

Deep Learning & Neural Networks

Lecture 3

Kevin Duh

Graduate School of Information Science
Nara Institute of Science and Technology

Jan 21, 2014

Applications of Deep Learning

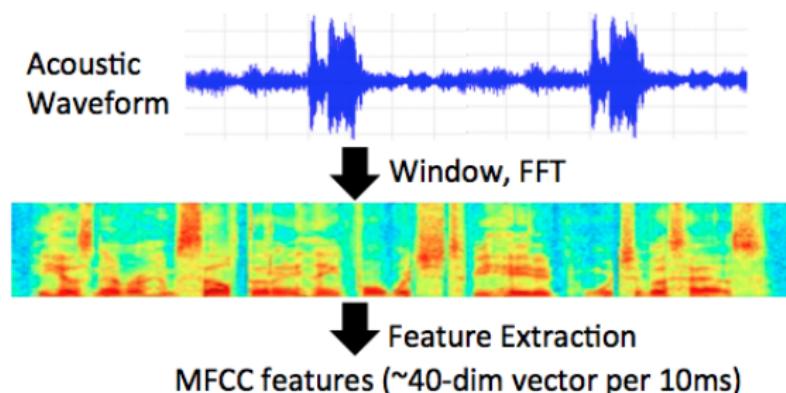
- Goal: To give a taste of how deep learning is used in practice, and how varied it is, e.g.:
 - ① Speech Recognition: hybrid DNN-HMM system
 - ② Computer Vision: local receptive field / pooling architecture
 - ③ Language Modeling: recurrent structure

Today's Topic

- 1 Deep Neural Networks for Acoustic Modeling in Speech Recognition
[Hinton et al., 2012]
- 2 Building High-Level Features using Large Scale Unsupervised Learning
[Le et al., 2012]
- 3 Recurrent Neural Network Language Models [Mikolov et al., 2010]

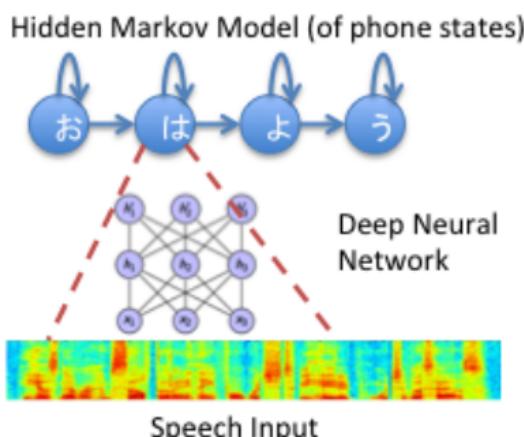
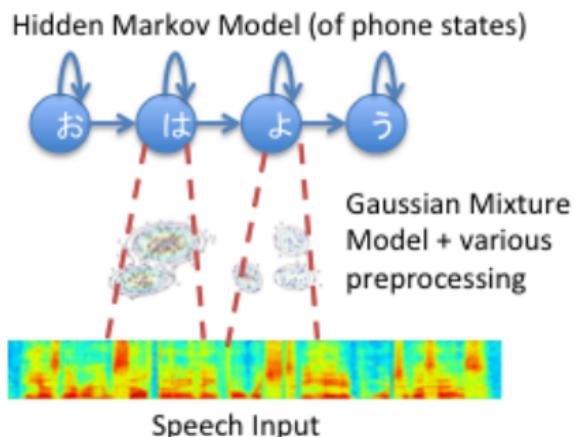
Background: Simplified View of Speech Recognition

- Task: Given input acoustic signal, predict word/phone sequence
- $\arg \max_{\text{phone_sequence}} p(\text{acoustics}|\text{phone})p(\text{phone}|\text{previous_phones})$
 - ▶ $p(\text{acoustics}|\text{phone})$ modeled by Gaussian Mixture Model (GMM)
 - ▶ $p(\text{phone}|\text{previous_phones})$ by transitions in Hidden Markov Model (HMM)
- Acoustic features:

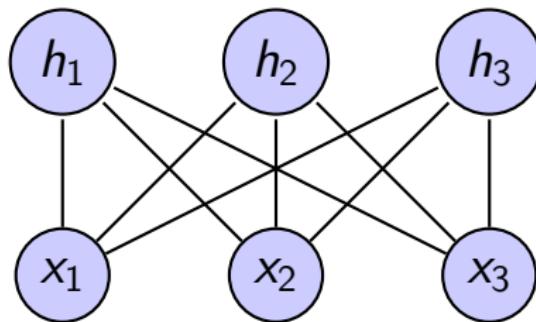


DNN-HMM Hybrid Architecture

- ① Train Deep Belief Nets on speech features: typically 3-8 layers, 2000 units/layer, 15 frames of input, 6000 output
- ② Fine-tune with frame-per-frame phone labels obtained from traditional Gaussian models
- ③ Further discriminative training in conjunction with higher-level Hidden Markov Model



