Deep Learning for NLP

(without Magic)

Richard Socher and Christopher Manning

Stanford University

NAACL 2013, Atlanta

http://nlp.stanford.edu/courses/NAACL2013/

*with a big thank you to Yoshua Bengio, with whom we participated in the previous ACL 2012 version of this tutorial

Deep autoencoders

Alternative to contrastive unsupervised word learning

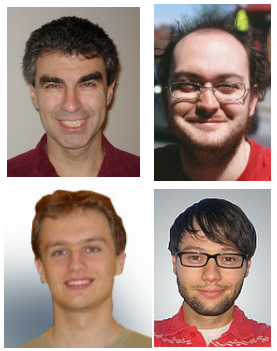
- Another is RBMs (Hinton et al. 2006), which we don't cover today

Works well for fixed input representations

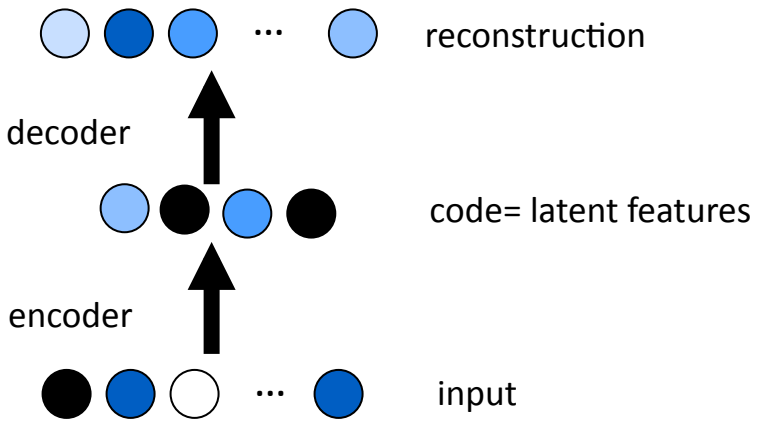
1. Definition, intuition and variants of autoencoders
2. Stacking for deep autoencoders
3. Why do autoencoders improve deep neural nets so much?

Auto-Encoders

- Multilayer neural net with target output = input
- Reconstruction=decoder(encoder(input))



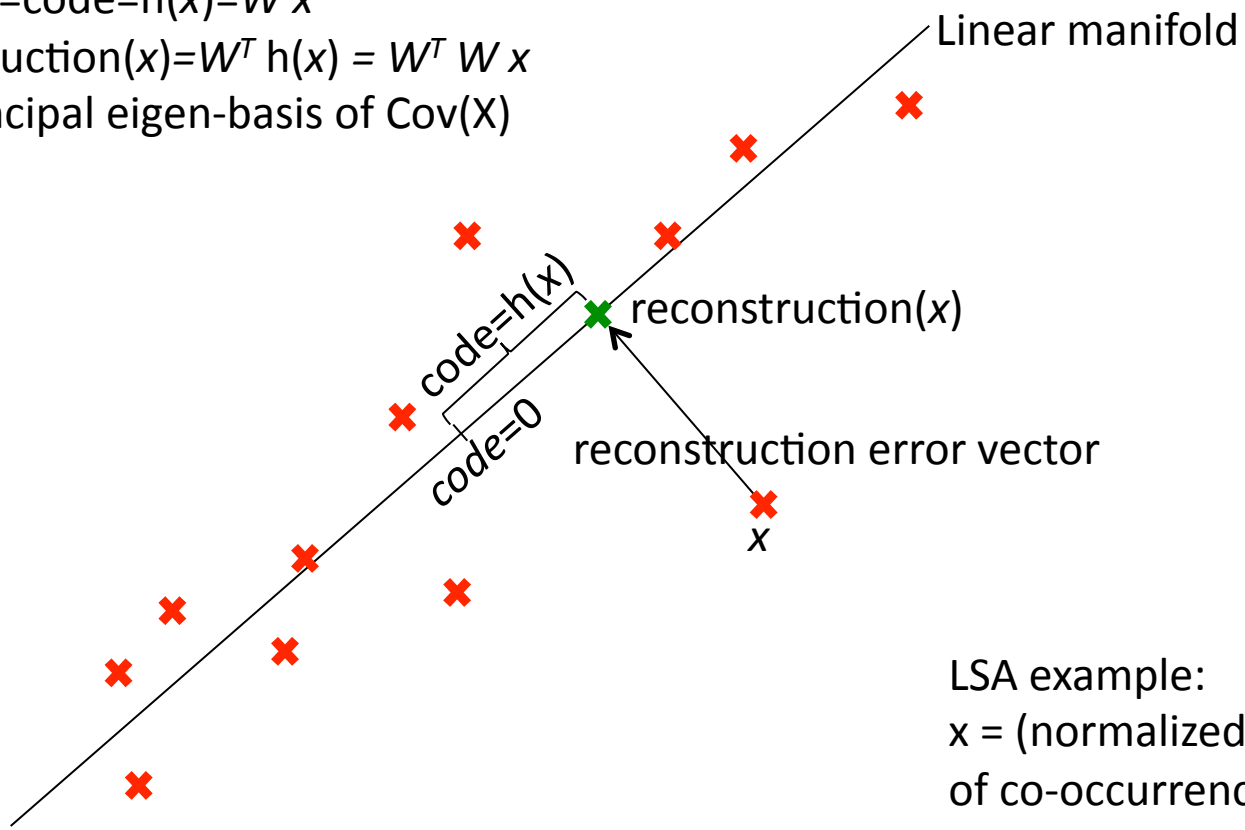
$$
\begin{aligned}
a &= \tanh(Wx + b) \\
x' &= \tanh(W^T a + c) \\
cost &= ||x' - x||^2
\end{aligned}
$$



