Computer Vision – TP14
Advanced Deep Learning Topics II

Miguel Coimbra, Francesco Renna

U.PORTO FC

Outline

- Generative models
- Variational autoencoders (VAEs)
- Generative adversarial networks (GANs)

Outline

- Generative models
- Variational autoencoders (VAEs)
- Generative adversarial networks (GANs)

Generative models

- Given a set of training data, learn their distribution and generate new data from a similar distribution

Training data ~ $p_{data}(x)$ Generated samples ~ $p_{model}(x)$

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

- Explicit: returns $p_{model}(x)$
- Implicit: generate samples only

Applications

- Generate new data for simulations
  - Reinforcement learning
- Generate new data for model training
  - Data augmentation
- Fill in the gaps of measured data
  - Super-resolution
  - Colorization
- Inference of latent representation

Outline

- Generative models
- Variational autoencoders (VAEs)
- Generative adversarial networks (GANs)

Variational Autoencoders

- Idea: we can use the autoencoder approach to generate data from a specific distribution
- Training: data sampled from such distribution
- Use autoencoder to generate the statistical description of the data

Variational Autoencoders

- Idea
  - Encoder and decoder provide **distributions** (their parameters), not data points!
- Assumptions
  - Training data $\{x_i\}_{i=1}^{N}$
  - $p(z)$ Gaussian distribution
  - $p(x|z)$ Gaussian distribution (Decoder)
  - $p(z|x)$ approximated by a Gaussian distribution (Encoder)

Variational Autoencoders

- Training
  - Use a variational lower bound of the log-likelihood $\log p_\theta(x_i)$
- Generate data
  - Sample z (latent representation) from a Gaussian prior
  - Use decoder to get (Gaussian) $p_\theta(x|z)$
  - Sample $x|z$ from $p_\theta(x|z)$

Training

- Data likelihood intractable to compute:

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

- Use a network to model encoder distribution

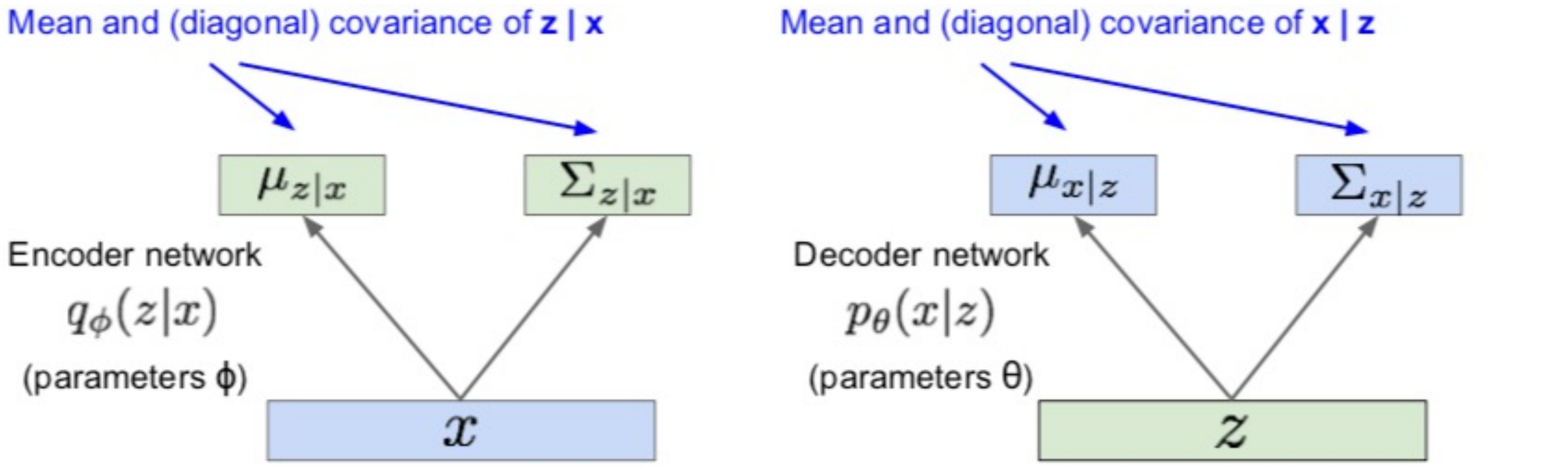
$$q_\phi(z|x) \approx p(z|x)$$

- Then, optimize a lower bound of the data likelihood:

$$\log p_\theta(x) \geq E_z[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) || p_\theta(z))$$

