

Unsupervised Neural Nets: Autoencoders and ICA

ICA vs. PCA

Autoencoder types

denoising

variational

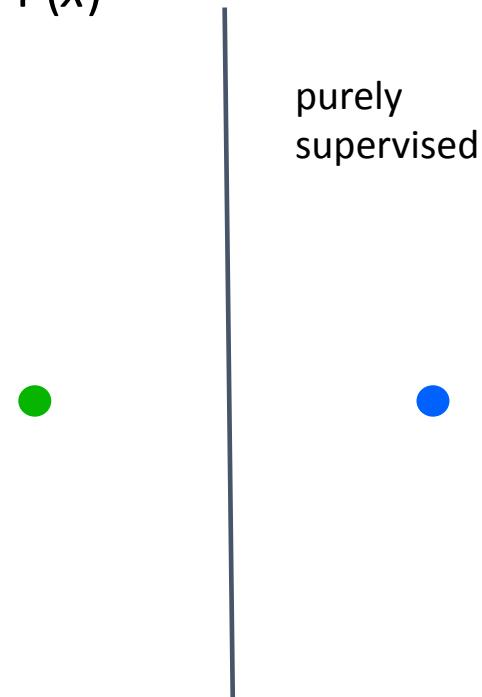
Lyle Ungar

with figures from

Quoc Le, Socher & Manning

Semi-Supervised Learning

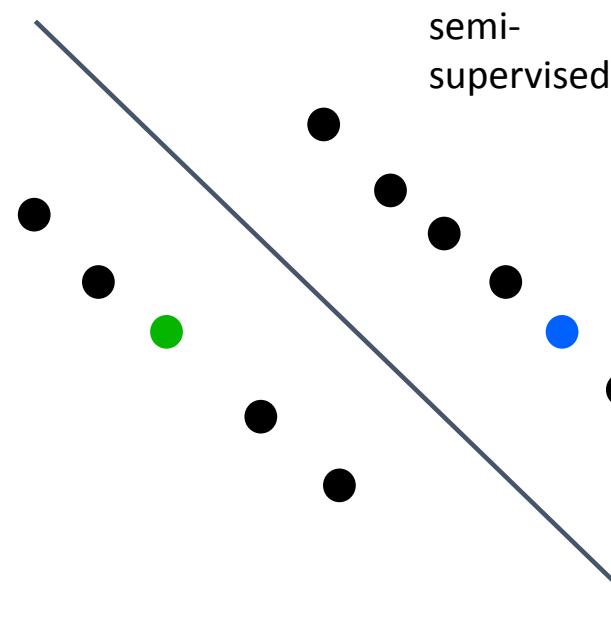
- Hypothesis: $P(c|x)$ can be more accurately computed using shared structure with $P(x)$



from Socher and Manning

Semi-Supervised Learning

- Hypothesis: $P(c|x)$ can be more accurately computed using shared structure with $P(x)$

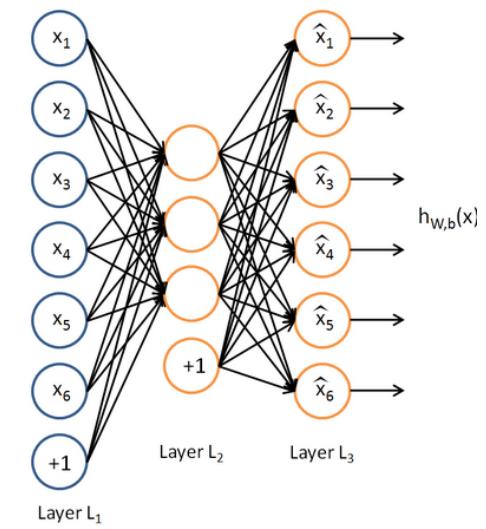


from Socher and Manning

Unsupervised Neural Nets

◆ Auto-encoders generalize PCA

- Take same image as input and output
- Learn weights to minimize the reconstruction error
- Avoid perfect fitting
 - Pass through a “bottleneck” or impose sparsity
 - Or add noise to the input
 - ◆ *Denoising auto-encoder*



http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity

Denoising Auto-encoder

- ◆ **Image reconstruction (in CNN)**

- X = image with noise added
- Y = original image

- ◆ **Intermediate neural outputs are an embedding**

Transformer

- ◆ **Masking in NLP (in LSTM)**

- X = sentence with words removed
- Y = the words that were removed

- ◆ **Intermediate neural outputs are an embedding**

Variational Auto-Encoder (VAE)

