Lecture 19: Autoencoders

CS 109B, STAT 121B, AC 209B, CSE 109B

Mark Glickman and Pavlos Protopapas

Supervised Learning

Given: (x, y)

Goal: Learn a mapping *h*: *X* -> *Y*

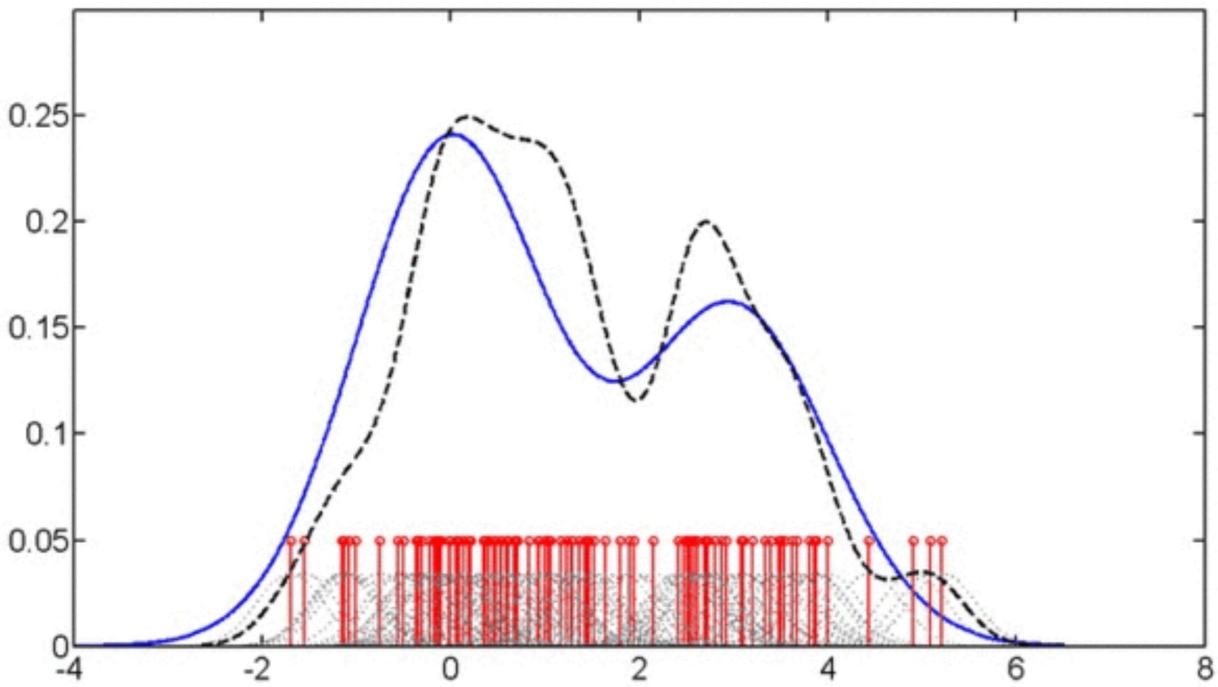
Unsupervised Learning

Given: x

Goal: Discover hidden structures from data

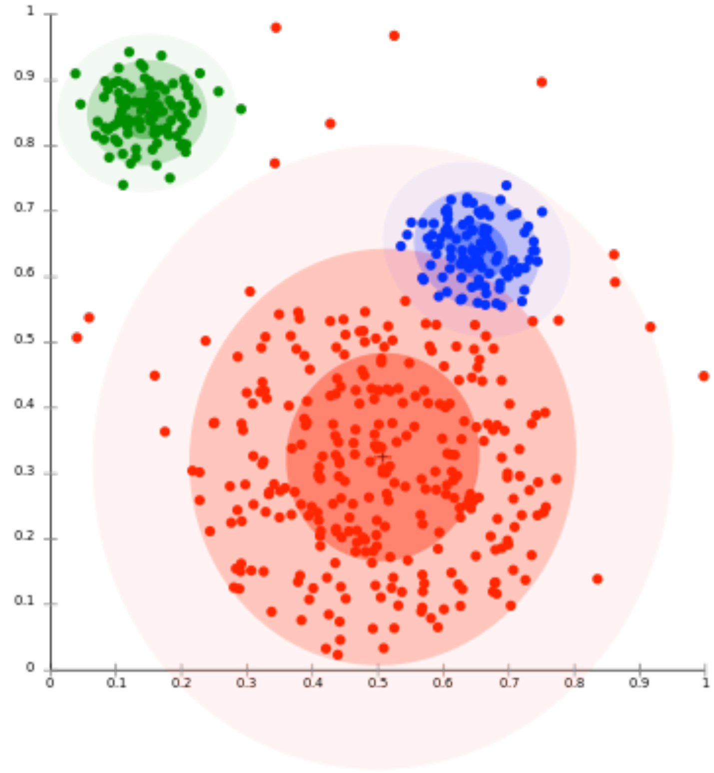
Example: Density Estimation