- Probable inputs have small reconstruction error

PCA = Linear Manifold = Linear Auto-Encoder

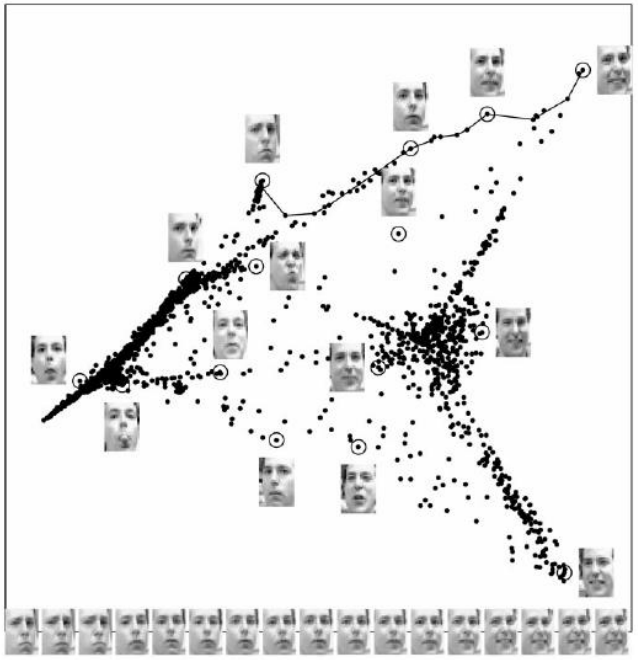
input x, 0-mean
features=code=h(*x*)=*W x*
reconstruction(*x*)=W^T h(*x*) = W^T *W x*
W = principal eigen-basis of Cov(X)

