Generative Models. VAE

Evgeny Burnaev

Skoltech, Moscow, Russia

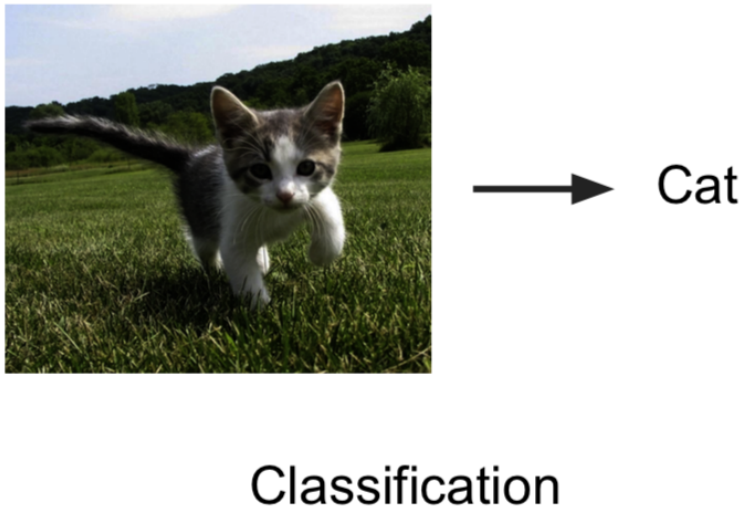
Skoltech

Skolkovo Institute of Science and Technology

Outline

Supervised vs Unsupervised Learning

- **Supervised Learning**
- **Data**: $(\mathbf{x}, y)$, $\mathbf{x}$ is a feature vector, y is a label
- **Goal**: Learn a function to map $\mathbf{x} \rightarrow y$
- **Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.



Supervised vs Unsupervised Learning

- **Supervised Learning**
- **Data**: $(\mathbf{x}, y)$, $\mathbf{x}$ is a feature vector, y is a label
- **Goal**: Learn a function to map $\mathbf{x} \rightarrow y$
- **Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

DOG, DOG, CAT

Object Detection

Supervised vs Unsupervised Learning

- **Supervised Learning**
- **Data**: $(\mathbf{x}, y)$, $\mathbf{x}$ is a feature vector, y is a label
- **Goal**: Learn a function to map $\mathbf{x} \to y$
- **Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

Semantic Segmentation

Supervised vs Unsupervised Learning

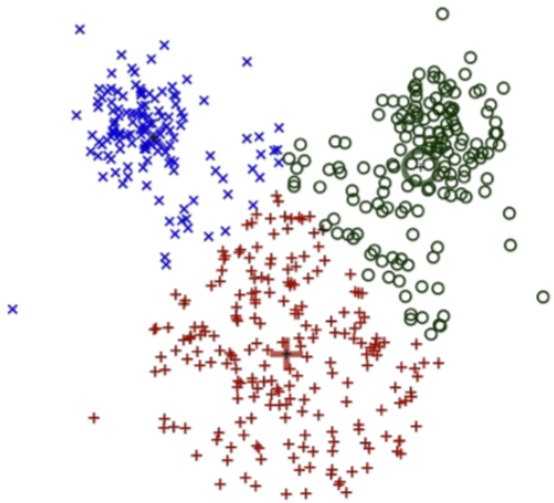
- **Supervised Learning**
- **Data**: $(\mathbf{x}, y)$, $\mathbf{x}$ is a feature vector, y is a label
- **Goal**: Learn a function to map $\mathbf{x} \rightarrow y$
- **Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

A cat sitting on a suitcase on the floor

Image captioning

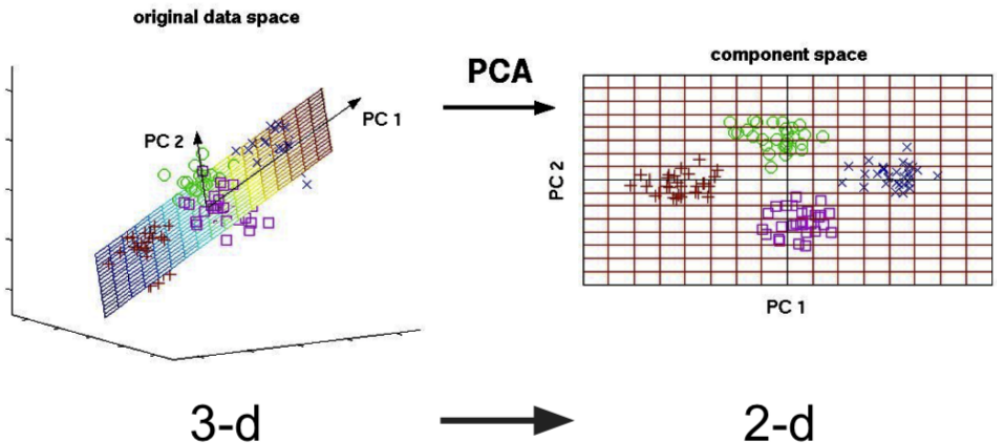
Supervised vs Unsupervised Learning

- **Unsupervised Learning**
- **Data**: $\mathbf{x}$, $\mathbf{x}$ is a feature vector
- **Goal**: Learn some underlying hidden structure of the data
- **Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.



K-means clustering

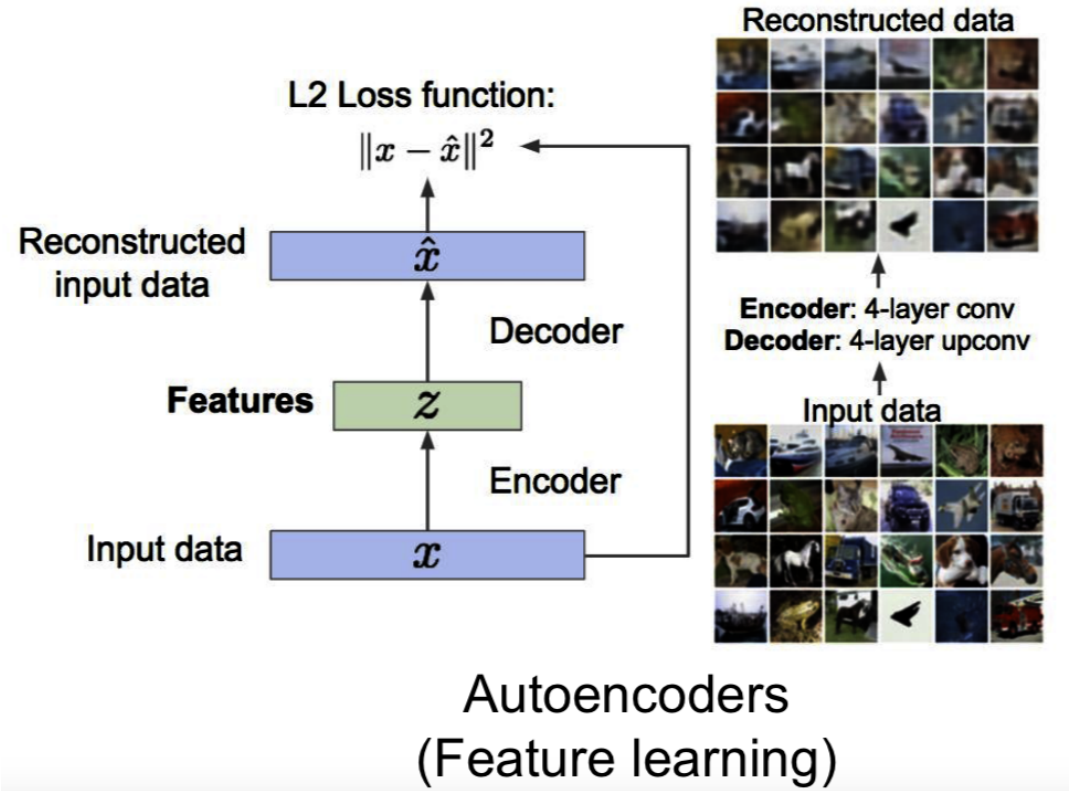
- **Unsupervised Learning**
- **Data**: $\mathbf{x}$, $\mathbf{x}$ is a feature vector
- **Goal**: Learn some underlying hidden structure of the data
- **Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.



Principal Component Analysis
(Dimensionality reduction)

Supervised vs Unsupervised Learning

- **Unsupervised Learning**
- **Data**: $\mathbf{x}$, $\mathbf{x}$ is a feature vector
- **Goal**: Learn some underlying hidden structure of the data
- **Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.

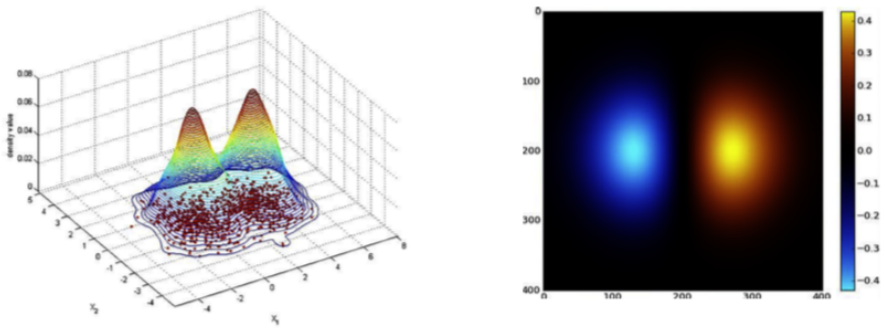


Supervised vs Unsupervised Learning

- **Unsupervised Learning**
- **Data**: $\mathbf{x}$, $\mathbf{x}$ is a feature vector
- **Goal**: Learn some underlying hidden structure of the data
- **Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Figure copyright Ian Goodfellow, 2016. Reproduced with permission.

