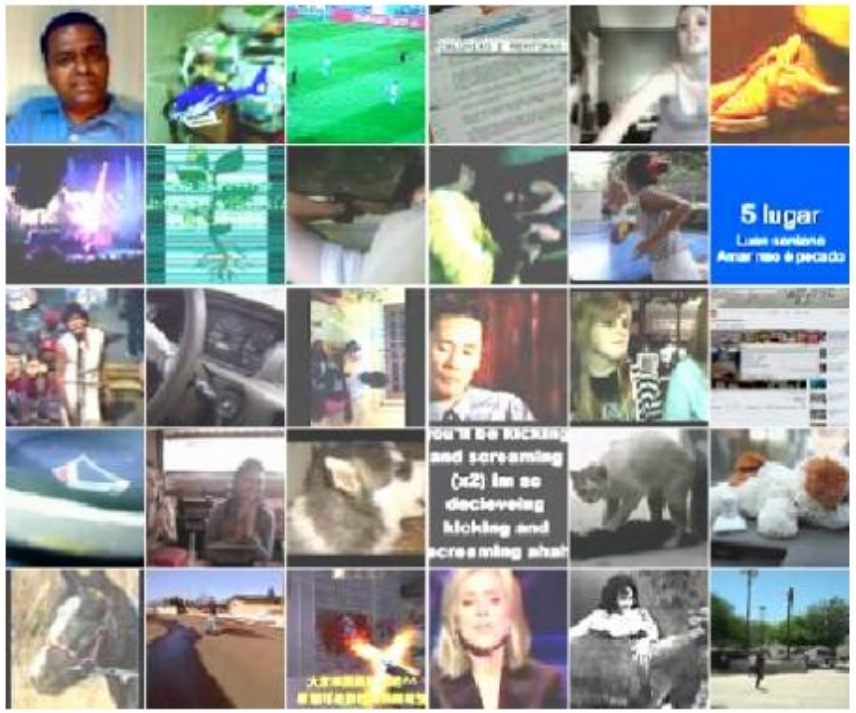
Unsupervised Pretraining

Based on the slides from Mike Mozer and Mitesh M Khapra

Discovering High-Level Features Via Unsupervised Learning
(Le et al., 2012)

Training set

- 10M YouTube videos
- single frame sampled from each
- 200 x 200 pixels
- fewer than 3% of frames contain faces (using OpenCV face detector)



Some Neurons Become Face Detectors

Look at all neurons in final layer and find the best face detector

Figure 3. Top: Top 48 stimuli of the best neuron from the test set. Bottom: The optimal stimulus according to numerical constraint optimization.

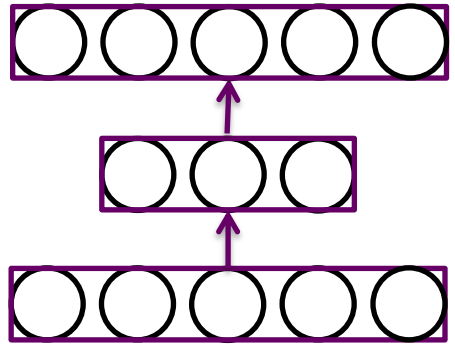
Some Neurons Become Cat and Body Detectors

Autoencoders

Self-supervised training procedure

Given a set of input vectors (no target outputs)

