The rest of the semester

- **HWs get less demanding**
  - Two weeks for HW7
  - Replaced by work on the project
- **Start thinking about final project**
  - 3 person teams
  - Real data set
    - with enough complexity to be interesting
    - Kaggle = minimal
    - But not too much data cleaning!
  - Details and rubric next week

Self-Supervised learning
Unsupervised Neural Nets: Autoencoders and ICA

ICA vs. PCA
Autoencoder types
denoising
variational

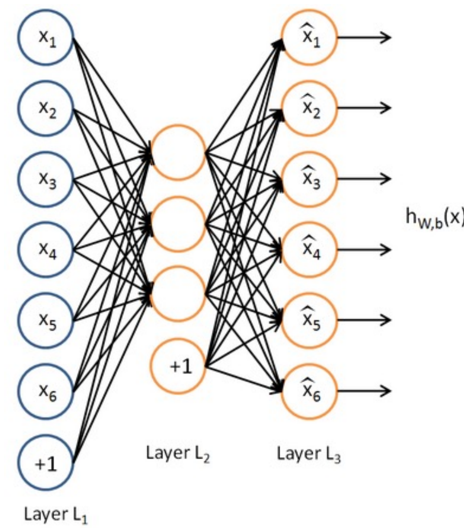
Lyle Ungar

with figures from

Quoc Le, Socher & 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**

