Autoencoders

A: The autoencoder formalism

Unnamed structure

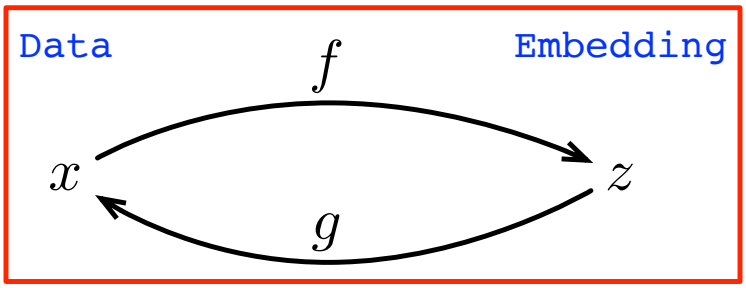
So far: representations that capture various types of structure:

- clusters
- linear subspaces
- manifolds
- explicit distance and similarity information

Can we find representations that capture:

- **multiple types of structure**, simultaneously
- **other structure for which we don't even have definitions**?

Deterministic autoencoders



- f: encoding map
- g: decoding map

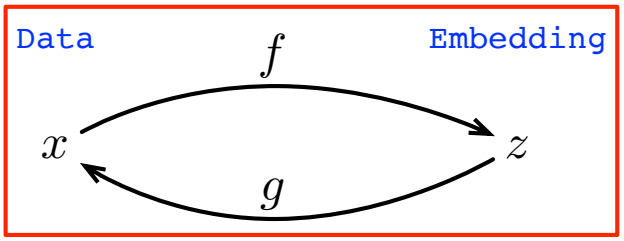
z should be *simpler* than x:

- e.g., lower-dimensional or sparser
- *expose* key structure in x

Typically: find f, g to minimize reconstruction loss $\mathbb{E}_X[\ell(X, g(f(X)))]$

PCA as an autoencoder

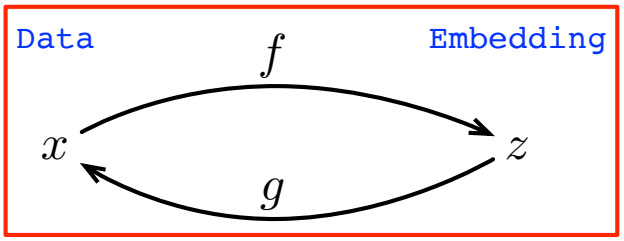
Use PCA to map data from $\mathbb{R}^d$ to $\mathbb{R}^k$.



- What are f and g?
- What loss function is being minimized?

K-means as an autoencoder

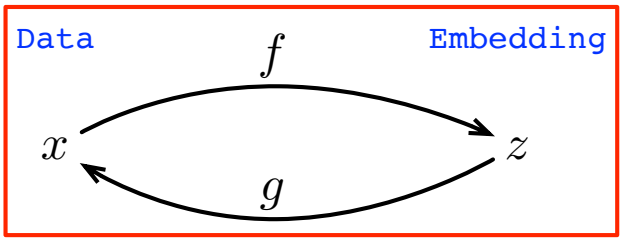
Apply k-means clustering to data in $\mathbb{R}^d$.



- What are f and g?
- What loss function is being minimized?

More general autoencoders

Allow f, g to be arbitrary functions, e.g. neural net with one hidden layer.



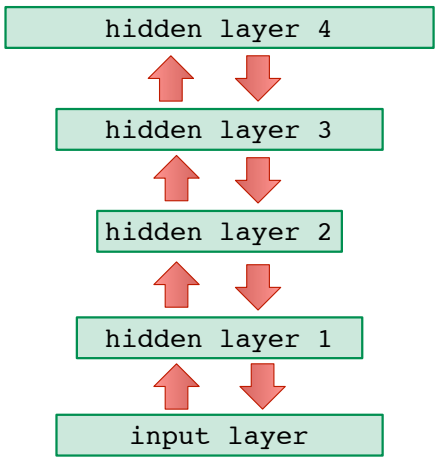
- f: encoding map
- g: decoding map

Common variant: **sparse autoencoder**. z is sparse, e.g.