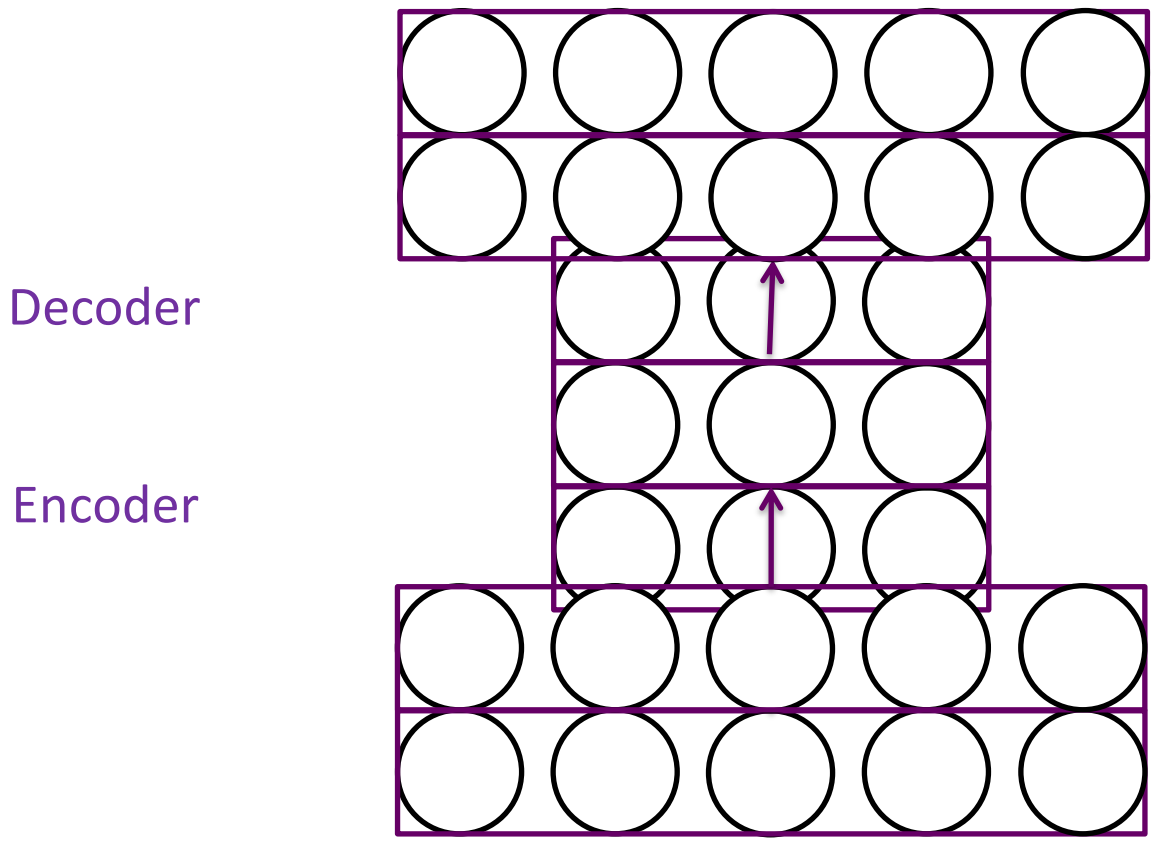
Map input back to itself via a hidden layer bottleneck



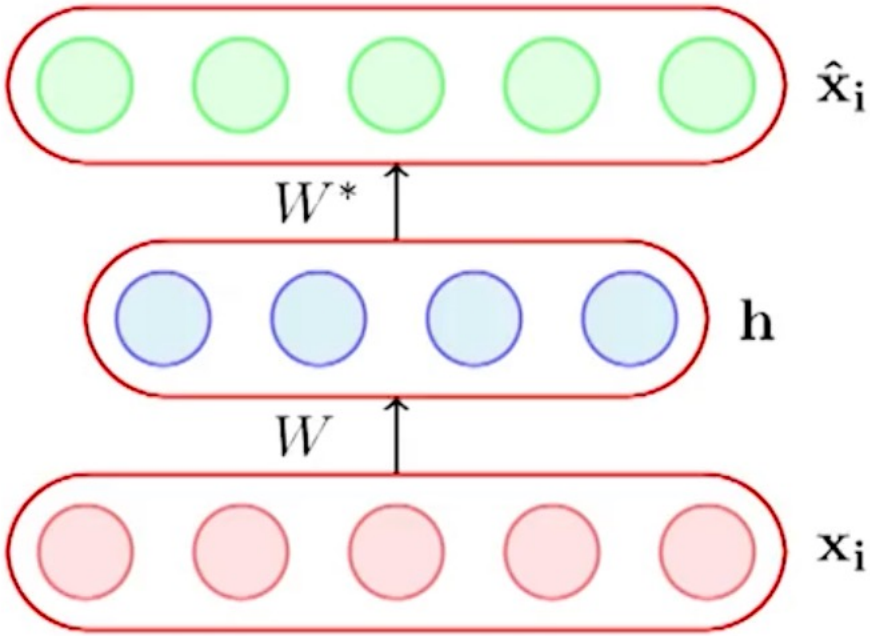
How to achieve bottleneck?