Training

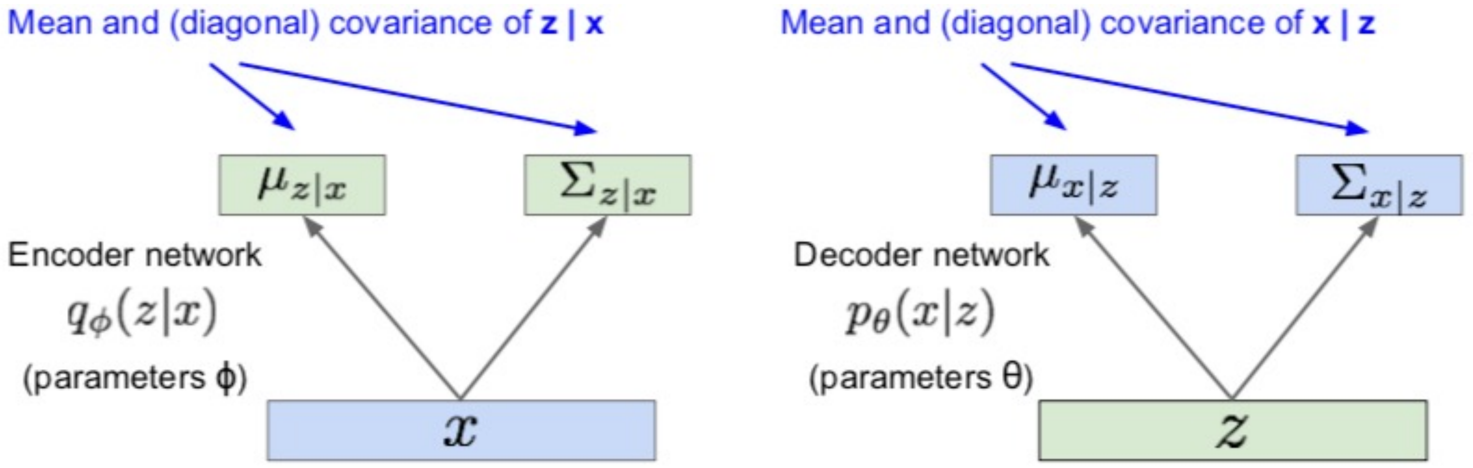
- Encoder and decoder networks return distribution parameters



Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Training

- Sample from these distributions to get latent z and image x



Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Training

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

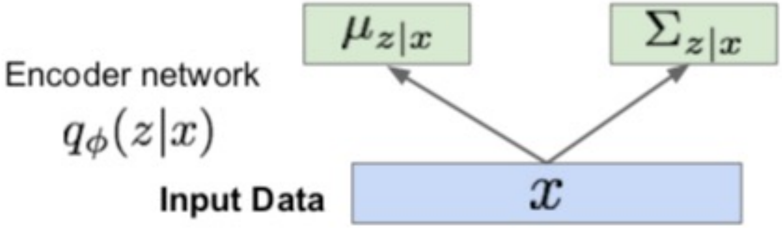
Let's look at computing the bound (forward pass) for a given minibatch of input data

Input Data x

Training

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

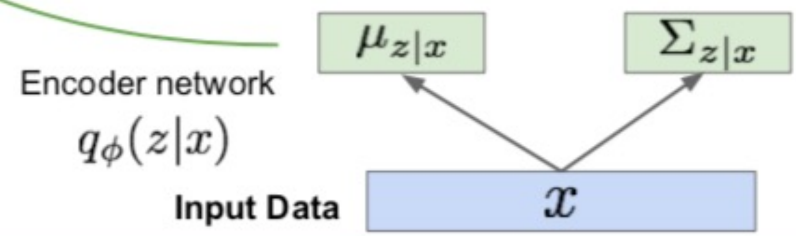


Training

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Make approximate posterior distribution close to prior

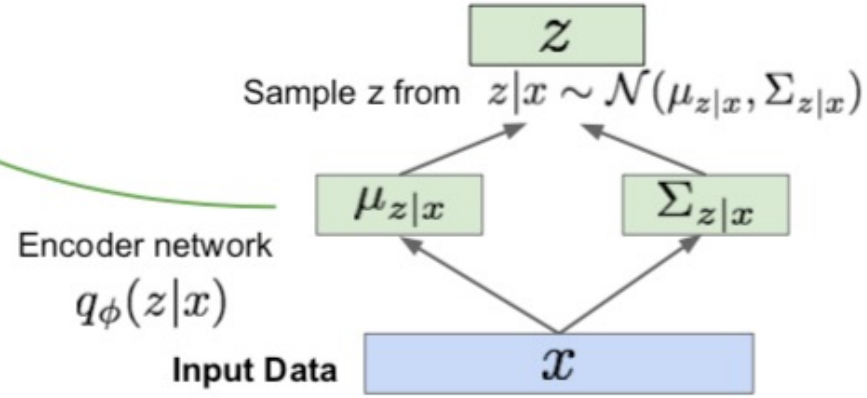


Training

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Make approximate posterior distribution close to prior