- Estimate probability density $p(x)$ from observations $\{x_1, \ldots, x_m\}$



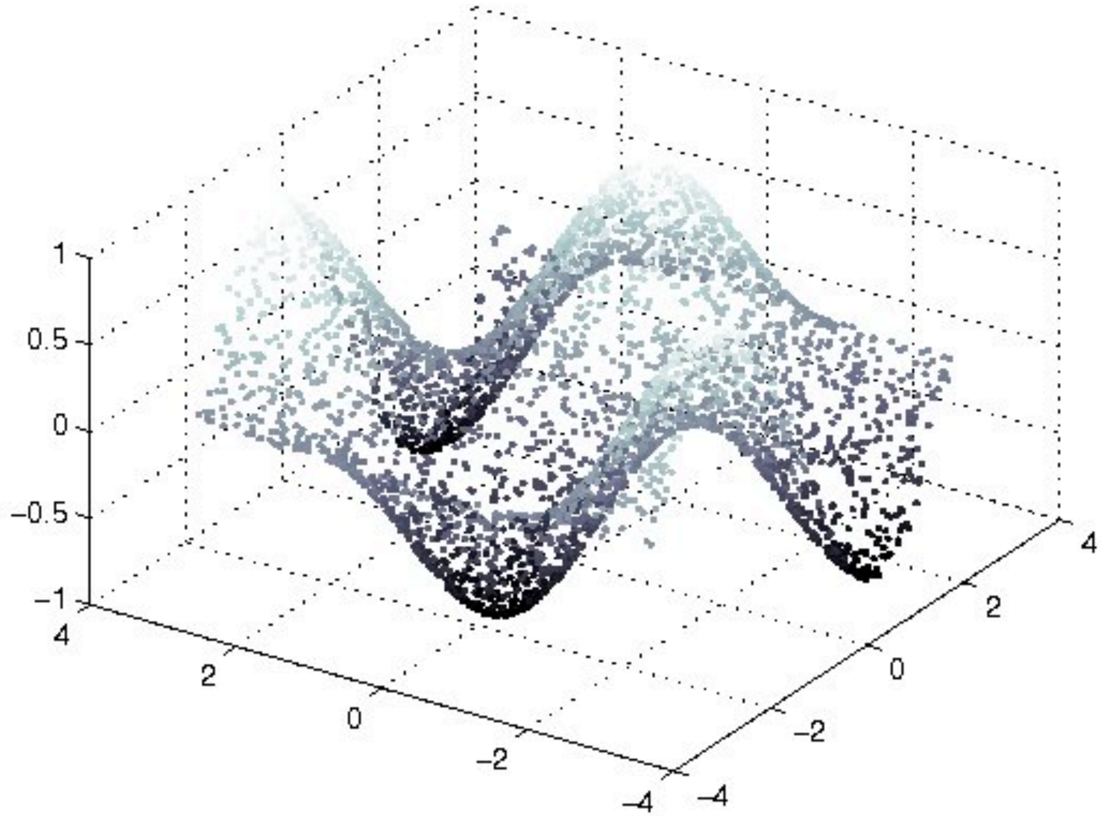
Example: Clustering

- Group data points based on similarity



Example: Representation Learning

- Data lies on a low-dimensional manifold



Linear Factor Model

- *h*: Explanatory factors / latent variables

$$h \sim p(h)$$

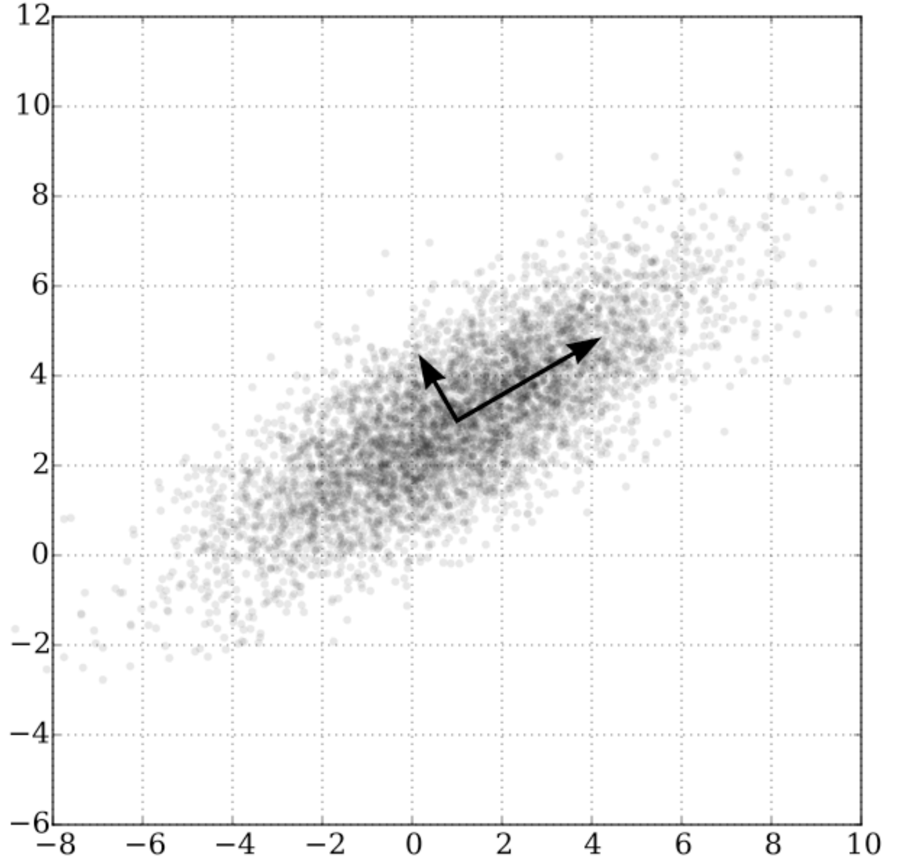
$$x \sim Wh + b + \varepsilon$$