- Fewer neurons
- Sparsity constraint
- Information transmission constraint (e.g., add noise to unit, or shut off randomly, a.k.a. dropout)

Autoencoder Combines An Encoder And A Decoder



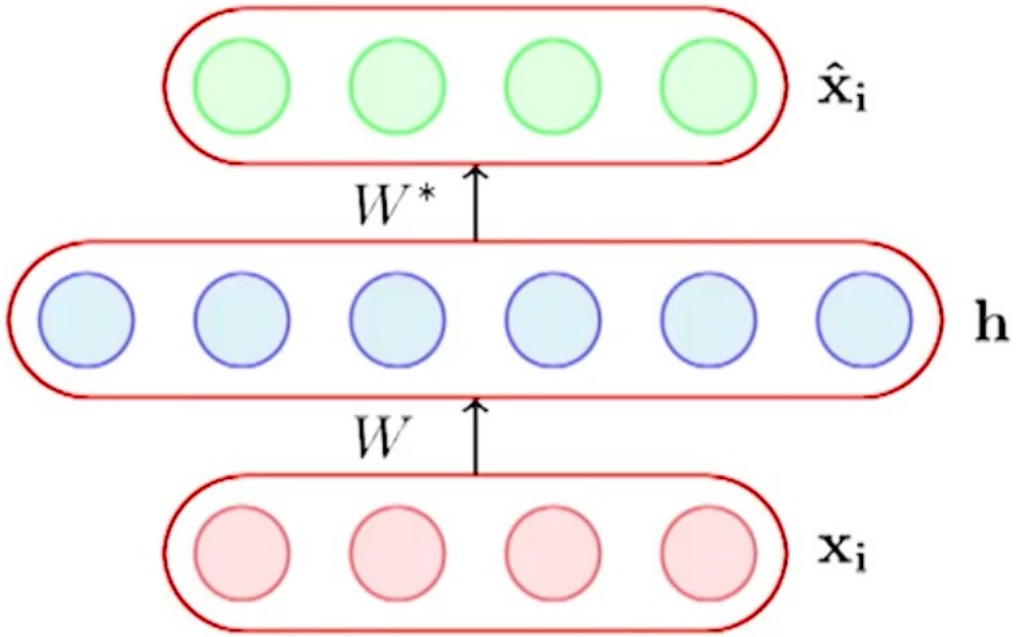
Autoencoder



$$\mathbf{h} = g(W\mathbf{x_i} + \mathbf{b})$$

$$\hat{\mathbf{x}}_\mathbf{i} = f(W^*\mathbf{h} + \mathbf{c})$$

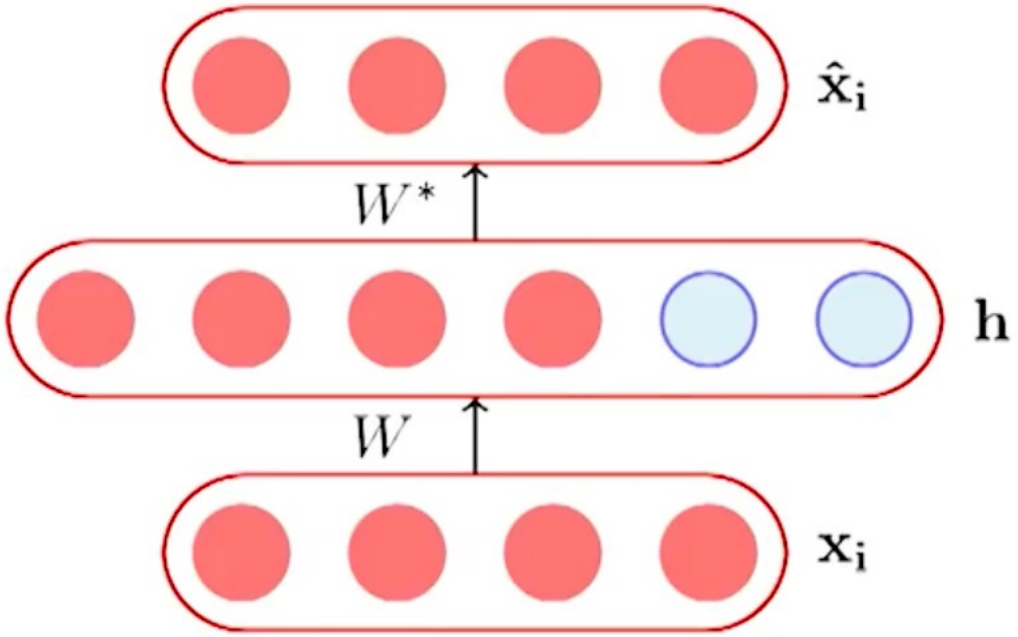
Autoencoder



$$\mathbf{h} = g(W\mathbf{x_i} + \mathbf{b})$$