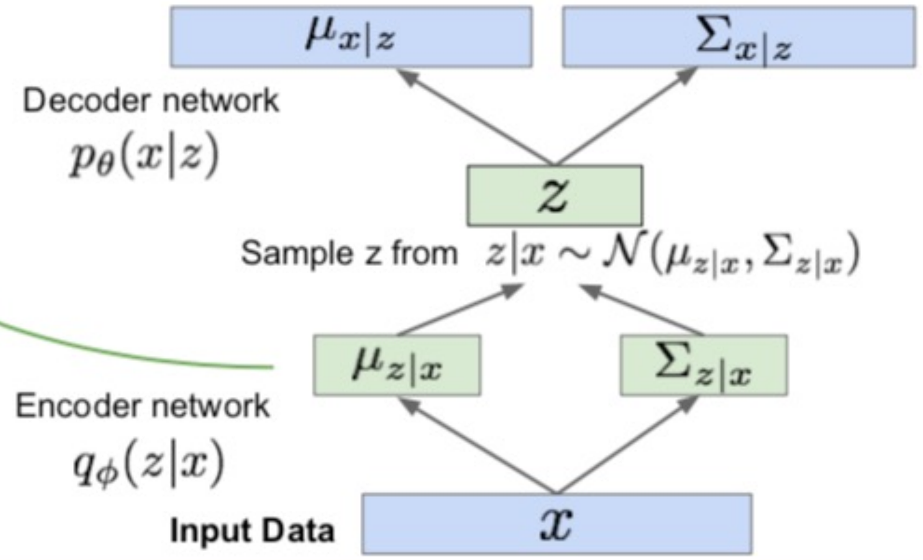


Training

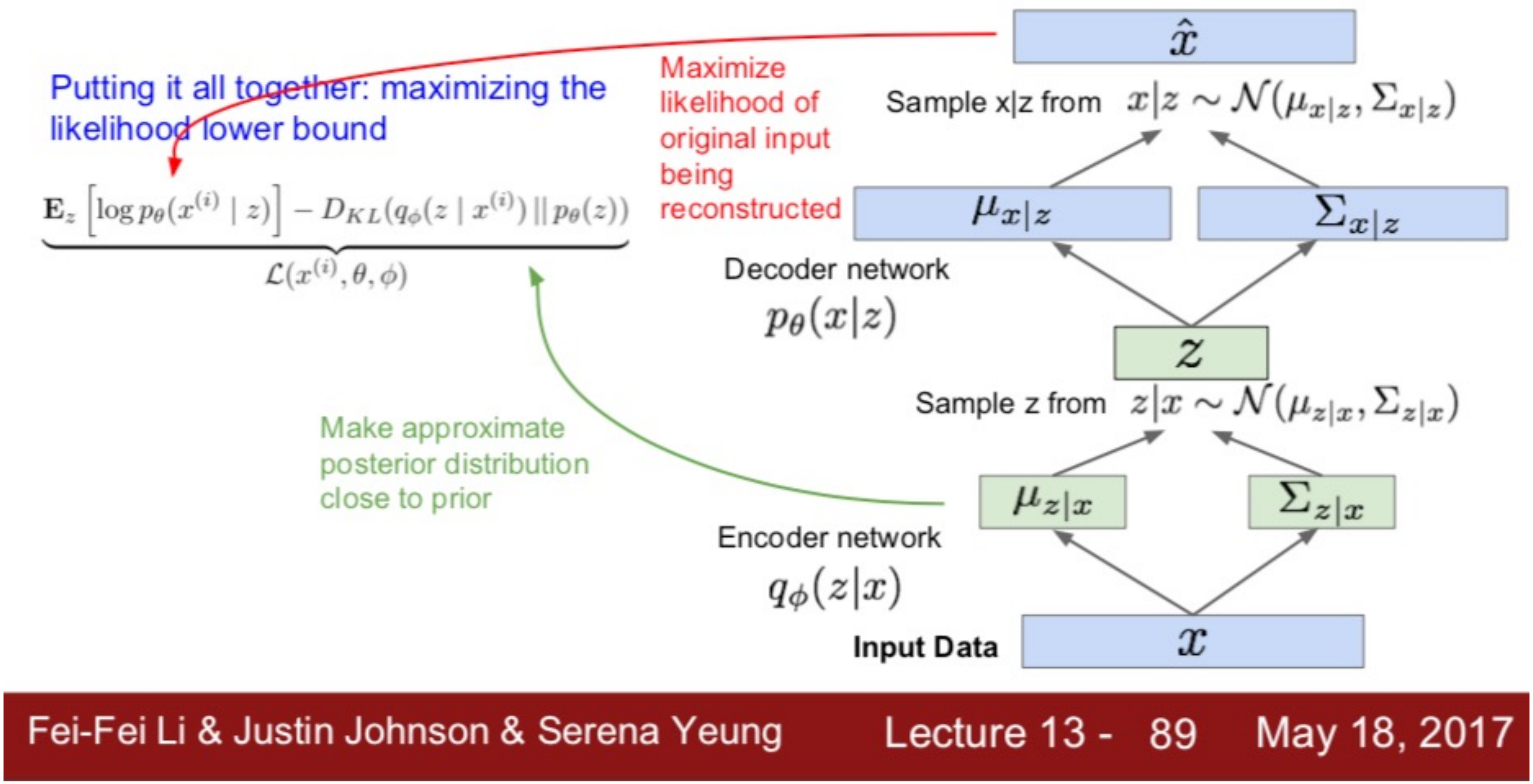
Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