- *Goal:* Infer *h* for a given *x*
  - Used as features for a learning task

Probabilistic Principal Component Analysis

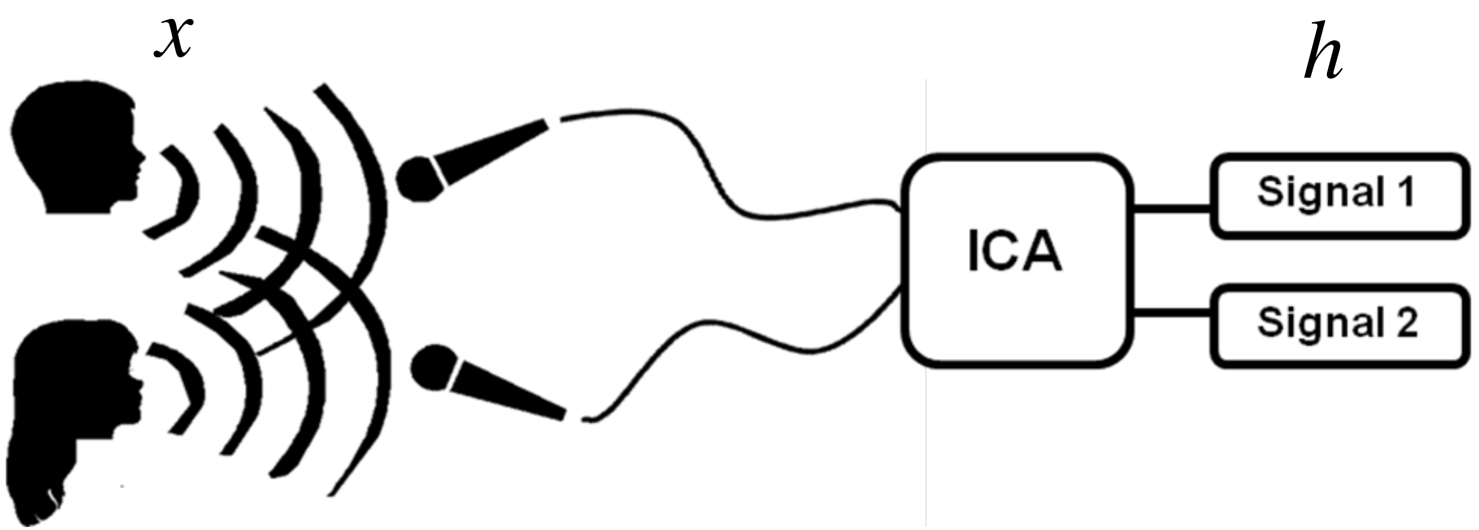
$$h \sim \mathcal{N}(0, I)$$

$$\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$$



Independent Component Analysis

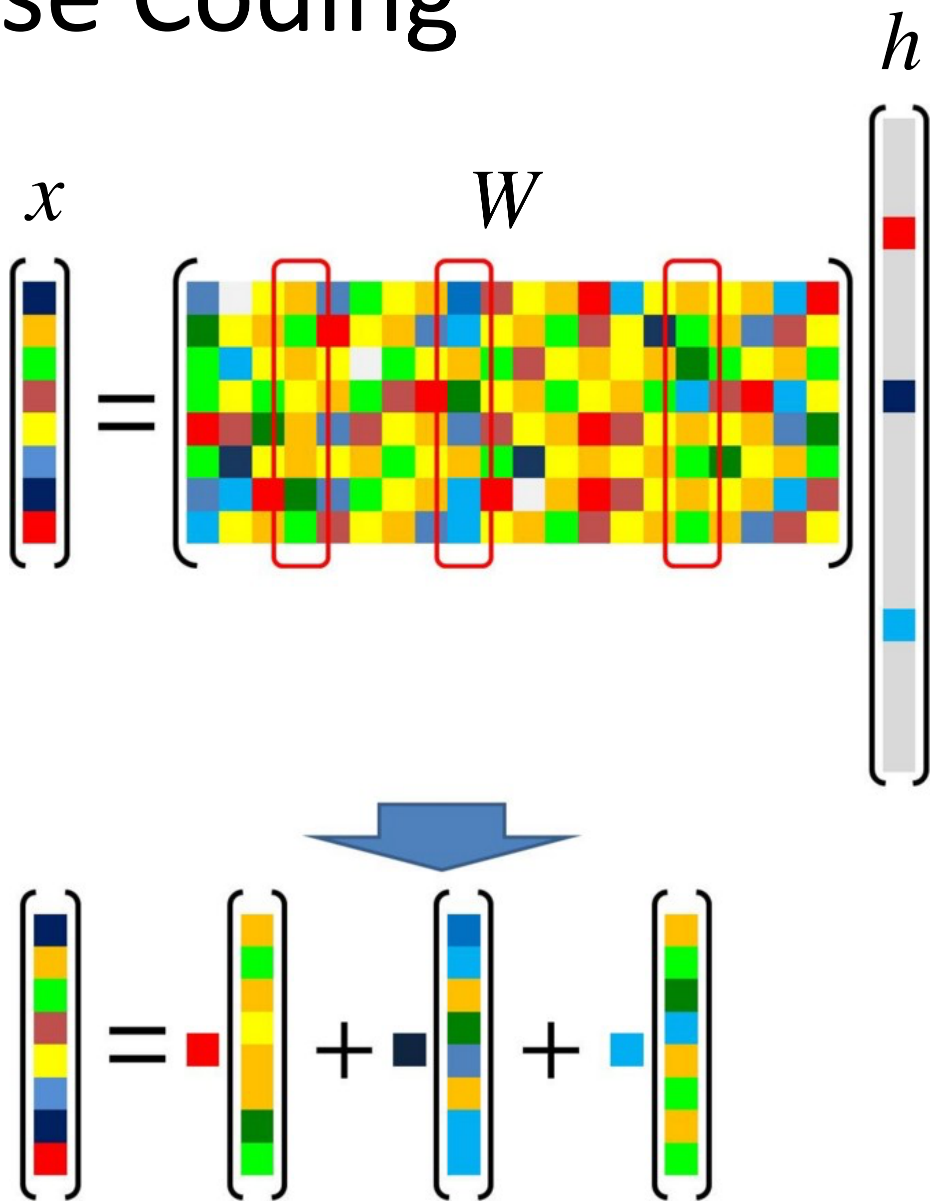
- *h* is drawn from a non-Gaussian distribution
- E.g. audio signal separation



