

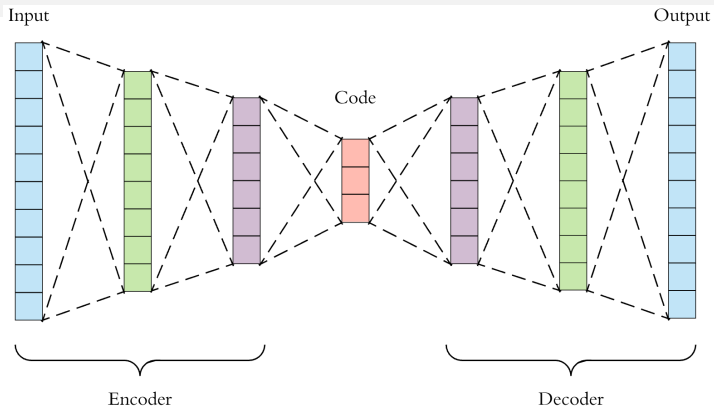
Variational Auto-Encoders

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad



Recall: Autoencoders

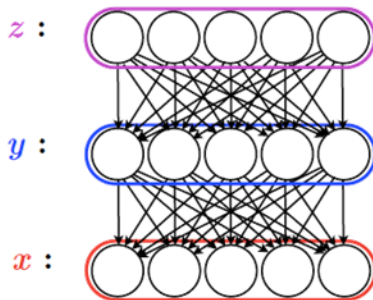


Autoencoders can reconstruct data, and can learn features to initialize a supervised model.
Can we generate images from an autoencoder?

Credit: [Arden Dertat, TowardsDataScience](#)

Variational Autoencoders

- Introduced around the same time by two groups of researchers:
 - Kingma and Welling, **Auto-Encoding Variational Bayes**, ICLR 2014
 - Rezende, Mohamed and Wierstra, **Stochastic Backpropagation and Variational Inference in Deep Latent Gaussian Models**, ICML 2014



Credit: Aaron Courville, Deep Learning Summer School, 2015

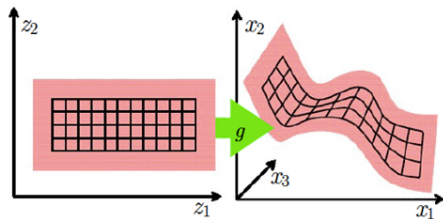
Variational Autoencoders

- **Latent Variable Model:** Learn a mapping from some latent variable z to a possibly complex distribution on x

$$p(x) = \int p(x, z) dz \quad \text{where} \quad p(x, z) = p(x|z)p(z)$$

$$p(z) = \text{something simple}; \quad p(x|z) = g(z)$$

- Can we learn to decouple the true explanatory factors (**latent variables**) underlying the data distribution (e.g. identity and expression in face images)? How?



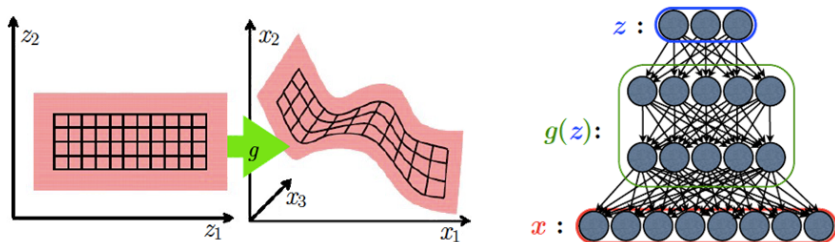
Credit: Aaron Courville, Deep Learning Summer School, 2015

Variational Autoencoders

- Leverage neural networks to learn a latent variable model!