1-d density estimation



2-d density estimation

Supervised vs Unsupervised Learning

- **Supervised Learning**
- **Data**: $(\mathbf{x}, y)$, $\mathbf{x}$ is a feature vector, y is a label
- **Goal**: Learn a function to map $\mathbf{x} \to y$
- **Examples**: Classification, regression, object detection, semantic segmentation, image captioning, etc.

- **Unsupervised Learning**
- **Data**: $\mathbf{x}$, $\mathbf{x}$ is a feature vector
- **Goal**: Learn some underlying hidden structure of the data
- **Examples**: Clustering, dimensionality reduction, feature learning, density estimation, etc.

Feature training data is cheap!

Holy grail: solve unsupervised learning $\Rightarrow$ understand structure of visual world

Generative Models

Given training data, generate new samples from same distribution

Training data $\sim p_{data}(\mathbf{x})$

Generated data $\sim p_{model}(\mathbf{x})$

Want to learn $p_{model}(\mathbf{x})$ similar to $p_{data}(\mathbf{x})$

Addresses density estimation, a core problem in unsupervised learning.
Issues:

- Explicit density estimation: explicitly define and solve for $p_{model}(\mathbf{x})$
- Implicit density estimation: learn model that can sample from $p_{model}(\mathbf{x})$ w/o explicitly defining it

Why Generative Models?

- Realistic samples for artwork, super-resolution, colorization, etc.

- Generative models of time-series data can be used for simulation and planning (reinforcement learning applications!)
- Training generative models can also enable inference of latent representations that can be useful as general features

Taxonomy of Generative Models

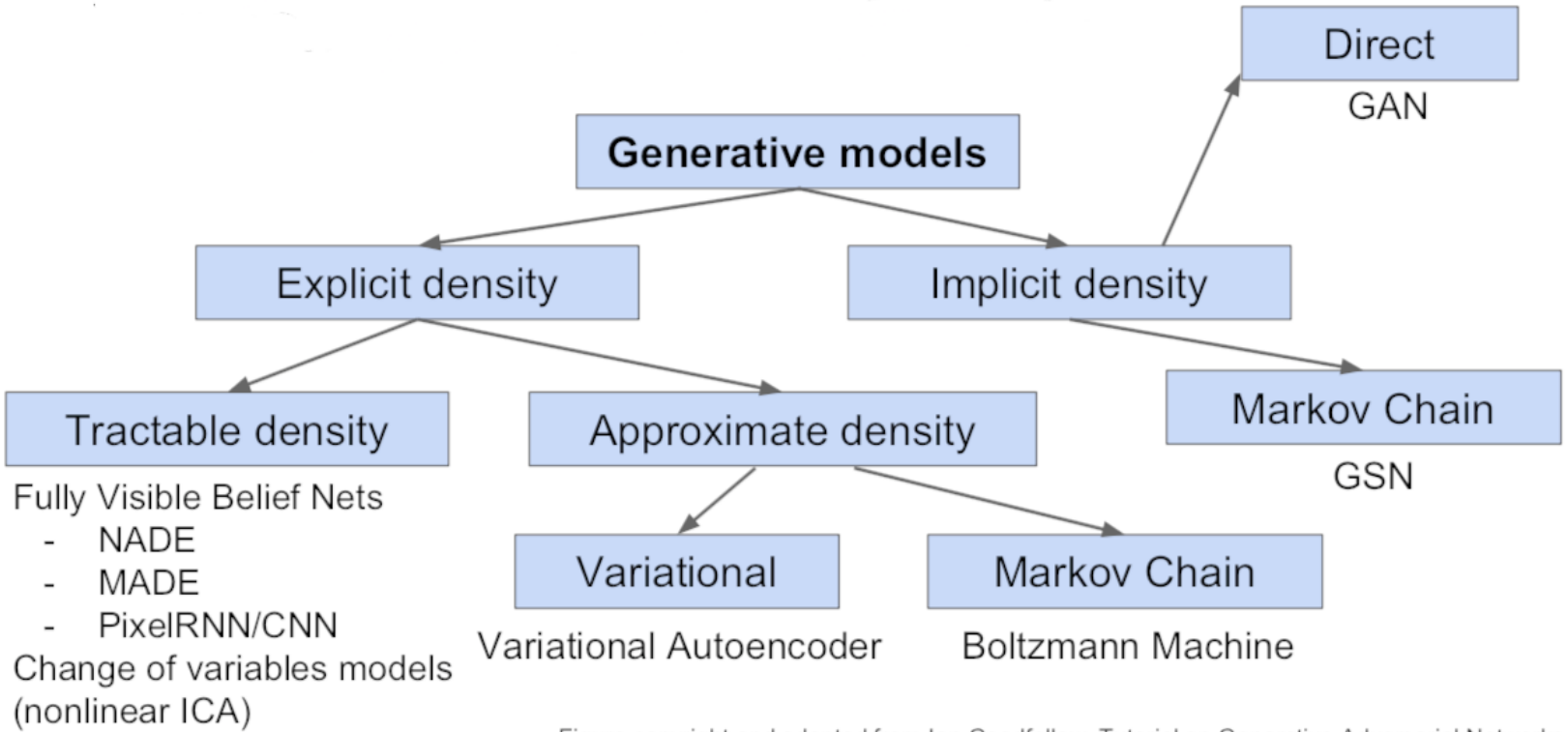
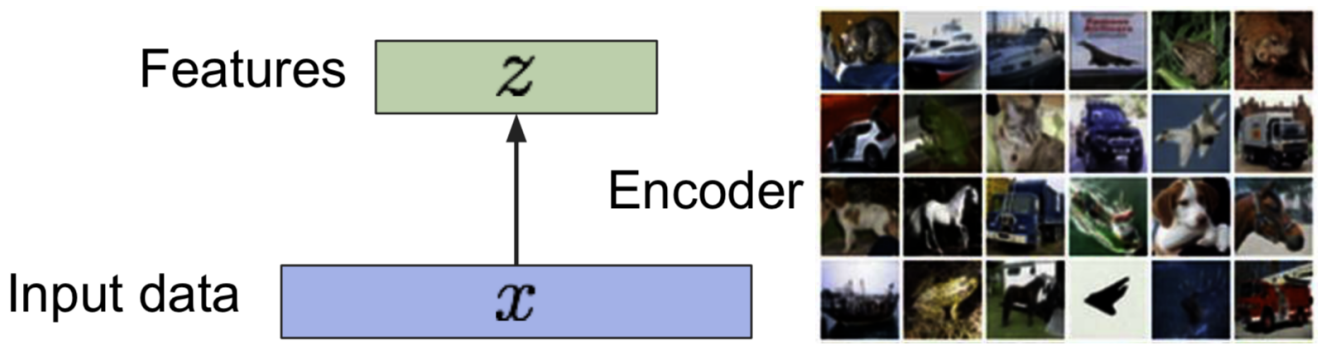
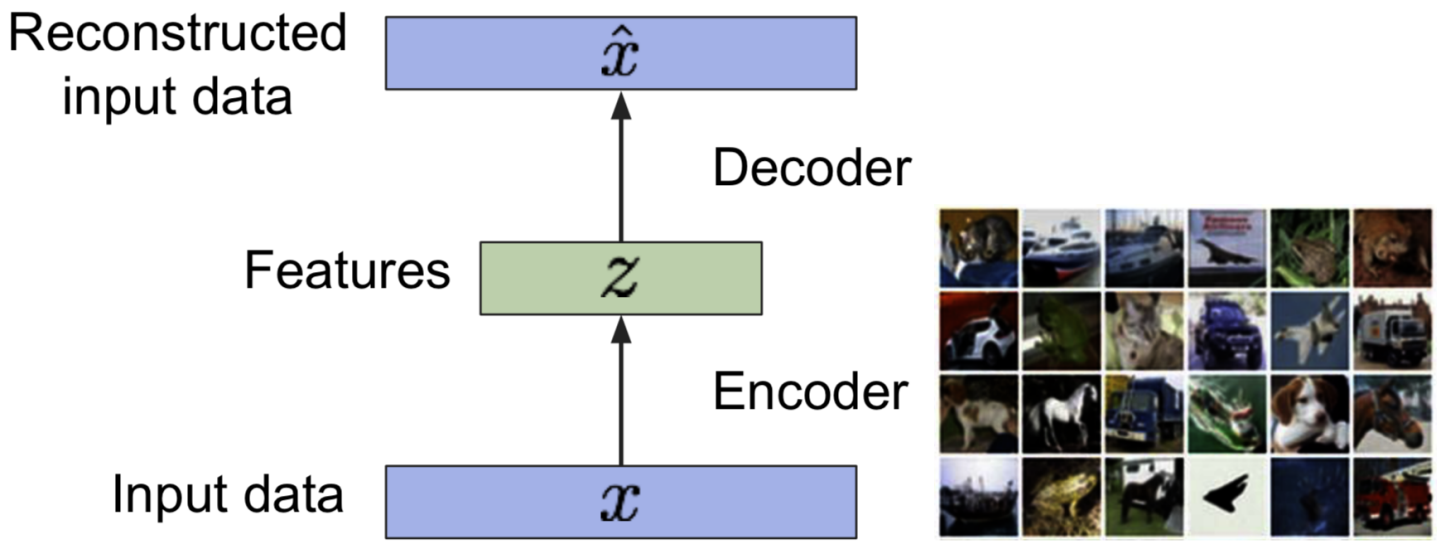


