{\text{Init State}} \prod_{t=1}^{T} \underbrace{a_{q_{t-1}q_t}}_{\text{Transistion}} \prod_{t=0}^{T} \underbrace{\mathbb{P}(x_t|q_t)}_{\text{Senone posterior}}$$

Where $\mathbb{P}(x_t|q_t)$ models the tied triphone senone posterior given mel-frequency cepstral coefficients (**MFCCs**) based on 11 sampled frames of audio.

Deep Neural Networks



Training

Training is complicated and time intensive! At a very high level two steps: initialization and DNN training.

Initialization:

- Find the best tying of triphone states
- Deal with some book keeping
- Train a CD-GMM-HMM using those states.
- Convert CD-GMM-HMM into CD-DNN-HMM keeping senone structure
- Apply pre-training algorithm on CD-DNN-HMM

Training

DNN Training:

- Generate raw alignment of states to senones
- Use alignment to refine by backpropagation
- Re-estimate prior senone probability given frames
- Refine HMM transitions probabilities
- Goto first step if no improvement against development set

Just enough time to talk about one of these steps in detail. Let's look at pre-training...

Pre-Training In-Depth

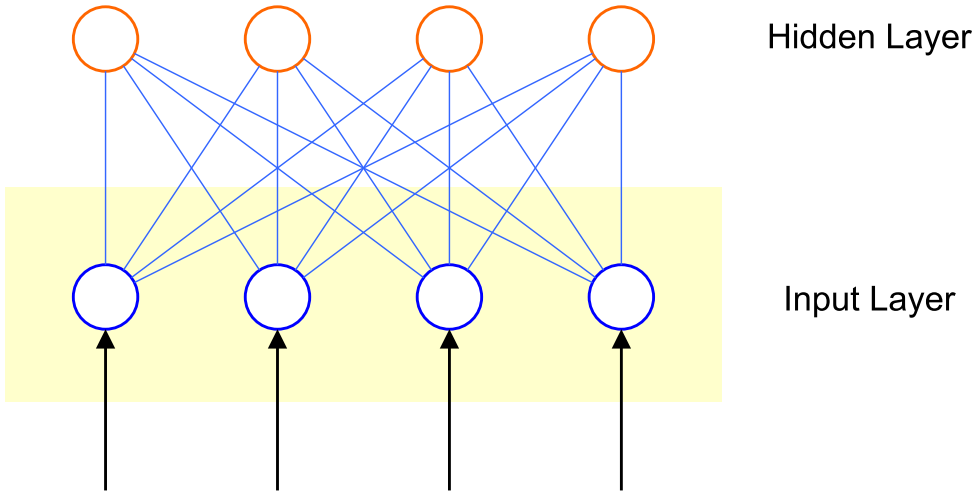
DNN training computationally intractable until Hinton et al. come to the rescue with *A Fast Learning Algorithm for Deep Belief Nets*.

Big idea: Use an approximate method, **contrastive divergence**, to get near an optimal solution, then use traditional methods, **backpropagation**, to finish the job.

Need to understand Restricted Boltzman Machines (RMBs) and Deep Belief Networks (DBNs)

Pre-Training: RBMs and DBNs

Bipartite arrangement of weights is assigned an energy:



$$E(v, h) = -b^T v - c^T h - v^T W h$$

For purpose of this talk, stack RBMs on top of one another to get a DBN

Pre-Training: Contrastive Divergence

Want to do vanilla Stochastic Gradient Descent, however, our model term takes **exponential** time to compute correctly.

$$-\frac{\partial \ell(\theta)}{\partial w_{ij}} = \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\text{data}} - \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\text{model}}$$

Instead, **approximate** it (essentially minimizing Kullback-Leibler divergence) with one step Gibbs sampling:

$$-\frac{\partial \ell(\theta)}{\partial w_{ij}} \approx \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_1$$

Pre-Training: Bringing it all together

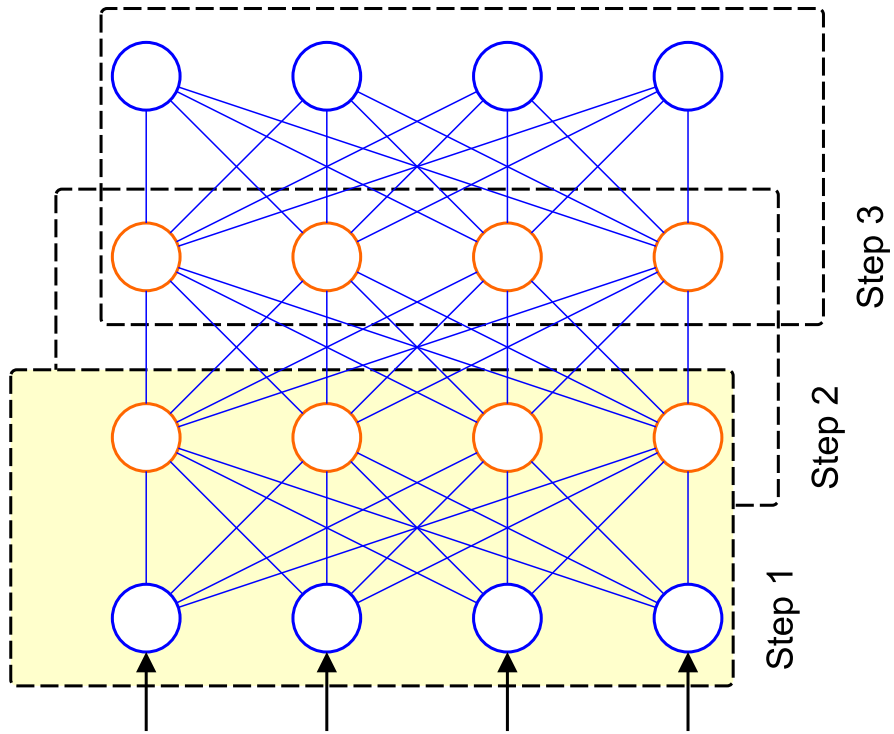
Hinton's Greedy Algorithm:

1) Train RBM consisting of first two layers

2) Move RBM frame up a layer and train

3) When out of layers, you have trained DBN

4) Refine with backpropagation



Outline

Experimental Results

Papers report different metrics (sentence accuracy, phone error rate) on different datasets (Bing, TIMIT)

	Bing Mobile		TIMIT	
Architecture	Dev.	Test	Dev.	Test
CD-GMM-HMMM	70.3%	68.4%		
CD-DNN-HMM	71.8%	69.6%		
CNN-HMM				20.07%

Take away: Both demonstrate significant improvement over GMM approach, with CNN approach giving the best performance.

Outline

Conclusions

TODO: Garrett

Conclusions (Cont)

TODO: Garrett

Questions?