- ◆ Minimize reconstruction error and maximize independence of "components"
 - Reminiscent of a mixture model
- ◆ Intermediate neural outputs (components) are an embedding

Independent Components Analysis (ICA)

- ◆ Given observations \mathbf{X} , find \mathbf{W} such that components \mathbf{s}_j of $\mathbf{S} = \mathbf{X}\mathbf{W}$ are “as independent of each other as possible”
 - E.g. have maximum KL-divergence or low mutual information
 - Alternatively, find directions in \mathbf{X} that are most skewed
 - farthest from Gaussian
 - Usually mean center and “whiten” the data (make unit covariance) first
 - whiten: $\mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1/2}$
- ◆ Very similar to PCA
 - But the loss function is not quadratic
 - So optimization cannot be done by SVD

Trendy deep learning generalization: “disentanglement”

Independent Components Analysis (ICA)

- ◆ Given observations \mathbf{X} , find \mathbf{W} and \mathbf{S} such that components s_j of $\mathbf{S} = \mathbf{X}\mathbf{W}$ are “as independent of each other as possible”
 - S_k = “sources” should be independent
- ◆ Reconstruct $\mathbf{X} \sim (\mathbf{X}\mathbf{W})\mathbf{W}^+ = \mathbf{S}\mathbf{W}^+$
 - \mathbf{S} like *principal component scores*
 - \mathbf{W}^+ like *loadings*
 - $\mathbf{x} \sim \sum_j s_j \mathbf{w}_j^+$
- ◆ **Auto-encoder** – nonlinear generalization that “encodes” \mathbf{X} as \mathbf{S} and then “decodes” it

Reconstruction ICA (RICA)

◆ Reconstruction ICA: find \mathbf{W} to minimize

- Reconstruction error
 - $\|\mathbf{X} - \mathbf{SW}^+\|_2 = \|\mathbf{X} - (\mathbf{XW})\mathbf{W}^+\|_2$

And minimize

- Mutual information between sources $\mathbf{S} = \mathbf{XW}$

$$I(s_1, s_2 \dots s_k) = \sum_{i=1}^k H(s_i) - H(s)$$

$$H(y) = -\int p(y) \log p(y) dy$$

Difference between the entropy of each “source” s_i and the entropy of all of them together

Note: this is a bit more complex than it looks, as we have real numbers, not distributions

Mutual information

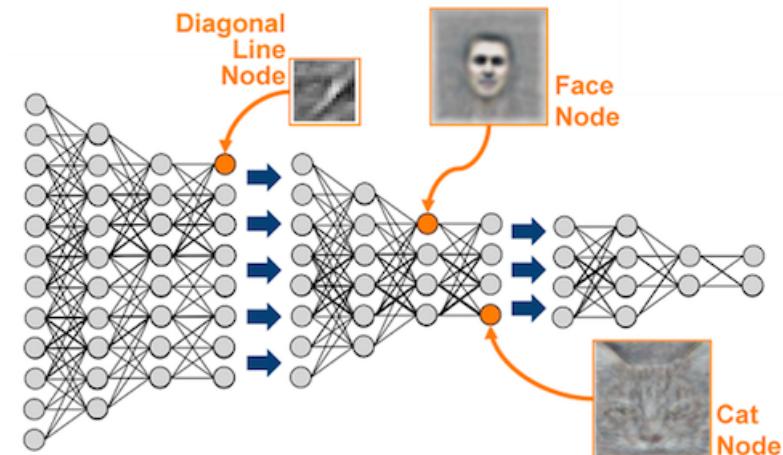
$$MI(y_1, y_2, \dots, y_m) = KL(p(y_1, y_2, \dots, y_m) || p(y_1)p(y_2) \dots p(y_m))$$

How well do the independent distributions approximate the joint distribution?