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data
- **Encoder**: Linear + nonlinearity (sigmoid) $\rightarrow$ Deep fully-connected $\rightarrow$ ReLU CNN
- $\mathbf{z}$ usually smaller than $\mathbf{x}$ (dimensionality reduction)
- Q: Why dimensionality reduction?
- A: Want features to capture meaningful factors of variation in data

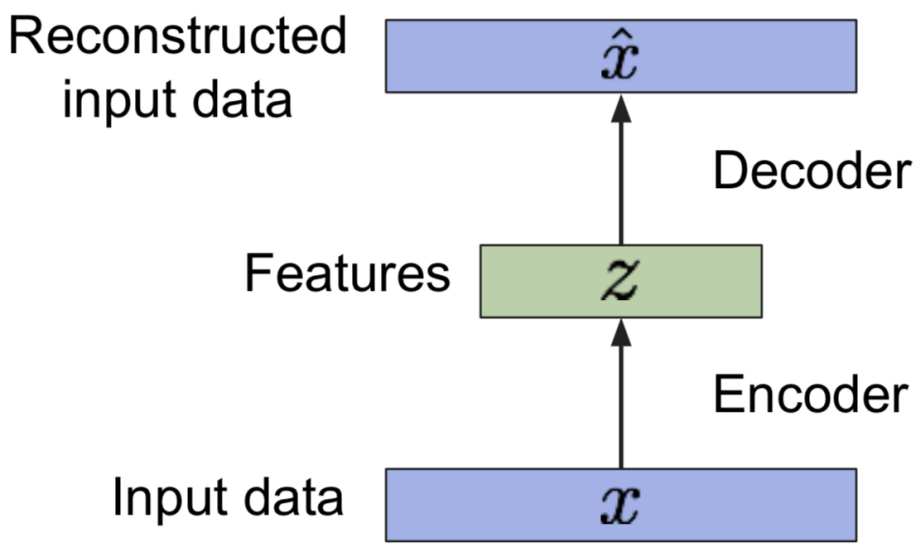


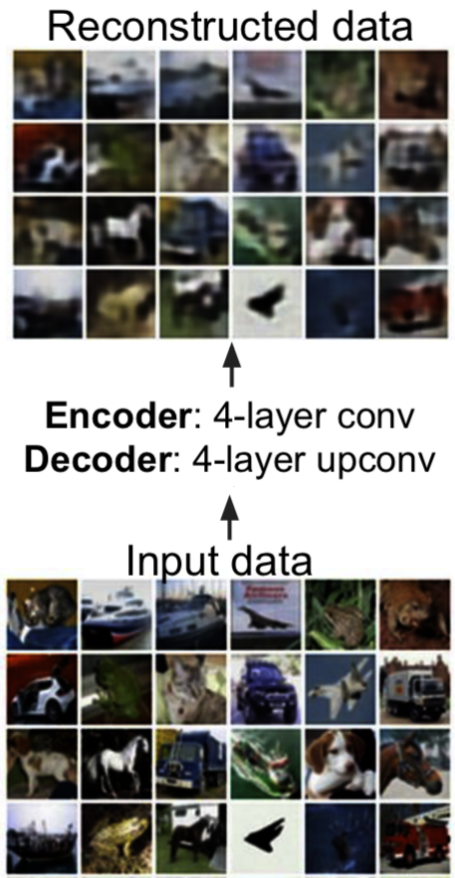
- How to learn this feature representation?
- Train such that features can be used to reconstruct original data
  "Autoencoding" – encoding itself
- **Decoder**: Linear + nonlinearity (sigmoid) $\rightarrow$ Deep fully-connected $\rightarrow$ ReLU CNN



Autoencoders

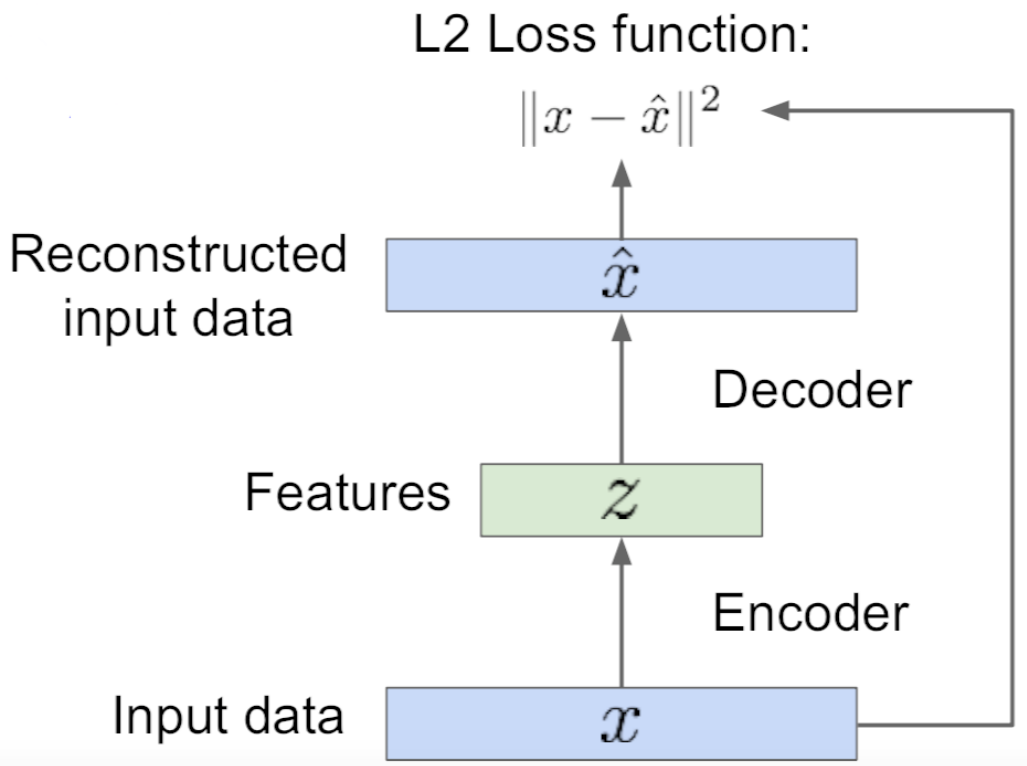
- How to learn this feature representation?
- Train such that features can be used to reconstruct original data
  "Autoencoding" – encoding itself
- **Decoder**: Linear + nonlinearity (sigmoid) $\rightarrow$ Deep fully-connected $\rightarrow$ ReLU CNN