Gaussian-Bernoulli RBM for Continuous Data



h_j are binary, x_i are continuous variables

$$p(x, h) = \frac{1}{Z_\theta} \exp(-E_\theta(x, h)) = \frac{1}{Z_\theta} \exp\left(\sum_i \frac{-(x_i - b_i)^2}{2v_i} + \sum_{ij} \frac{x_i w_{ij} h_j}{\sqrt{v_i}} + d^T h\right)$$

$$p(h_j = 1|x) = \sigma\left(\sum_i \frac{w_{ij} x_i}{\sqrt{v_i}} + d_j\right)$$

$$p(x_i|h) \sim \text{Gaussian with mean } b_i + \sqrt{v_i} \sum_j w_{ij} h_j \text{ and variance } v_i$$

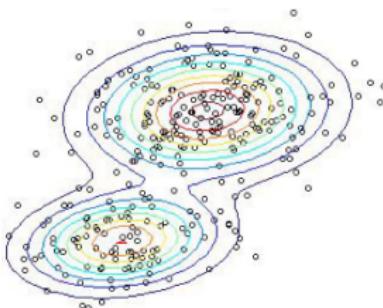
Usually, x is normalized to zero mean, unit variance beforehand

GMM vs. DNN in modeling speech

- Speech is produced by modulating a small number of parameters in a dynamical system (e.g vocal tract)
 - ▶ True structure should be in low-dimensional space

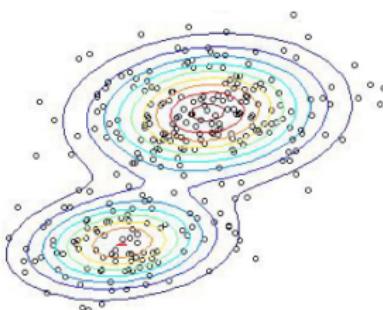
GMM vs. DNN in modeling speech

- Speech is produced by modulating a small number of parameters in a dynamical system (e.g vocal tract)
 - ▶ True structure should be in low-dimensional space
- GMM's: $p(x) = \sum_j p(h_j)p(x|h_j)$ with $p(x|h_j)$ as Gaussian
 - ▶ High model expressiveness: can model any non-linear data
 - ▶ But may require large **full**-covariance Gaussians or **many** diagonal-covariance Gaussians → statistically inefficient



GMM vs. DNN in modeling speech

- Speech is produced by modulating a small number of parameters in a dynamical system (e.g vocal tract)
 - ▶ True structure should be in low-dimensional space
- GMM's: $p(x) = \sum_j p(h_j)p(x|h_j)$ with $p(x|h_j)$ as Gaussian
 - ▶ High model expressiveness: can model any non-linear data
 - ▶ But may require large **full**-covariance Gaussians or **many** diagonal-covariance Gaussians → statistically inefficient



- RBM & DNN's distributed factor representation is more efficient
 - ▶ Also: no need to worry about feature correlation → exploit larger temporal window as input

Results

DNN-HMM outperforms GMM-HMM on various datasets
Already commercialized!

Word Error Rate Results:

TASK	HOURS OF TRAINING DATA	DNN-HMM	GMM-HMM WITH SAME DATA	GMM-HMM WITH MORE DATA
SWITCHBOARD (TEST SET 1)	309	18.5	27.4	18.6 (2,000 H)
SWITCHBOARD (TEST SET 2)	309	16.1	23.6	17.1 (2,000 H)
ENGLISH BROADCAST NEWS	50	17.5	18.8	
BING VOICE SEARCH (SENTENCE ERROR RATES)	24	30.4	36.2	
GOOGLE VOICE INPUT	5,870	12.3		16.0 (>> 5,870 H)
YOUTUBE	1,400	47.6	52.3	

Why it works: Larger context and less hand-engineered preprocessing

More details on Switchboard result [Seide et al., 2011]

Basic Setup:

- Input: 39-dim derived from PLP, HLDA transform
- Output: 9304 cross-word triphone states (tied)

Baseline GMM-HMM:

- GMM with 40 Gaussians.
- Training: (1) max-likelihood (EM), (2) discriminative BMMI

DNN-HMM:

- 7 stacked RBM's with 2048 units per layer
- Pre-training on 2 passes over training data (300 hours of speech)
- Mini-batch size: 100-300 (pre-training), 1000 (backpropagation)

acoustic model	#params	WER (r. chg.)
GMM 40 mix, BMMI	29.4M	23.6
CD-DNN 1 layer \times 4634 nodes	43.6M	26.0 (+10%)
+ 2 \times 5 neighbor frames	45.1M	22.4 (-14%)
CD-DNN 7 layers \times 2048 nodes	45.1M	17.1 (-24%)
+ updated state alignment	45.1M	16.4 (-4%)
+ sparsification 66%	15.2M nz	16.1 (-2%)

Today's Topic

- 1 Deep Neural Networks for Acoustic Modeling in Speech Recognition
[Hinton et al., 2012]
- 2 Building High-Level Features using Large Scale Unsupervised Learning
[Le et al., 2012]
- 3 Recurrent Neural Network Language Models [Mikolov et al., 2010]

Motivating Question: Is it possible to learn high-level features (e.g. face detectors) using only unlabeled images?

Motivating Question: Is it possible to learn high-level features (e.g. face detectors) using only unlabeled images?