$$\hat{\mathbf{x}}_\mathbf{i} = f(W^*\mathbf{h} + \mathbf{c})$$

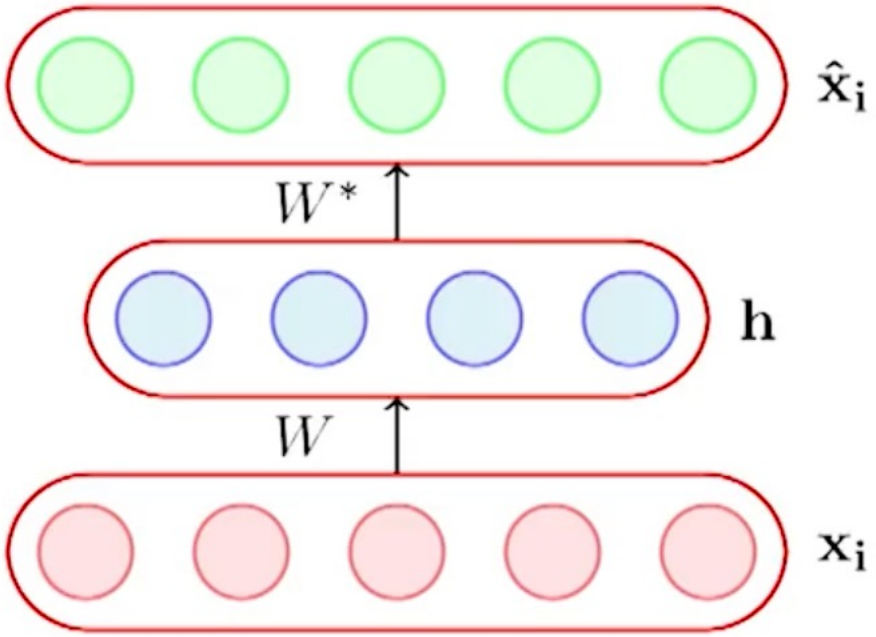
Autoencoder



$$\mathbf{h} = g(W\mathbf{x_i} + \mathbf{b})$$

$$\hat{\mathbf{x}}_\mathbf{i} = f(W^*\mathbf{h} + \mathbf{c})$$

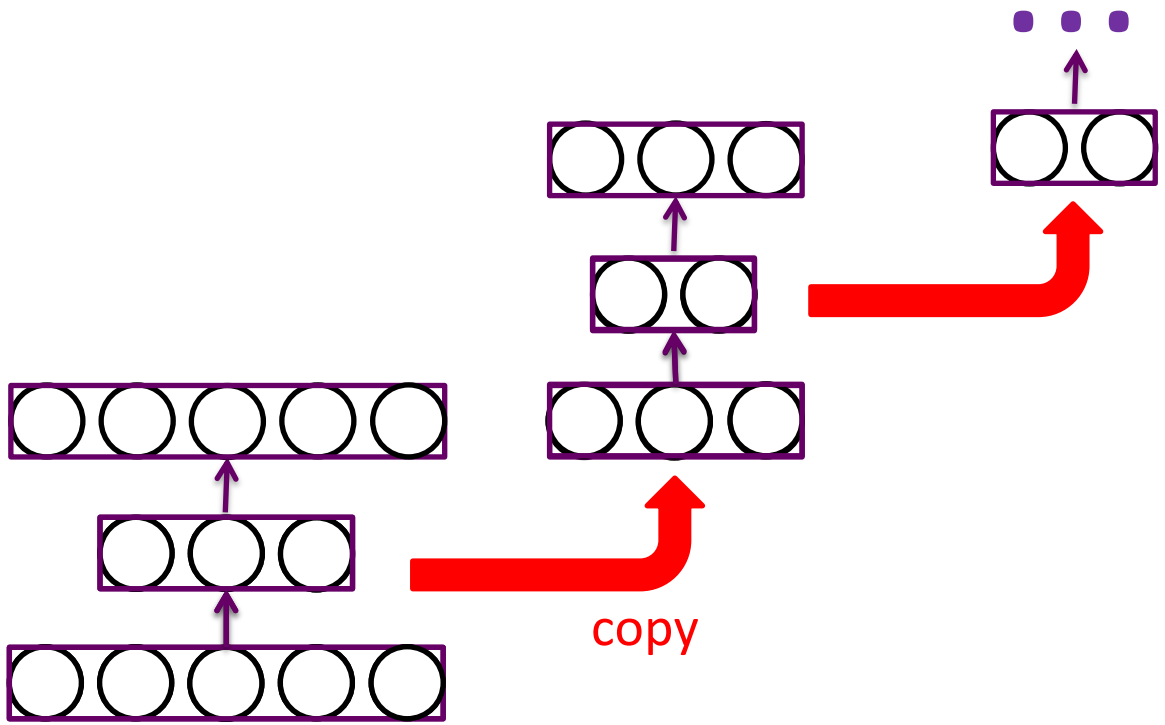
Autoencoder



$$\mathbf{h} = g(W\mathbf{x_i} + \mathbf{b})$$
$$\hat{\mathbf{x}}_\mathbf{i} = f(W^*\mathbf{h} + \mathbf{c})$$

$$\min_{W,W^*,\mathbf{c},\mathbf{b}} \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{x}_{ij} - x_{ij})^2$$