Auto-encoders

- ◆ Take same image as input and output
- ◆ often adding noise to the input (*denoising auto-encoder*)
- ◆ Learn weights to minimize the reconstruction error
- ◆ This can be done repeatedly (reconstructing features)
- ◆ Used for semi-supervised learning

Unsupervised deep learning



from Socher and Manning

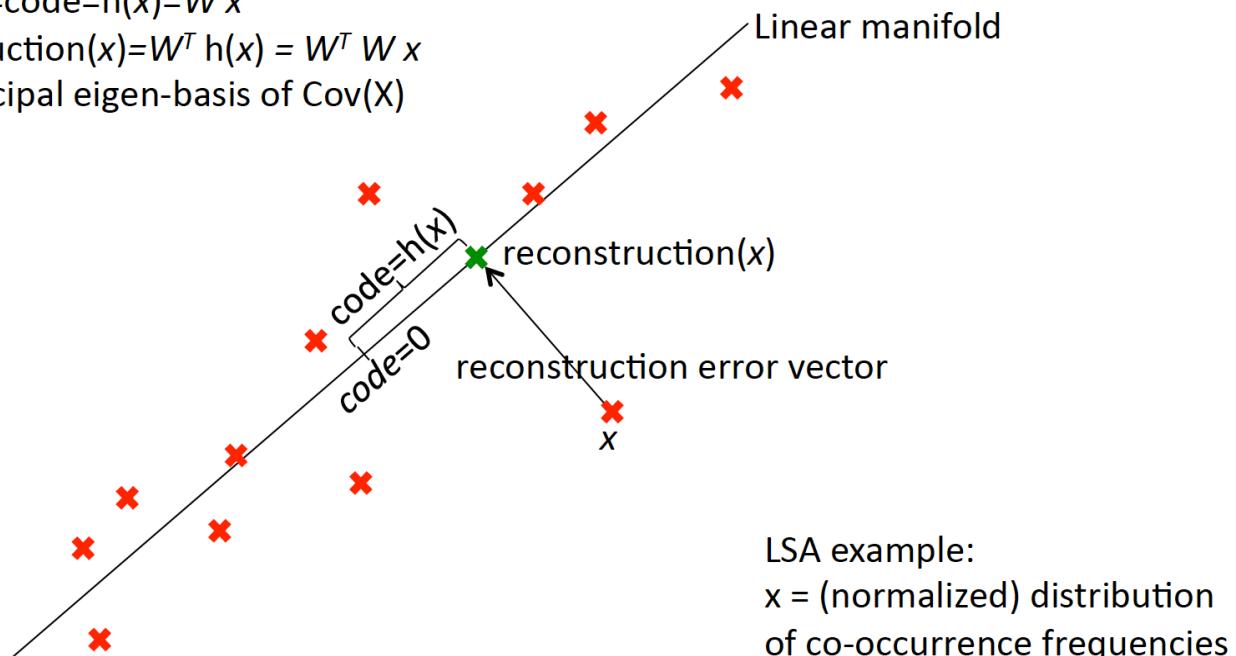
PCA = Linear Manifold = Linear Auto-encoder