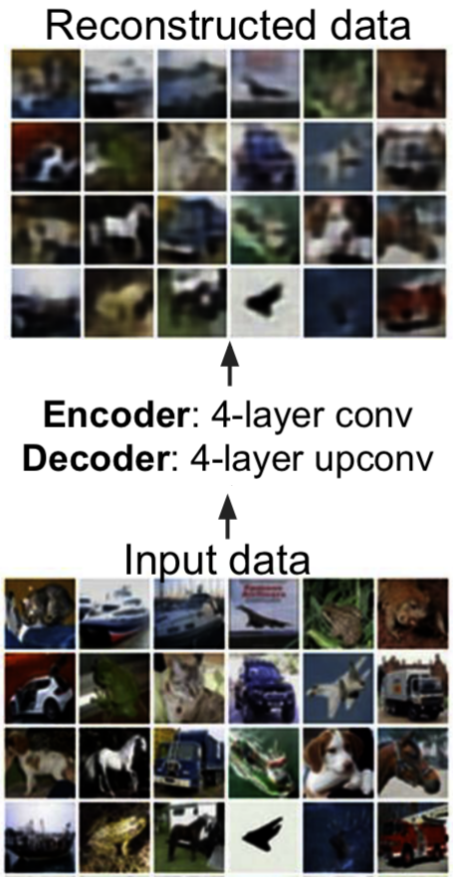




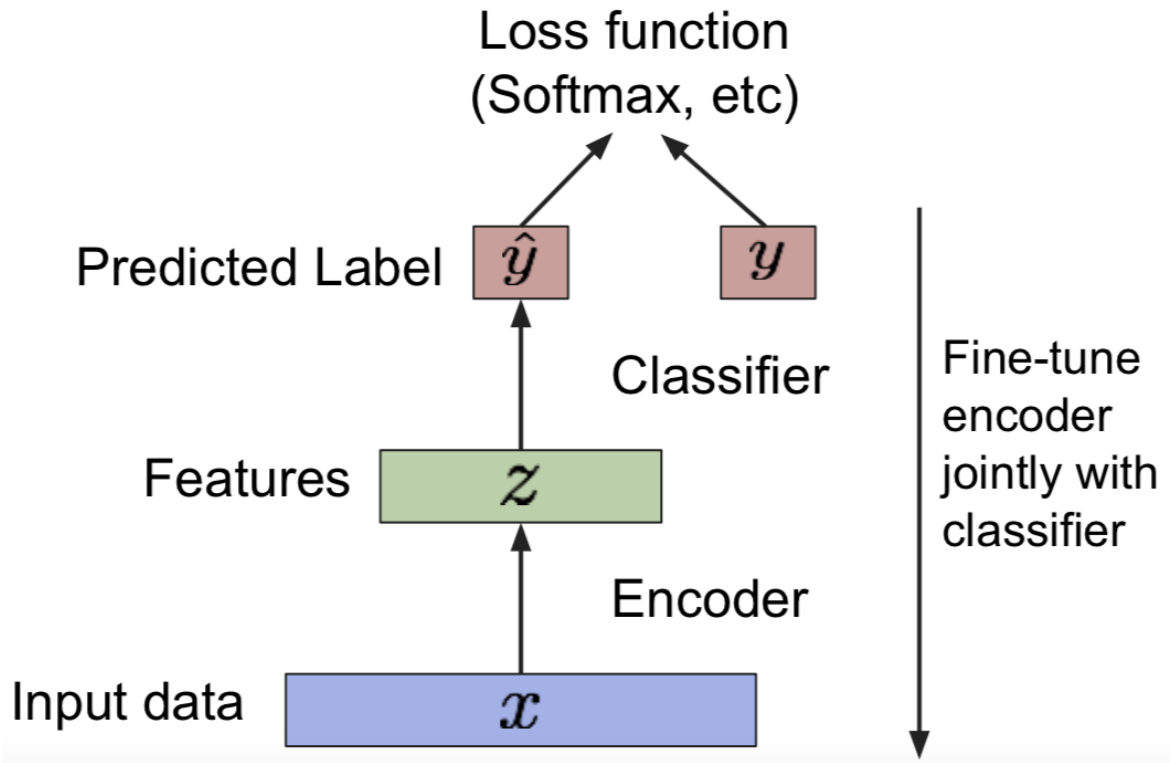
Autoencoders

- Train such that features can be used to reconstruct original data
- Doesn't use labels!
- After training we can throw away decoder and ...





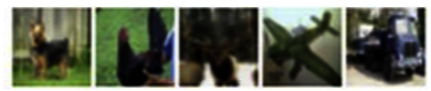
- Encoder can be used to initialize a supervised model



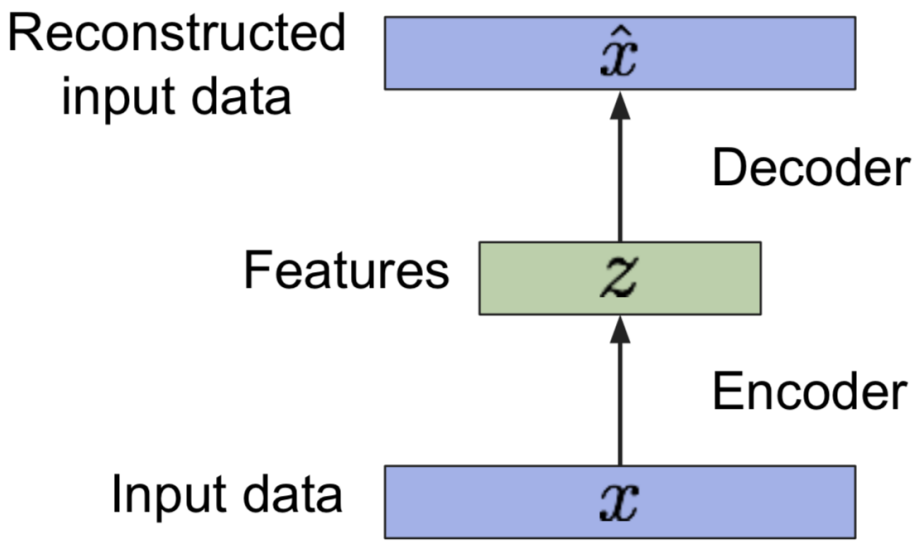
bird plane

dog deer truck

Train for final task
(sometimes with
small data)

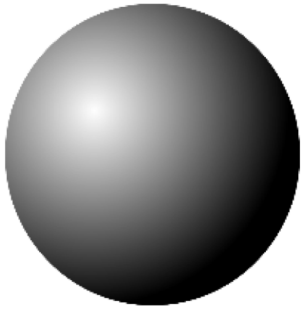


- Autoencoders can reconstruct data, and can learn features to initialize a supervised model
- Features capture factors of variation in training data. Can we generate new images from an autoencoder?



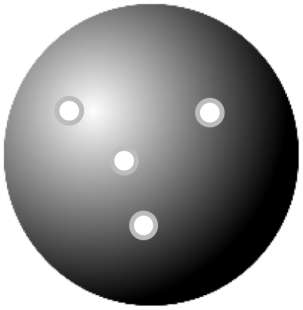
Probabilistic model for Data on manifolds

$$z \sim p(z)$$



Probabilistic model for Data on manifolds

$$z \sim p(z)$$



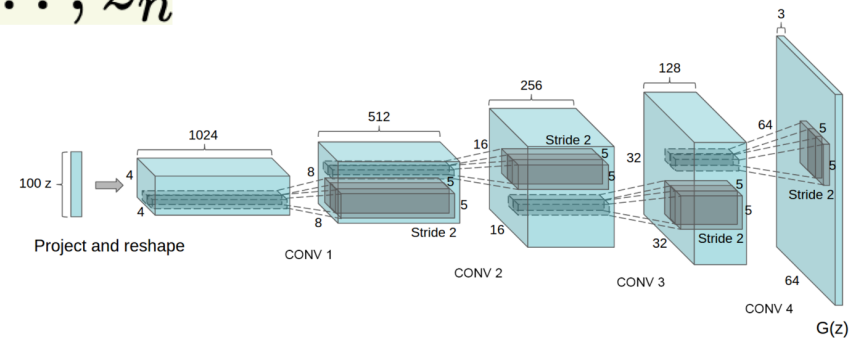
$$z_1, z_2, \ldots, z_n$$

Probabilistic model for Data on manifolds

$$z \sim p(z)$$

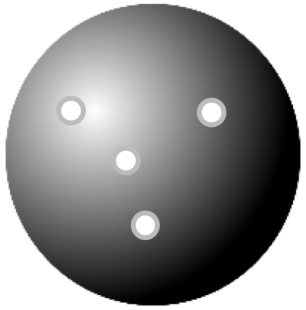
$$g_\theta$$

$$z_1, z_2, \ldots, z_n$$



Probabilistic model for Data on manifolds

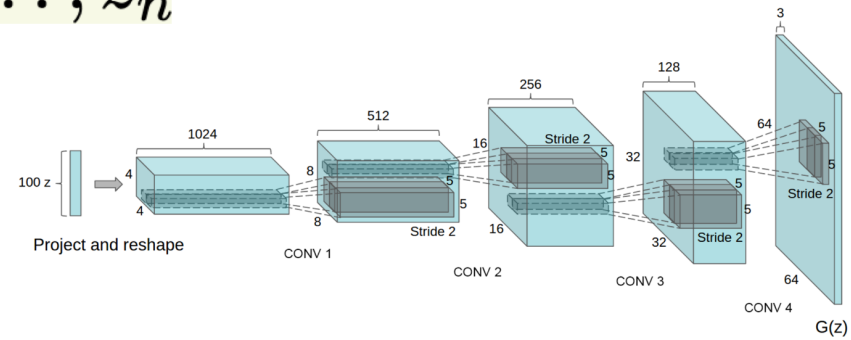
$$z \sim p(z)$$



$$g_\theta$$

$$z_1, z_2, \ldots, z_n$$

$$x_1, x_2, \ldots, x_n$$



Probabilistic model for Data on manifolds

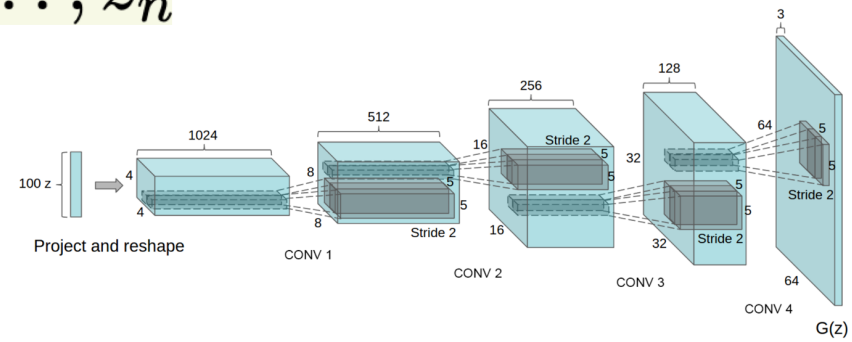
$$z \sim p(z)$$

$$x \sim p(x|g_\theta(z)) \cdot p(z)$$

g_θ

$z_1, z_2, \ldots, z_n$

$x_1, x_2, \ldots, x_n$



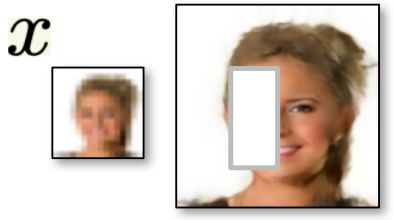
Why do I really need a latent model?

Answer: image restoration/editing/enhancement

x

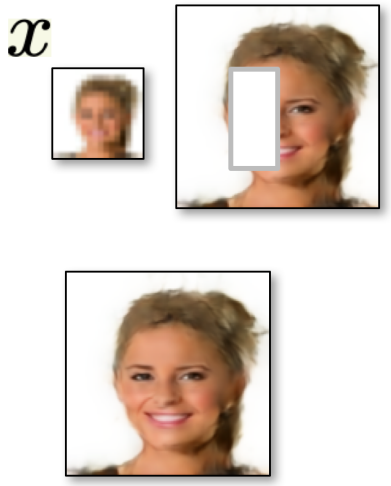
Why do I really need a latent model?