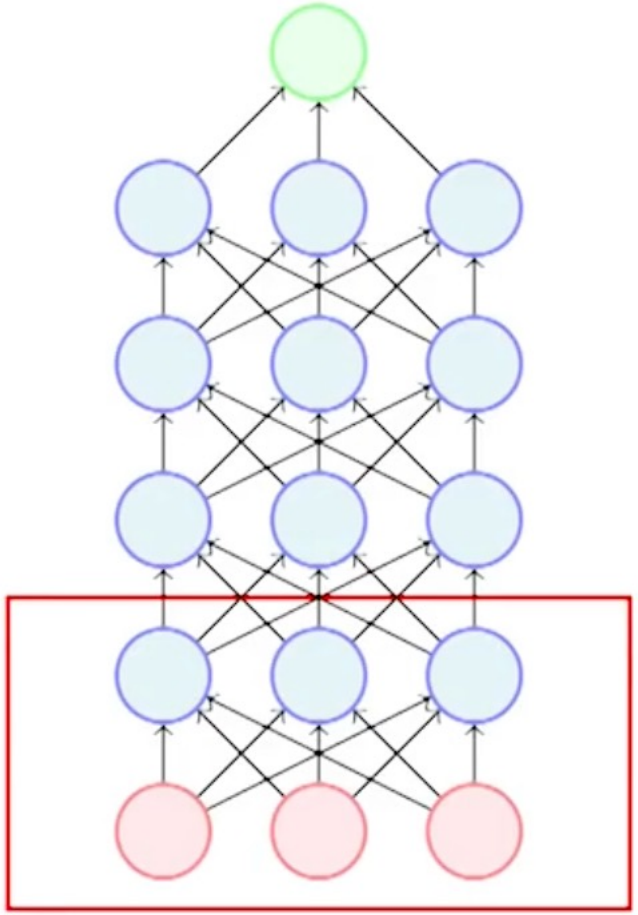
Stacked Autoencoders

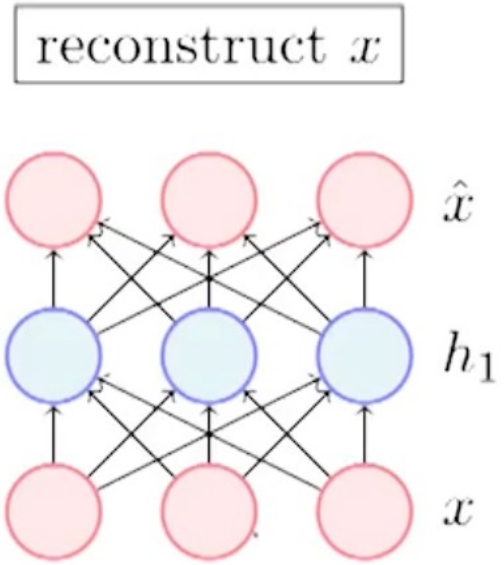


deep network

Note that decoders can be stacked to produce a generative domain model

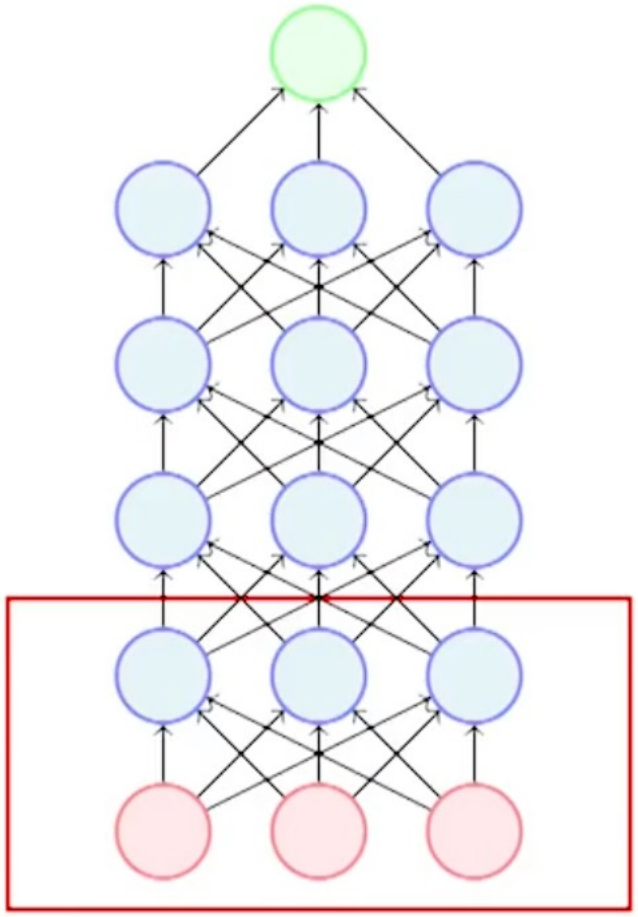
Unsupervised Pretraining

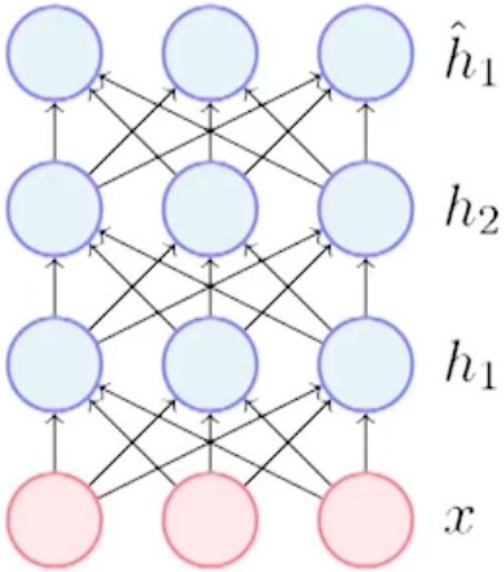




$$\min \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{x}_{ij} - x_{ij})^2$$