Sparse Coding

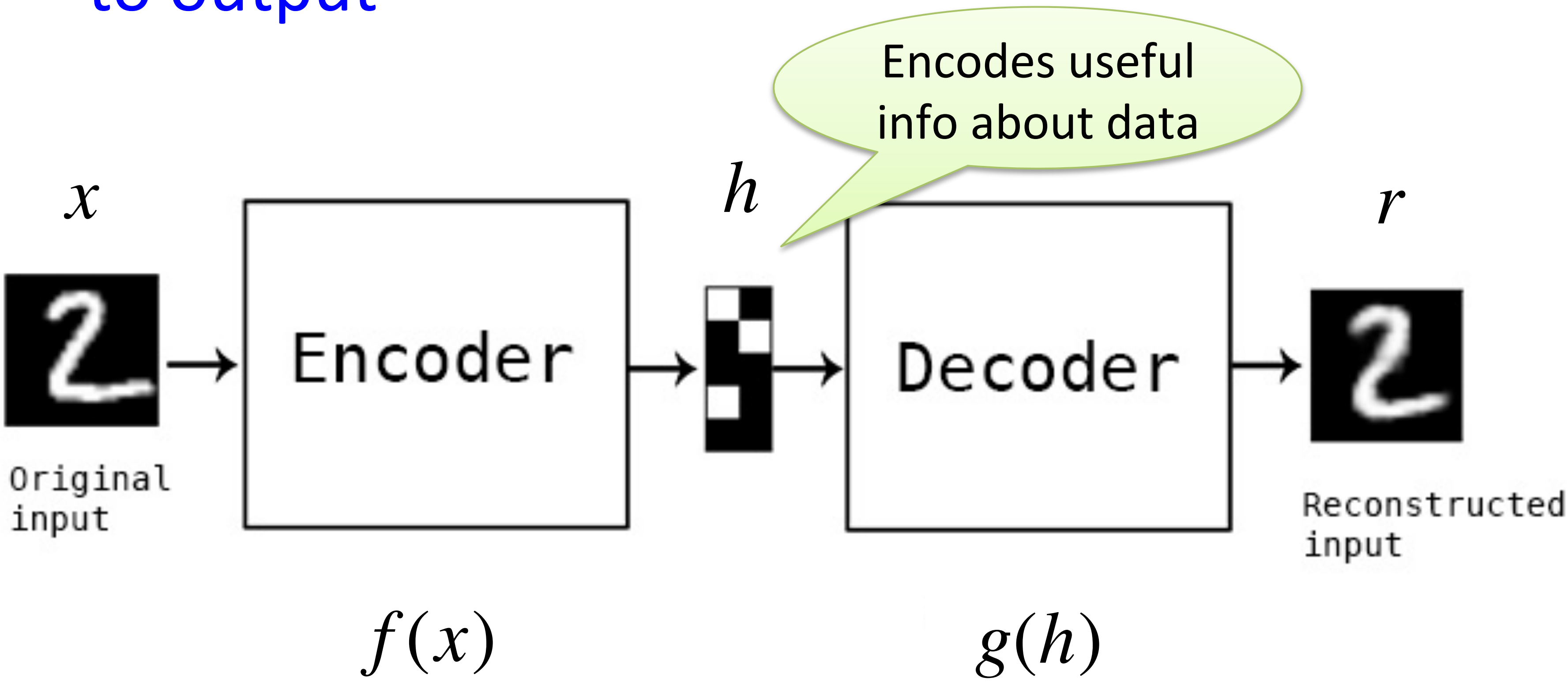
Sparse latent variables: e.g.