Answer: image restoration/editing/enhancement



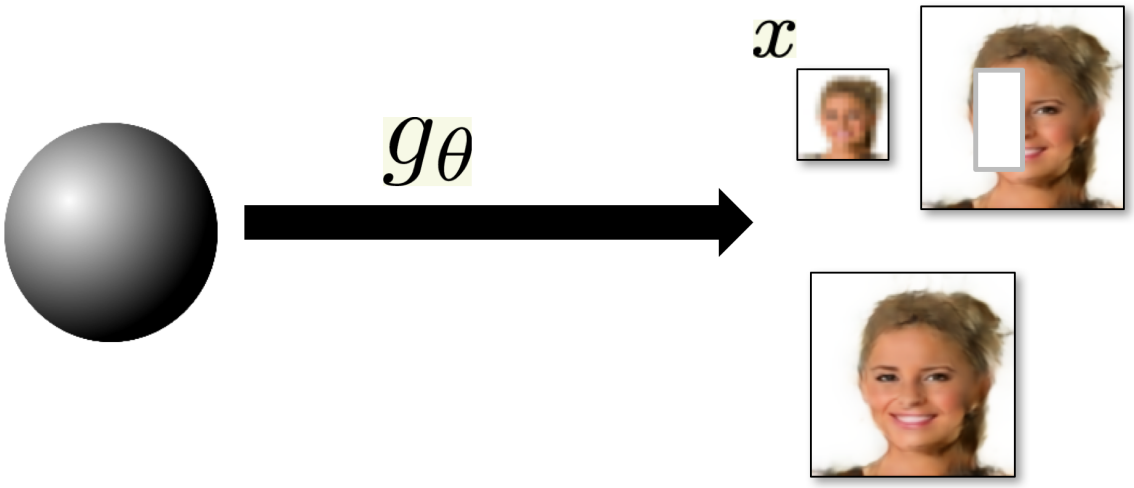
Why do I really need a latent model?

Answer: image restoration/editing/enhancement



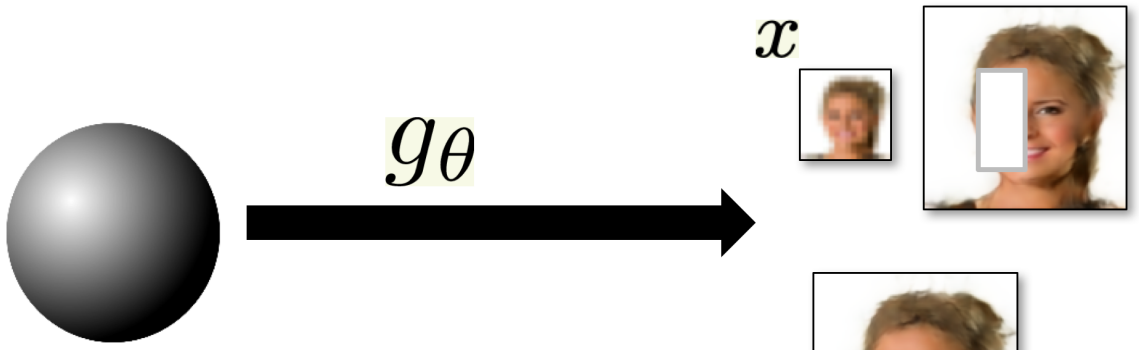
Why do I really need a latent model?

Answer: image restoration/editing/enhancement



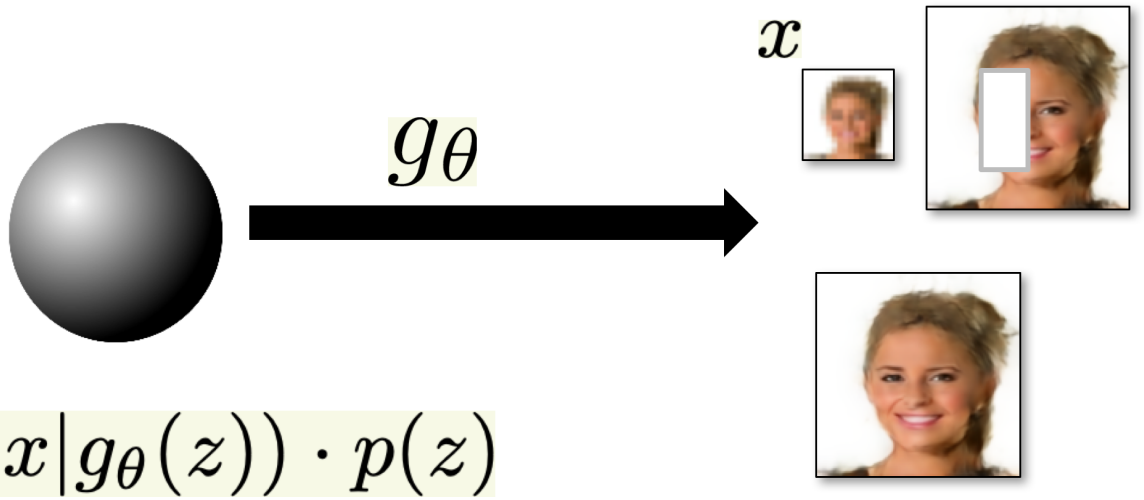
Why do I really need a latent model?

Answer: image restoration/editing/enhancement



$$x \sim p(x|g_\theta(z)) \cdot p(z)$$

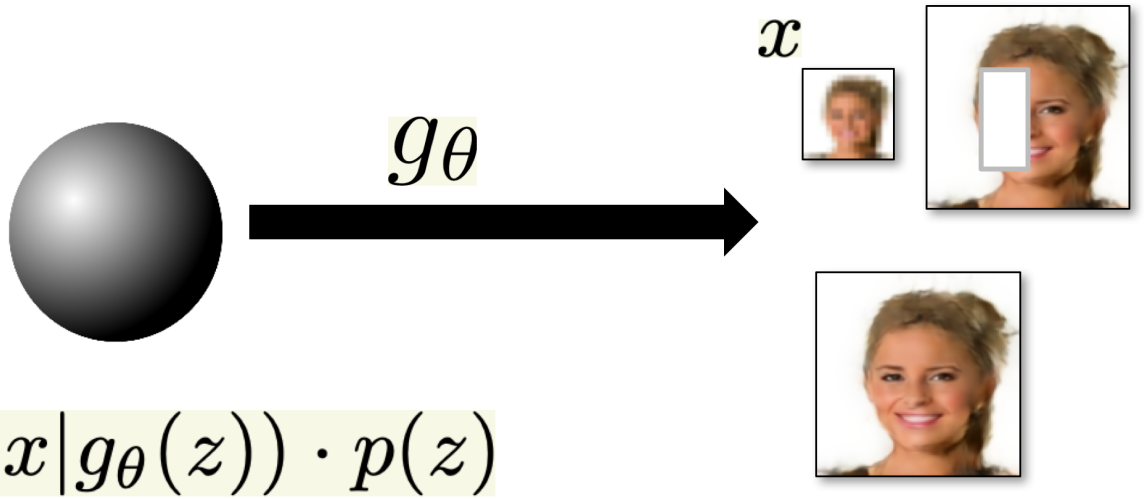
Answer: image restoration/editing/enhancement



$$x \sim p(x|g_\theta(z)) \cdot p(z)$$

$$\hat{z} = \arg\max_z \left[\log p(x|g_\theta(z)) + \log p(z)\right]$$

Answer: image restoration/editing/enhancement



$$x \sim p(x|g_\theta(z)) \cdot p(z)$$

$$\hat{z} = \arg\max_z \left[\log p(x|g_\theta(z)) + \log p(z)\right]$$

$$\hat{x} = g_\theta(\hat{z})$$

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

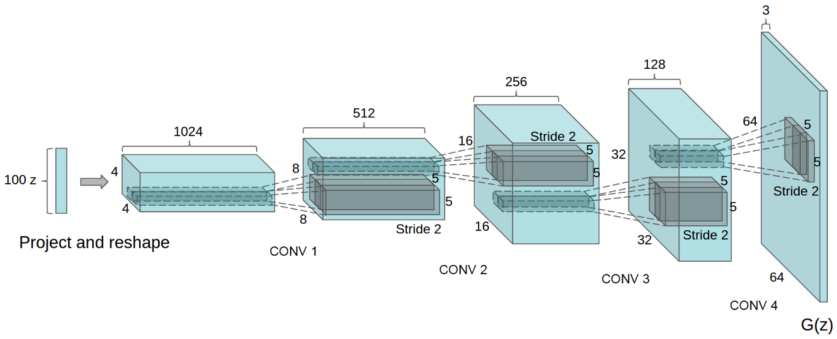
Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Decoder neural network

Decoder neural network:

$$p_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \sigma^2_\theta(z) \cdot \mathrm{I})$$



Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

✔ ✔ ✗

Intractible to compute!