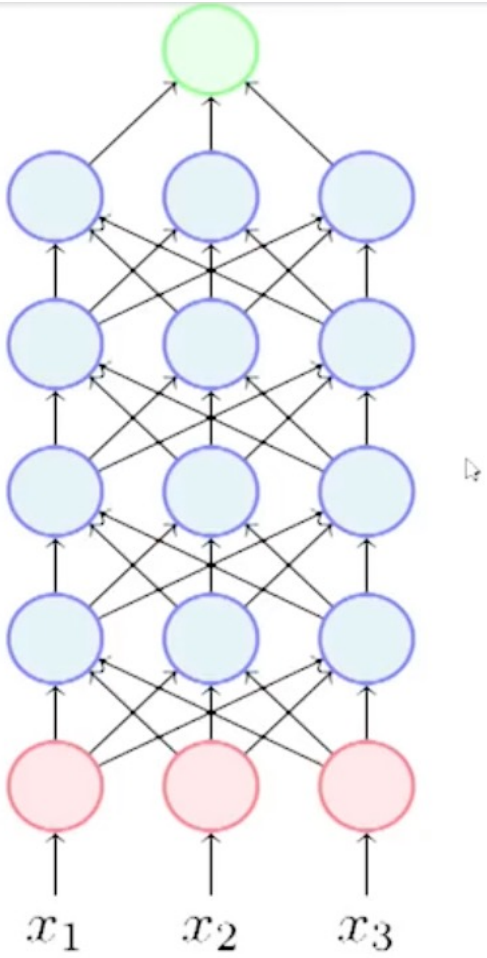
Unsupervised Pretraining





$$\min \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\hat{h}_{1_{ij}} - h_{1_{ij}})^2$$