- by explicitly keeping just the k highest activations
- by adding a sparsity-encouraging penalty term in the optimization

Stacked autoencoders

Successively higher-level representations



One way to fit these models:

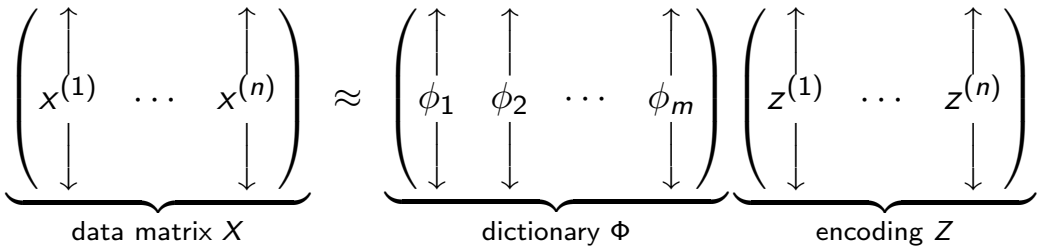
- Fit one layer at a time to the previous layer's activations
- Then fine-tune the whole structure to minimize reconstruction error

B: Special case: Dictionary learning

Linear decoding: dictionary learning

Given data points $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$, and an integer m:

- Choose m dictionary vectors $\phi_1, \ldots, \phi_m \in \mathbb{R}^d$.
- Approximate each $x^{(i)}$ by a linear combination of these dictionary elements.



- Principal component analysis: the ϕ_i are orthogonal and $m \leq d$
- Nonnegative matrix factorization: Φ and Z are non-negative
- K-means: $m = k$ and the encodings are in $\{e_1, \ldots, e_k\}$

More dictionary learning

Given data points $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$, and an integer m:

- Choose m dictionary vectors $\phi_1, \ldots, \phi_m \in \mathbb{R}^d$.
- Approximate each $x^{(i)}$ by a linear combination of these dictionary elements.

$$\underbrace{\begin{pmatrix} \uparrow & & \uparrow \\ x^{(1)} & \cdots & x^{(n)} \\ \downarrow & & \downarrow \end{pmatrix}}_{\text{data matrix } X} \approx \underbrace{\begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \phi_1 & \phi_2 & \cdots & \phi_m \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}}_{\text{dictionary } \Phi} \underbrace{\begin{pmatrix} \uparrow & & \uparrow \\ z^{(1)} & \cdots & z^{(n)} \\ \downarrow & & \downarrow \end{pmatrix}}_{\text{encoding } Z}$$

- Independent component analysis: the rows of Z are (approximately) statistically independent and $m \leq d$
- Sparse coding: columns of Z are sparse and often $m > d$ ("overcomplete basis")

Sparse coding

Given $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$, find dictionary vectors $\phi_1, \ldots, \phi_m$ and sparse representations $z^{(1)}, \ldots, z^{(n)} \in \mathbb{R}^m$ such that

$$x^{(i)} \approx \Phi z^{(i)}.$$

Optimization problem: find matrices Φ, Z that minimize

$$\begin{aligned} L(\Phi, Z) &= \|X - \Phi Z\|_F^2 - \lambda \cdot \text{sparsity}(Z) \\ &= \sum_{i=1}^{n} \left(\|x^{(i)} - \Phi z^{(i)}\|^2 - \lambda \cdot \text{sparsity}(z^{(i)}) \right) \end{aligned}$$

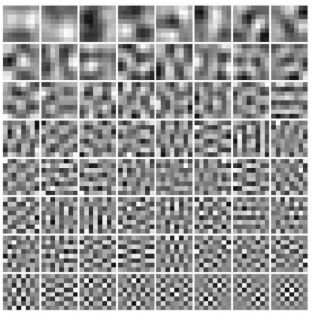
Alternating minimization procedure:

- Initialize Φ somehow
- Repeat until convergence:
  - Fixing Φ, minimize $L(\cdot)$ over Z
  - Fixing Z, minimize $L(\cdot)$ over Φ

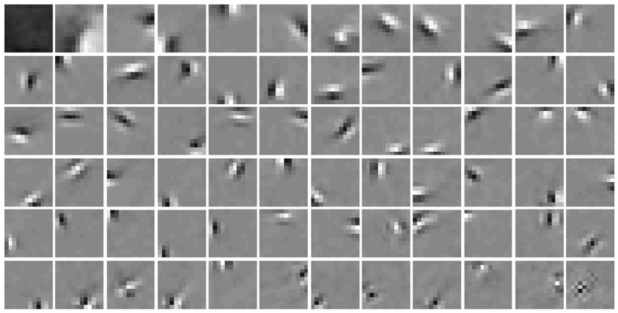
Example: image patches

Olshausen-Field (1996), Lewicki-Olshausen (1999): PCA versus sparse coding for natural image patches.

PCA



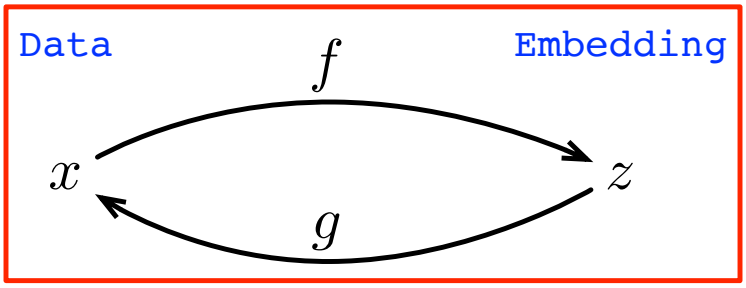
sparse coding



Sparse coding does a much better job at finding a basis that resembles the receptive fields of simple cells in visual cortex.

