Deep generative models of natural images

Emily Denton

Spring 2016
1 Motivation

2 Deep generative models: Intro

3 Deep generative models: Recent algorithms
   - Variational autoencoders
   - Generative adversarial networks
   - Generative moment matching networks
   - Evaluating generative models

4 Extensions
Outline

1. Motivation
2. Deep generative models: Intro
3. Deep generative models: Recent algorithms
   - Variational autoencoders
   - Generative adversarial networks
   - Generative moment matching networks
   - Evaluating generative models
4. Extensions
Generative models

- Have access to $x \sim p_{data}(x)$ through training set
- Want to learn a model $x \sim p_{model}(x)$
- Want $p_{model}$ to be similar to $p_{data}$
  - Samples drawn from $p_{model}$ reflect structure of $p_{data}$
  - Samples from true data distribution have high likelihood under $p_{model}$
Why do generative modeling?

- Unsupervised representation learning
  - Can transfer learned representation so discriminative tasks, retrieval, clustering, etc.

- Train network with both discriminative and generative criterion
  - Utilize unlabeled data, regularize

- Understand data

- Density estimation

- Data augmentation

- ...
Focus of this talk

Generative modeling is a HUGE field...I will focus on (a selected set of) deep directed models of natural images
Outline

1. Motivation

2. Deep generative models: Intro

3. Deep generative models: Recent algorithms
   - Variational autoencoders
   - Generative adversarial networks
   - Generative moment matching networks
   - Evaluating generative models

4. Extensions
Directed graphical models

We assume data is generated by:

\[ z \sim p(z) \quad x \sim p(x|z) \]

- \( z \) is latent/hidden \( x \) is observed (image)
- Use \( \theta \) to denote parameters of the generative model
Deep directed graphical models

\[ P(x) = \sum_h P(x|h)P(h) \]

- Intractable
- Can’t optimize data likelihood directly
Bounding the marginal likelihood

Recall Jenson’s inequality: When $f$ is concave, $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$

$$\log p(x) = \log \int_z p(x, z)$$

$$= \log \int_z q(z) \frac{p(x, z)}{q(z)}$$

$$\geq \int_z q(z) \log \frac{p(x, z)}{q(z)} = L(x; \theta, \phi) \quad \text{(by Jenson's inequality)}$$

$$= \int_z q(z) \log p(x, z) - \int_z q(z) \log q(z)$$

$$= \mathbb{E}_{q(z)}[\log p(x, z)] + \mathbb{H}(q(z))$$

Expectation of joint distribution + Entropy
Evidence Lower BOund (ELBO)

- Bound is tight when variational approximation matches true posterior:

\[
\log p(x) - L(x; \theta, \phi) = \log p(x) - \int_z q(z) \log \frac{p(x, z)}{q(z)}
\]

\[
= \int_z q(z) \log p(x) - \int_z q(z) \log \frac{p(x, z)}{q(z)}
\]

\[
= \int_z q(z) \log \frac{q(z)p(x)}{p(x, z)}
\]

\[
= D_{KL}(q(z; \phi) \| p(z|x))
\]
Assume existence of $q(z; \phi)$

Bound $\log p(x; \theta)$ with $L(x; \theta, \phi)$

Bound is tight when:

$$D_{KL}(q(z; \phi) || p(z|x)) = 0 \iff q(z; \phi) = p(z|x)$$
Learning directed graphical models

- Maximize bound on likelihood of data:
  \[
  \max_{\theta} \sum_{i=1}^{N} \log p(x_i; \theta) \geq \max_{\theta,\phi_1,...,\phi_N} \sum_{i=1}^{N} L(x_i; \theta, \phi_i)
  \]

- Historically, used different \( \phi_i \) for every data point
  - But we’ll move away from this soon..

- \( q(z; \phi) \) typically factorized distribution

- For more info see Blei et al. (2003)
New method of learning: approximate inference model

- Instead of having different variational parameters for each data point, fit a conditional parametric function

- The output of this function will be the parameters of the variational distribution $q(z|x)$

- Instead of $q(z)$ we have $q_\phi(z|x)$

- ELBO becomes:

$$L(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x, z)] + H(q_\phi(z|x))$$

  Expectation of joint distribution

  Entropy
Outline

1 Motivation

2 Deep generative models: Intro

3 Deep generative models: Recent algorithms
   - Variational autoencoders
   - Generative adversarial networks
   - Generative moment matching networks
   - Evaluating generative models

