Generative Models

CMPT 726
Mo Chen
SFU Computing Science
23 Nov., 2020

Outline

Outline

Generative Models

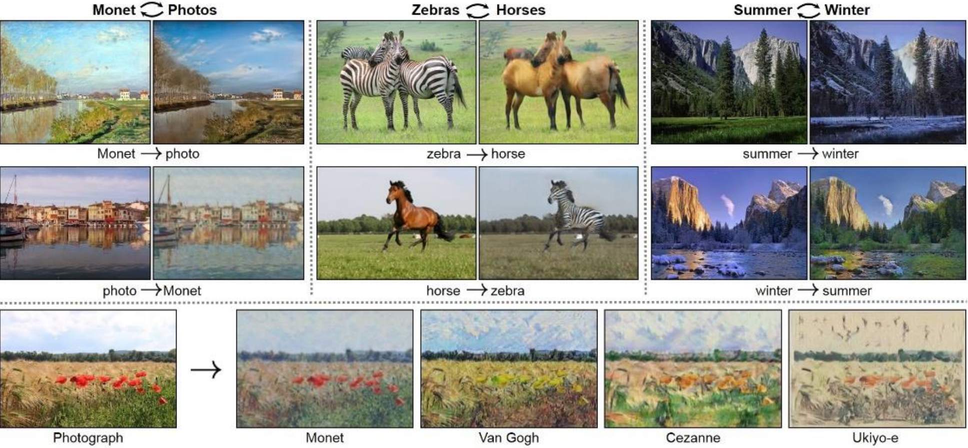
- Start with training data, with **unknown** distribution $p(x)$
  - No labels!

- Generate new samples from a similar distribution $\hat{p}(x)$
  - $\hat{p}(x)$ is learned from data

Generative Models

https://arxiv.org/abs/1406.5298

Generative Models

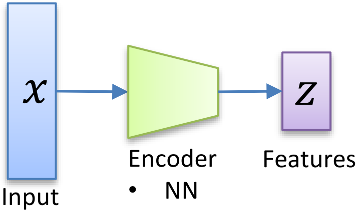


https://junyanz.github.io/CycleGAN/

Outline

Autoencoders

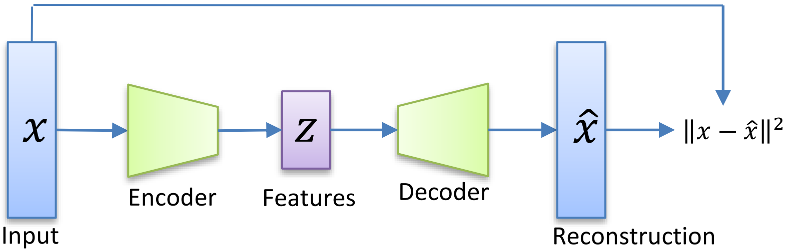
- Unsupervised learning method
    - Learns lower-dimensional features from unlabeled data
    - Labels can be expensive!



- Features z
    - Lower-dimensional
    - Can be useful for other tasks
    - How to learn?

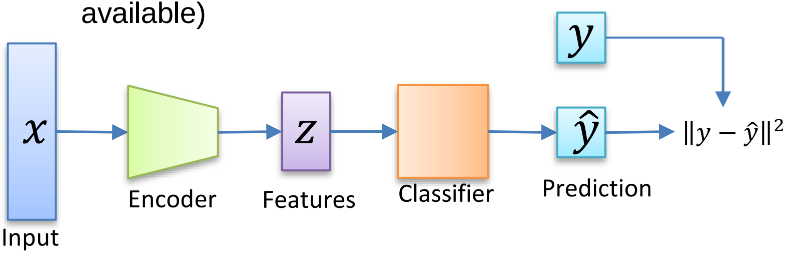
Autoencoders

- Unsupervised learning method
  - Learn features that allow reconstruction of original data
  - Reconstruction (decoding) done by a decoder
    - Decoder: another neural network
  - Loss function: $||x - \hat{x}||^2$



After Training

- Replace decoder with another network (e.g. classifier)
    - Supervised learning can be more efficient when features are "pre-learned"
    - Example: train autoencoder on ImageNet dataset, and then train only classifier on bird classification (less data available)



