

Outline

- Generative Models Basics
- Autoregressive Models
- **Autoencoder and Variational Autoencoder**
- Generative Adversarial Network
- Diffusion Models

So far...

PixelCNNs define tractable density function, optimize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^n p_{\theta}(x_i|x_1, \dots, x_{i-1})$$

VAEs define intractable density function with latent \mathbf{z} :

$$p_{\theta}(x) = \int p_{\theta}(\mathbf{z})p_{\theta}(x|\mathbf{z})d\mathbf{z}$$

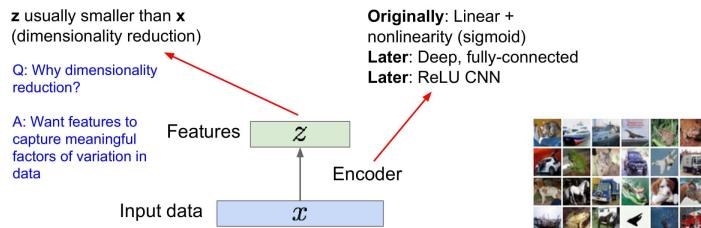
Cannot optimize directly, derive and optimize lower bound on likelihood instead

31

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Some background first: Autoencoders

Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



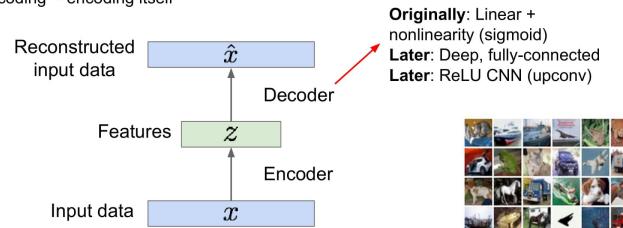
37

Some background first: Autoencoders

How to learn this feature representation?

Train such that features can be used to reconstruct original data

"Autoencoding" - encoding itself

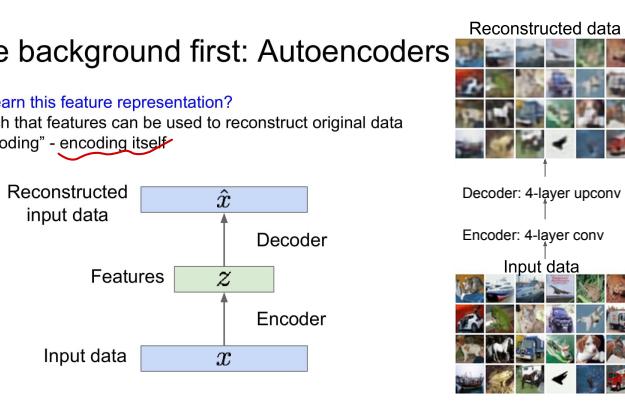


40

Some background first: Autoencoders

How to learn this feature representation?

Train such that features can be used to reconstruct original data
 "Autoencoding" - encoding itself



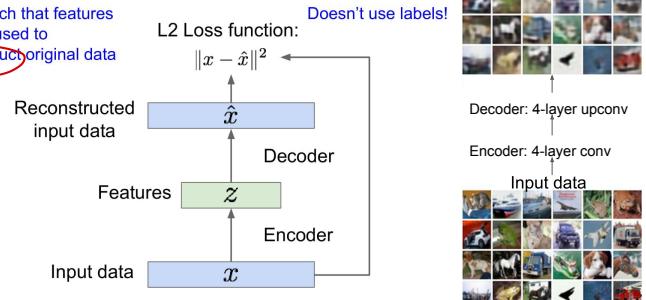
41

Some background first: Autoencoders

Train such that features can be used to reconstruct original data

Doesn't use labels!

L2 Loss function: $\|x - \hat{x}\|^2$

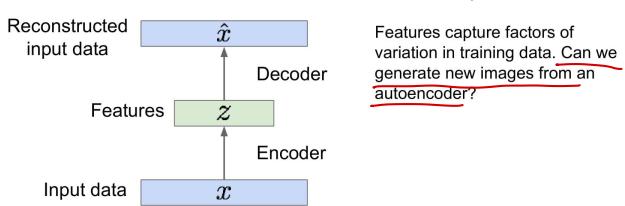


43

Some background first: Autoencoders

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?

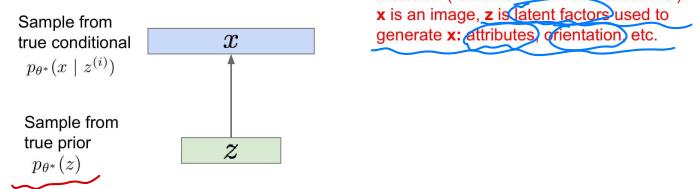


45

Variational Autoencoders

Probabilistic spin on autoencoders - will let us sample from the model to generate data!

Assume training data $\{\mathbf{x}^{(i)}\}_{i=1}^N$ is generated from underlying unobserved (latent) representation \mathbf{z}

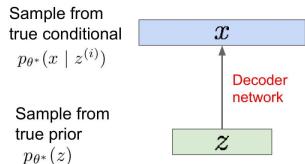


Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

48

Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders



We want to estimate the true parameters θ^* of this generative model.

How should we represent this model?

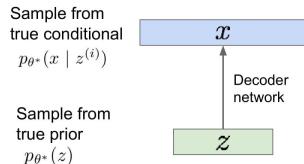
Choose prior $p(z)$ to be simple, e.g. Gaussian.

Conditional $p(x|z)$ is complex (generates image) => represent with neural network

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

52 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders



We want to estimate the true parameters θ^* of this generative model.

How to train the model?