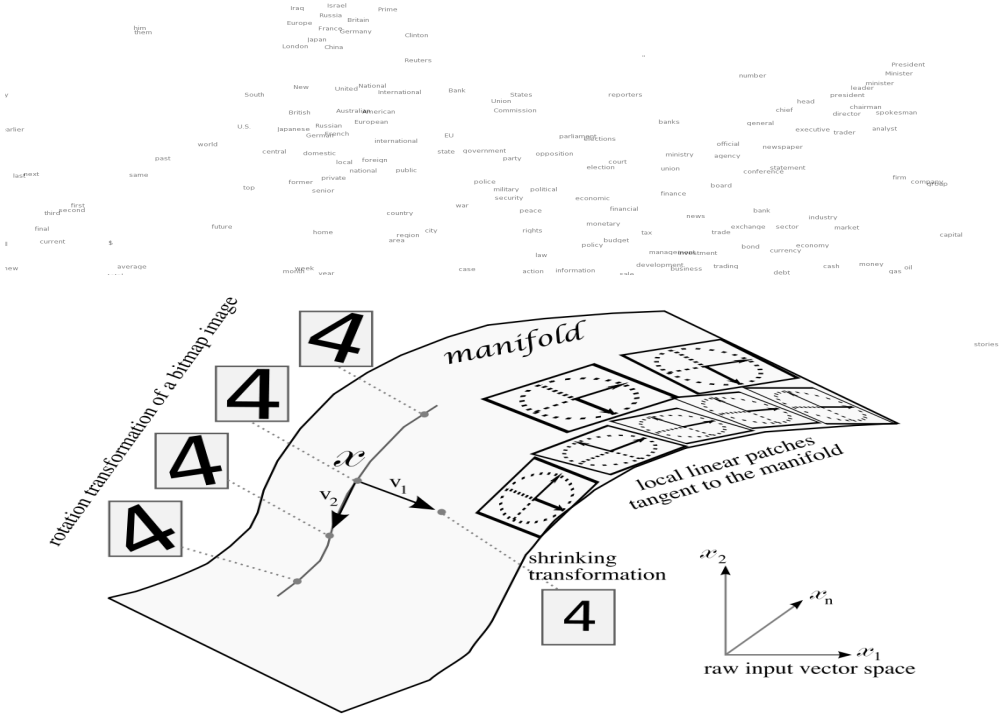


LSA example:
x = (normalized) distribution
of co-occurrence frequencies

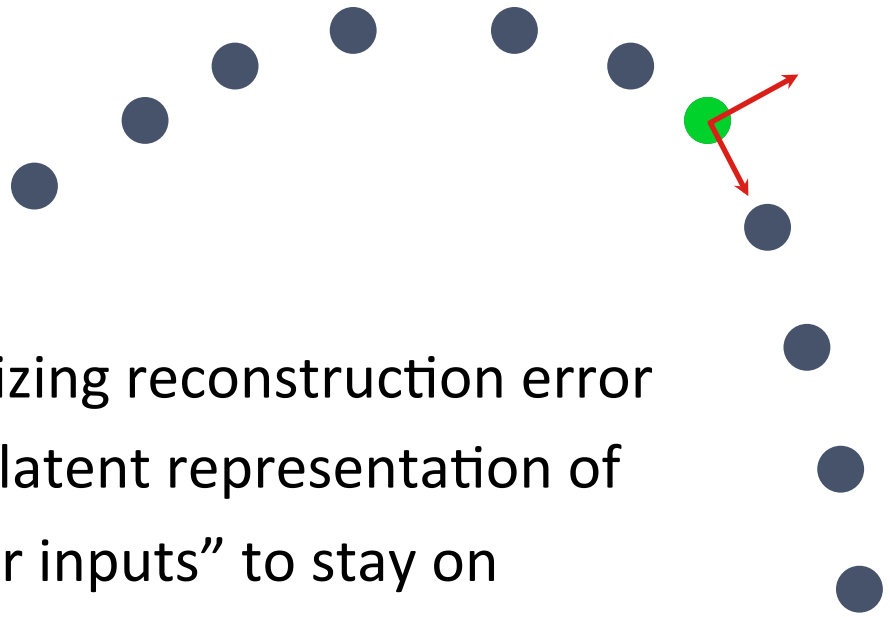
The Manifold Learning Hypothesis

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





Auto-Encoders Learn Salient Variations, like a non-linear PCA

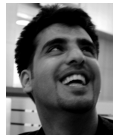


Minimizing reconstruction error forces latent representation of "similar inputs" to stay on manifold

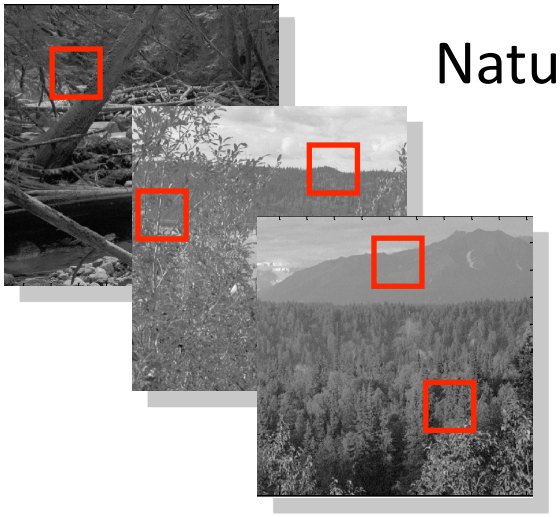
Auto-Encoder Variants

- Discrete inputs: cross-entropy or log-likelihood reconstruction criterion (similar to used for discrete targets for MLPs)