input x , 0-mean

features=code= $h(x) = Wx$

reconstruction(x)= $W^T h(x) = W^T Wx$

W = principal eigen-basis of $\text{Cov}(X)$

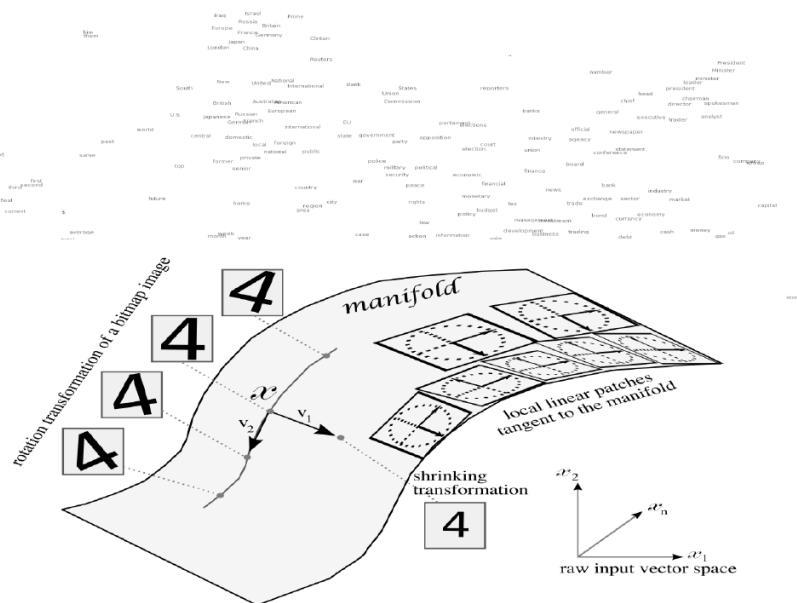
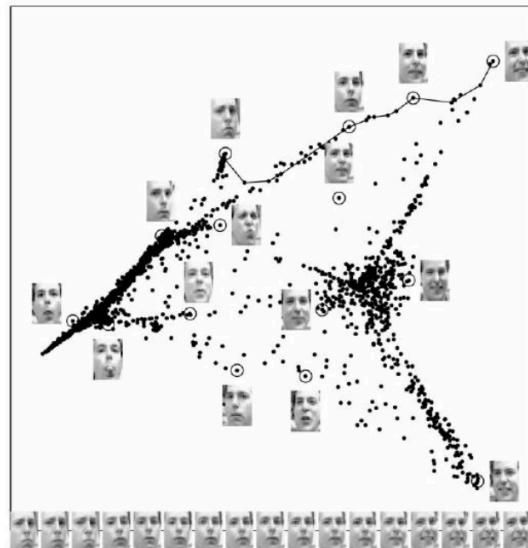


LSA example:
 x = (normalized) distribution
of co-occurrence frequencies

from Socher and Manning

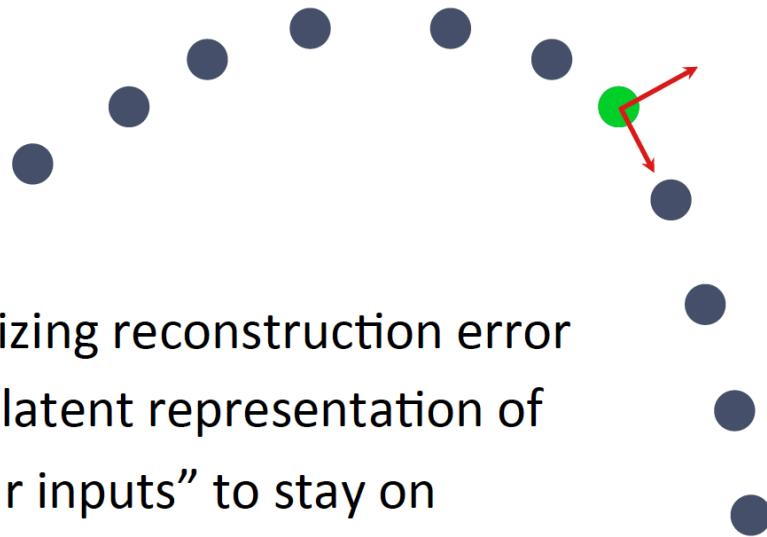
The Manifold Learning Hypothesis

- Examples concentrate near a lower dimensional “manifold” (region of high density where small changes are only allowed in certain directions)¹