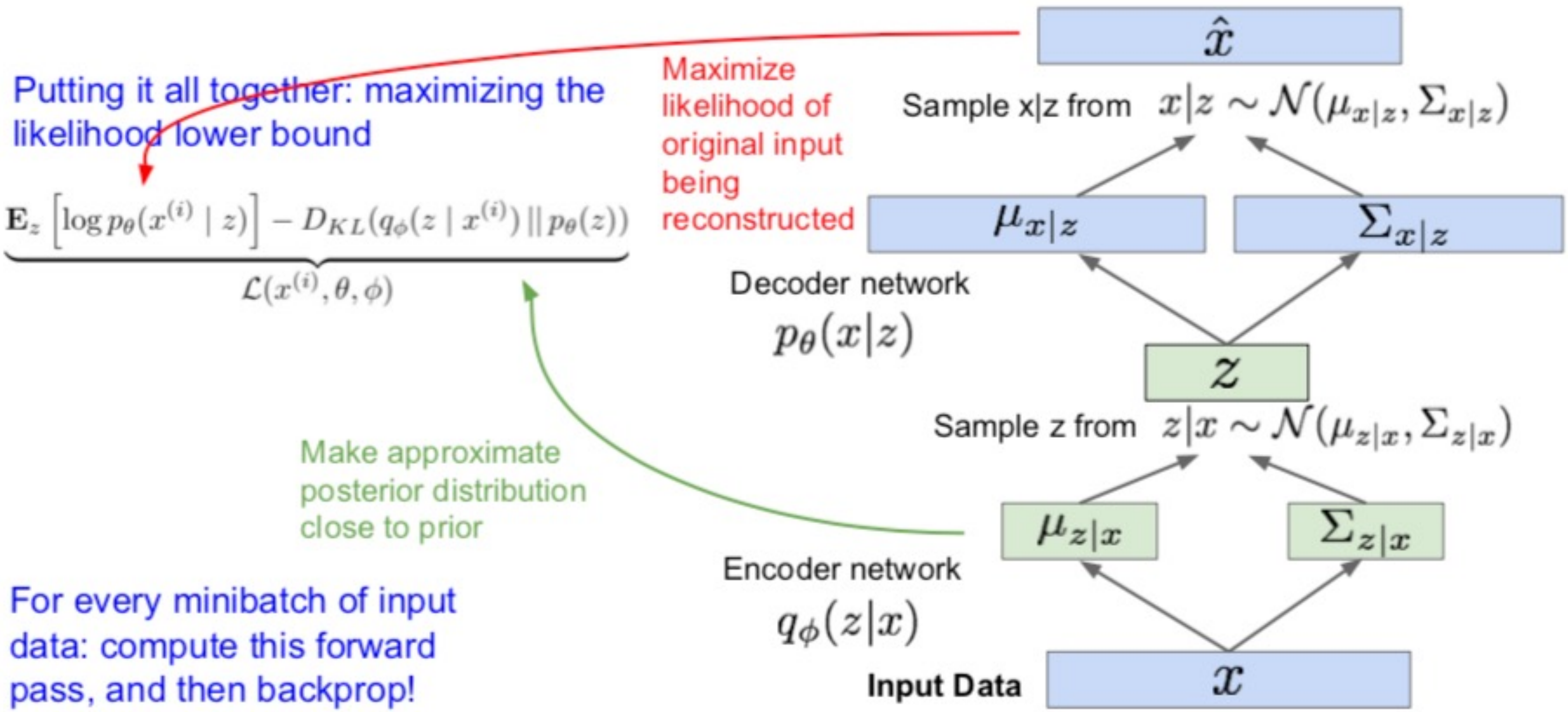
Make approximate
posterior distribution
close to prior