Unsupervised Pretraining



$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i))^2$$

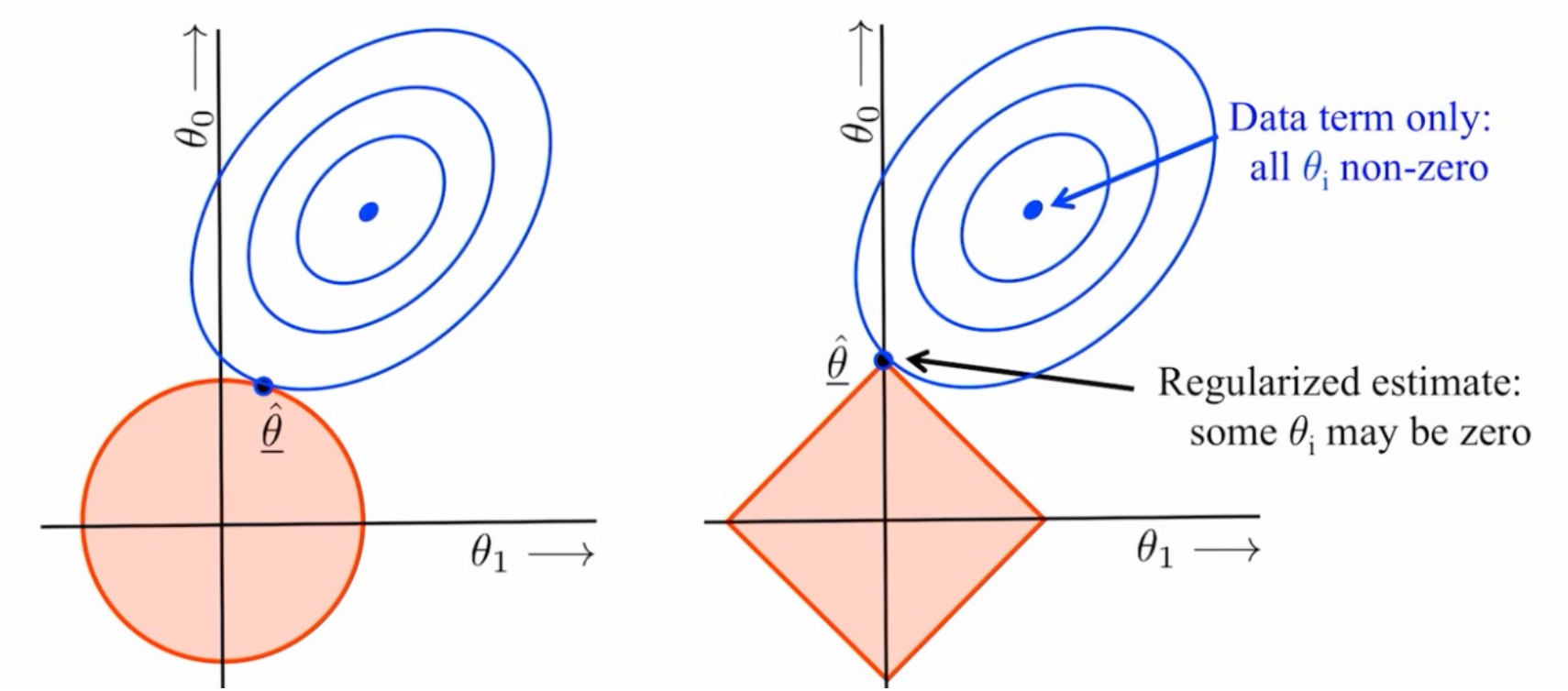
Unsupervised Pretraining

Why does this work better?

- Is it because of better optimization?
- Is it because of better regularization?

[1]The difficulty of training deep architectures and effect of unsupervised pre-training - Erhan et al,2009

Unsupervised Pretraining



Unsupervised Pretraining

Deep Learning has evolved

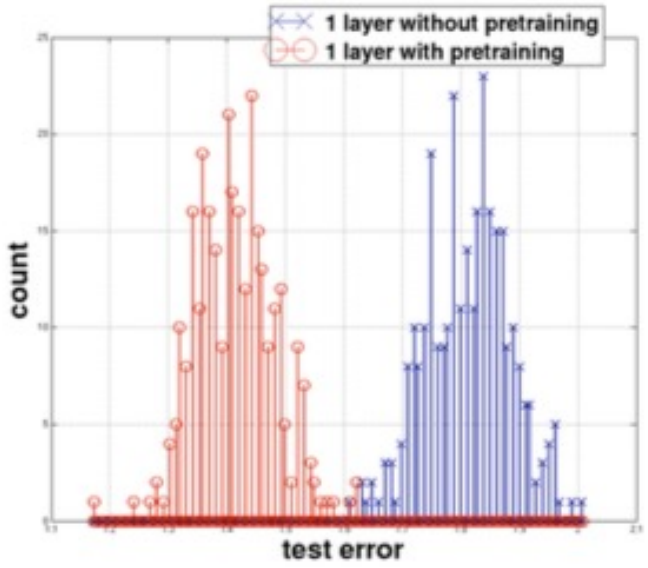
- Better optimization algorithms
- Better regularization methods
- Better activation functions
- Better weight initialization strategies

Why Does Unsupervised Pretraining Help Deep Learning?

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

"Unsupervised training guides the learning toward basins of attraction of minima that support better generalization from the training set"