- Preventing them to learn the identity everywhere:
  - Undercomplete (eg PCA): bottleneck code smaller than input
  - Sparsity: penalize hidden unit activations so at or near 0
    [Goodfellow et al 2009]
  - Denoising: predict true input from corrupted input
    [Vincent et al 2008]
  - Contractive: force encoder to have small derivatives
    [Rifai et al 2011]

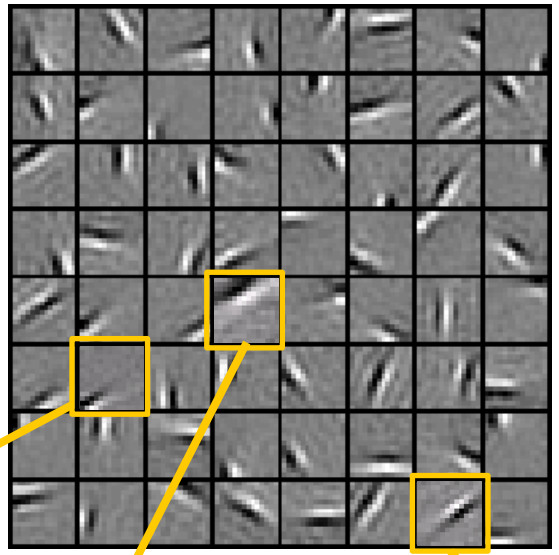


Sparse autoencoder illustration for images

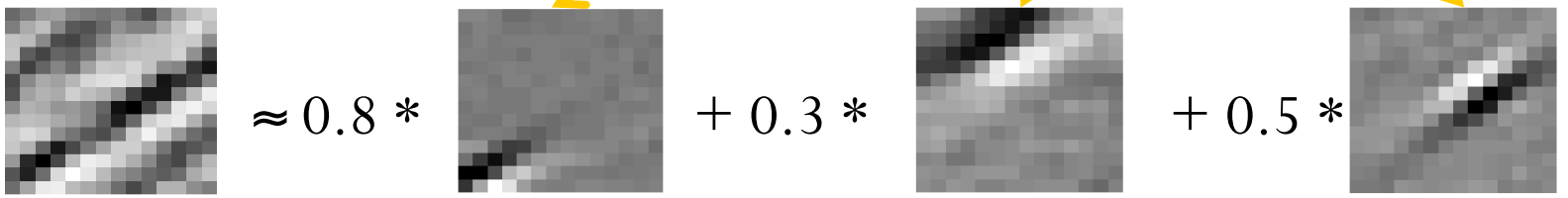


Natural Images

Learned bases:



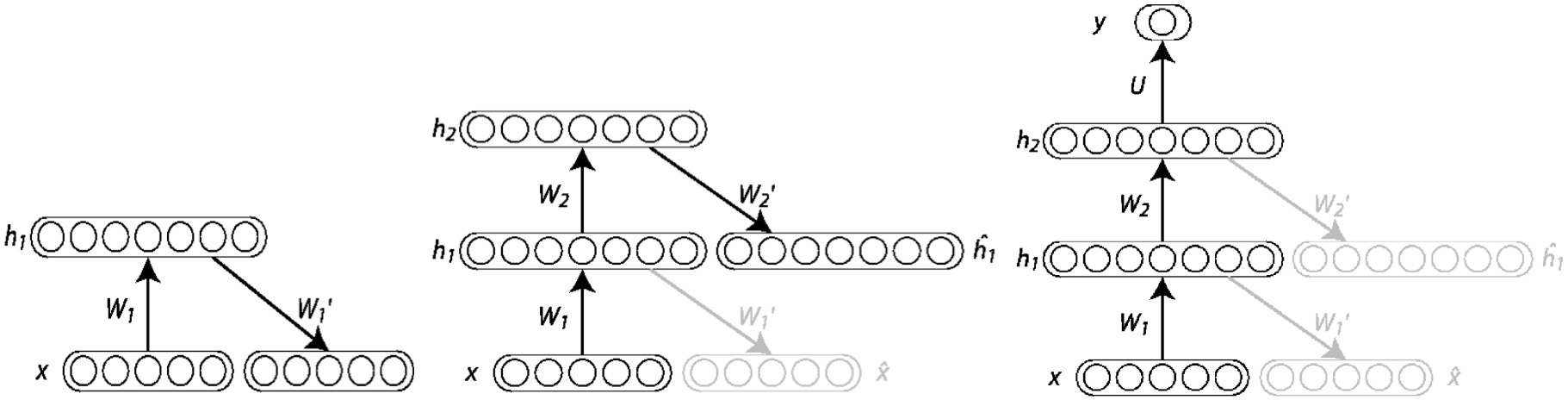
Test example



$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

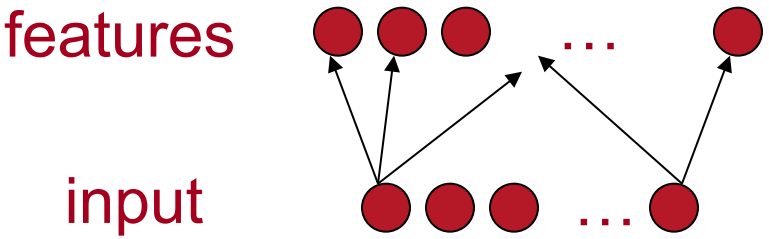
Stacking Auto-Encoders

- Can be stacked successfully (Bengio et al NIPS'2006) to form highly non-linear representations