from Socher and Manning

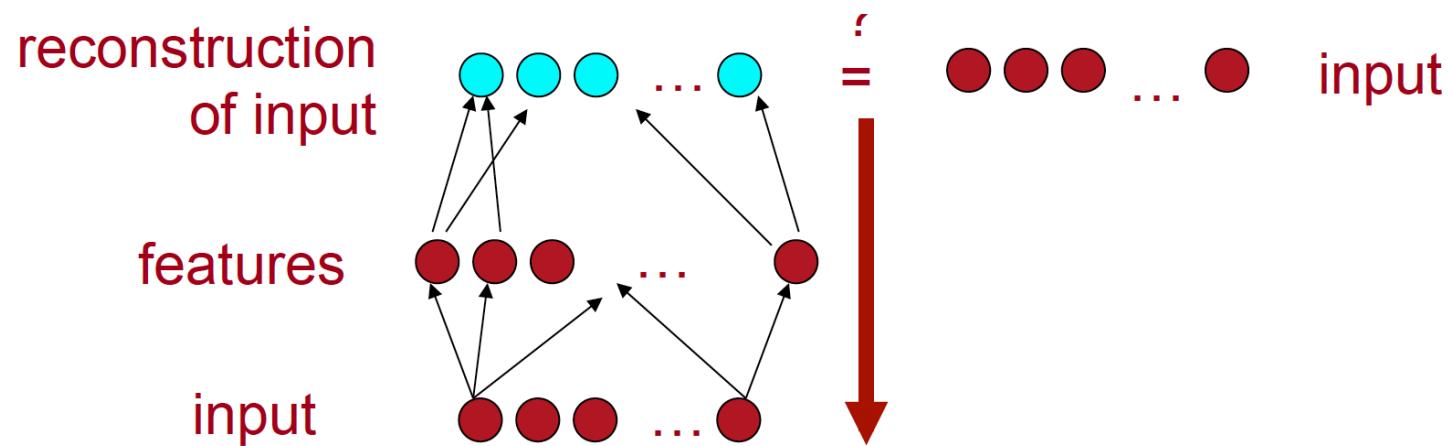
Auto-Encoders are like nonlinear PCA



Minimizing reconstruction error
forces latent representation of
“similar inputs” to stay on
manifold