- More hidd
  increase li
  poor local
- result is ro
  initializati



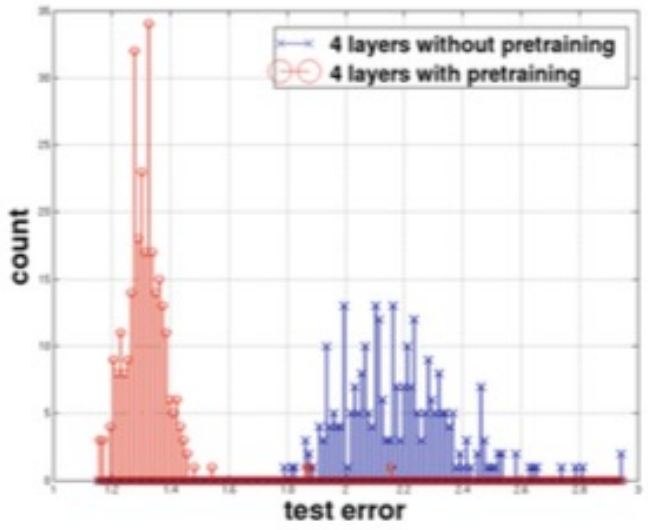


Figure 2: Histograms presenting the test errors obtained on MNIST using models trained with or without pre-training (400 different initializations each). **Left**: 1 hidden layer. **Right**: 4 hidden layers.

Using Autoencoders To Initialize Weights For Supervised Learning

Effective pretraining of weights should act as a regularizer

- **Limits the region of weight space that will be explored by supervised learning**

Autoencoder must be learned well enough to move weights away from origin

Acknowledgements

Some contents in the slides are borrowed from online resources for academic purpose

Author does not claim originality of these