- Answer: yes.

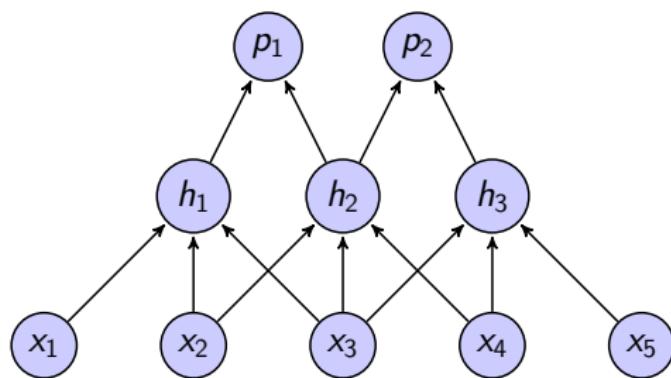
Motivating Question: Is it possible to learn high-level features (e.g. face detectors) using only unlabeled images?

- Answer: yes.
 - ▶ Using a deep network of 1 billion parameters
 - ▶ 10 million images (sampled from Youtube)
 - ▶ 1000 machines (16,000 cores) x 1 week.

"Grandmother Cell" Hypothesis

- Grandmother cell: A neuron that lights up when you see or hear your grandmother
 - ▶ Lots of interesting (controversial) discussions in the neuroscience literature
- For our purposes: is it possible to learn such high-level concepts from raw pixels?

Previous work: Convolutional Nets [LeCun et al., 1998]



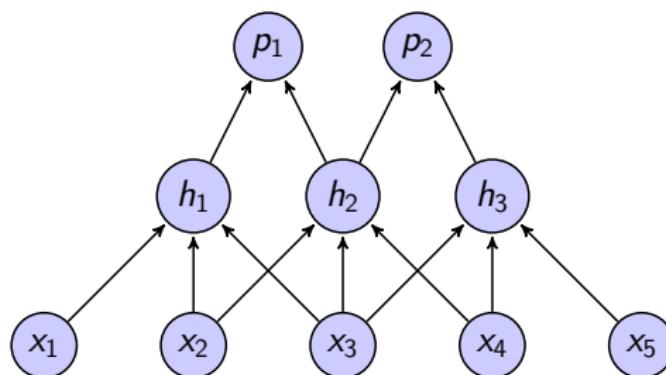
Receptive Field (RF): each h_j only connects to small input region.
Tied weights \rightarrow convolution

Pooling: e.g. $p_1 = \max(h_1, h_2)$ or
$$p_1 = \sqrt{h_1^2 + h_2^2}$$

Advantages:

- ① Fewer weights
- ② Shift invariance

Previous work: Convolutional Nets [LeCun et al., 1998]



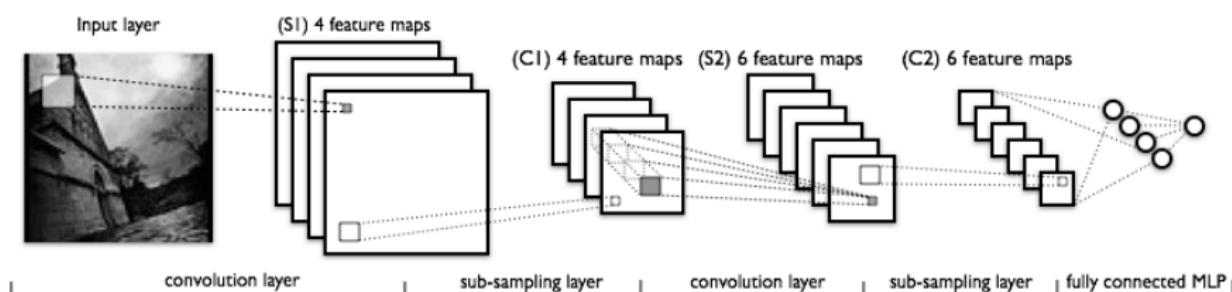
Receptive Field (RF): each h_j only connects to small input region.

