Variational autoencoders (VAEs)
Outline

• Basic VAE formulation
• Highlights of recent work
Recall: GANs

- **Training:**
  - Discriminator: low scores for fake data, high scores for real data
  - Generator: increase discriminator score on fake data
- **Test time:** discard discriminator and use generator to sample from learned distribution
Variational autoencoders: Overview

- Probabilistic formulation based on variational Bayes framework
- At training time, jointly learn encoder and decoder by maximizing (a bound on) the data likelihood
- At test time, discard encoder and use decoder to sample from the learned distribution

Variational autoencoders: Overview

- Probabilistic generative model of the data distribution:

  - Sample \( z \) from prior \( p(z) \) (usually Gaussian)
  - Sample \( x \) from conditional \( p(x|z) \)
  - Try to approximate the conditional with neural network
Variational autoencoders: Overview

- At training time, jointly learn encoder and decoder
- **Encoder**: given inputs $x$, output $q_\phi(z \mid x)$
  - Specifically, output mean and (diagonal) covariance, or $\mu_{z \mid x}$ and $\Sigma_{z \mid x}$, so that $q_\phi(z \mid x) = N(\mu_{z \mid x}, \Sigma_{z \mid x})$
- **Decoder**: given $z$, output $p_\theta(x \mid z)$
  - Specifically, output $\mu_{x \mid z}$ and $\Sigma_{x \mid z}$ so that $p_\theta(x \mid z) = N(\mu_{x \mid z}, \Sigma_{x \mid z})$
- **Training objective**: (a bound on) data likelihood under the model
Variational autoencoders: Overview

- At test time, discard encoder and use decoder to sample from $p_\theta(x \mid z) = N(\mu_{x|z}, \Sigma_{x|z})$
Variational autoencoders: Training

- Objective: maximize the variational lower bound on the data likelihood:

$$\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)$$

Adapted from J. Johnson
Variational autoencoders: Training

• Objective: maximize the variational lower bound on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))
\]

1. Run training point \(x\) through encoder to get a distribution over latent codes \(z\)

Adapted from J. Johnson
Variational autoencoders: Training

- Objective: maximize the variational lower bound on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))
\]

1. Run training point \(x\) through encoder to get a distribution over latent codes \(z\)
2. Encoder output should match the prior \(p(z)\)
   - Closed form solution when \(q_\phi\) is diagonal Gaussian and \(p\) is unit Gaussian (Assume \(z\) has dimension \(J\)):
     \[
     -D_{KL}(q_\phi(z|x), p(z)) = \sum_{j=1}^{J} \left( 1 + \log \left( \Sigma_{z|x} \right)_j^2 \right) - \left( \mu_{z|x} \right)_j^2 - \left( \Sigma_{z|x} \right)_j^2
     \]

Adapted from J. Johnson
Variational autoencoders: Training

- Objective: maximize the *variational lower bound* on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))
\]

1. Run training point \( x \) through encoder to get a distribution over latent codes \( z \)
2. Encoder output should match the prior \( p(z) \)
3. Sample code \( z \) from encoder output

Adapted from J. Johnson
Variational autoencoders: Training

- Objective: maximize the \textit{variational lower bound} on the data likelihood:

\[
\log p_{\theta}(x) \geq \mathbb{E}_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}(q_{\phi}(z|x), p(z))
\]

1. Run training point \( x \) through encoder to get a distribution over latent codes \( z \)
2. Encoder output should match the prior \( p(z) \)
3. Sample code \( z \) from encoder output
4. Run sampled \( z \) through decoder to get a distribution over data samples

Adapted from J. Johnson
Variational autoencoders: Training

- Objective: maximize the *variational lower bound* on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))
\]

1. Run training point \( x \) through encoder to get a distribution over latent codes \( z \)
2. Encoder output should match the prior \( p(z) \)
3. Sample code \( z \) from encoder output
4. Run sampled \( z \) through decoder to get a distribution over data samples
5. Original input should be likely under the distribution output in (4)

Adapted from J. Johnson
Variational autoencoders: Training

- Objective: maximize the \textit{variational lower bound} on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x), p(z)\right)
\]

- Data likelihood
- Regularization
Variational autoencoders: Training

- Objective: maximize the \textit{variational lower bound} on the data likelihood:

\[
\log p_\theta(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z))
\]

\begin{itemize}
\item Data likelihood
\item Regularization
\end{itemize}

- Objective for the entire dataset:

\[
\mathbb{E}_{x \sim D} \left[ \mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x), p(z)) \right]
\]

For further details, see: C. Doersch, \textit{Tutorial on Variational Autoencoders}, 2016
Original results

- Learned 2D manifolds:

Variational autoencoders: Generating data

Image source
Basic VAE framework: Summary

• **Pros:**
  • Principled mathematical formalism for generative models
  • Allows inference of code given image, can be useful for controlling the latent space

• **Cons:**
  • Samples blurrier and lower quality compared to GANs

• **Active areas of research:**
  • More powerful and flexible approximations for relevant probability distributions
  • Combining VAEs and GANs
  • Incorporating structure in latent variables, e.g., hierarchical or categorical distributions

Adapted from J. Johnson
Combining VAEs and GANs

- Define decoder probability model $p_\theta(x|z)$ not in terms of reconstruction errors in pixel space, but in terms of errors in discriminator feature space.
Combining VAEs and GANs: BicycleGANs

*Toward Multimodal Image-to-Image Translation*, NIPS 2017
Combining VAEs and GANs: BicycleGANs

- Key ideas:
  - Image-to-image translation is a *one-to-many* problem. Need to model conditional distribution of output given input parametrized by $z$.
  - To prevent mode collapse (or many-to-one mapping from $z$ to output), need to encourage the mapping between output and latent code to be invertible.
  - Propose BicycleGAN framework to simultaneously learn mappings in both directions.

Combining VAEs and GANs: BicycleGANs

Generating better samples: VQ-VAE-2

- Combining VAE and autoregressive models:

  Train a VAE-like model to generate multiscale grids of latent codes

  Use a multiscale autoregressive model (PixelCNN) to sample in latent code space

Generating better samples: VQ-VAE-2

- 256 x 256 class-conditional samples, trained on ImageNet:
Generating better samples: VQ-VAE-2

- 256 x 256 class-conditional samples, trained on ImageNet:

  - Redshank
  - Pekinese
  - Papillon
  - Drake
  - Spotted Salamander
Generating better samples: VQ-VAE-2

- 1024 x 1024 generated faces, trained on FFHQ:
Generating better samples: Hierarchical VAE

Figure 1: 256×256-pixel samples generated by NVAE, trained on CelebA HQ [28].