4 Extensions
**Variational autoencoder**

- **Encoder** network maps from image space to latent space
  - Outputs parameters of $q_\phi(z|x)$

- **Decoder** maps from latent space back into image space
  - Outputs parameters of $p_\theta(x|z)$

[Kingma & Welling (2013)]
Rearranging the ELBO:

\[ L(x; \theta, \phi) = \int_z q_\phi(z|x) \log \int_z \frac{p(x, z)}{q_\phi(z|x)} \]

\[ = \int_z q_\phi(z|x) \log \int_z \frac{p(x|z)p(z)}{q_\phi(z|x)} \]

\[ = \int_z q_\phi(z|x) \log p(x|z) + \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} \]

\[ = \mathbb{E}_{q(z|x)} \log p(x|z) - \mathbb{E}_{q(z|x)} \log \frac{q(z|x)}{p(z)} \]

\[ = \underbrace{\mathbb{E}_{q(z|x)} \log p(x|z)}_{\text{Reconstruction term}} - D_{KL}(q(z|x)\|p(z)) \]
Variational autoencoder

- Inference network outputs parameters of $q_{\phi}(z|x)$
- Generative network outputs parameters of $p_{\theta}(x|z)$
- Optimize $\theta$ and $\phi$ jointly by maximizing ELBO:

$$L(x; \theta, \phi) = \mathbb{E}_{q(z|x)} \log p(x|z) - D_{KL}(q(z|x)||p(z))$$

- Reconstruction term
- Prior term
Stochastic gradient variation bayes (SGVB) estimator

- Reparameterization trick: re-parameterize $z \sim q_\phi(h|z)$ as
  
  $$z = g_\phi(x, \epsilon) \text{ with } \epsilon \sim p(\epsilon)$$

- For example, with a Gaussian can write $z \sim \mathcal{N}(\mu, \sigma^2)$ as
  
  $$z = \mu + \epsilon \sigma^2 \text{ with } \epsilon \sim \mathcal{N}(0, 1)$$

[Kingma & Welling (2013); Rezende et al. (2014)]
Stochastic gradient variation bayes (SGVB) estimator

\[ L(x; \theta, \phi) = \mathbb{E}_{q(z|x)} \log p(x|z) - D_{KL}(q(z|x) \| p(z)) \]

- **Reconstruction term**
- **Prior term**

- Using reparameterization trick we form Monte Carlo estimate of reconstruction term:

\[
\mathbb{E}_{q\phi(z|x)} \log p_\theta(x|z) = \mathbb{E}_{p(\epsilon)} \log p_\theta(x|g_\phi(x, \epsilon))
\]

\[
\approx \frac{1}{L} \sum_{i=1}^{L} \log p_\theta(x|g_\phi(x, \epsilon)) \quad \text{where } \epsilon \sim p(\epsilon)
\]

- KL divergence term can often be computed analytically (eg. Gaussian)
VAE learned manifold

[Kingma & Welling (2013)]
**Motivation**

Deep generative models: Intro
Deep generative models: Recent algorithms
Extensions

**Variational autoencoders**
Generative adversarial networks
Generative moment matching networks
Evaluating generative models

**VAE samples**

(a) 2-D latent space  (b) 5-D latent space  (c) 10-D latent space  (d) 20-D latent space

[Kingma & Welling (2013)]
**VAE tradeoffs**

- **Pros:**
  - Theoretically pleasing
  - Optimizes bound on likelihood
  - Easy to implement

- **Cons:**
  - Samples tend to be blurry
    - Maximum likelihood minimizes $D_{KL}(p_{data}\|p_{model})$

[Tweis et al. (2016)]
Generative adversarial networks

- Don’t focus on optimizing $p(x)$, just learn to sample
- Two networks pitted against one another:
  - Generative model $G$ captures data distribution
  - Discriminative model $D$ distinguishes between real and fake samples

[Goodfellow et al. (2014)]
Generative adversarial networks

- $D$ is trained to estimate the probability that a sample came from data distribution rather than $G$

- $G$ is trained to maximize the probability of $D$ making a mistake

$$\min_G \max_D \mathbb{E}_{x \sim p_{data(x)}} \log D(x) + \mathbb{E}_{z \sim p_{noise(z)}} \log (1 - D(G(z)))$$
GAN samples

MNIST

CIFAR-10 (fully connected)

TFD

CIFAR-10 (convolutional)
GAN tradeoffs

- **Pros:**
  - Very powerful model
  - High quality samples

- **Cons:**
  - Tricky to train (no clear objective to track, instability)
  - Can ignore large parts of image space
    - Because approximately minimizing Jenson-Shannon divergence [Goodfellow et al. (2014); Theis et al. (2016); Huszar (2015)]