Training


Putting it all together: maximizing the likelihood lower bound
$\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))$
$\mathcal{L}(x^{(i)}, \theta, \phi)$
Maximize likelihood of original input being reconstructed
Make approximate posterior distribution close to prior
$\hat{x}$
Sample x|z from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$
$\mu_{x|z}$
$\Sigma_{x|z}$
Decoder network
$p_\theta(x|z)$
z
Sample z from $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$
$\mu_{z|x}$
$\Sigma_{z|x}$
Encoder network
$q_\phi(z|x)$
Input Data
x
Fei-Fei Li & Justin Johnson & Serena Yeung
Lecture 13 - 89
May 18, 2017

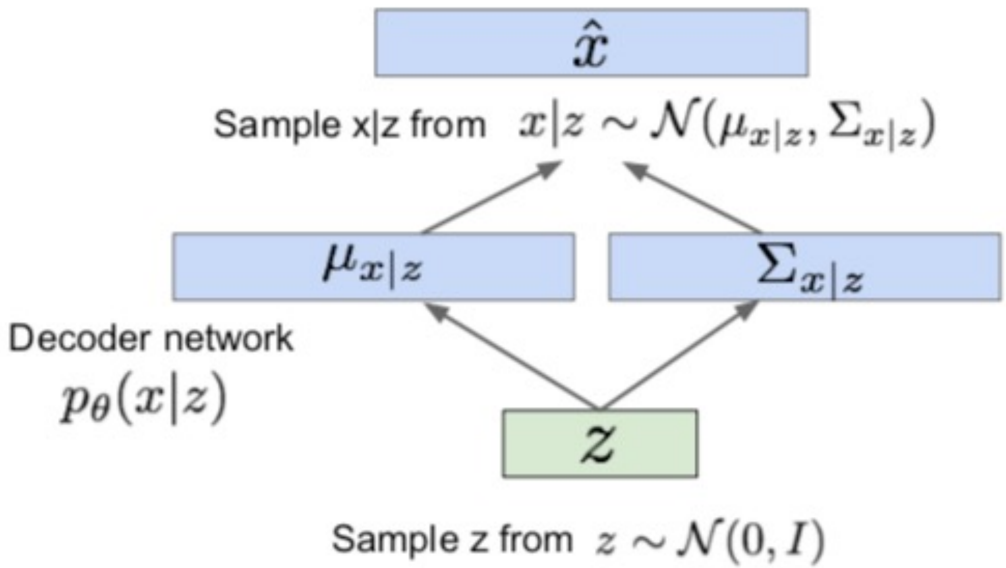
Training



$\hat{x}$
Putting it all together: maximizing the likelihood lower bound
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$
Maximize likelihood of original input being reconstructed
Sample x|z from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$
$\mu_{x|z}$
$\Sigma_{x|z}$
Decoder network
$p_\theta(x|z)$
z
Sample z from $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$
Make approximate posterior distribution close to prior
$\mu_{z|x}$
$\Sigma_{z|x}$
Encoder network
$q_\phi(z|x)$
For every minibatch of input data: compute this forward pass, and then backprop!
Input Data
x