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

Posterior density is also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$

(✔ $p_\theta(z)$, ✔ $p_\theta(x|z)$, ✗ $\int$)

Posterior density is also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$

(✔ $p_\theta(x|z)$, ✔ $p_\theta(z)$, ✗ $p_\theta(x)$)

Intractable data likelihood

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$ (✔ ✔ ✗)

Posterior density is also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$ (✔ ✔ ✗)

Solution: construct an encoder network $q_\phi(z|x)$ to approximate $p_\theta(z|x)$

Data likelihood: $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$ (✔ ✔ ✗)

Posterior density is also intractable: $p_\theta(z|x) = p_\theta(x|z) p_\theta(z) / p_\theta(x)$ (✔ ✔ ✗)

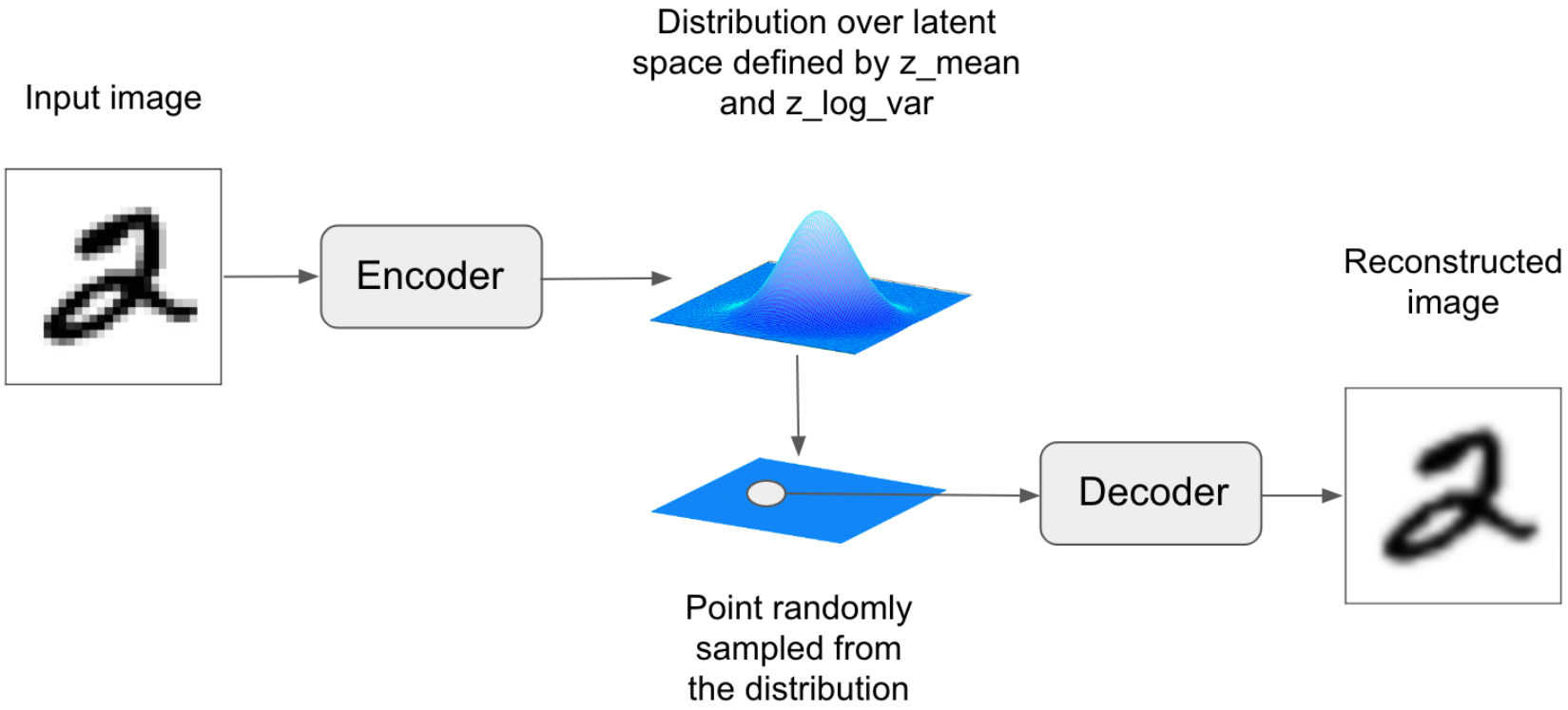
Solution: construct an encoder network $q_\phi(z|x)$

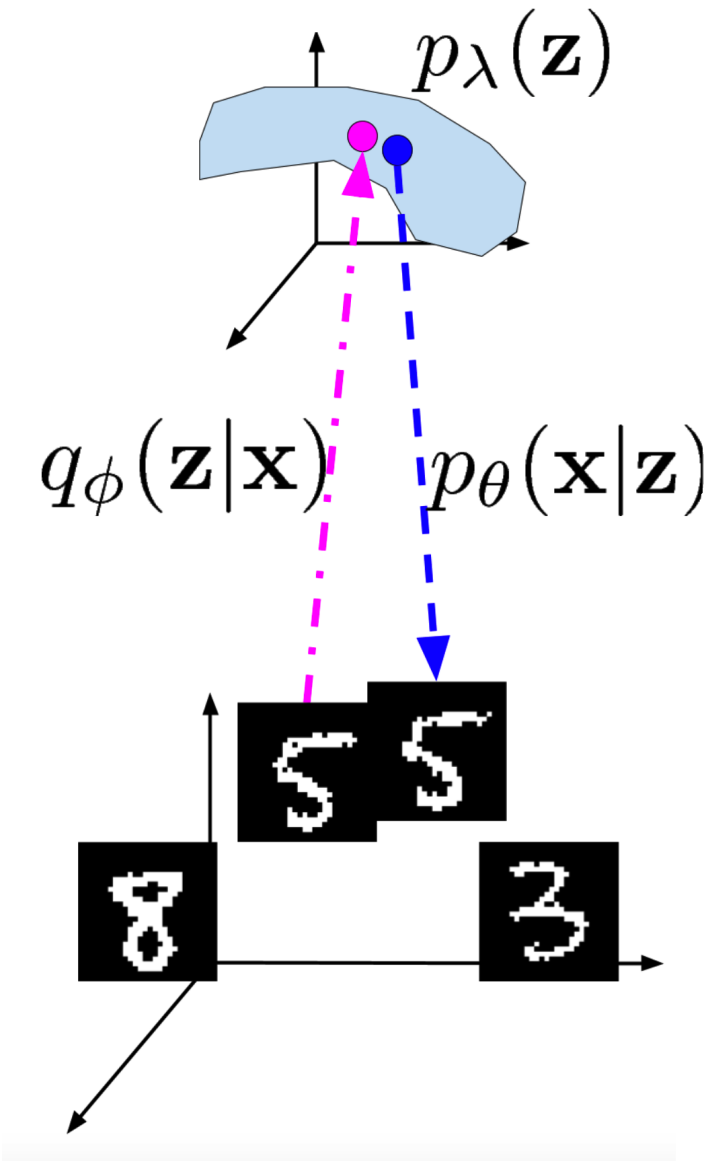
to approximate $p_\theta(z|x)$

Encoder neural network:

$$q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \sigma^2_\phi(x) \cdot \mathrm{I})$$

Variational AutoEncoders




$p_\lambda(\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$p_\theta(\mathbf{x}|\mathbf{z})$

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$
$$= KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i)$$

always ≥ 0

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$

$$= KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i)$$

always ≥ 0

$$\geq L(\theta, \phi, x_i)$$

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$
$$= KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i)$$

always ≥ 0

$$\geq L(\theta, \phi, x_i)$$

Variational lower-bound:

$$L(\theta, \phi, x_i) =$$
$$= -KL[q_\phi(z|x_i)||p(z)] + \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]$$

Marginal log-likelihood:

$$\log p_\theta(x_i) = \\ = KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i) \\ \geq L(\theta, \phi, x_i)$$

always ≥ 0

Variational lower-bound:

$$L(\theta, \phi, x_i) = \\ = -KL[q_\phi(z|x_i)||p(z)] + \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]$$

Regularization!

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$
$$= KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i)$$

always ≥ 0

$$\geq L(\theta, \phi, x_i)$$

Variational lower-bound:

$$L(\theta, \phi, x_i) =$$
$$= -KL[q_\phi(z|x_i)||p(z)] + \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]$$

Regularization!

Latent representation

Marginal log-likelihood:

$$\log p_\theta(x_i) =$$

$$= KL[q_\phi(z|x)||p_\theta(z|x)] + L(\theta, \phi, x_i)$$

always ≥ 0

$$\geq L(\theta, \phi, x_i)$$

Variational lower-bound:

$$L(\theta, \phi, x_i) =$$

$$= -KL[q_\phi(z|x_i)||p(z)] + \mathbb{E}_{q_\phi(z|x_i)}[\log p_\theta(x_i|z)]$$

Reconstruction error

Regularization!