Tied weights \rightarrow convolution

Pooling: e.g. $p_1 = \max(h_1, h_2)$ or
$$p_1 = \sqrt{h_1^2 + h_2^2}$$

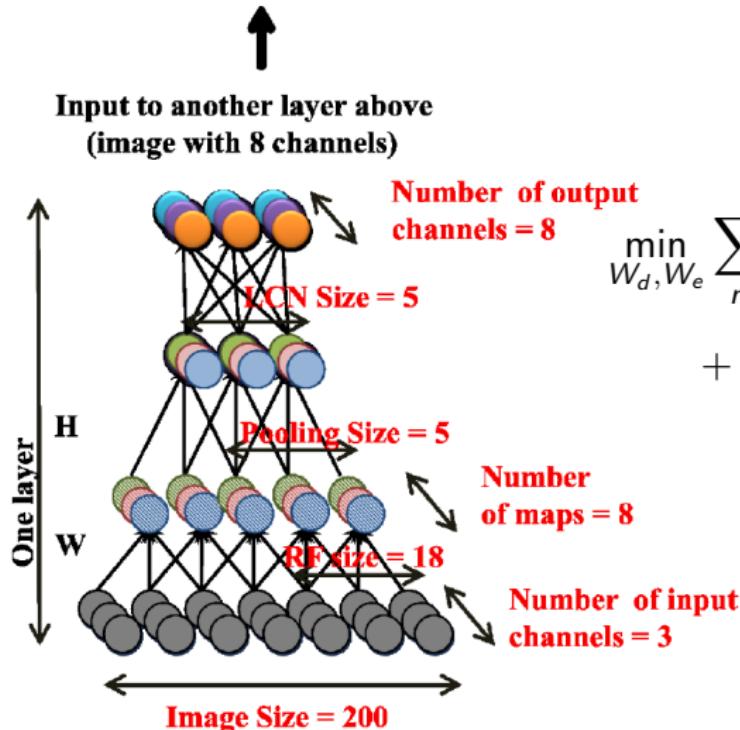
Advantages:

- ① Fewer weights
- ② Shift invariance



(Figure from <http://deeplearning.net/tutorial/lenet.html>)

Architecture



$$\min_{W_d, W_e} \sum_m \|W_d W_e x^{(m)} - x^{(m)}\| \quad (1)$$

$$+ \sum_{m,k} \sqrt{\epsilon + P_k(W_e x^{(m)})^2} \quad (2)$$

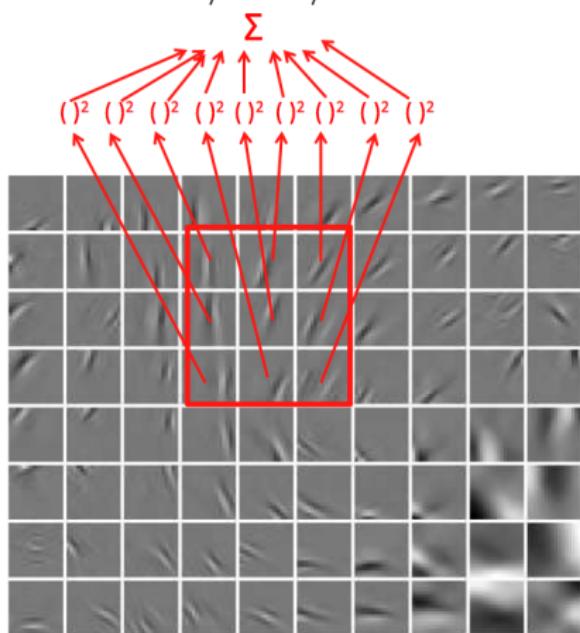
(1): auto-encoder
(2): pooling

Repeated 3 times to form Deep Architecture
 $x^{(m)}$ = image of 200x200 pixels x3 channels

Feature learning by Topographic ICA

[Hyvärinen et al., 2001]

Learns shift/scale/rotation-invariant features



Reconstruction version

[Le et al., 2011] can be trained faster

$$\min_{W_d, W_e} \sum_m ||W_d W_e x^{(m)} - x^{(m)}|| + \sum_{m,k} \sqrt{\epsilon + P_k(W_e x^{(m)})^2}$$

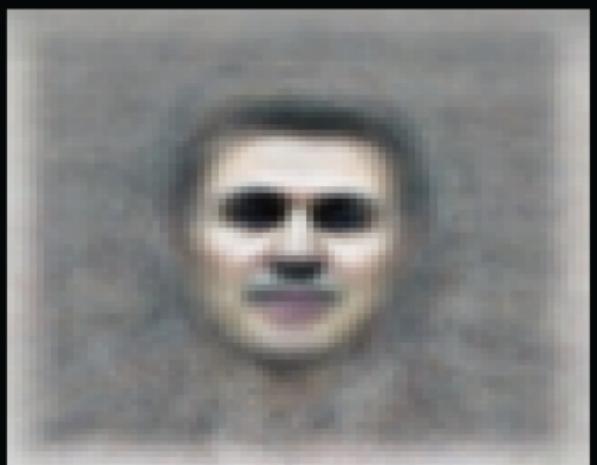
Training Setup

- 3-layer network, 1 billion parameters (trained jointly)
- 10 million 200x200 pixel images from 10 million Youtube videos
- 1000 machines (16,000 cores) \times 1 week
- Lots of tricks for data/model parallelization (next lecture)