Generating data

1. Sample z from prior
2. Use decoder network
3. Sample from Gaussian posterior



Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

VAEs results

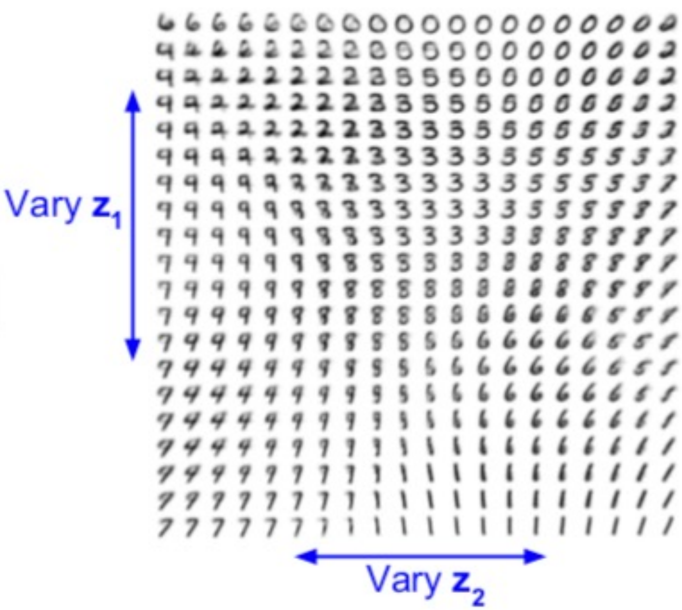
32x32 CIFAR-10

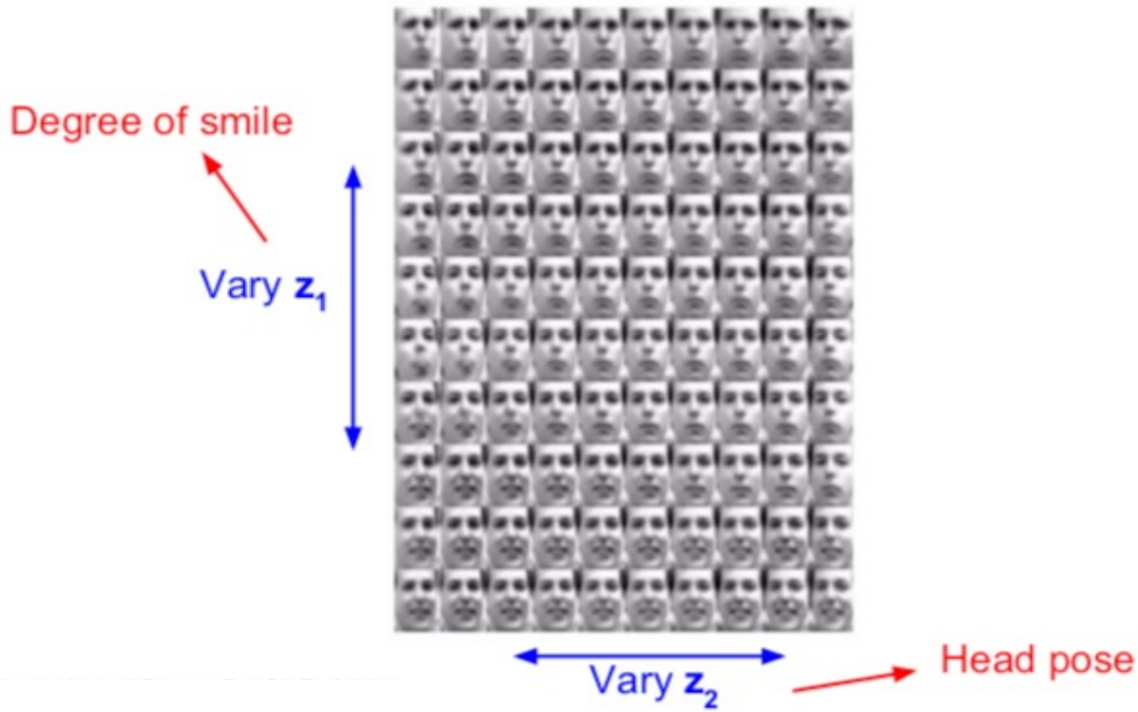
Labeled Faces in the Wild

Figures copyright (L) Dirk Kingma et al. 2016; (R) Anders Larsen et al. 2017. Reproduced with permission.

Latent representations

- z contains independent factors!



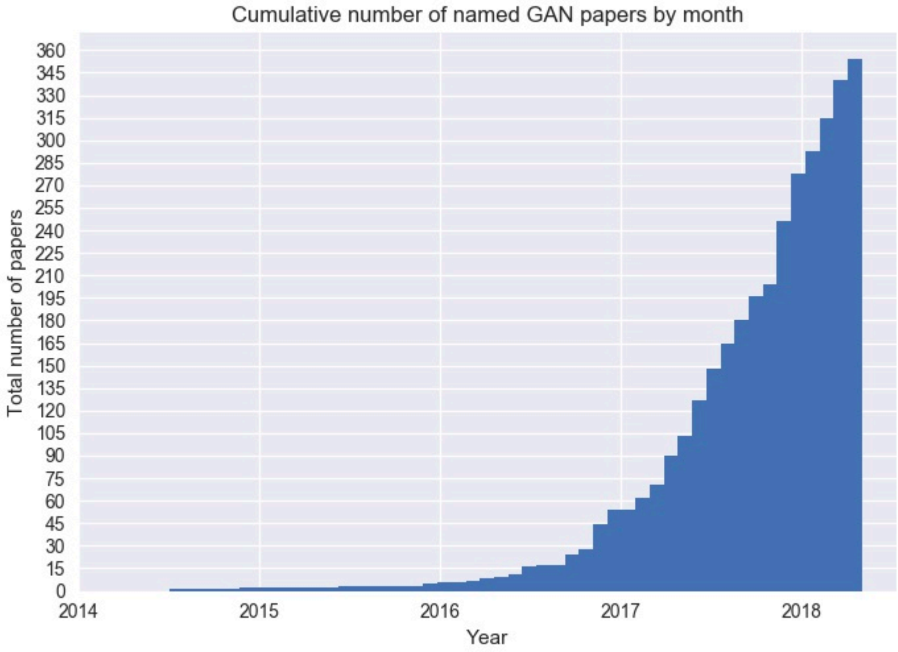


Outline

- Generative models
- Variational autoencoders (VAEs)
- Generative adversarial networks (GANs)