$$h_i \sim Laplace\left(0, \frac{1}{\lambda}\right)$$

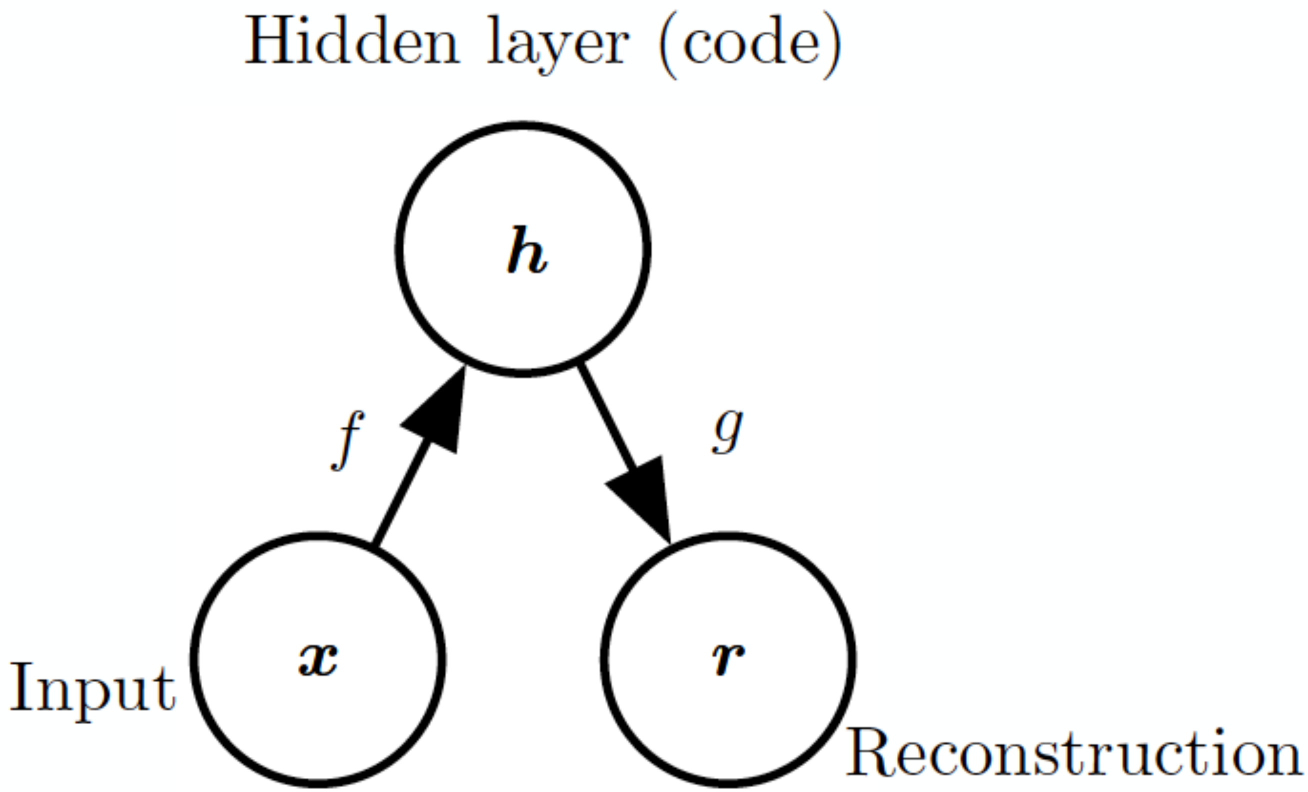


Beyond Linear: Autoencoders

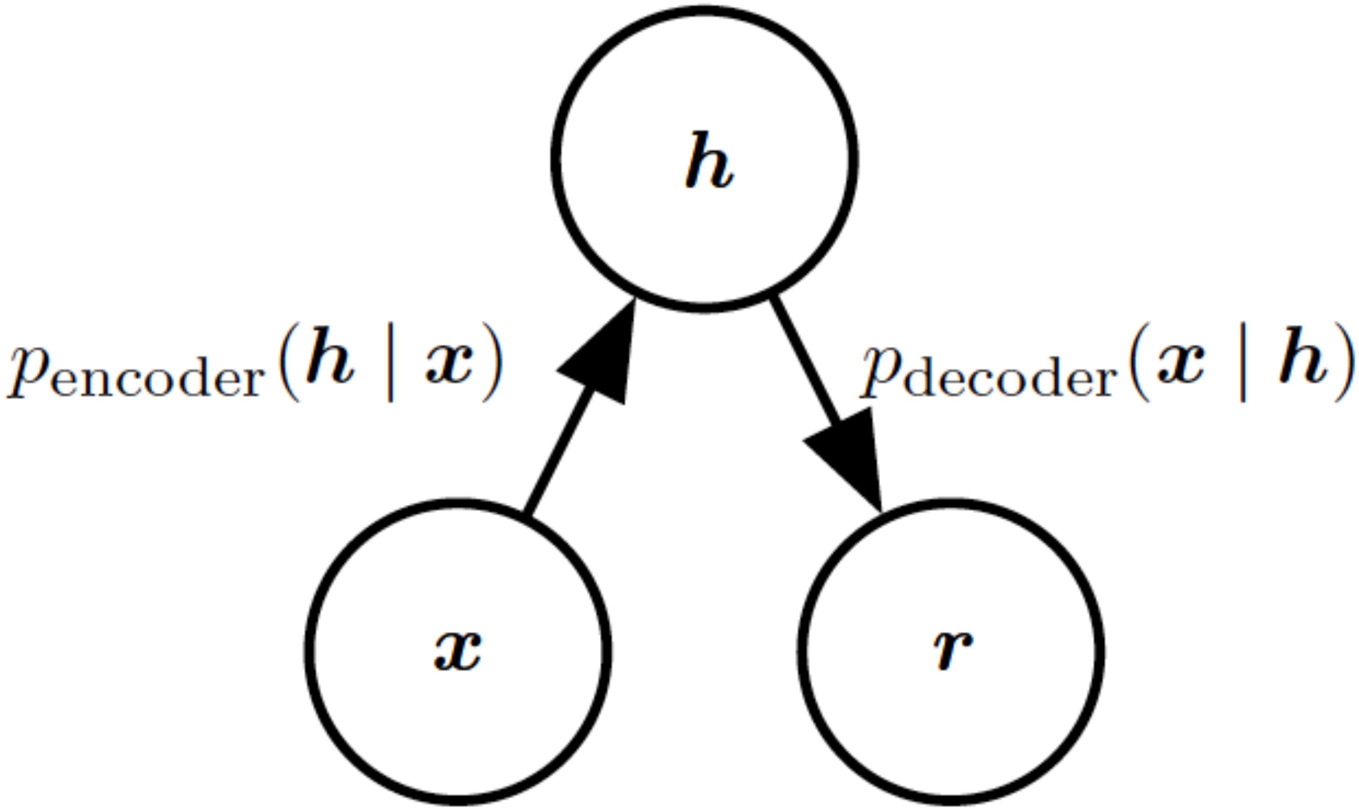
- Neural net that approximately copies its input to output