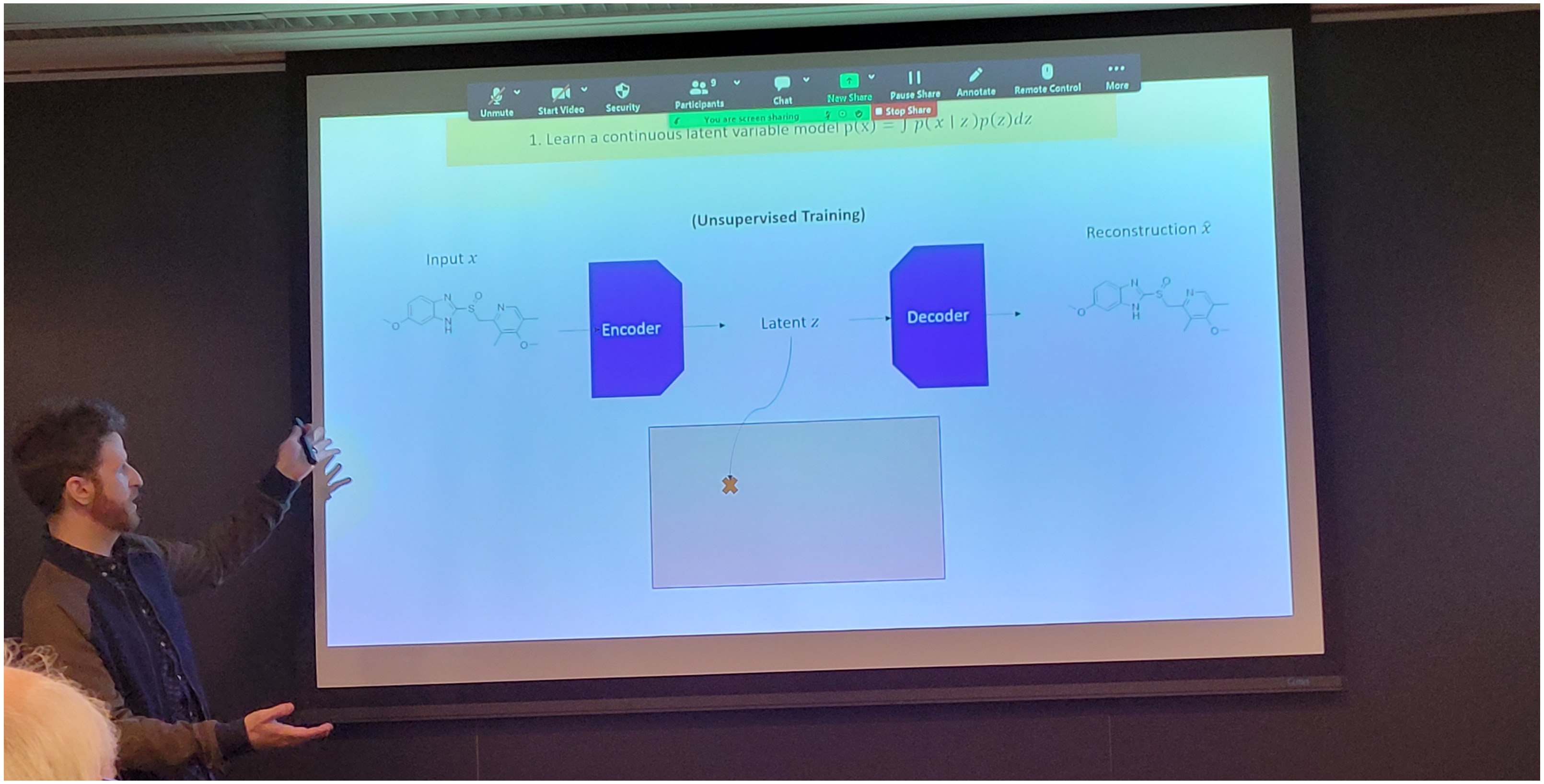
Transformer

- **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)

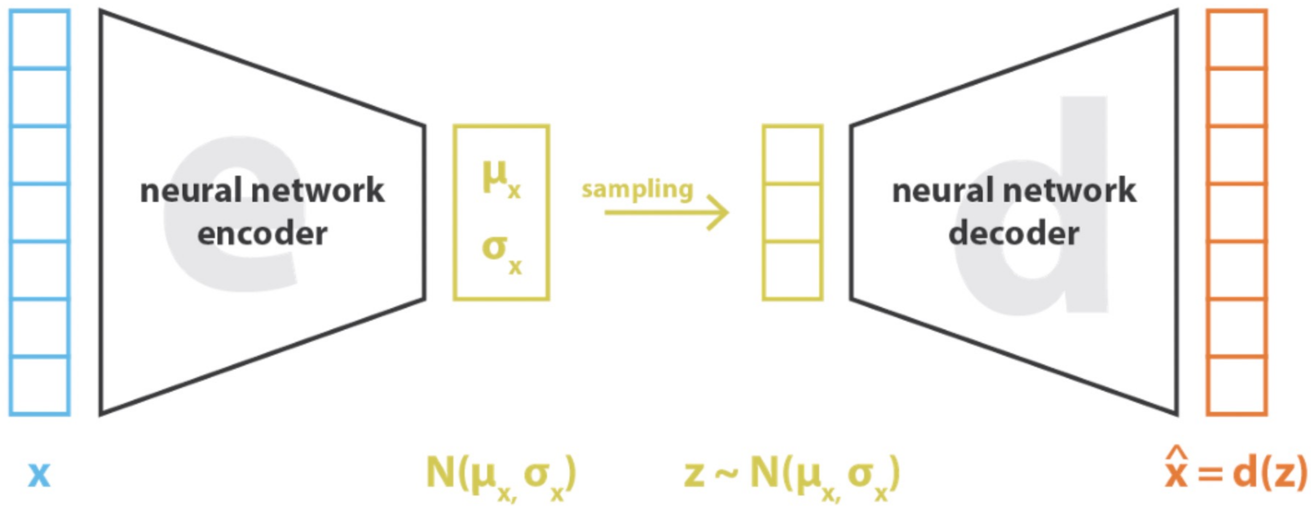
- **Minimize reconstruction error and maximize independence of ”components”**
  - Reminiscent of a mixture model (which we haven’t covered yet)
- **Intermediate neural outputs (components) are an embedding**

Variational Auto-Encoder (VAE)



Jake Gardner, Penn

Variational Auto-Encoder (VAE)



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Independent Components Analysis (ICA)