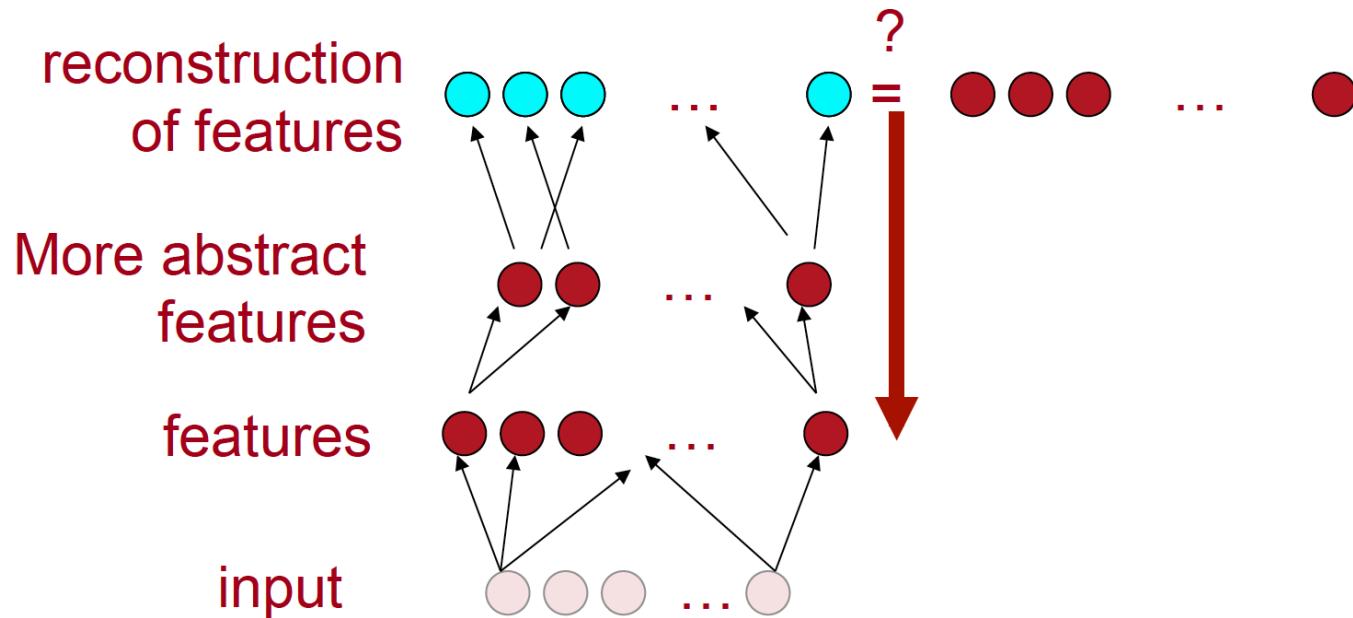
from Socher and Manning

Stacking for deep learning



from Socher and Manning

Stacking for deep learning

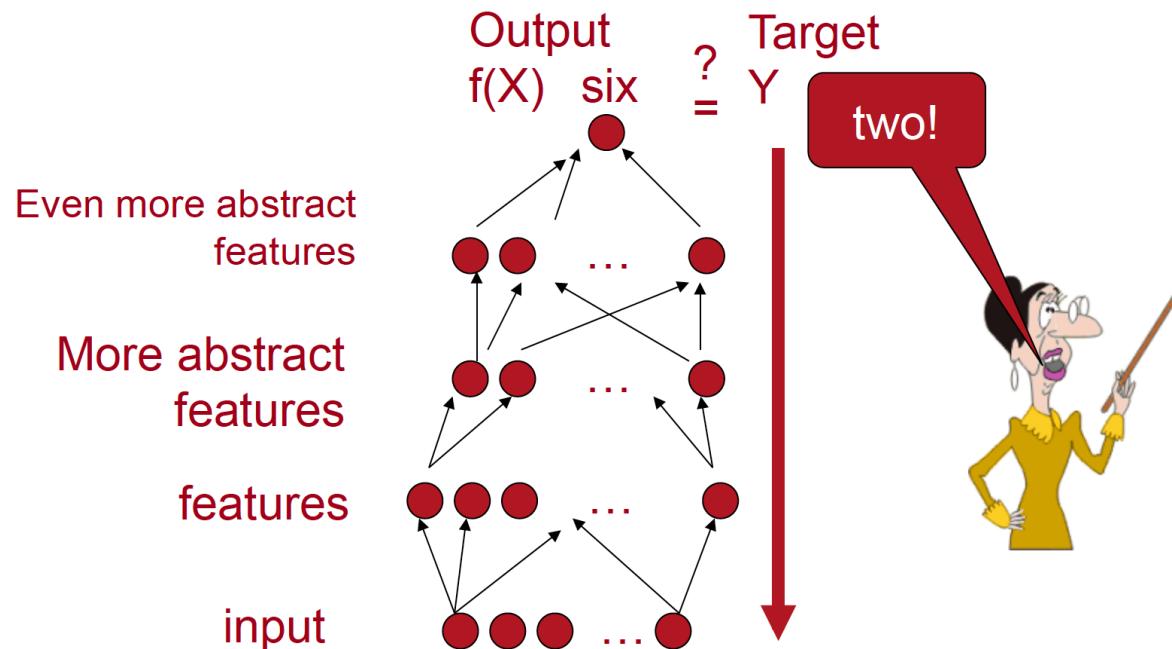


Now learn to reconstruct the features
(using more abstract ones)

from Socher and Manning

Stacking for deep learning

- ◆ Recurse – many layers deep
- ◆ Gives embeddings



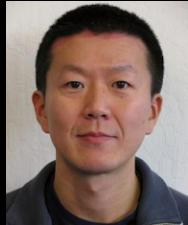
from Socher and Manning

Tera-scale deep learning

Quoc V. Le
Stanford University and Google

Now at google

Joint work with



Kai Chen



Greg Corrado



Jeff Dean



Matthieu Devin



Rajat Monga



Andrew Ng



Marc' Aurelio
Ranzato



Paul Tucker



Ke Yang

Additional
Thanks:

Samy Bengio, Zhenghao Chen, Tom Dean, Pangwei Koh,
Mark Mao, Jiquan Ngiam, Patrick Nguyen, Andrew Saxe,
Mark Segal, Jon Shlens, Vincent Vanhoucke, Xiaoyun Wu,
Peng Xe, Serena Yeung, Will Zou

Warning: this x and W are
the transpose of what we use

TICA:

$$\min_W \sum_j \sum_i h_j(W; x^{(i)})$$

$$s.t. \quad WW^T = I$$

Reconstruction ICA:

$$\min_W \left[\frac{\lambda}{m} \sum_{i=1}^m \|W^T W x^{(i)} - x^{(i)}\|_2^2 \right] + \sum_j \sum_i h_j(W; x^{(i)})$$

Lemma 3.1 When the input data $\{x^{(i)}\}_{i=1}^m$ is whitened, the reconstruction cost $\frac{\lambda}{m} \sum_{i=1}^m \|W^T W x^{(i)} - x^{(i)}\|_2^2$ is equivalent to the orthonormality cost $\lambda \|W^T W - \mathbf{I}\|_{\mathcal{F}}^2$.

Lemma 3.2 The column orthonormality cost $\lambda \|W^T W - \mathbf{I}_n\|_{\mathcal{F}}^2$ is equivalent to the row orthonormality cost $\lambda \|WW^T - \mathbf{I}_k\|_{\mathcal{F}}^2$ up to an additive constant.

- Equivalence between Sparse Coding, Autoencoders, RBMs and ICA
- Build deep architecture by treating the output of one layer as input to another layer