$$p(x) = \int p(x, z) dz \quad \text{where} \quad p(x, z) = p(x|z)p(z)$$

$$p(z) = \text{something simple}; \quad p(x|z) = g(z)$$



Credit: Aaron Courville, Deep Learning Summer School, 2015

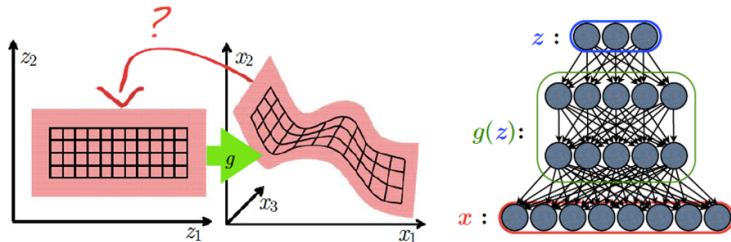
Variational Autoencoders

- Leverage neural networks to learn a latent variable model!

$$p(x) = \int p(x, z) dz \quad \text{where} \quad p(x, z) = p(x|z)p(z)$$

$$p(z) = \text{something simple}; \quad p(x|z) = g(z)$$

- Where does z come from? Computing the posterior $p(z|x)$ is intractable, and we need it to train the directed model

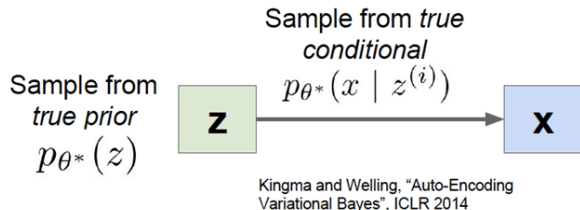


Credit: Aaron Courville, Deep Learning Summer School, 2015

Variational Autoencoders

A Bayesian spin on an autoencoder!

Assume our data $\{x^{(i)}\}_{i=1}^N$ is generated like this:

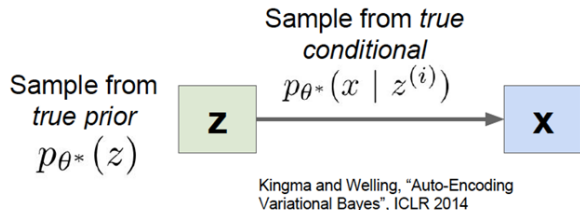


Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders

A Bayesian spin on an autoencoder!

Assume our data $\{x^{(i)}\}_{i=1}^N$ is generated like this:



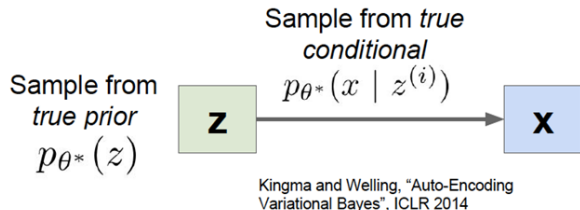
- **Intuition:** x is an image, z gives class, orientation, attributes, etc

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders

A Bayesian spin on an autoencoder!

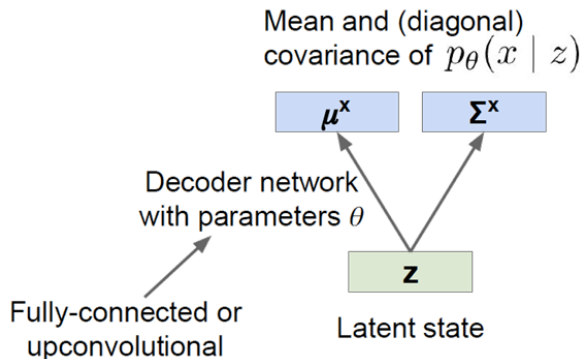
Assume our data $\{x^{(i)}\}_{i=1}^N$ is generated like this:



- **Intuition:** x is an image, z gives class, orientation, attributes, etc
- **Problem:** Estimate θ without access to latent states $z^{(i)}$