Face neuron



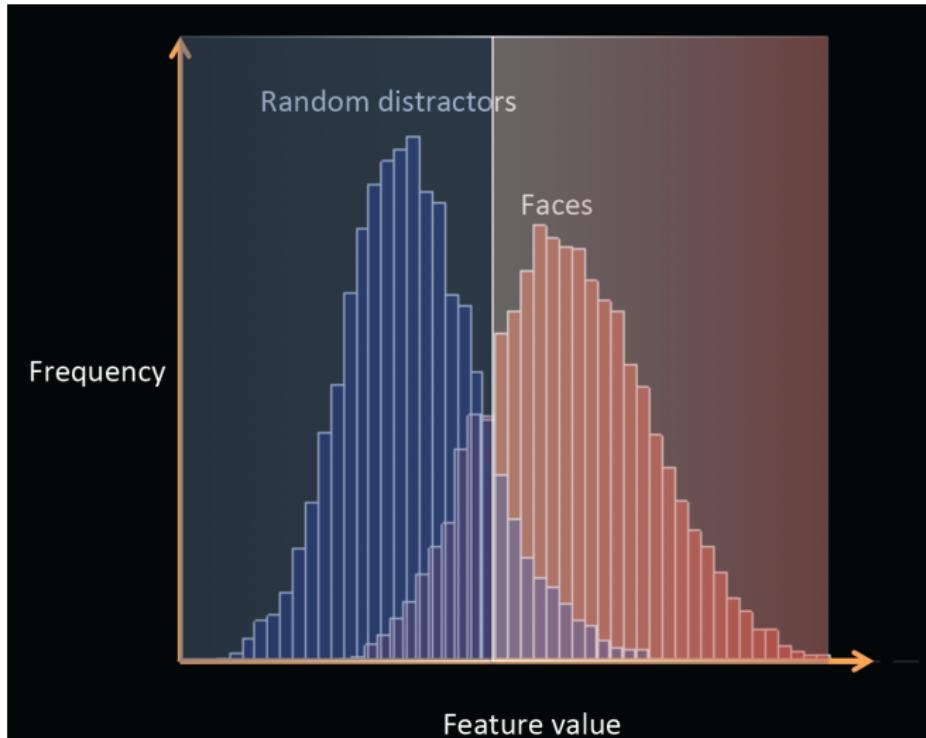
Top stimuli from the test set



Optimal stimulus
by numerical optimization

*Graphics from [Le et al., 2012]

Face neuron



*Graphics from [Le et al., 2012]