GANs taking over

Track updates at the GAN Zoo



https://github.com/hindupuravinash/the-gan-zoo

(Goodfellow 2018)

Generative adversarial networks

- Main idea
  - In contrast with other models (e.g., VAEs), it does not provide explicit information about data distribution (i.e., density function)
  - Generate samples only
- How?
  - Define an adversarial two-player game between a **generator** and a **discriminator**

Generator

Learns a transformation from a simple noise distribution to training distribution (e.g., images)

Training samples

z

Input noise vector

Generator

G_{θ_G}

Synthetic sample

Discriminator

Distinguish between real and fake images

Real

Synthetic

Discriminator

D_{θ_D}

REAL (1)
or
FAKE (0)?

Training GANs

- Train jointly the generator and the discriminator in a minimax game
  - Generator: try to fool the discriminator by generating real-looking images
  - Discriminator: try to distinguish between real and generated images

Training GANs

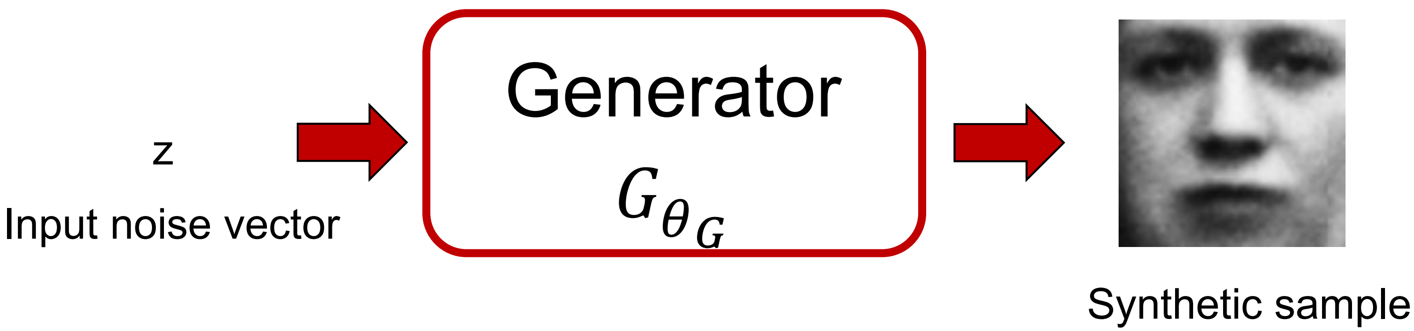
- Minimax objective function

$$\min_{\theta_G} \max_{\theta_D} E_{p_{train}} \log D_{\theta_D}(x) + E_{p\,(z)} \log(1 - D_{\theta_D}\left(G_{\theta_G}(z)\right))$$

  - **Discriminator**: maximize the objective so that D(x) is close to 1 (real) and D(G(z)) close to 0 (fake)
  - **Generator**: minimize the objective so that D(G(z)) is close to 1 (the discriminator is fooled)
  - Solved via alternating gradient optimization