Latent representation

ELBO:

ELBO:

$$\sum_{i=1}^{n} \log p_\theta(x_i) \geq$$

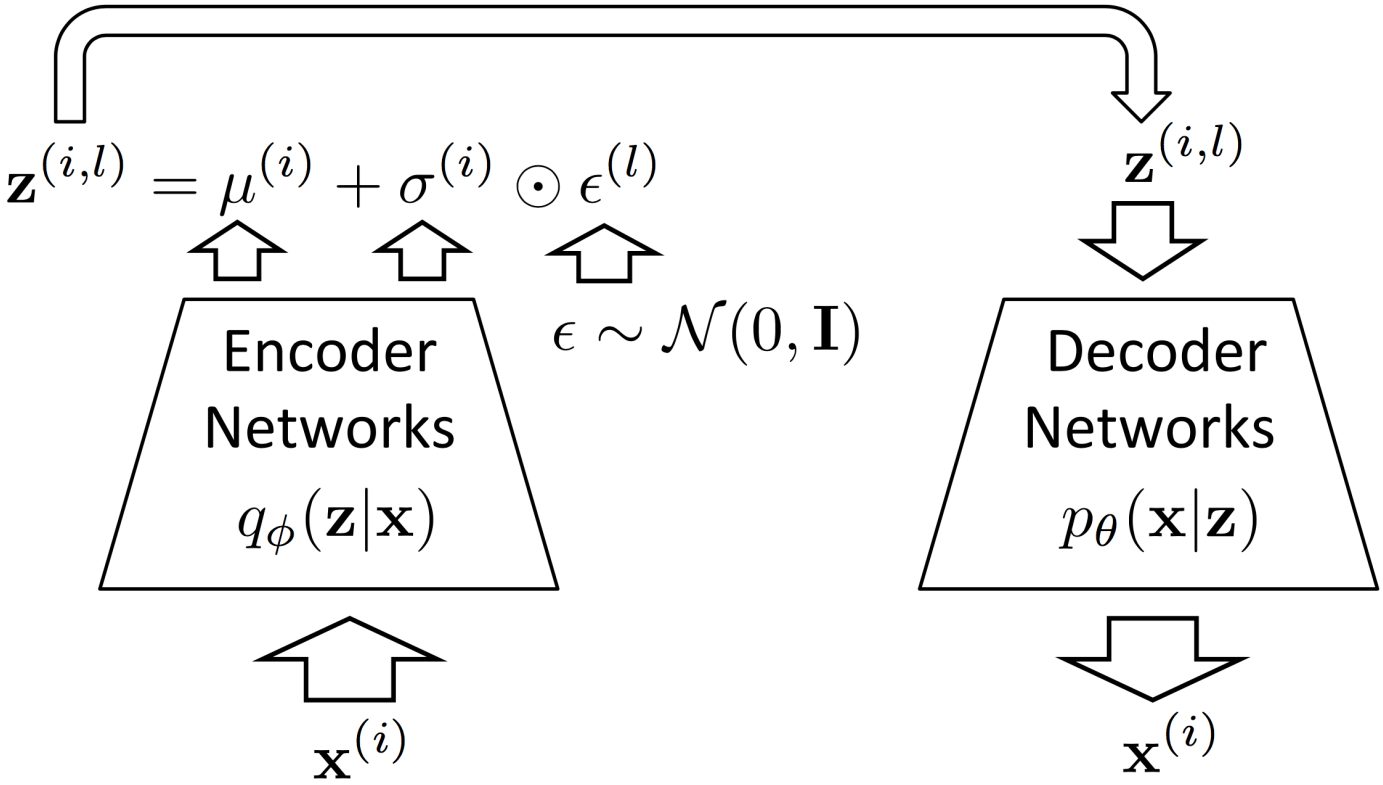
$$\geq \sum_{i=1}^{n} L(\theta, \phi, x_i)$$

ELBO:

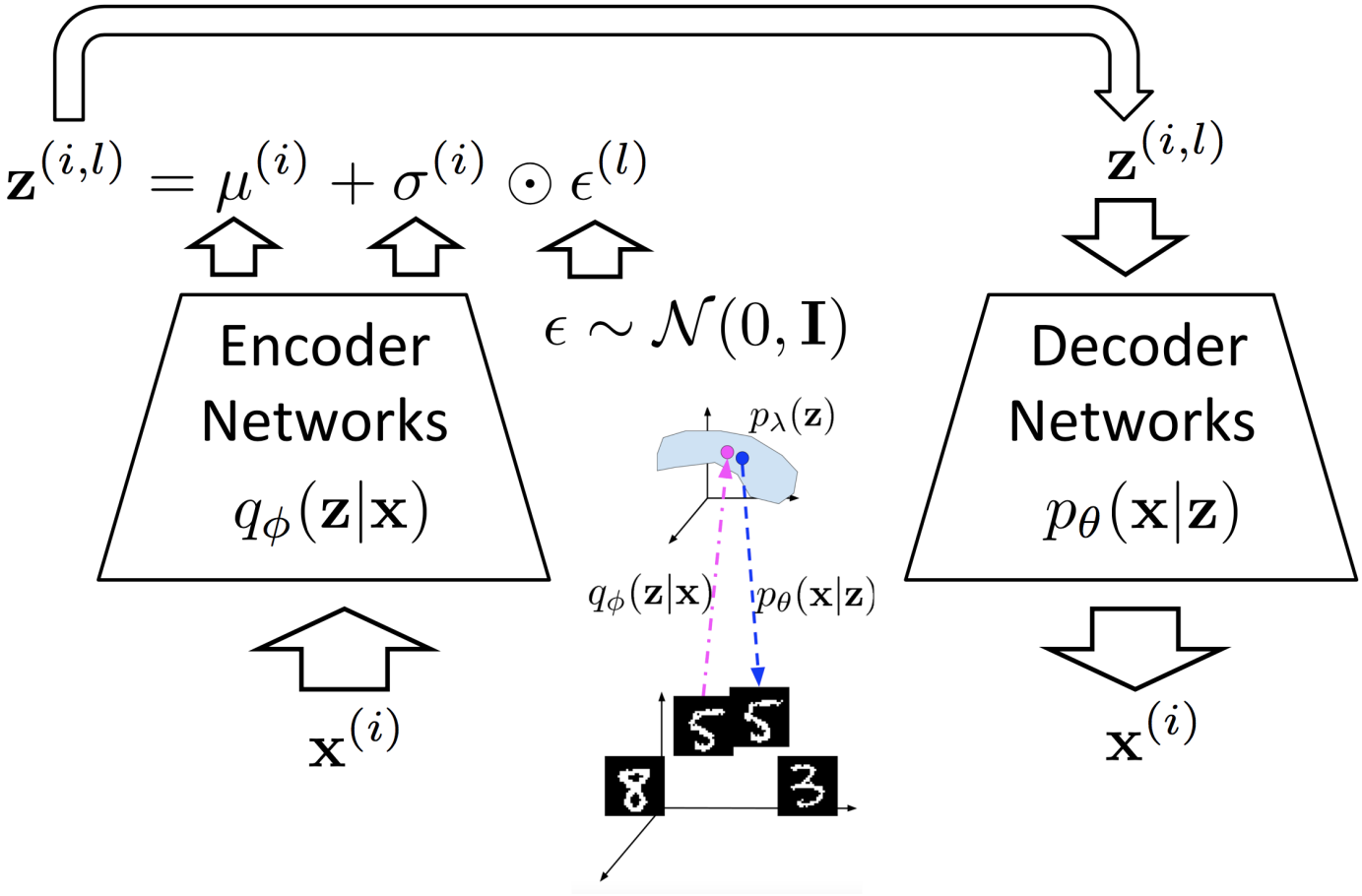
$$\sum_{i=1}^{n} \log p_\theta(x_i) \geq$$

$$\geq \sum_{i=1}^{n} L(\theta, \phi, x_i) \rightarrow \max_{\theta,\phi}$$

Variational AutoEncoders: Reparameterization Trick



Variational AutoEncoders: Reparameterization Trick



Empirical variational lower-bound:

Empirical variational lower-bound:

$$p(z) = \mathcal{N}(z|0, \mathrm{I})$$

Empirical variational lower-bound:

$$p(z) = \mathcal{N}(z|0, \mathrm{I})$$

$$z_{i,l} \sim q_\phi(z|x_i)$$

Empirical variational lower-bound:

$$p(z) = \mathcal{N}(z|0, \mathrm{I})$$

$$z_{i,l} \sim q_\phi(z|x_i)$$

$$\hat{L}(\theta, \phi, x_i) = \frac{1}{2}\sum_{j=1}^{d}[1 + \log\sigma^2_{j,\phi}(x_i) - \mu^2_{j,\phi}(x_i) - \sigma^2_{j,\phi}(x_i)]$$

$$+ \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x_i|z_{i,l}) \to \max_{\theta,\phi}$$

Empirical variational lower-bound:

$$p(z) = \mathcal{N}(z|0, \mathrm{I})$$

$$z_{i,l} \sim q_\phi(z|x_i)$$

Regularization!

$$\hat{L}(\theta, \phi, x_i) = \frac{1}{2}\sum_{j=1}^{d}[1 + \log \sigma^2_{j,\phi}(x_i) - \mu^2_{j,\phi}(x_i) - \sigma^2_{j,\phi}(x_i)]$$

$$+ \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x_i|z_{i,l}) \to \max_{\theta,\phi}$$

Empirical variational lower-bound:

$$p(z) = \mathcal{N}(z|0, \mathrm{I})$$

$$z_{i,l} \sim q_\phi(z|x_i)$$

Regularization!

$$\hat{L}(\theta, \phi, x_i) = \frac{1}{2}\sum_{j=1}^{d}[1 + \log\sigma^2_{j,\phi}(x_i) - \mu^2_{j,\phi}(x_i) - \sigma^2_{j,\phi}(x_i)]$$

$$+ \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(x_i|z_{i,l}) \to \max_{\theta,\phi}$$