C: Variational autoencoder

Probabilistic autoencoders



- f: encoding map
- g: decoding map

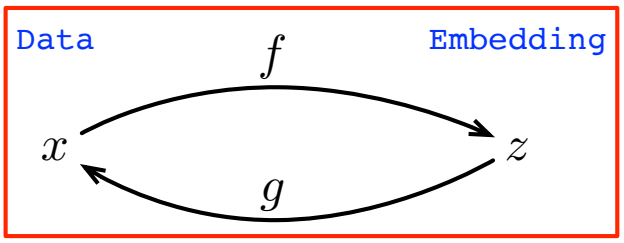
Now $g(z) \in \Theta$, where $\{p_\theta : \theta \in \theta\}$ is a family of probability distributions over x-space

Typically: find f, g to minimize $\mathbb{E}_X[-\ln p_{g(f(X))}(X)]$

In fact, f can also be a probabilistic map

Topic model as probabilistic autoencoder

A document is given by a count-vector $x \in \mathbb{N}^V$

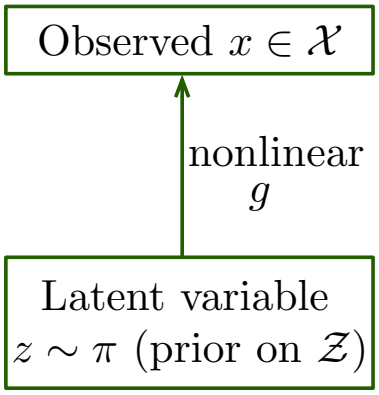


What are f and g?

Aside: PCA AND k-means

How can we view these as probabilistic autoencoders?

Nonlinear decoding



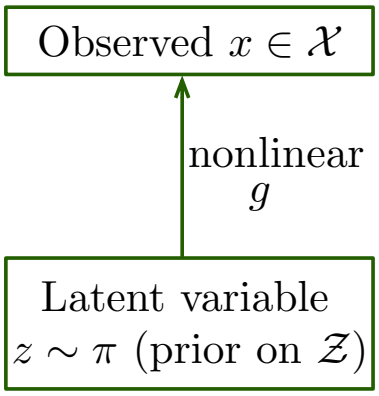
- Let $\{p_\theta : \theta \in \Theta\}$ be a family of distributions on $\mathcal{X}$
- Learn nonlinear map $g : \mathcal{Z} \to \Theta$ (e.g. neural net)
- $\Pr(x, z|g) = \pi(z) p_{g(z)}(x)$

Example:

- $\mathcal{Z} = \mathbb{R}^k$ and π is Gaussian $N(0, I_k)$
- $\mathcal{X} = \{0, 1\}^d$
- $\Theta = [0, 1]^d$, p_θ is product-Bernoulli:

$$p_\theta(x) = \prod_{i=1}^{d} \theta_i^{x_i} (1 - \theta_i)^{1 - x_i}$$

Approximate encoder



- $\{p_\theta : \theta \in \Theta\}$, family of distributions on $\mathcal{X}$
- Neural net $g : \mathcal{Z} \to \Theta$
- $\Pr(x, z|g) = \pi(z) p_{g(z)}(x)$

In order to learn g, need $\Pr(x|g)$: intractable!

$$\Pr(x|g) = \int_z \pi(z)\, p_{g(z)}(x)\, dz$$

Also intractable: $\Pr(z|x, g) = \Pr(x, z|g)/\Pr(x|g)$

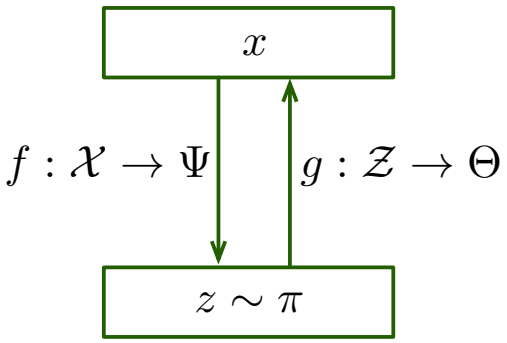
Learn another neural net to approximate $\Pr(z|x, g)$

- Let $\{q_\psi : \psi \in \Psi\}$ be a family of distributions on $\mathcal{Z}$
- Encoding map $f : \mathcal{X} \to \Psi$
- Want: $q_{f(x)}(z) \approx \Pr(z|x, g)$