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

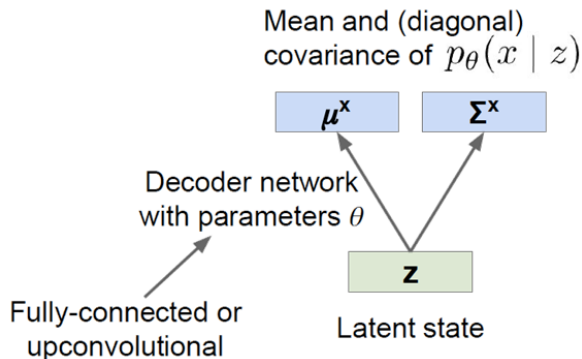
Variational Autoencoders



- **Prior:** Assume $p_{\theta}(z)$ is a unit Gaussian

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders



- **Prior:** Assume $p_{\theta}(z)$ is a unit Gaussian
- **Conditional:** Assume $p_{\theta}(x|z)$ is a diagonal Gaussian, predict mean and variance with neural network

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders

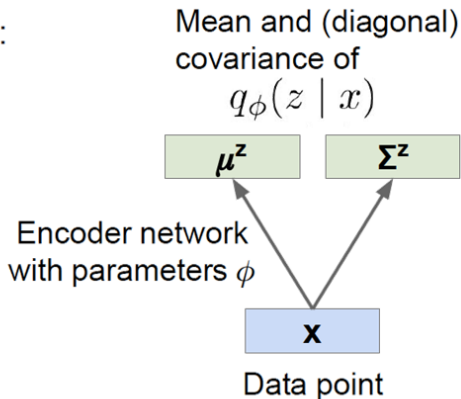
By Bayes Rule the posterior is:

$$p_{\theta}(z | x) = \frac{p_{\theta}(x | z) p_{\theta}(z)}{p_{\theta}(x)}$$

Use decoder network 😊

Gaussian 😊

Intractable integral ☹️



Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders

By Bayes Rule the posterior is:

$$p_{\theta}(z | x) = \frac{p_{\theta}(x | z) p_{\theta}(z)}{p_{\theta}(x)}$$

Use decoder network 😊

Gaussian 😊

Intractible integral ☹️

Fully-connected
or convolutional

Encoder network
with parameters ϕ

Mean and (diagonal)
covariance of
 $q_{\phi}(z | x)$

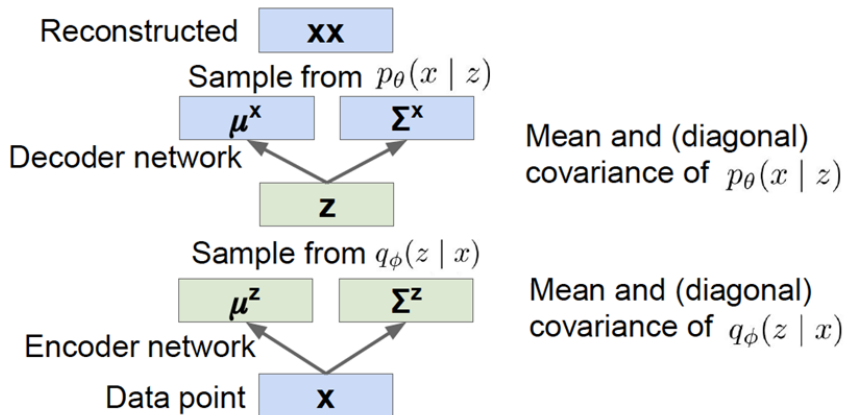


Approximate posterior with
encoder network $q_{\phi}(z | x)$

Data point

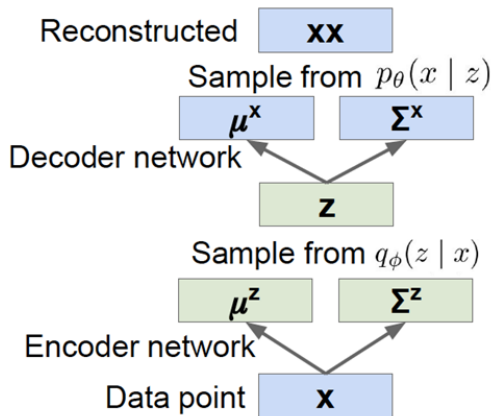
Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders



Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoders



Training like a normal autoencoder:
reconstruction loss at the end,
regularization toward prior in middle