Outline

Generative Models

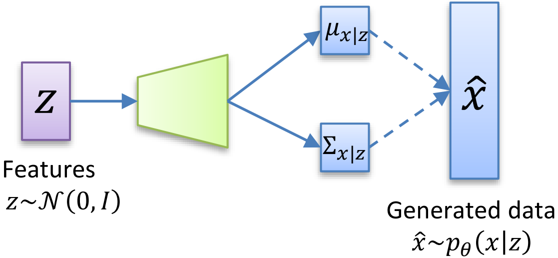
Autoencoders

Variational Auto Encoders

Generative Adversarial Networks

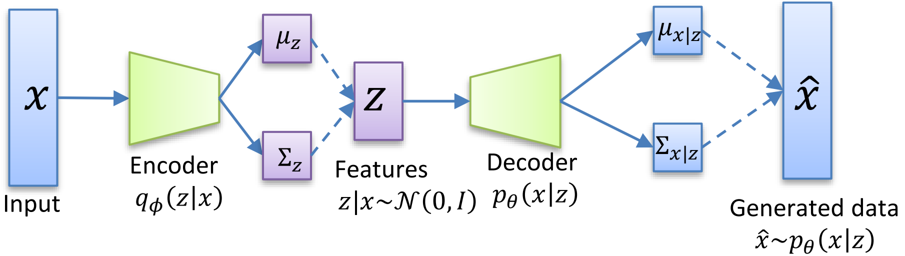
Variational Autoencoders

- Make autoencoders probabilistic to allows us to generate a variety of new data
- New data generated from features z
    - Feature variations: $z \sim p_\theta(z) = \mathcal{N}(0, I)$ for simplicity
    - Synthetic data variations: $\hat{x} \sim p_\theta(x|z)$, represented by a neural network



Variational Autoencoders: Training

- We still need to learn features that allow reconstruction of original data
  - This time, since there is variation in z and x, we cannot simply minimize $||x - \hat{x}||^2$
- Instead, maximize the likelihood $p_\theta(\hat{x})$ that a training image will be generated
  - fit a distribution $p_\theta(\hat{x})$ to data, which has distribution $p(x)$



ELBO Loss

- How to maximize $p_\theta(x^{(i)})$?
  - Data sample $x^{(i)}$ is given; we are finding θ

$$\log p_\theta(x^{(i)}) = \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (\mathbb{E}_b[a] = a \text{ if } a \text{ does not depend on } b)$$

$$= \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}\right] \qquad (\text{Bayes' rule: } p(a|b) = \frac{p(b|a)p(a)}{p(b)})$$

$$= \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log \frac{p_\theta(x^{(i)}|z)p_\theta(z)}{p_\theta(z|x^{(i)})}\frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})}\right]$$

$$= \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right] - \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)}\right] + \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})}\right]$$

$$= \mathbb{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right] - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)})\|p_\theta(z)\right) + D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)})\|p_\theta(z|x^{(i)})\right)$$

- D_{KL} is the Kullback-Leibler divergence
  - a measure of closeness of probability distributions
  - Always ≥ 0

ELBO Loss

- D_{KL} is the Kullback-Leibler divergence
  - a measure of closeness of probability distributions
  - Always ≥ 0

$$\log p_\theta\left(x^{(i)}\right) = \mathbb{E}_{z\sim q_\phi\left(z|x^{(i)}\right)}\left[\log p_\theta\left(x^{(i)}|z\right)\right] - D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right)||p_\theta(z)\right) + D_{\mathrm{KL}}\left(q_\phi\left(z|x^{(i)}\right)||p_\theta\left(z|x^{(i)}\right)\right)$$

- First term: Estimate by sampling from decoder (yet more Math tricks, see [1])
- Second term: Closed-form expression if we assume $p_\theta(z)$ is Gaussian
- Third term: intractable to compute but always ≥ 0, so ignore…
- Maximize first two terms, the evidence lower bound (ELBO)

[1] Kingma, Welling 2013. "*Auto-Encoding Variational Bayes*." http://arxiv.org/abs/1312.6114