Structure of Autoencoders



Stochastic Autoencoders



Undercomplete Autoencoders

- *h* has lower dimension than *x*
- Must discard some information in *h*
- Learning involves minimizing loss:

$$L(x, g(f(x)))$$

- Equivalent to PCA when *f* is linear, *L* is MSE

Overcomplete Autoencoders

- h has greater dimension than x
- Autoencoder may simply copy input to output without learning anything useful
- Regularization to limit model capacity

Regularized Autoencoders

- Sparse autoencoders
- Denoising autoencoders
- Autoencoders with dropout on *h*
- Contractive autoencoders

Sparse Autoencoders

- Cost on *h* that penalizes code from being large

e.g. L_1 penalty $|h|$

$$L(x, g(f(x))) + \Omega(h)$$

$$h = f(x)$$

- Regularization on output of encoder, not on network parameters

Denoising Autoencoders

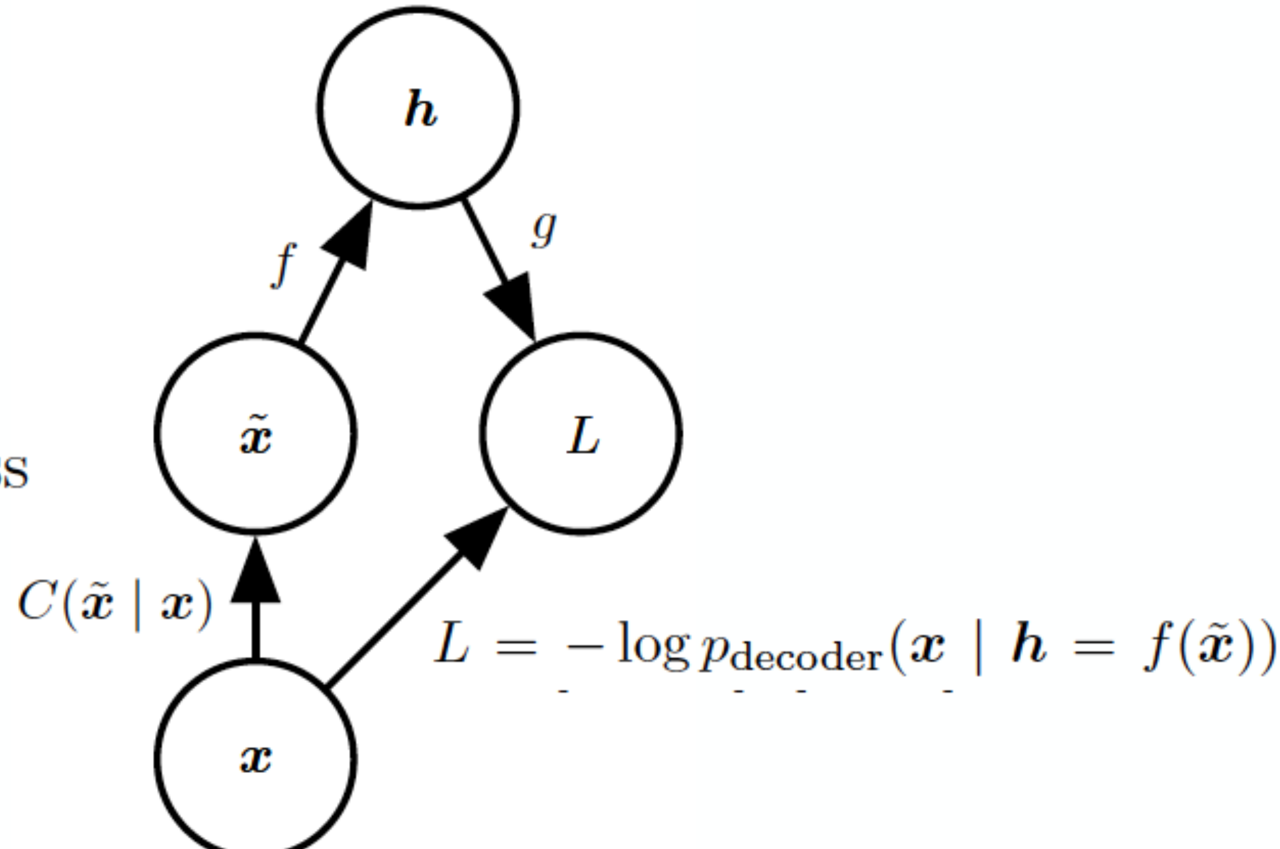
- Trained with corrupted data points, but to reconstruct original data points