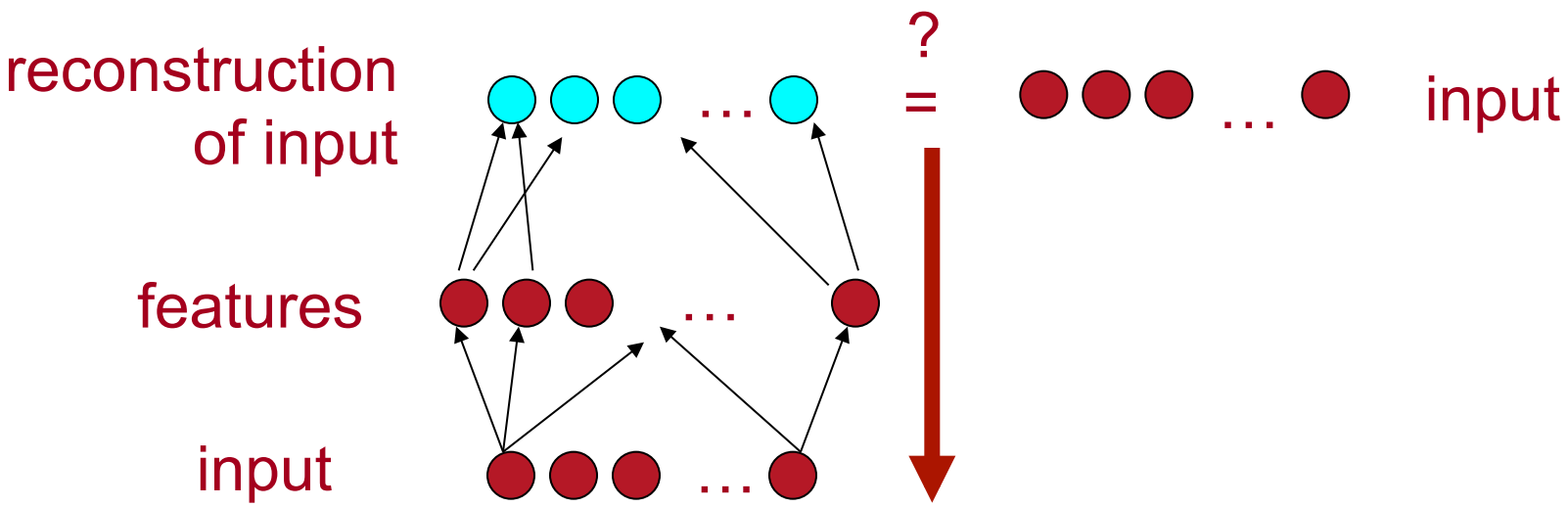


Layer-wise Unsupervised Learning

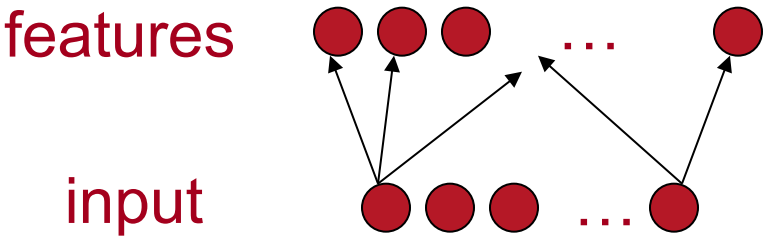
Layer-wise Unsupervised Pre-training



Layer-wise Unsupervised Pre-training



Layer-wise Unsupervised Pre-training

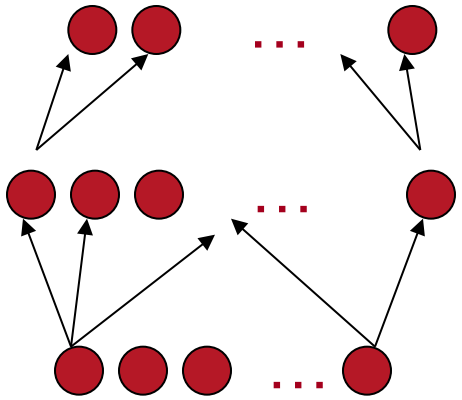


Layer-wise Unsupervised Pre-training

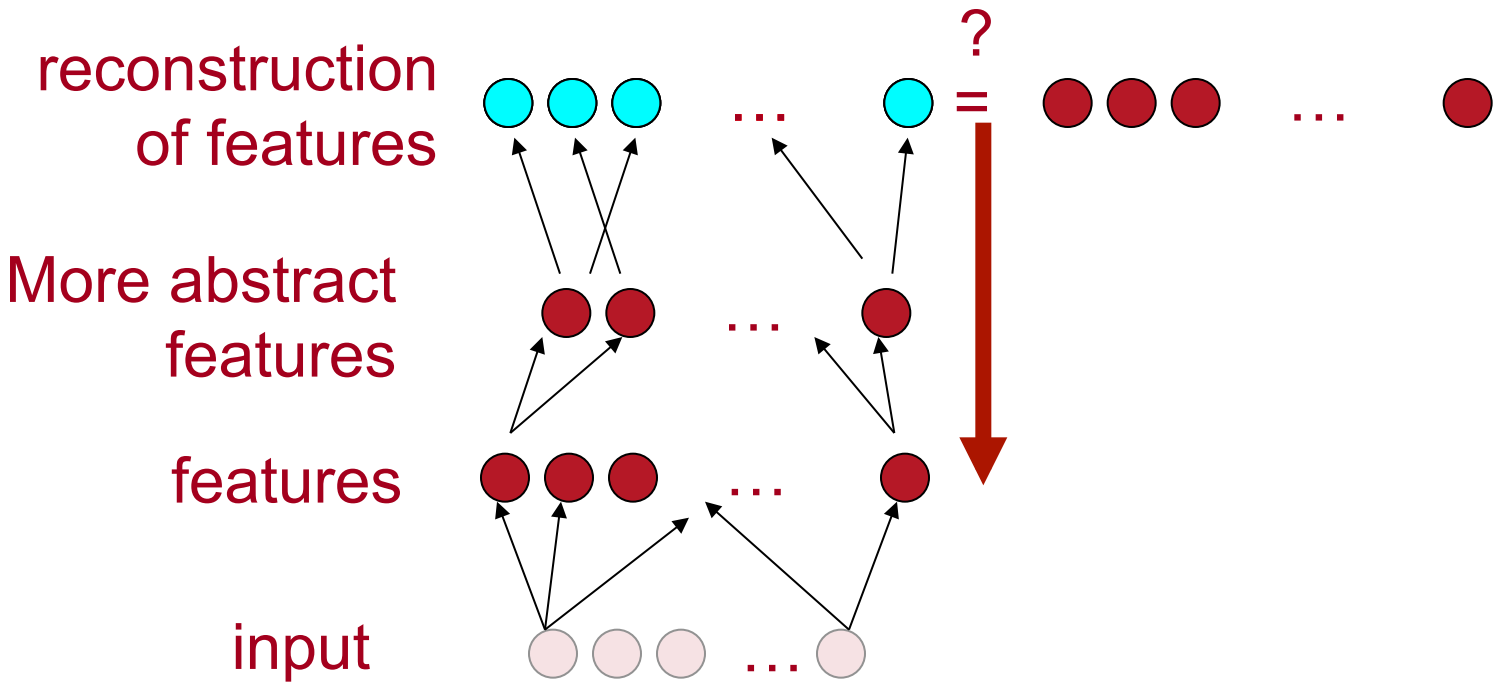
More abstract features