Training



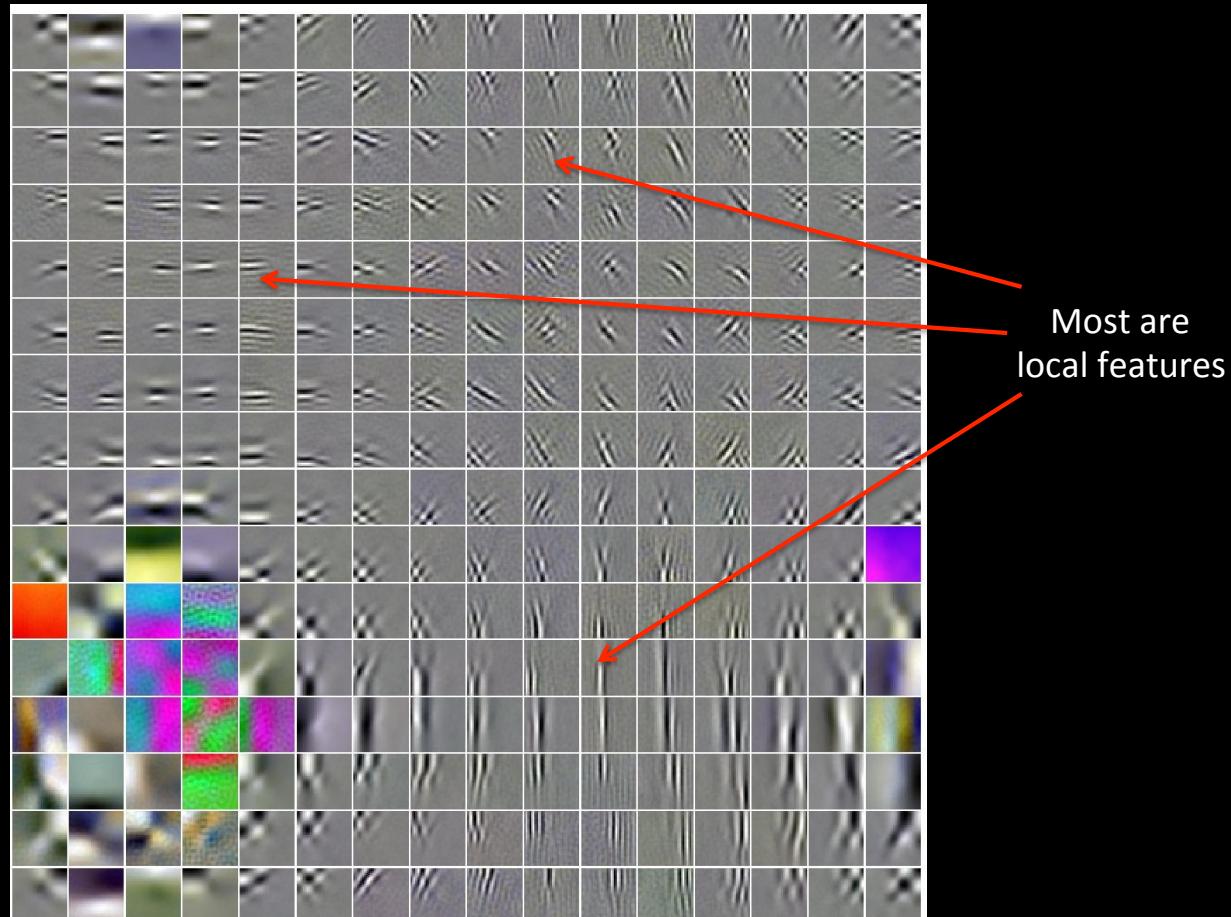
Dataset: 10 million 200x200 unlabeled images from YouTube/Web

Train on 2000 machines (16000 cores) for 1 week

1.15 billion parameters

- 100x larger than previously reported
- Small compared to visual cortex

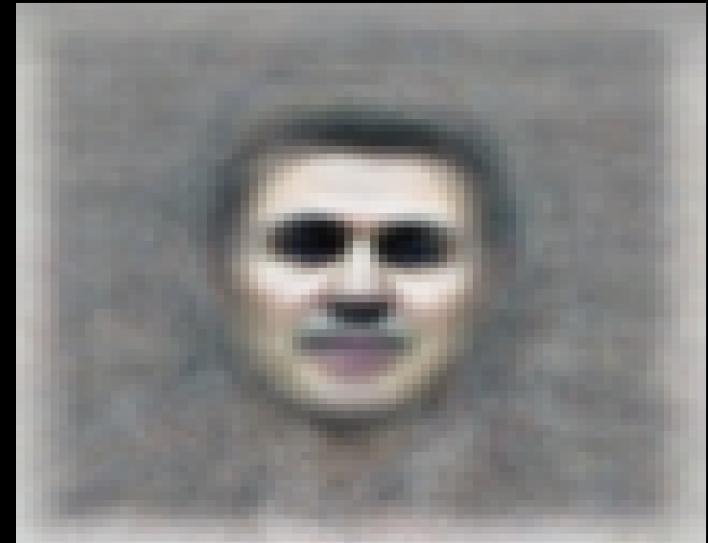
Visualization of features learned



The face neuron



Top stimuli from the test set



Optimal stimulus
by numerical optimization

The cat neuron



Top stimuli from the test set



Optimal stimulus
by numerical optimization

What you should know

- ◆ ICA
 - Like PCA but does *disentanglement* as well as reconstruction
- ◆ Unsupervised neural nets (auto-encoders)
 - Generalize PCA or ICA
 - Denoising or variational
 - Often trained recursively
 - Often learn an “overcomplete basis”
 - Used in semi-supervised learning
- ◆ Transformers use masking