$$L(x, g(f(\tilde{x})))$$

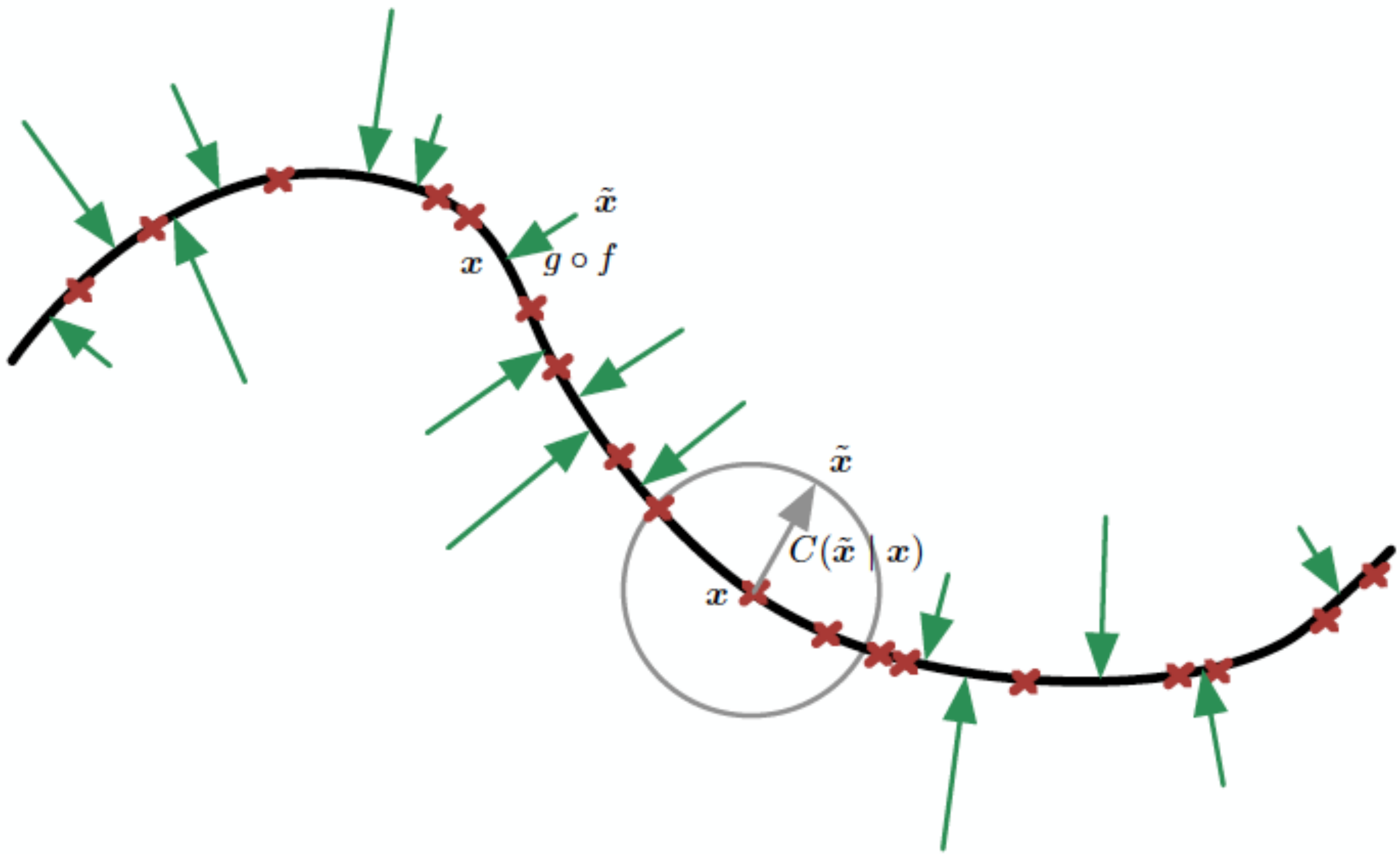
Corrupted copy of *x*

Denoising Autoencoders

C: corruption process
(introduce noise)