ELBO Loss

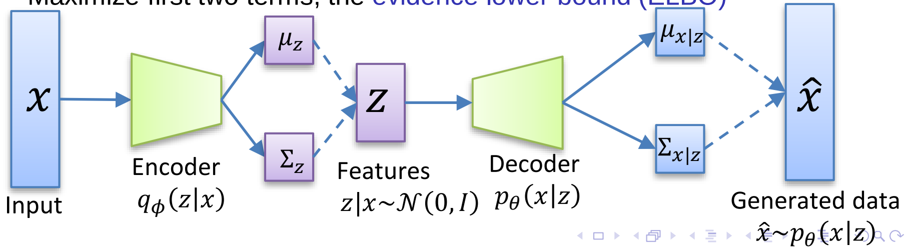
$$l(x^{(i)}, \theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right] - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)}) \| p_\theta(z)\right)$$

- First term: Estimate by sampling from decoder (yet more Math tricks, see [1])
  - Maximize data likelihood, given latent features
- Second term: Closed-form expression if we assume $p_\theta(z)$ is Gaussian
  - Make encoder $q_\phi(z|x^{(i)})$ close to $p_\theta(z) = \mathcal{N}(0, I)$
- Maximize first two terms, the evidence lower bound (ELBO)

ELBO Loss

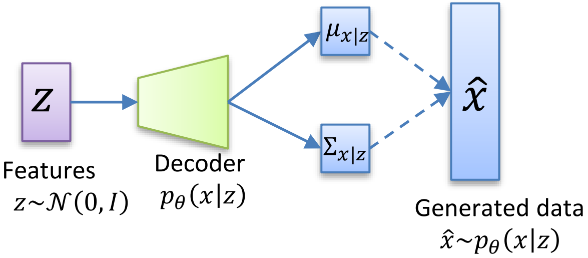
$$l(x^{(i)}, \theta, \phi) = \mathbb{E}_{z \sim q_\phi(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - D_{\mathrm{KL}}\left(q_\phi(z|x^{(i)}) \| p_\theta(z)\right)$$

- First term: Estimate by sampling from decoder (yet more Math tricks, see [1])
  - Maximize data likelihood, given latent features
- Second term: Closed-form expression if we assume $p_\theta(z)$ is Gaussian
  - Make encoder $q_\phi(z|x^{(i)})$ close to $p_\theta(z) = \mathcal{N}(0, I)$
- Maximize first two terms, the evidence lower bound (ELBO)

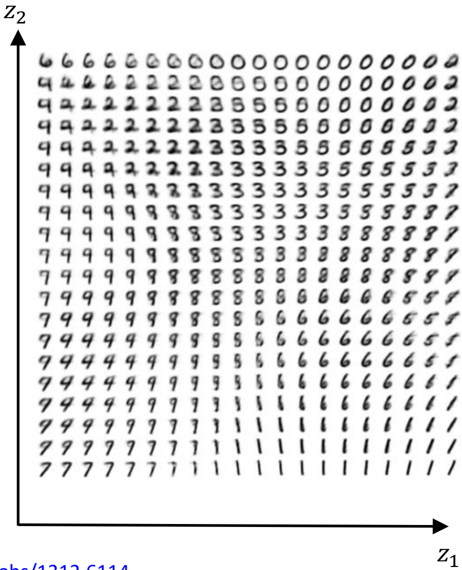


Test Time

- Throw away encoder
- Sample $z \sim \mathcal{N}(0, I)$
- Compute $p_\theta(x|z)$ using decoder
- Sample from decoder to generate new data $\hat{x}$