Mean and (diagonal)
covariance of $p_\theta(x | z)$
(should be close to data x)

Mean and (diagonal)
covariance of $q_\phi(z | x)$
(should be close to prior $p_\theta(z)$)

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^N p_{\theta}(x^{(i)})$$

Maximize likelihood of dataset $\{x^{(i)}\}_{i=1}^N$

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^N p_{\theta}(x^{(i)})$$

Maximize likelihood of dataset $\{x^{(i)}\}_{i=1}^N$

$$= \arg \max_{\theta} \sum_{i=1}^N \log p_{\theta}(x^{(i)})$$

Maximize log-likelihood instead
because sums are nicer

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^N p_{\theta}(x^{(i)})$$

Maximize likelihood of dataset $\{x^{(i)}\}_{i=1}^N$

$$= \arg \max_{\theta} \sum_{i=1}^N \log p_{\theta}(x^{(i)})$$

Maximize log-likelihood instead
because sums are nicer

$$p_{\theta}(x^{(i)}) = \int p_{\theta}(x^{(i)}, z) dz$$

Marginalize joint
distribution

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^N p_{\theta}(x^{(i)})$$

Maximize likelihood of dataset $\{x^{(i)}\}_{i=1}^N$

$$= \arg \max_{\theta} \sum_{i=1}^N \log p_{\theta}(x^{(i)})$$

Maximize log-likelihood instead because sums are nicer

$$p_{\theta}(x^{(i)}) = \int p_{\theta}(x^{(i)}, z) dz$$

Marginalize joint distribution

$$= \int p_{\theta}(x^{(i)} | z) p_{\theta}(z) dz$$

Intractable integral ☹️

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\begin{aligned}\log p_{\theta}(x^{(i)}) &= \mathbf{E}_{z \sim q_{\phi}(z|x^{(i)})} \left[\log p_{\theta}(x^{(i)}) \right] && (p_{\theta}(x^{(i)}) \text{ Does not depend on } z) \\ &= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \right] && (\text{Bayes' Rule}) \\ &= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \frac{q_{\phi}(z | x^{(i)})}{q_{\phi}(z | x^{(i)})} \right] && (\text{Multiply by constant})\end{aligned}$$

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\begin{aligned}\log p_{\theta}(x^{(i)}) &= \mathbf{E}_{z \sim q_{\phi}(z|x^{(i)})} \left[\log p_{\theta}(x^{(i)}) \right] \quad (p_{\theta}(x^{(i)}) \text{ Does not depend on } z) \\&= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Bayes' Rule}) \\&= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \frac{q_{\phi}(z | x^{(i)})}{q_{\phi}(z | x^{(i)})} \right] \quad (\text{Multiply by constant}) \\&= \mathbf{E}_z \left[\log p_{\theta}(x^{(i)} | z) \right] - \mathbf{E}_z \left[\log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z)} \right] + \mathbf{E}_z \left[\log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Logarithms}) \\&= \underbrace{\mathbf{E}_z \left[\log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi) \text{ "Elbow"}} + \underbrace{D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z | x^{(i)}))}_{\geq 0}\end{aligned}$$

Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: The Math

$$\begin{aligned}\log p_{\theta}(x^{(i)}) &= \mathbf{E}_{z \sim q_{\phi}(z|x^{(i)})} \left[\log p_{\theta}(x^{(i)}) \right] \quad (p_{\theta}(x^{(i)}) \text{ Does not depend on } z) \\&= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Bayes' Rule}) \\&= \mathbf{E}_z \left[\log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \frac{q_{\phi}(z | x^{(i)})}{q_{\phi}(z | x^{(i)})} \right] \quad (\text{Multiply by constant}) \\&= \mathbf{E}_z \left[\log p_{\theta}(x^{(i)} | z) \right] - \mathbf{E}_z \left[\log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z)} \right] + \mathbf{E}_z \left[\log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Logarithms}) \\&= \underbrace{\mathbf{E}_z \left[\log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi) \text{ "Elbow"}} + \underbrace{\mathbf{E}_z \left[\log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z | x^{(i)})} \right]}_{\geq 0}\end{aligned}$$