[Theis et al. (2016)]
Generative moment matching networks

- Same idea as GANs, but different optimization method
- Match moments of data and generative distributions
- Maximum mean discrepancy
  - Estimator for answering whether two samples come from same distribution
- Evaluate MMD on generated samples

[Li et al. (2015); Dziugaite et al. (2015)]
Generative moment matching networks

\[ L_{MMD^2} = \left\| \frac{1}{N} \sum_{i=1}^{N} \phi(x_i) - \frac{1}{M} \sum_{j=1}^{M} \phi(x_j) \right\|^2 \]

\[ = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \phi(x_i)^\top \phi(x_{i'}) - \frac{1}{M^2} \sum_{j=1}^{M} \sum_{j'=1}^{M} \phi(x_j)^\top \phi(x_{j'}) \]

\[ - \frac{2}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \phi(x_i)^\top \phi(x_j) \]

- Can make use of kernel trick
- If \( \phi \) is identity, then matching means
- Complex \( \phi \) can match higher order moments

Emily Denton
Deep generative models of natural images
GMMN samples

(a) GMMN MNIST samples

(b) GMMN TFD samples

(c) GMMN+AE MNIST samples

(d) GMMN+AE TFD samples

Emily Denton
Deep generative models of natural images
GMMN tradeoffs

- **Pros:**
  - Theoretically pleasing

- **Cons:**
  - Batch size very important
  - Samples aren’t great (get better when combined with autoencoder)

[Theis *et al.* (2016)]
How to evaluate a generative model?

- Log likelihood on held out data
  - Makes sense when goal is density estimation
  - Many approaches don’t have tractable likelihood or it isn’t explicitly represented
  - Have to resort to Parzen window estimates [Breuleux et al. (2009)]

- Quality of samples
  - But a lookup table of training images will succeed here...

- Best: evaluate in context of particular application

- See Theis et al. (2016) for more details
Outline

1 Motivation

2 Deep generative models: Intro

3 Deep generative models: Recent algorithms
   - Variational autoencoders
   - Generative adversarial networks
   - Generative moment matching networks
   - Evaluating generative models

4 Extensions
**DRAW: Deep Recurrent Attentive Writer**

**Basic idea:**
- Iteratively construct image
- Observe image through sequence of glimpses

- Recurrent encoder and decoder
- Optimizes variational bound
- Attention mechanism determines:
  - Input region observed by encoder
  - Output region modified by decoder

---

[Gregor et al. (2015)]
DRAW samples
Generating images from captions

- Language model: bidirectional RNN
- Image model: conditional DRAW network
- Image sharpening with Laplacian pyramid adversarial network

[Mansimov et al. (2016)]
Generating images from captions

A yellow school bus parked in a parking lot.
A red school bus parked in a parking lot.
A green school bus parked in a parking lot.
A blue school bus parked in a parking lot.

The decadent chocolate desert is on the table.
A bowl of bananas is on the table.
A vintage photo of a cat.
A vintage photo of a dog.

[Mansimov et al. (2016)]
Laplacian pyramid of adversarial networks

- Difficult to generate large images in one shot
- Break problem up into sequence of manageable steps
- Samples drawn in coarse-to-fine fashion

- Each scale is a convnet trained using GAN framework

[Denton et al. (2015)]
LAPGAN training procedure
LAPGAN samples
LAPGAN samples
Deep convolutional generative adversarial networks (DCGAN)

- Radford et al. (2016) propose several tricks to make GAN training more stable
DCGAN vector arithmetic

man with glasses

man without glasses

woman without glasses

woman with glasses
Texture synthesis with spatial LSTMs

- Two dimensional LSTM
- Sequentially predict pixels in an image conditioned on previous pixels

[Theis & Bethge (2015)]
Texture synthesis with spatial LSTMs

[Theis & Bethge (2015)]
Pixel recurrent neural networks

- Sequentially predict pixels in an image conditioned on previous pixels
- Uses spatial LSTM

[van den Oord et al. (2016)]


References II


Mansimov, Elman, Parisotto, Emilio, Ba, Jimmy, & Salakhutdinov, Ruslan. 2016. Generating Images from Captions with Attention. *In: ICLR*.


van den Oord, Aaron, Kalchbrenner, Nal, & Kavukcuoglu, Koray. 2016. Pixel Recurrent Neural Networks. In: ICML.