Cat neuron



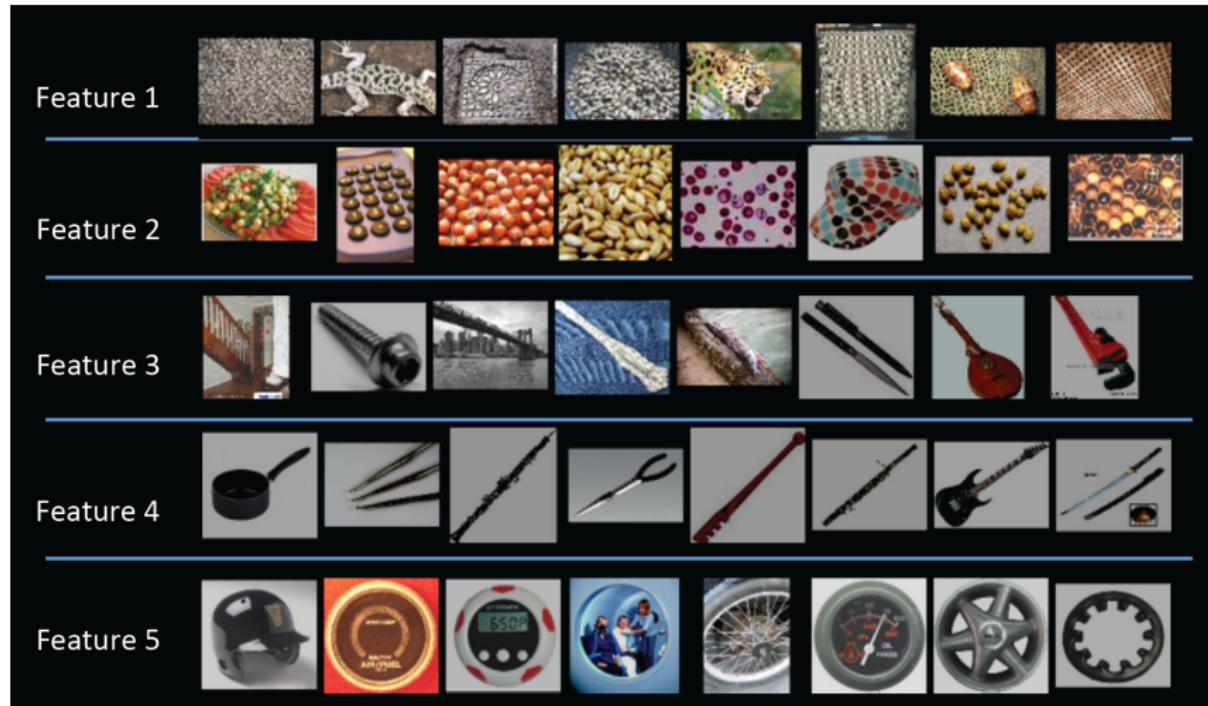
Top stimuli from the test set



Optimal stimulus
by numerical optimization

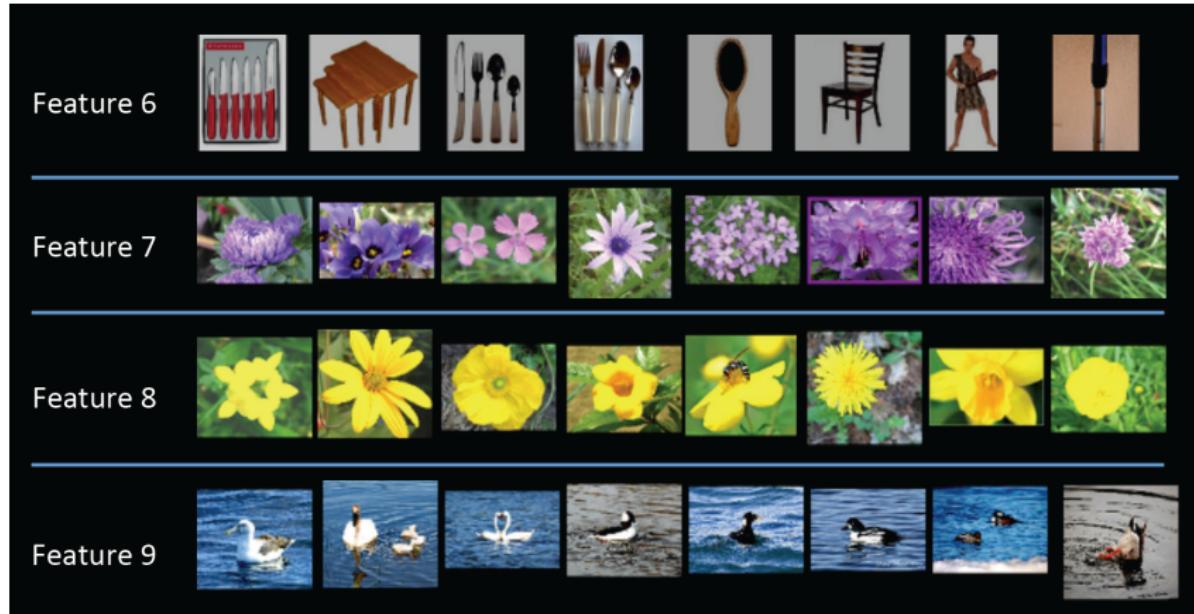
*Graphics from [Le et al., 2012]

More examples



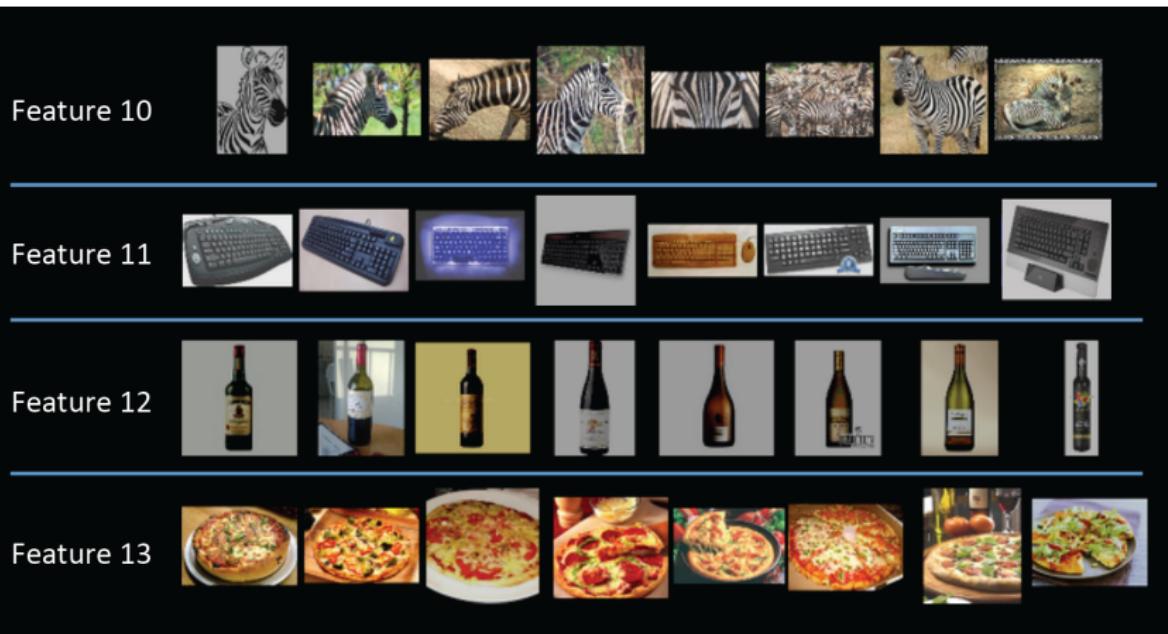
*Graphics from [Le et al., 2012]

More examples



*Graphics from [Le et al., 2012]

More examples



*Graphics from [Le et al., 2012]

ImageNet Classification Results



mite

container ship

motor scooter

leopard

- Add logistic regression on top of final layer
- Supervised learning on ImageNet dataset

Test Accuracy (22K categories):

Method	Accuracy
Random	0.005%
Previous State-of-the-art	9.3%
[Le et al., 2012] without pre-training on Youtube data	13.6%
[Le et al., 2012] with pre-training on Youtube data	15.8%