- Given observations **X**, find **W** such that components $\mathbf{s}_j$ of **S** = **XW** are "as independent of each other as possible"
  - E.g. have maximum KL-divergence or low mutual information
  - Alternatively, find directions in **X** that are most skewed
    - farthest from Gaussian
  - Usually mean center and "whiten" the data
    - whiten: make unit covariance
    - whiten: $X (X^T X)^{-1/2}$
- **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 **X**, find **W** and **S** such that components $\mathbf{s}_j$ of $\mathbf{S} = \mathbf{XW}$ are "as independent of each other as possible"
  - S_k = "sources" should be independent
- Reconstruct $\mathbf{X} \sim (\mathbf{XW})\mathbf{W}^+ = \mathbf{SW}^+$
  - **S** like *principal component scores*
  - $\mathbf{W}^+$ like *loadings*
  - $\boldsymbol{x} \sim \Sigma_j \; s_j \, \boldsymbol{w}_j^+$
- **Auto-encoder** – nonlinear generalization that "encodes" **X** as **S** and then "decodes" it

Reconstruction ICA (RICA)

- **Reconstruction ICA: find W to minimize**
  - Reconstruction error
    - $||\mathbf{X} - \mathbf{S}\mathbf{W}^{+}||_2 = ||\mathbf{X} - (\mathbf{X}\mathbf{W})\mathbf{W}^{+}||_2$

And minimize

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

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

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

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

http://mathworld.wolfram.com/DifferentialEntropy.html

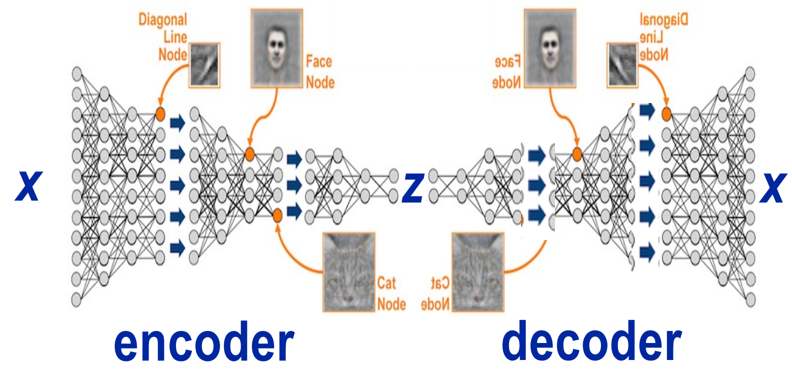
Mutual information

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

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



from Socher and Manning

Unsupervised deep learning

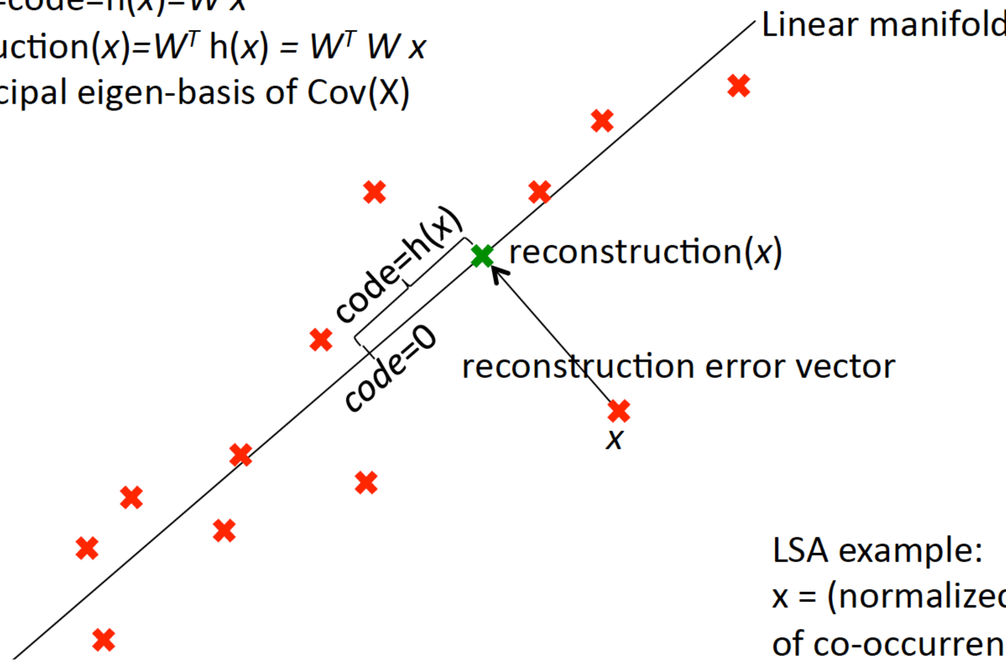
PCA = Linear Manifold = Linear Auto-encoder