E.g. for $\mathcal{Z} = \mathbb{R}^k$, could take $\Psi = \mathbb{R}^{2k}$ with

- $\psi = (\mu_1, \ldots, \mu_k, \ln \sigma_1, \ldots, \ln \sigma_k)$
- $q_\psi = N(\mu, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_k^2))$

The loss function



- $\{p_\theta : \theta \in \Theta\}$, distributions on $\mathcal{X}$
- $\{q_\psi : \psi \in \Psi\}$, distributions on $\mathcal{Z}$
- $\Pr(x, z|g) = \pi(z) p_{g(z)}(x)$
- Want $q_{f(x)}(z) \approx \Pr(z|x, g)$

Should choose g to maximize

$$\Pr(x|g), \text{ which equals } \frac{\Pr(x, z|g)}{\Pr(z|x, g)}$$

Instead choose g, f to maximize

$$L_{f,g}(x) = \mathbb{E}_{z \sim q_{f(x)}} \ln \frac{\Pr(x, z|g)}{q_{f(x)}(z)}$$

This is called the ELBO.

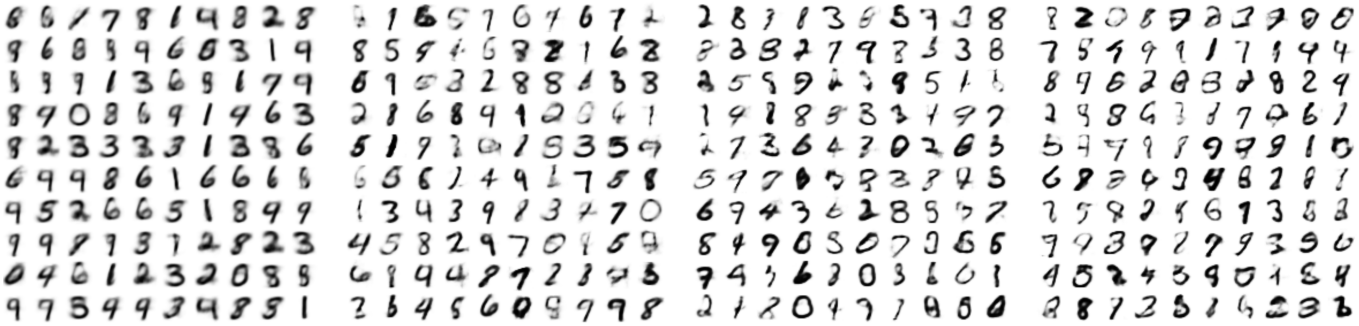
Given data $x_1, \ldots, x_n \in \mathcal{X}$, solve:

$$\arg\max_{f,g} \sum_{i=1}^{n} L_{f,g}(x_i)$$

Property of the ELBO

Claim: $L_{f,g}(x) = \ln \Pr(x|g) - K(q_{f(x)}, \Pr(\cdot|x, g))$

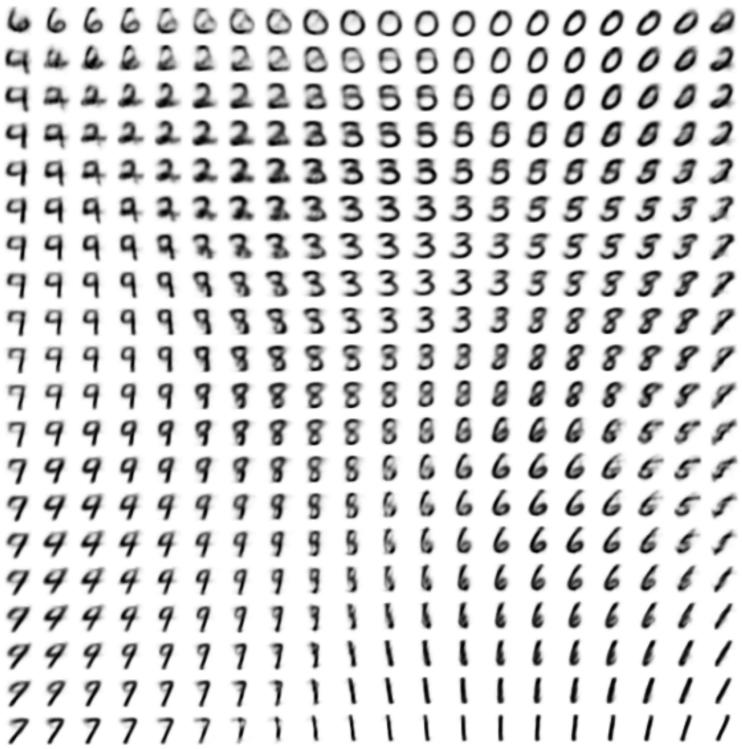
VAE example: MNIST



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Kingma and Welling (2014). Autoencoding variational Bayes.

VAE example: MNIST



Kingma and Welling (2014). Autoencoding variational Bayes.