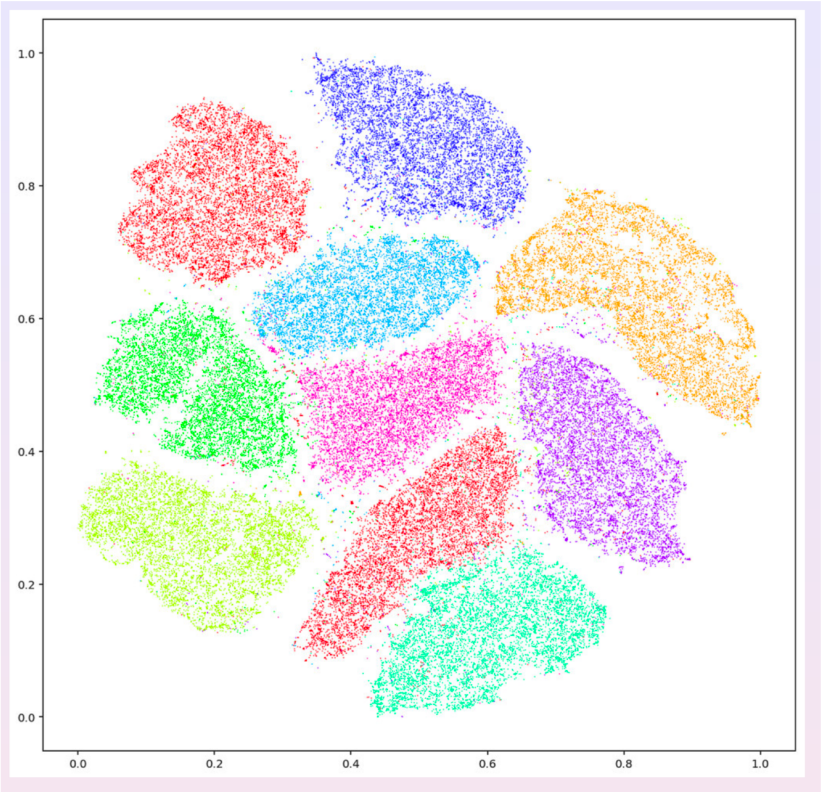
Reconstruction error!

Example: MNIST

Use decoder network

Example: MNIST

Use decoder network



Each digit corresponds to a distribution on the manifold

Example: MNIST

Use decoder network. Now sample $\boldsymbol{z}$ from prior!

Use decoder network. Now sample $\boldsymbol{z}$ from prior!

Diagonal prior on $\boldsymbol{z}$
=> independent latent variables

Example: MNIST

Use decoder network. Now sample $\boldsymbol{z}$ from prior!

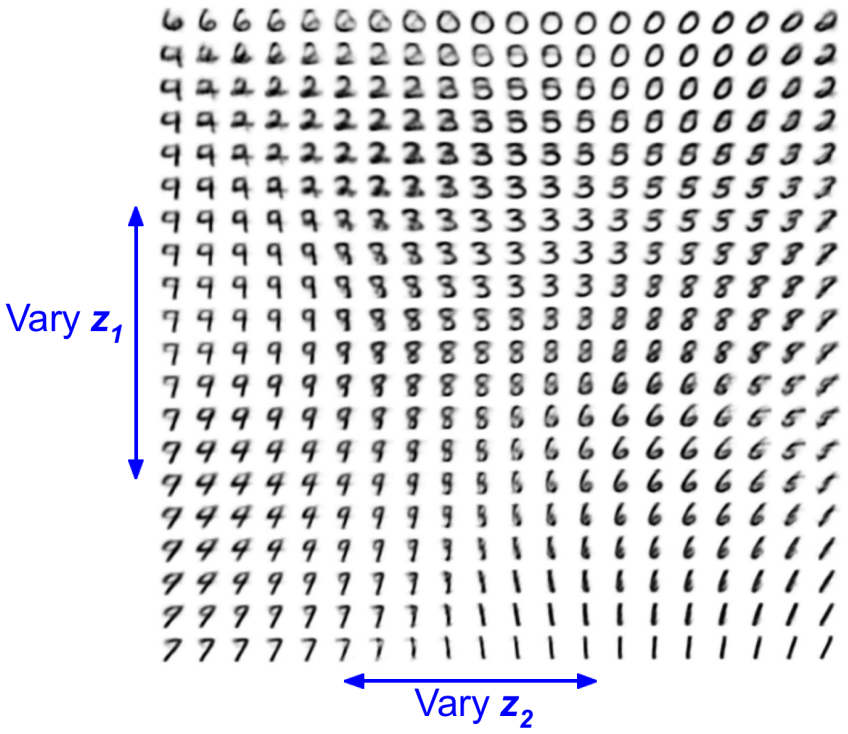
Diagonal prior on $\boldsymbol{z}$
=> independent latent variables

Different dimensions of $\boldsymbol{z}$ encode interpretable factors of variation

Example: MNIST

Use decoder network. Now sample z from prior!

Data manifold for 2-d z

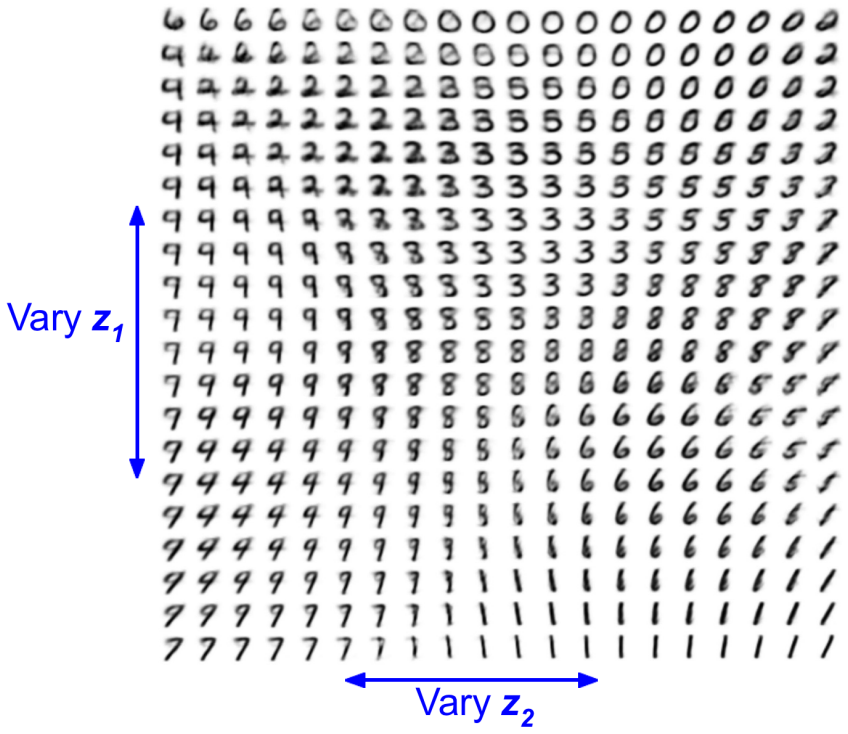


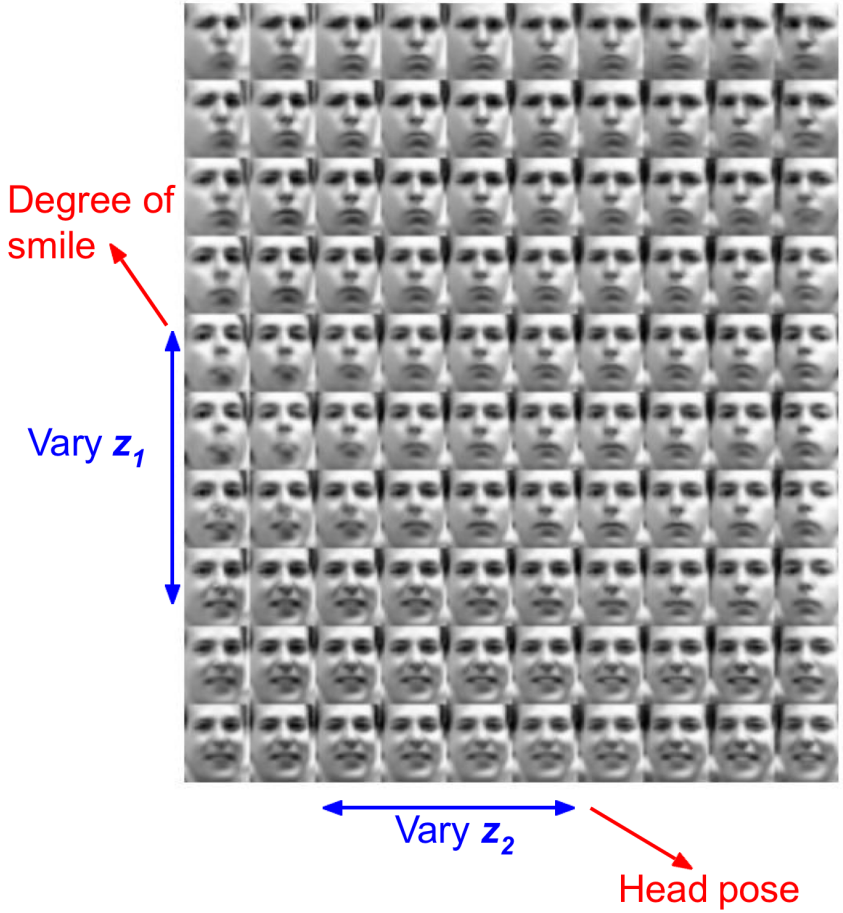
Diagonal prior on z
=> independent latent variables

Different dimensions of z encode interpretable factors of variation

Use decoder network. Now sample z from prior!

Data manifold for 2-d z





Use decoder network. Now sample z from prior!

Labeled Faces in the Wild