input x, 0-mean

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

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

W = principal eigen-basis of Cov(X)



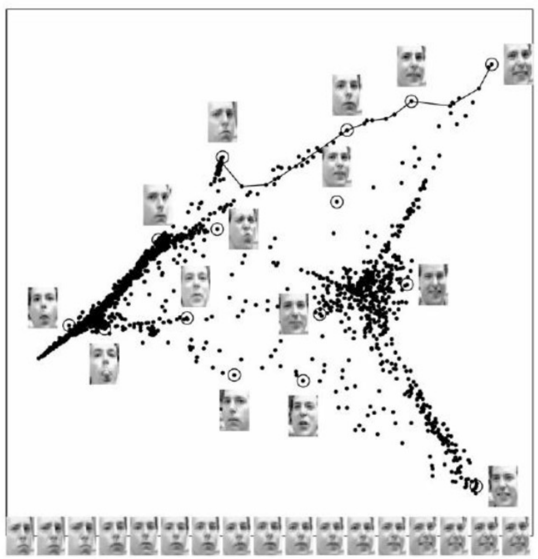
LSA example:

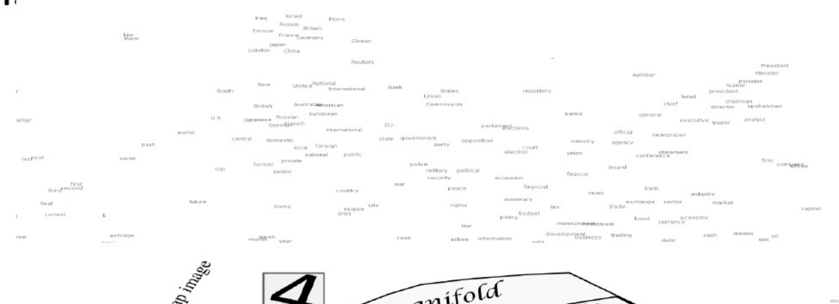
x = (normalized) distribution of co-occurrence frequencies