Today's Topic

- 1 Deep Neural Networks for Acoustic Modeling in Speech Recognition
[Hinton et al., 2012]
- 2 Building High-Level Features using Large Scale Unsupervised Learning
[Le et al., 2012]
- 3 Recurrent Neural Network Language Models [Mikolov et al., 2010]

Goal of Language Modeling

- Give probabilities to word sequences (e.g. sentences)
 - ▶ Likely sentences in the world (e.g. "let's recognize speech") → high probability
 - ▶ Unlikely sentences in the world (e.g. "let's wreck a nice beach") → low probability
- Useful for various applications involving natural language

Goal of Language Modeling

- Give probabilities to word sequences (e.g. sentences)
 - ▶ Likely sentences in the world (e.g. "let's recognize speech") → high probability
 - ▶ Unlikely sentences in the world (e.g. "let's wreck a nice beach") → low probability
- Useful for various applications involving natural language
- N-gram model decomposes sentence probability, e.g.
 $p(w^{(1)}, w^{(2)}, w^{(3)}, w^{(4)}) =$
 - ▶ $p(w^{(4)}|w^{(3)})p(w^{(3)}|w^{(2)})p(w^{(2)}|w^{(1)})p(w^{(1)})$ (2-gram)
 - ▶ $p(w^{(4)}|w^{(3)}, w^{(2)})p(w^{(3)}|w^{(2)}, w^{(1)})p(w^{(2)}|w^{(1)})p(w^{(1)})$ (3-gram)

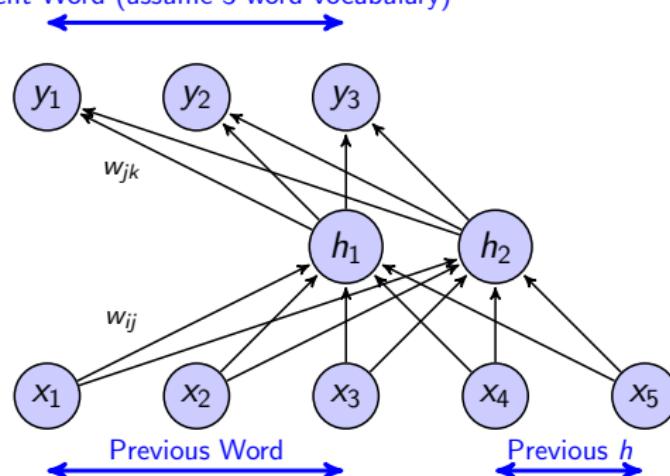
Goal of Language Modeling

- Give probabilities to word sequences (e.g. sentences)
 - ▶ Likely sentences in the world (e.g. "let's recognize speech") → high probability
 - ▶ Unlikely sentences in the world (e.g. "let's wreck a nice beach") → low probability
- Useful for various applications involving natural language
- N-gram model decomposes sentence probability, e.g.
 $p(w^{(1)}, w^{(2)}, w^{(3)}, w^{(4)}) =$
 - ▶ $p(w^{(4)}|w^{(3)})p(w^{(3)}|w^{(2)})p(w^{(2)}|w^{(1)})p(w^{(1)})$ (2-gram)
 - ▶ $p(w^{(4)}|w^{(3)}, w^{(2)})p(w^{(3)}|w^{(2)}, w^{(1)})p(w^{(2)}|w^{(1)})p(w^{(1)})$ (3-gram)
- Estimate from text data:
 $p(w^{(2)}|w^{(1)}) = \text{count}(w^{(1)}, w^{(2)})/\text{count}(w^{(1)})$, plus smoothing to account for unknown words and word sequences

Recurrent Neural Net Architecture for Language Modeling

Model $p(\text{current_word} | \text{previous_words})$ with a recurrent hidden layer

Current Word (assume 3-word vocabulary)



- Probability of word k :

$$y_k = \frac{\exp(W_{jk}^T h)}{\sum_{k'} \exp(W_{jk'}^T h)}$$

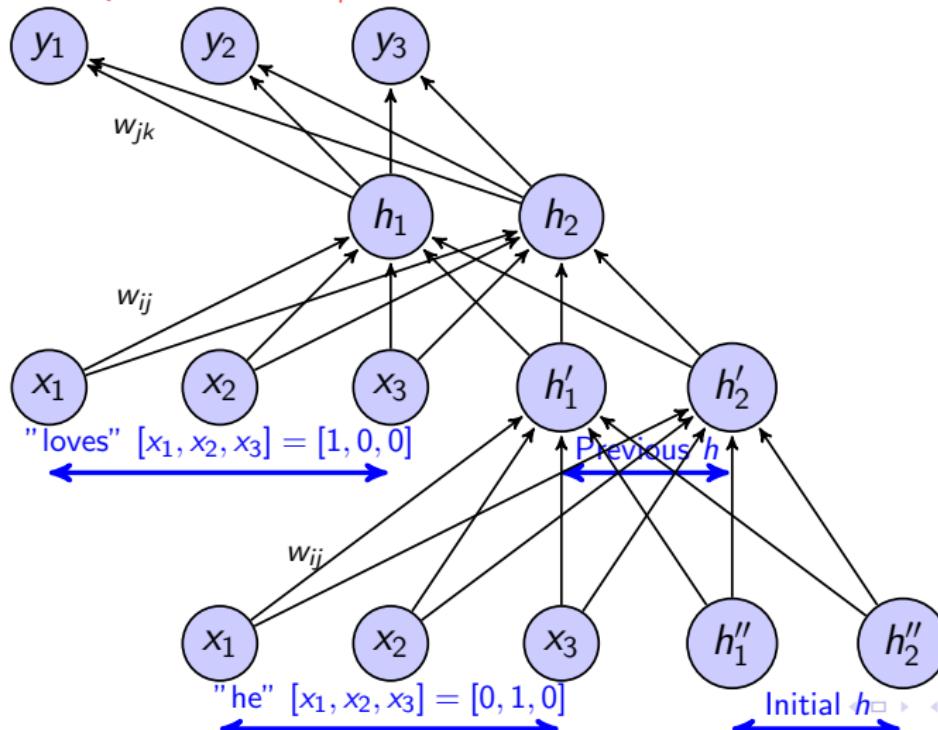
- $[x_1, x_2, x_3]$ is binary vector with 1 at current vocabulary & 0 otherwise
- $[x_4, x_5]$ is a copy of $[h_1, h_2]$ from the previous time-step
- $h_j = \sigma(W_{jj}^T x_i)$ is hidden "state" of the system