features

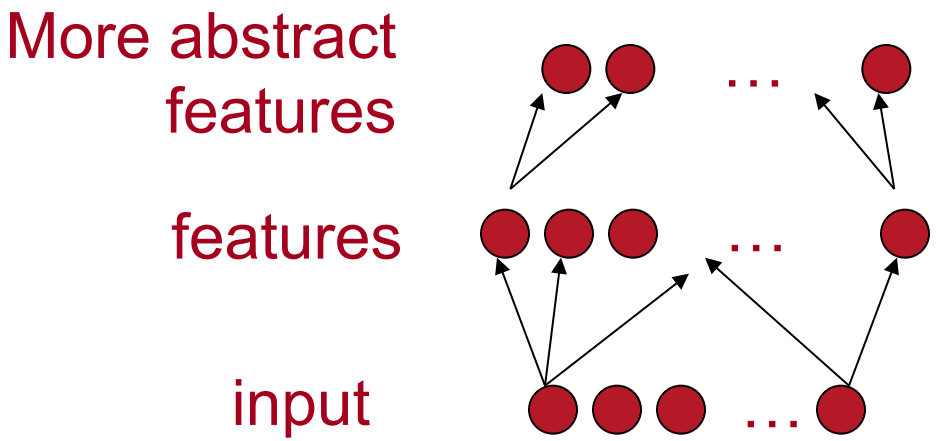
input



Layer-wise Unsupervised Learning



Layer-wise Unsupervised Pre-training



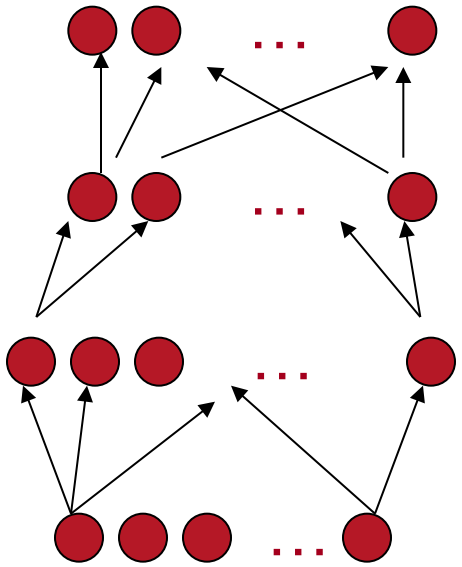
Layer-wise Unsupervised Learning

Even more abstract features

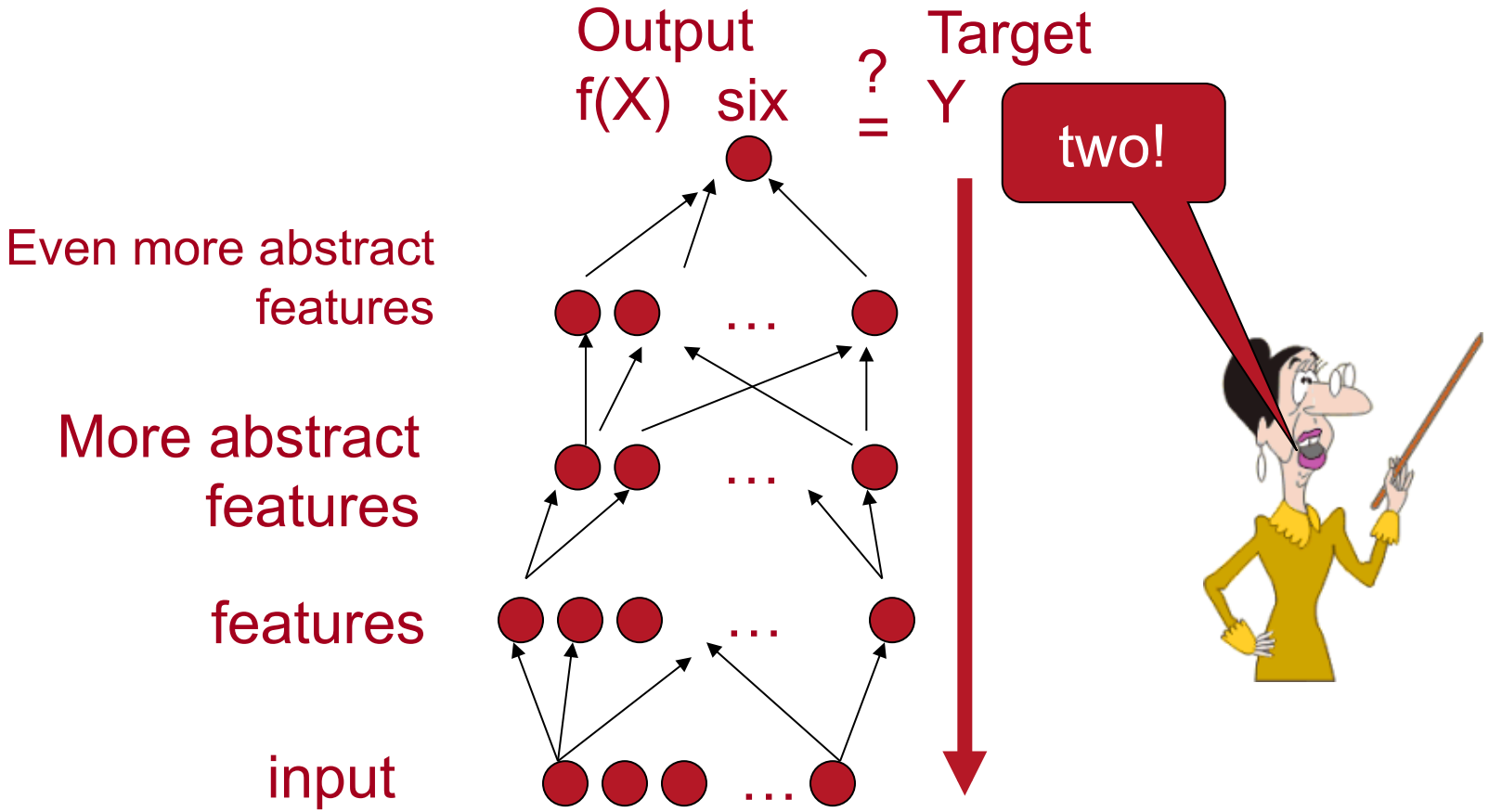
More abstract features

features

input

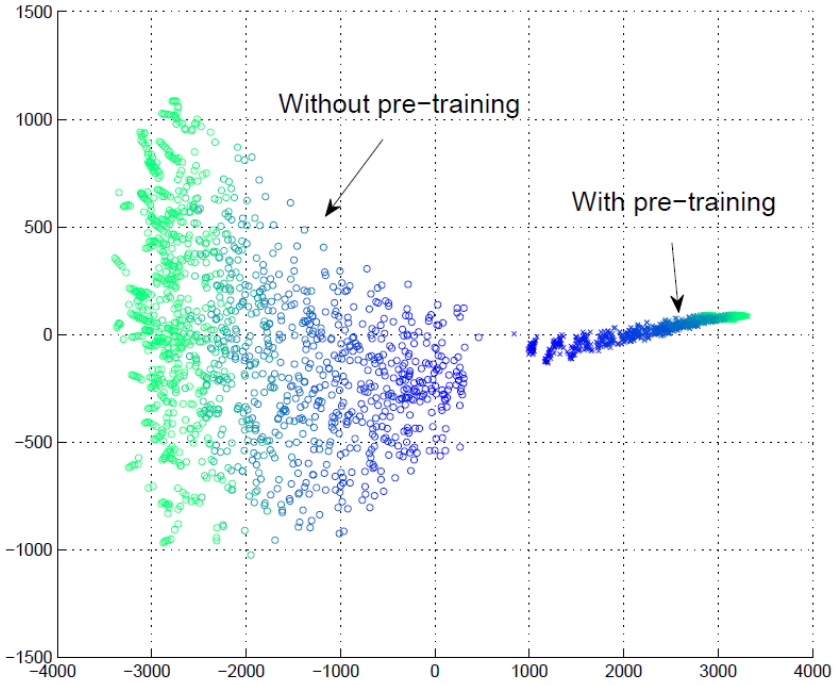


Supervised Fine-Tuning



Why is unsupervised pre-training working so well?

- Regularization hypothesis:
  - Representations good for P(x) are good for P(y|x)
- Optimization hypothesis:
  - Unsupervised initializations start near better local minimum of supervised training error
  - Minima otherwise not achievable by random initialization



Erhan, Courville, Manzagol, Vincent, Bengio (JMLR, 2010)