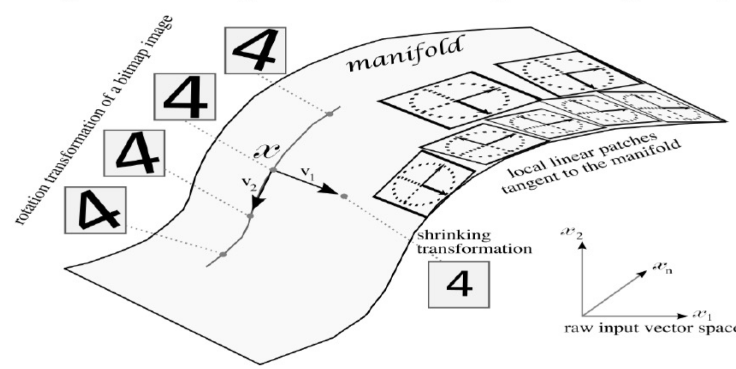
from Socher and Manning

The Manifold Learning Hypothesis

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

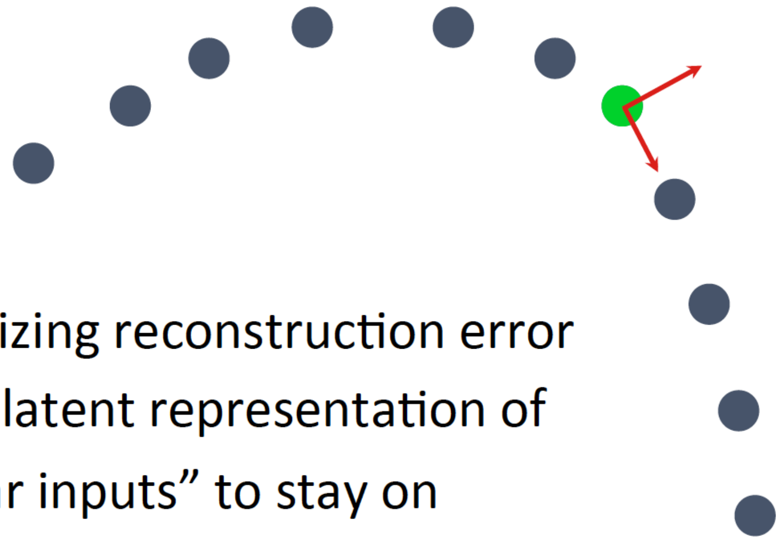






from Socher and Manning

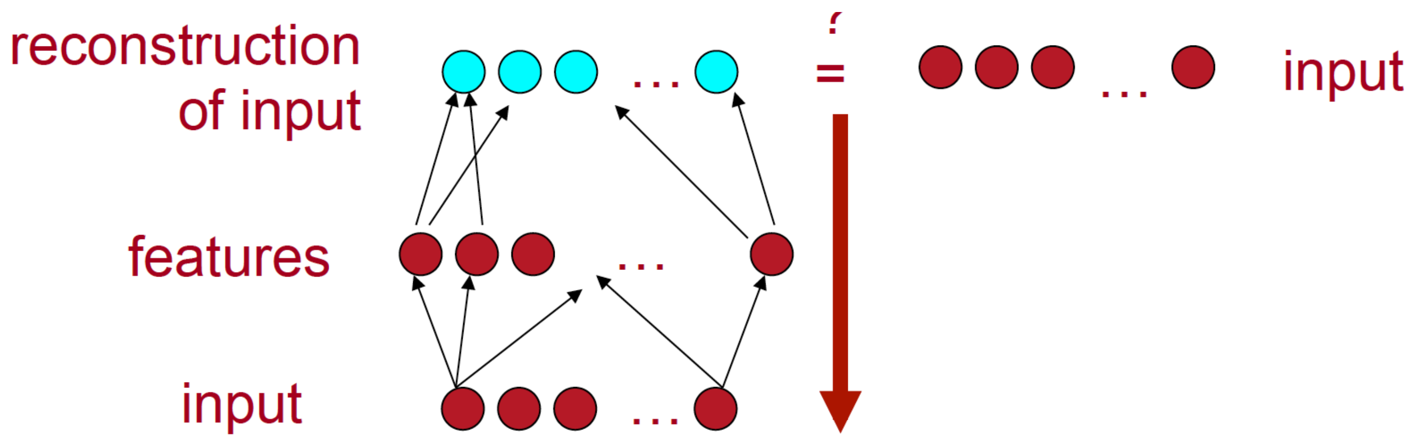
Auto-Encoders are like nonlinear PCA



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

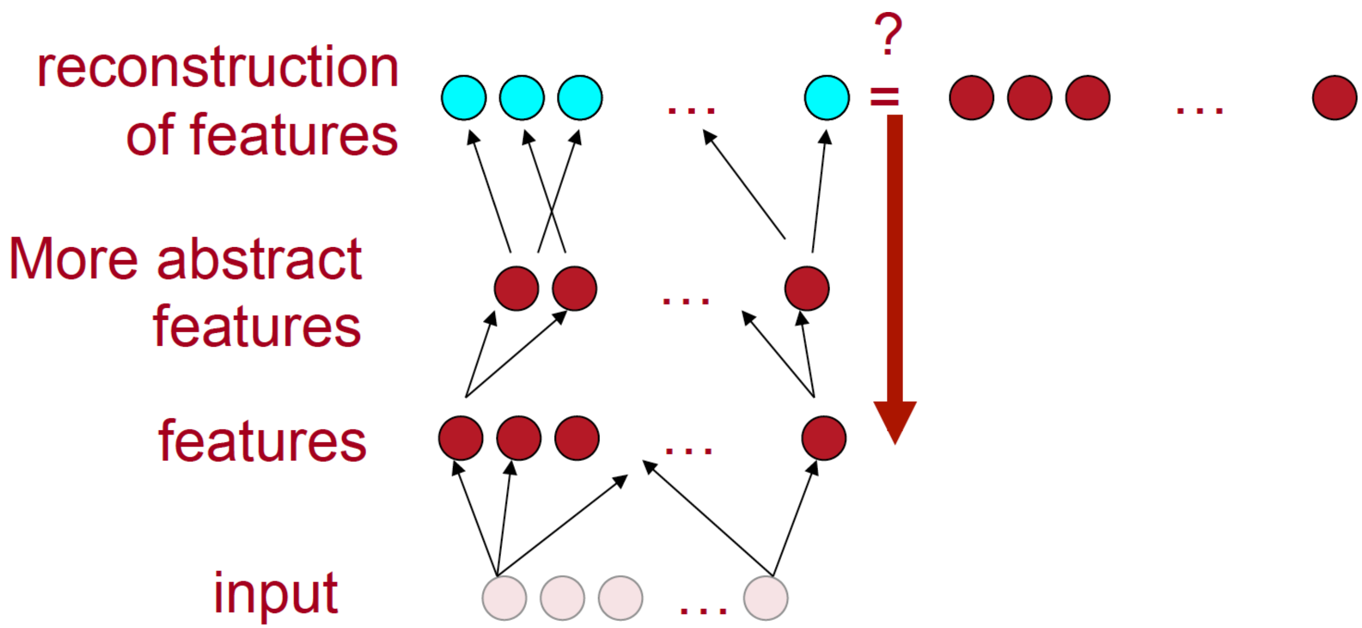
from Socher and Manning

Stacking for deep learning



from Socher and Manning

Stacking for deep learning

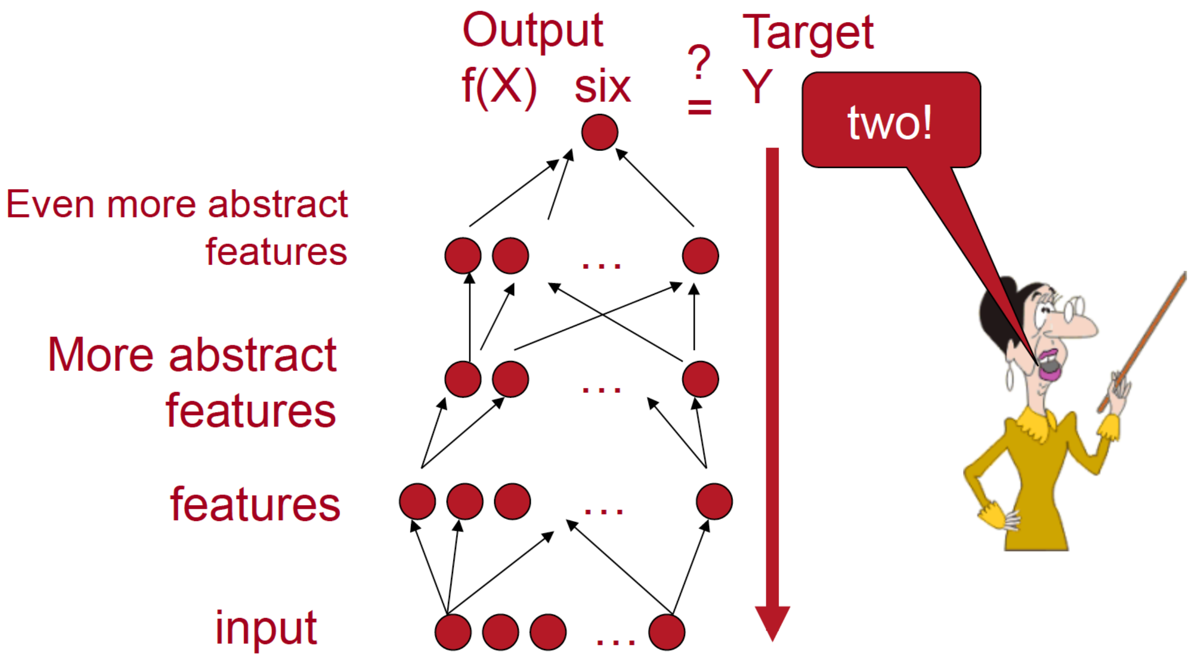


Now learn to reconstruct the features (using more abstract ones)