Training: Backpropagation through Time

Unroll the hidden states for certain time-steps.

Given error at y , update weights by backpropagation

Example: he loves | her



Advantages of Recurrent Nets

- Hidden nodes h form a distributed representation of partial sentence
 - ▶ h is a succinct conditioning factor for predicting current word
 - ▶ Arbitrarily-long history is (theoretically) kept through recurrence

Advantages of Recurrent Nets

- Hidden nodes h form a distributed representation of partial sentence
 - ▶ h is a succinct conditioning factor for predicting current word
 - ▶ Arbitrarily-long history is (theoretically) kept through recurrence
- In practice:
 - ▶ Backpropagation through Time forms a deep network; may be hard to train. Fixed to < 10 previous time-steps/words
 - ▶ $y_k = \frac{\exp(W_{jk}^T h)}{\sum_{k'} \exp(W_{jk'}^T h)}$ requires summation k over vocabulary size, which is large. There are shortcuts to reduce computation.

Advantages of Recurrent Nets

- Hidden nodes h form a distributed representation of partial sentence
 - ▶ h is a succinct conditioning factor for predicting current word
 - ▶ Arbitrarily-long history is (theoretically) kept through recurrence
- In practice:
 - ▶ Backpropagation through Time forms a deep network; may be hard to train. Fixed to < 10 previous time-steps/words
 - ▶ $y_k = \frac{\exp(W_{jk}^T h)}{\sum_{k'} \exp(W_{jk'}^T h)}$ requires summation k over vocabulary size, which is large. There are shortcuts to reduce computation.
- By-product: $[w_{ij}]_i$ can be used as "word embeddings". Useful for various natural language processing tasks
[Zhila et al., 2013, Turian et al., 2010]

Results [Mikolov et al., 2010]

Trained on 6 million words (300K sentences) of New York Times data.

Evaluation on held-out data:

$$\text{perplexity} = 2^{\text{entropy}} = 2^{-\frac{1}{|\text{data}|} \sum_{\text{data}} \log p_{\text{model}}(\text{data})}$$

Model	Perplexity
N-gram (N=5)	221
Recurrent Net $ h = 60$	229
Recurrent Net $ h = 90$	202
Recurrent Net $ h = 250$	173
Recurrent Net $ h = 400$	171
Combining 3 Recurrent Nets	151
Combining 3 Recurrent Nets, dynamic update on held-out	128

References I

-  Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012).
Deep neural networks for acoustic modeling in speech recognition.
IEEE Signal Processing Magazine, 29.
-  Hyvärinen, A., Hoyer, P., and Inki, M. (2001).
Topographic independent component analysis.
Neural Computation, 13(7):1527–1558.
-  Le, Q., Karpenko, A., Ngiam, J., and Ng, A. (2011).
ICA with reconstruction cost for efficient overcomplete feature learning.
In *NIPS*.

References II

-  Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. (2012).
Building high-level features using large scale unsupervised learning.
In *ICML*.
-  LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998).
Gradient-based learning applied to document recognition.
Proc, 86(11):2278–2324.
-  Mikolov, T., Karafiat, S., Burget, L., Černocký, J., and Khudanpur, S. (2010).
Recurrent neural network based language models.
In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*.

References III

-  Seide, F., Li, G., and Yu, D. (2011).
 Conversational speech transcription using context-dependent deep neural networks.
 In *Proc. Interspeech 2011*, pages 437–440.
-  Turian, J., Ratinov, L.-A., and Bengio, Y. (2010).
 Word representations: A simple and general method for semi-supervised learning.
 In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.
 Association for Computational Linguistics.

References IV



Zhila, A., Yih, W.-t., Meek, C., Zweig, G., and Mikolov, T. (2013). Combining heterogeneous models for measuring relational similarity. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1000–1009, Atlanta, Georgia. Association for Computational Linguistics.