Test Time



"Auto-Encoding Variational Bayes." http://arxiv.org/abs/1312.6114

Outline

Generative Models

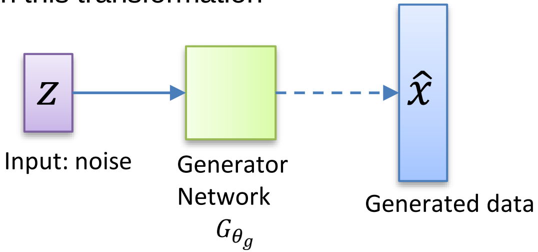
Autoencoders

Variational Auto Encoders

Generative Adversarial Networks

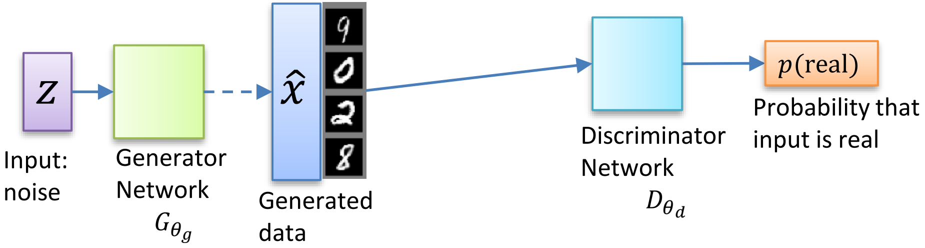
Generative Adversarial Networks (GANs)

- No explicit representation of probability densities (mean and variance)
- Data generation can be thought of as sampling from data distribution a complex distribution
    - Sample from simple distribution (noise)
    - Transform the samples so that they become distributed according to a more complex distribution
- GANs: Learn this transformation



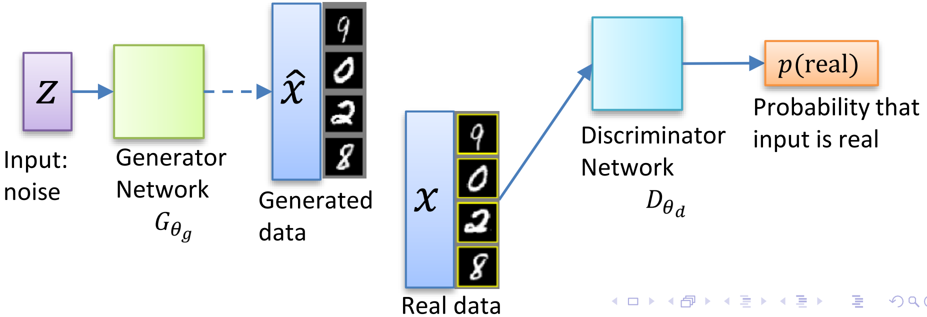
Generative Adversarial Networks (GANs)

- Generator network needs to output realistic images, i.e. those from the same distribution as some dataset
    - But need to learning the transformation *without labels*
    - Idea: learn to fool a discriminator network
- Discriminator network: discriminate between real and fake data
    - "Labels": real or not real



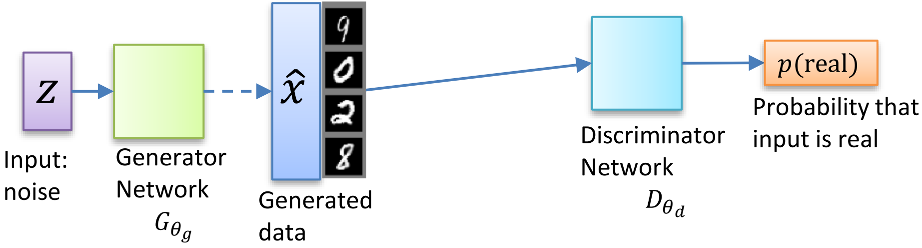
Generative Adversarial Networks (GANs)

- Generator network needs to output realistic images, i.e. those from the same distribution as some dataset
  - But need to learning the transformation *without labels*
  - Idea: learn to fool a discriminator network
- Discriminator network: discriminate between real and fake data
  - “Labels”: real or not real