Learn model parameters to maximize likelihood of training data

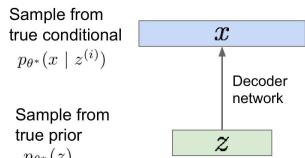
$$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$$

Now with latent z

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

55 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders



We want to estimate the true parameters θ^* of this generative model.

How to train the model?

Learn model parameters to maximize likelihood of training data

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

Q: What is the problem with this?

Intractable!

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

57 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders: Intractability

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

Intractable to compute $p(x|z)$ for every z !

decoder NN, defined Gaussian

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

61 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders: Intractability

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

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

Intractable data likelihood

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

63 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders: Intractability

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

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

Solution: In addition to decoder network modeling $p_{\theta}(x|z)$, define additional encoder network $q_{\phi}(z|x)$ that approximates $p_{\theta}(z|x)$

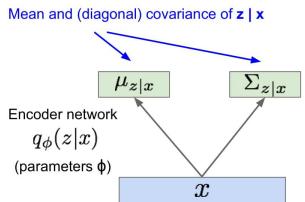
Will see that this allows us to derive a lower bound on the data likelihood that is tractable, which we can optimize

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

64 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Since we're modeling probabilistic generation of data, encoder and decoder networks are probabilistic

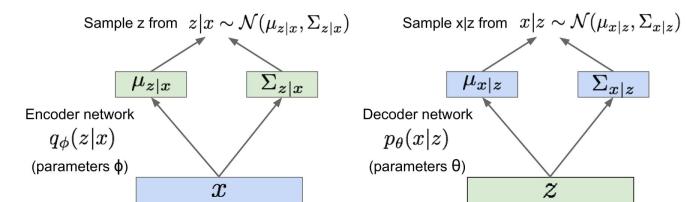


Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

65 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Since we're modeling probabilistic generation of data, encoder and decoder networks are probabilistic



Encoder and decoder networks also called "recognition"/"inference" and "generation" networks

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

67 Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z \sim q_\phi(z|x^{(i)})} [\log p_\theta(x^{(i)})] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

Taking expectation wrt. z
(using encoder network) will
come in handy later

69
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

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

The expectation wrt. z (using
encoder network) let us write
nice KL terms

74
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

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

Decoder network gives $p_\theta(x|z)$, can
compute estimate of this term through
sampling. (Sampling differentiable
through reparam. trick, see paper.)

This KL term (between
Gaussians for encoder and z
prior) has nice closed-form
solution!

$p_\theta(z|x)$ intractable (saw
earlier), can't compute this KL
term! But we know KL
divergence always ≥ 0 .

75
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

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

Tractable lower bound which we can take
gradient of and optimize! ($p_\theta(x|z)$ differentiable,
KL term differentiable)

76
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

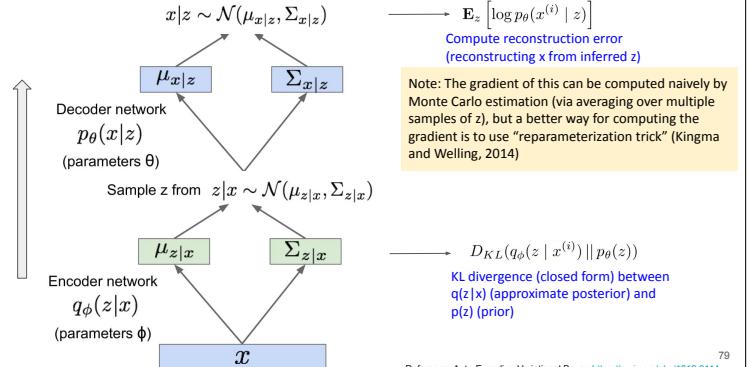
$$\begin{aligned} \log p_\theta(x^{(i)}) &= \mathbf{E}_{z \sim q_\phi(z|x^{(i)})} [\log p_\theta(x^{(i)})] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\ &= \mathbf{E}_z \left[\log \frac{p_\theta(x^{(i)} | z)p_\theta(z)}{p_\theta(z | x^{(i)})} \right] \quad (\text{Bayes' Rule}) \\ &= \mathbf{E}_z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

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

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

78
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

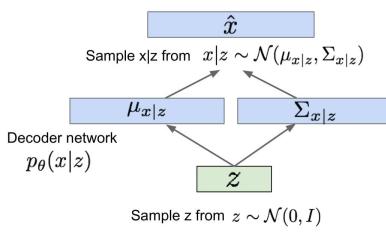
Training objective of VAE: ELBO



79
Reference: Auto-Encoding Variational Bayes <https://arxiv.org/abs/1312.6114>

Variational Autoencoders: Generating Data!

Use decoder network. Now sample z from prior!



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

81
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoders: Generating Data!

Diagonal prior on z
=> independent
latent variables

Different
dimensions of z
encode
interpretable factors
of variation

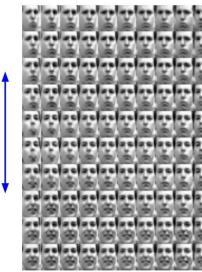
Also good feature representation that
can be computed using $q_\phi(z|x)$!

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

Degree of smile

Vary z_1

Vary z_2



83
Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

Variational Autoencoder: Summary

Probabilistic spin to traditional autoencoders => allows generating data
Defines an intractable density => derive and optimize a (variational) lower bound

Pros:

- Principled approach to generative models
- Allows inference of $q(z|x)$, can be useful feature representation for other tasks

Cons:

- Maximizes lower bound of likelihood: okay, but not as good evaluation as PixelRNN/PixelCNN
- Samples blurrier and lower quality compared to state-of-the-art (GANs)

Further Extensions:

- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian, e.g., Gaussian Mixture Models (GMMs)
- Incorporating structure in latent variables, e.g., Categorical Distributions

84
Slide credit: Justin Johnson