from Socher and Manning

Stacking for deep learning

- Recurse – many layers deep
- Gives embeddings to use in supervised learning



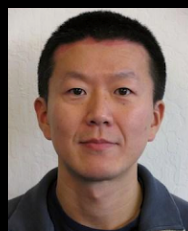
from Socher and Manning

Tera-scale deep learning

Quoc V. Le
Stanford University and Google

Now at google

Joint work with


Kai Chen


Greg Corrado


Jeff Dean

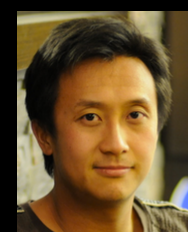

Matthieu Devin


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 Vanhouke, Xiaoyun Wu, Peng Xe, Serena Yeung, Will Zou

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

TICA:

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

$$s.t. \quad \boxed{WW^T = I}$$

Reconstruction ICA:

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

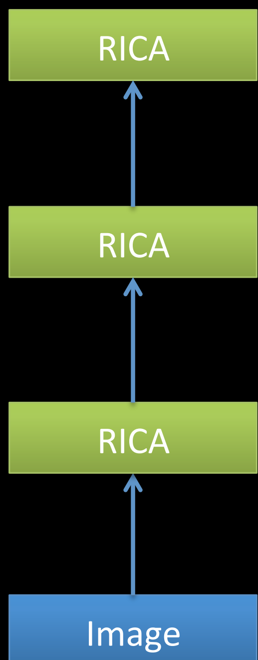
Lemma 3.1 *When the input data* $\{x^{(i)}\}_{i=1}^m$ *is whitened, the reconstruction cost* $\frac{\lambda}{m} \sum_{i=1}^m \|W^T W x^{(i)} - x^{(i)}\|_2^2$ *is equivalent to the orthonormality cost* $\lambda \|W^T W - \mathbf{I}\|_{\mathcal{F}}^2$.

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

Le, et al., *ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning*. NIPS 2011