Discriminator Loss Function

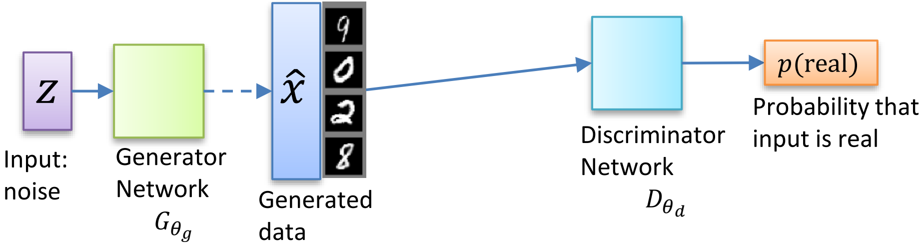
- Maximize $l_d(x^{(i)}, z^{(i)}; \theta_d) = \log D_{\theta_d}(x^{(i)}) + \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(z^{(i)})\right)\right)$
  - Fix generator parameters θ_g
- If the input is real data $x^{(i)}$, output something close to 1
- If the input is fake data $\hat{x}^{(i)}$, output something close to 0
  - Fake data is from generator: $\hat{x}^{(i)} = G_{\theta_g}(z^{(i)})$



Generator Loss Function

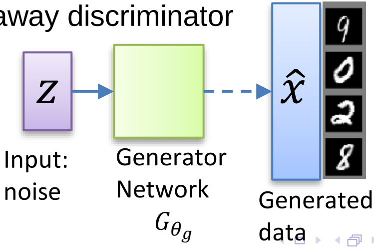
- Maximize $l_g(z^{(i)};\theta_g) = \log\left(D_{\theta_d}(G_{\theta_g}(z^{(i)}))\right)$
    - Fix discriminator parameters θ_d

- Try to make discriminator output 1 when taking generated data as input

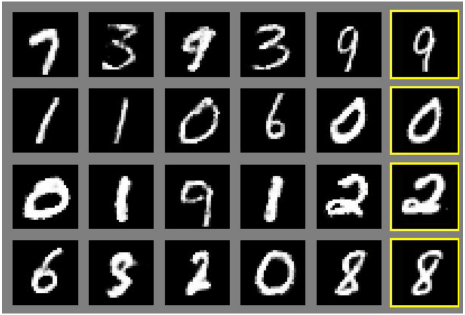


GANs: Training and Using

- For every training iteration
  - Perform k steps of discriminator updates
    - Sample $\{z^{(i)}\}_{i=1}^{M}$ (random noise), $\{x^{(i)}\}_{i=1}^{M}$ (from data)
    - Take ascent step in the $\nabla_{\theta_d} \sum_{i=1}^{M} l_d\left(x^{(i)}, z^{(i)}; \theta_d\right)$ direction
  - Perform a step of generator update
    - Sample $\{z^{(i)}\}_{i=1}^{M}$ (random noise)
    - Take ascent step in the $\nabla_{\theta_g} \sum_{i=1}^{M} l_g\left(z^{(i)}; \theta_g\right)$ direction
- After training, throw away discriminator

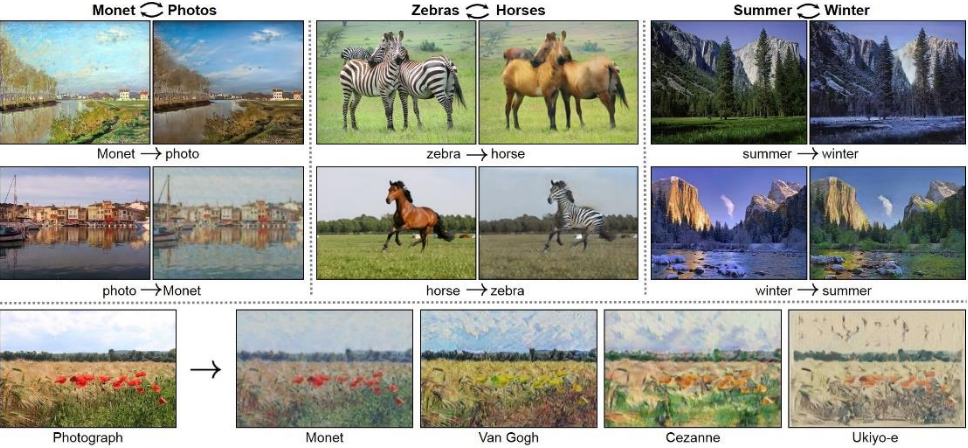


Generative Models



Goodfellow et al., 2014. "*Generative Adversarial Networks*."
http://arxiv.org/abs/1406.2661

Generative Models



https://junyanz.github.io/CycleGAN/