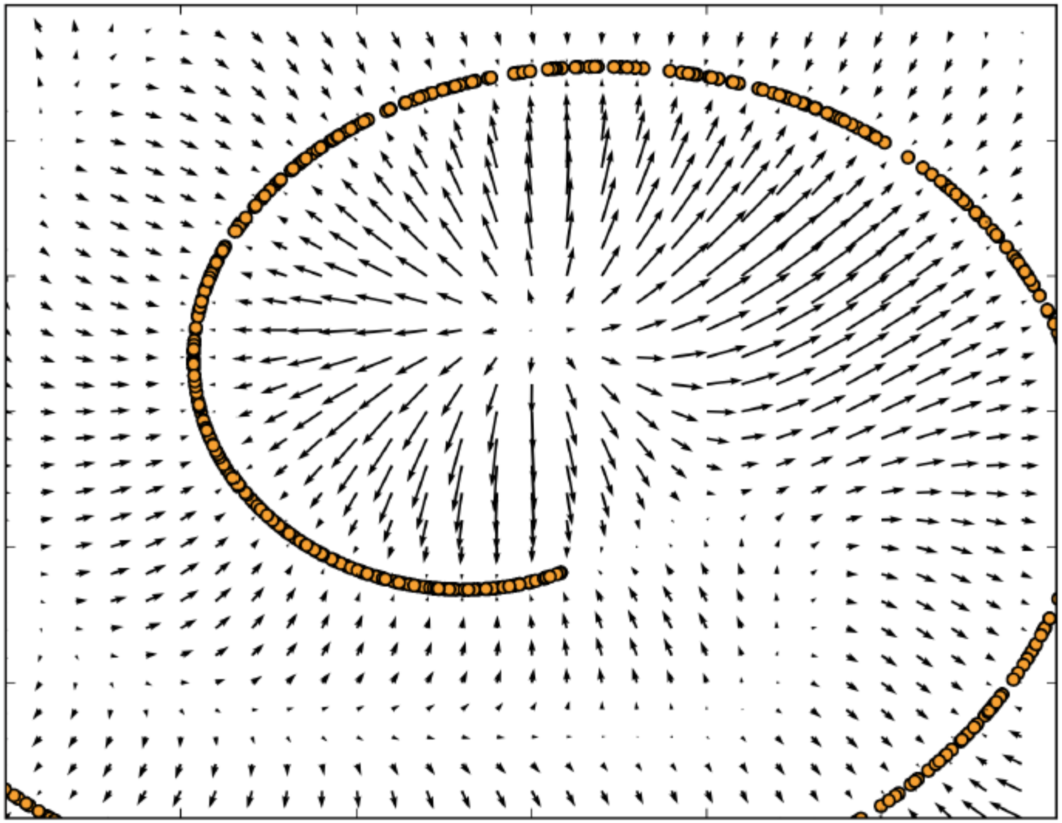


Denoising autoencoders learn a manifold

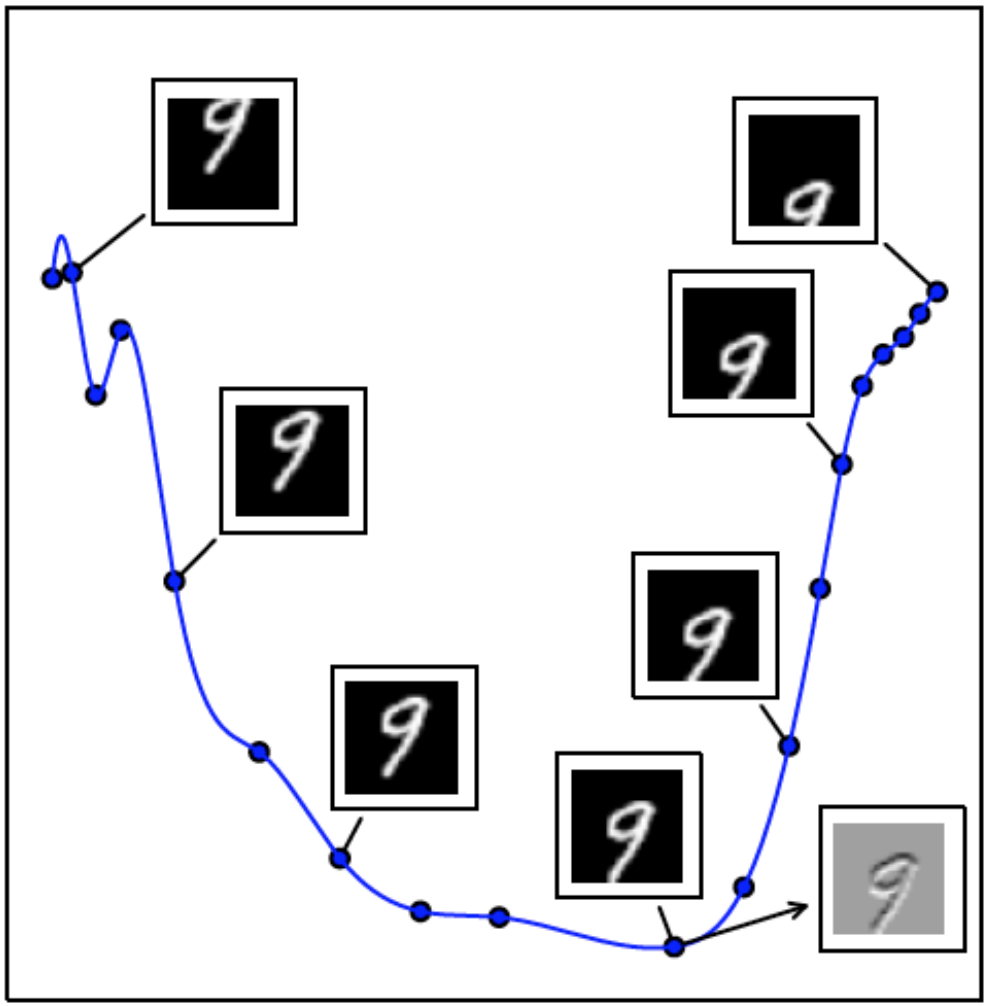


Vector field learned by denoising autoencoder

Each arrow is proportional to $g(f(x)) - x$



Tangent hyperplane of a manifold



Contractive Autoencoders

- Penalizes derivatives of f

$$L(x, g(f(x))) + \lambda \left\| \frac{\partial f(x)}{\partial x} \right\|_F^2$$

- Makes encoder resistant to small perturbations in input
- Identifies directions with most local variance

Contractive Autoencoders