$\log p_{\theta}(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)$
Variational lower bound (elbow)

$\theta^*, \phi^* = \arg \max_{\theta, \phi} \sum_{i=1}^N \mathcal{L}(x^{(i)}, \theta, \phi)$
Training: Maximize lower bound

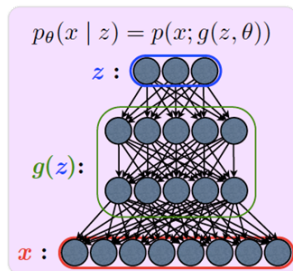
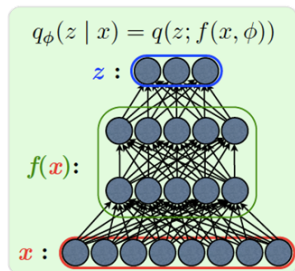
Credit: Fei-Fei Li, Andrej Karpathy and Justin Johnson, CS231n, Stanford Univ

Variational Autoencoder: Inference

- Introduce an inference model $q_{\phi}(z|x)$ that learns to approximate the intractable posterior $p_{\theta}(z|x)$ by optimizing the variational lower bound:

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

- We parametrize $q_{\phi}(z|x)$ with another neural network:



Credit: Aaron Courville, Deep Learning Summer School, 2015

Variational Autoencoder: How to train?

$$\begin{aligned}\mathcal{L}_{VAE} &= \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{p_\theta(z, x)}{q_\phi(z|x)} \right] \\ &= -D_{KL}(q_\phi(z|x) || p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x|z)]\end{aligned}$$

- $z \sim q_\phi(z|x)$: need to differentiate through the sampling process; how to update ϕ ? (encoder is probabilistic)

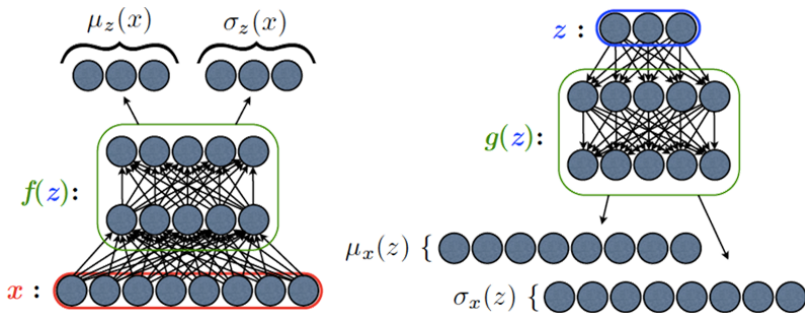
Variational Autoencoder: How to train?

$$\begin{aligned}\mathcal{L}_{VAE} &= \mathbb{E}_{q_{\phi}(z|x)} \left[\log \frac{p_{\theta}(z, x)}{q_{\phi}(z|x)} \right] \\ &= -D_{KL}(q_{\phi}(z|x) || p_{\theta}(z)) + \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x|z)]\end{aligned}$$

- $z \sim q_{\phi}(z|x)$: need to differentiate through the sampling process; how to update ϕ ? (encoder is probabilistic)
- **Solution:** Make the randomness independent of encoder output, thus making the encoder deterministic; how?

Reparametrization Trick

- Let's consider z to be real and $q_\phi(z|x) = \mathcal{N}(z; u_z(x), \sigma_z(x))$
- Parametrize z as $z = \mu_z(z) + \sigma_z(x)\epsilon_z$ where $\epsilon_z = \mathcal{N}(0, 1)$