Training



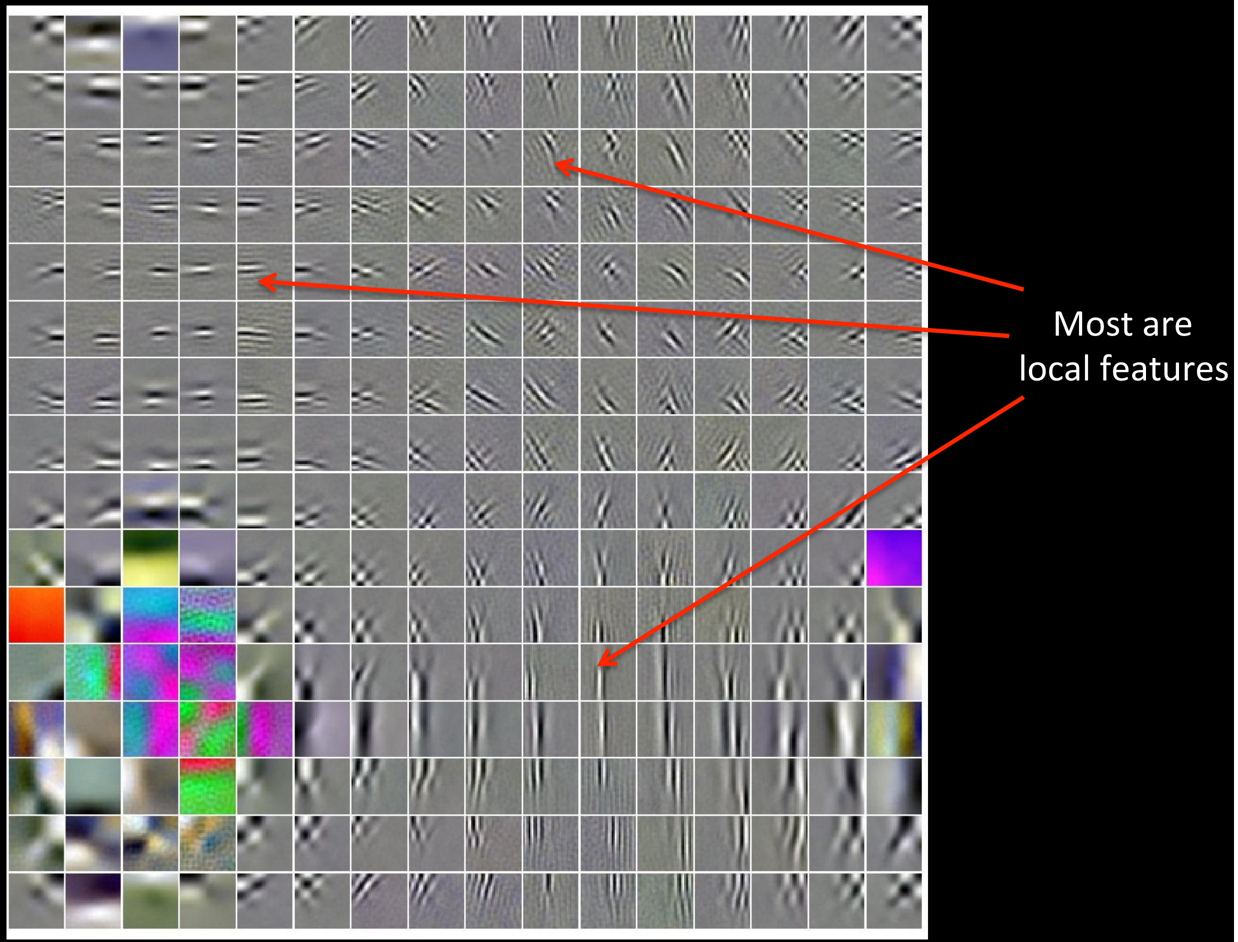
Dataset: 10 million 200x200 unlabeled images from YouTube/Web

Train on 2000 machines (16000 cores) for 1 week

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

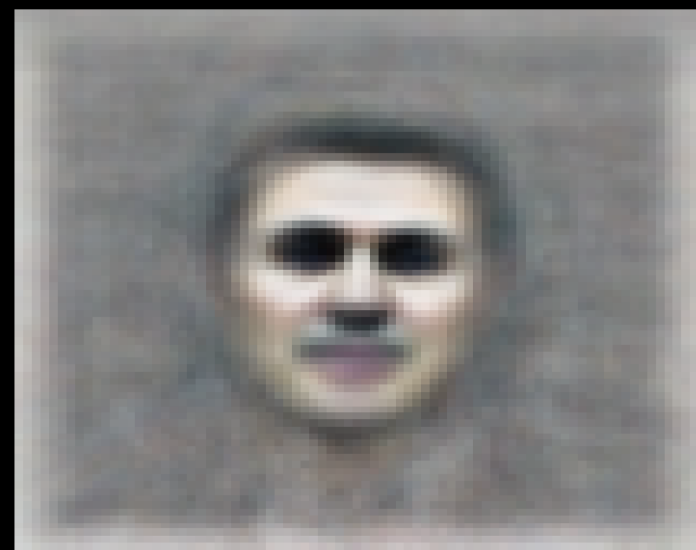
Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

Visualization of features learned



The face neuron

Top stimuli from the test set



Optimal stimulus
by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

The cat neuron

Top stimuli from the test set

Optimal stimulus
by numerical optimization

Le, et al., *Building high-level features using large-scale unsupervised learning*. ICML 2012

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**