Generating data

- After training, just use the generator to generate new samples

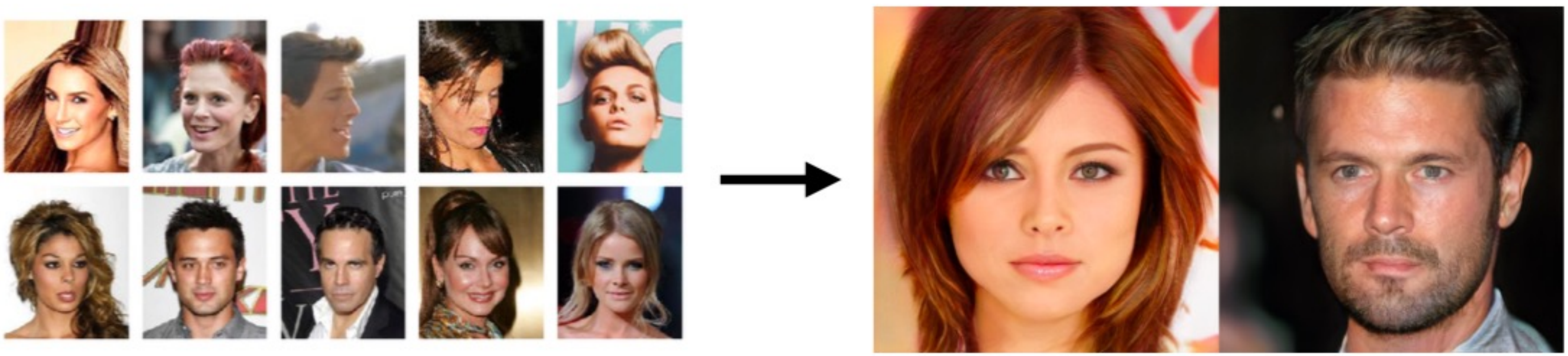


- Useful side-effect:
  - Discriminator can be used for transfer learning (why?)

GANs architectures

- Image GANs mainly use CNNs
- Generator
  - Upsampling + convolutional layers: similar to decoder in AEs
- Discriminator
  - Similar to classification CNN

GANs results

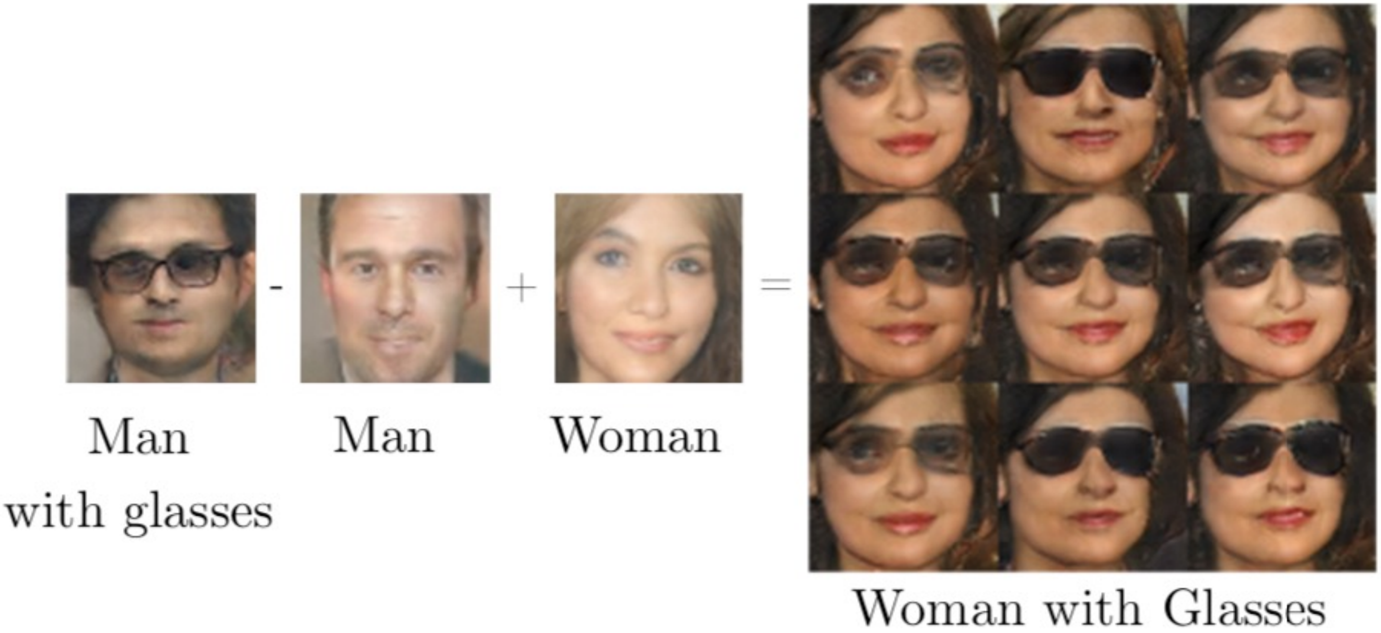


Training Data (CelebA) | Sample Generator (Karras et al, 2017)

Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196. 2017 Oct 27.

Latent representations

- Interpretable vector math



Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. 2015 Nov 19.

Generative models: VAEs vs GANs

- VAEs
  - ☺ Principled (max likelihood) approach
  - ☺ Provide explicit distributions
  - ☹ Provide blurry samples
- GANs
  - ☺ Beautiful, state-of-the-art samples
  - ☹ Trickier to train (unstable)
  - ☹ Do not provide explicit distributions

Resources

- F.F. Li, J. Johnson, S. Young. Convolutional Neural Networks for Visual Recognition, Stanford University, 2017
  - Lecture 13- ”Generative models”
  - http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
- I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Cambridge: MIT press, 2016.
  - Chapter 14 – “Autoencoders”
  - Chapter 20 – “Deep Generative Models”