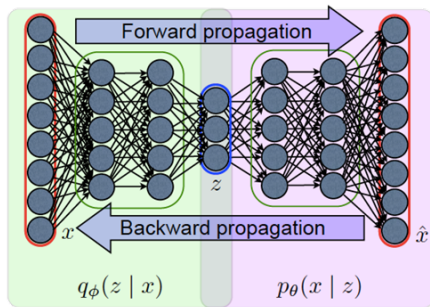


Credit: Aaron Courville, Deep Learning Summer School, 2015

Training with Backpropagation

With the **reparametrization trick**, we can simultaneously train both the **generative model** $p_{\theta}(x|z)$ and the **inference model** $q_{\phi}(z|x)$ using backpropagation

Objective function: $\mathcal{L}(\theta, \phi, x) = -D_{KL}(q_{\phi}(z|x)||p_{\theta}(z)) + \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)]$



Credit: Aaron Courville, Deep Learning Summer School, 2015

VAE: Summary

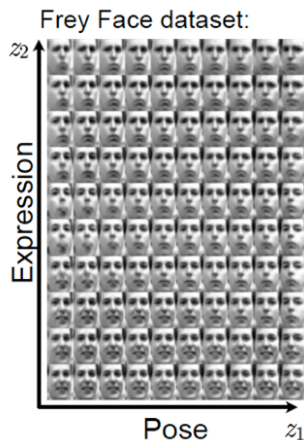
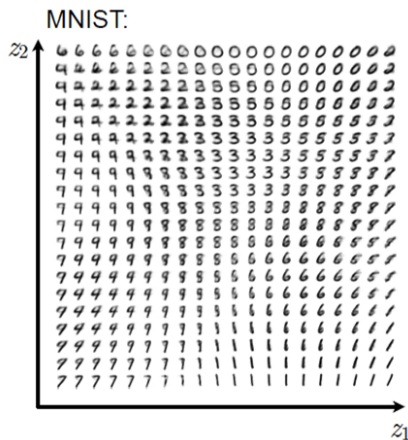
Traditional Autoencoders

- Learned by reconstructing input
- Used to learn features, initialize supervised models (not much anymore though)

Variational Autoencoders

- Bayesian learning meets deep learning
- Sample from model to generate images

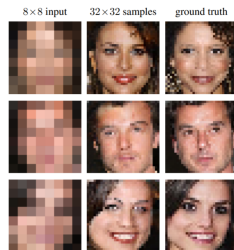
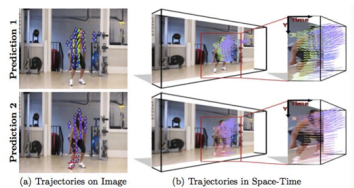
VAE: What can they do?



Credit: Aaron Courville, Deep Learning Summer School, 2015

Applications of VAEs

- Image and video generation
- Superresolution
- Forecasting from static images
- Image inpainting
- many more...



Credit: Dahl et al, Pixel Recursive Super Resolution, ICCV 2017

A Few Variants and Extensions

- Semi-Supervised VAEs
 - Kingma et al, Semi-Supervised Learning with Deep Generative Models, NeurIPS 2014
- Conditional VAE
 - Sohn et al, Learning Structured Output Representation using Deep Conditional Generative Models, NeurIPS 2015
- Importance-Weighted VAE
 - Burda et al, Importance Weighted Autoencoders, ICLR 2016
- Denoising VAE
 - Jiwoong et al, Denoising Criterion for Variational Auto-encoding Framework, AAAI 2017
- Inverse Graphics Network
 - Kulkarni et al, Deep Convolutional Inverse Graphics Network, NeurIPS 2015
- Adversarial Autoencoders
 - Makhzani et al, Adversarial Autoencoders, ICLR 2016

Homework

Readings

- Carl Doersch, [Tutorial on Variational Autoencoder](#), arXiv 2016
- VAE [example](#) in PyTorch
- Kingma and Welling, [Auto-Encoding Variational Bayes](#), ICLR 2014

Question

- Why does the encoder of a VAE map to a vector of means and a vector of standard deviations? Why does it not instead map to a vector of means and a covariance matrix?
- What about the decoder? If we assume a Mean Squared Error for the reconstruction loss, what is the covariance of the $p(x|z)$ Gaussian distribution?