Topics:

- Variational Auto-Encoders (VAEs)
- Variational Inference, ELBO
Administrativia

• Project submission instructions released
  – Due: 12/03, 11:55pm
  – Last deliverable in the class
  – Can’t use late days
  – [https://www.cc.gatech.edu/classes/AY2020/cs7643_fall/](https://www.cc.gatech.edu/classes/AY2020/cs7643_fall/)
Recap from last time
Variational Autoencoders (VAE)
PixelCNNs define tractable density function, optimize likelihood of training data:

\[
p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i \mid x_1, \ldots, x_{i-1})
\]
So far...

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

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

VAEs define intractable density function with latent \( z \):

\[ \hat{p}_\theta(x) = \int p_\theta(z)p_\theta(x | z)dz \]

\[ \sum_{z} p_\theta(z)p_\theta(x | z) \quad \text{z continuous} \]

\[ \sum_{z} p_\theta(z)p_\theta(x | z) \quad \text{z discrete} \]

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
So far...

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

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

VAEs define intractable density function with latent $z$:

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

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

Gaussian Mixture Model

\( 2 \in \{1, 2\} \)

\( P(2) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \)

\( \mathcal{N}(\mu_1, \sigma_1^2) \)

\( \mu_1 = -10 \)

\( \mathcal{N}(\mu_2, \sigma_2^2) \)

\( \mu_2 = 10 \)

\( \mathcal{N}(\mu_1, \sigma_1^2) \)

\( \mu_1 = -10 \)

\( \mathcal{N}(\mu_2, \sigma_2^2) \)

\( \mu_2 = 10 \)

(C) Dhruv Batra

Figure Credit: Kevin Murphy
Gaussian Mixture Model

\[ Z \sim \text{Cat}(\pi) \]

\[ \pi_c = P(Z = c) \]

\[ X \mid Z = c \sim N(\mu_c, \sigma_c^2) \]

\[ p(\tilde{x}) = \sum_p(\tilde{z}) p(\tilde{x} \mid \tilde{z}) \]

\[ p(x, \tilde{z}) \]
Gaussian Mixture Model

\[ P(z) = \pi_z \]

\[ P(x|z) = N(\mu_z, \sigma_z^2) \]

\[ P(\mathbf{x}) = \sum_z P(z) P(x|z) = \text{Marginalization} \]

\[ P(z|\mathbf{x}) = \frac{P(z, \mathbf{x})}{P(\mathbf{x})} = \frac{P(x|z) P(z)}{\sum_z P(x|z) \pi_z} = \frac{\sum_z P(x|z) \pi_z}{\sum_z P(x|z) \pi_z} \]

"Inference"
Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders
2. Variational Approximation
   • Variational Lower Bound / ELBO
3. Amortized Inference Neural Networks
4. “Reparameterization” Trick
Autoencoders

Train such that features can be used to reconstruct original data

L2 Loss function:

Doesn't use labels!

Encoded: 4-layer conv
Decoder: 4-layer upconv

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
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?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Autoencoders

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

Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/
Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

2. Variational Approximation
   - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

4. “Reparameterization” Trick
Key problem

\[ P(z|x) = \frac{p(z, x)}{p(x)} = \frac{p(x|z) p(z)}{\sum_z p(x|z) p(z)} \]

\[ q_i(z) \quad \text{Hard} \]
What is Variational Inference?

- Key idea
  - Reality is complex
  - Can we approximate it with something “simple”?
  - Just make sure simple thing is “close” to the complex thing.
Intuition

\[ KL(p || q) = \sum_{z \in \mathcal{Z}} p(z) \log \frac{p(z)}{q(z)} \]

\[ KL(q || p) = \sum_{z \in \mathcal{Z}} q(z) \log \frac{q(z)}{p(z)} \]

\[ d(p, q) = \min_{q \in \mathcal{Q}} KL(p || q) \]

\[ \text{Support} \]

\[ \text{Error} \]
Find simple approximate distribution

- Suppose $p$ is intractable posterior
- Want to find simple $q$ that approximates $p$
- KL divergence not symmetric

- $D(p||q)$
  - true distribution $p$ defines support of diff.
  - the “correct” direction
  - will be intractable to compute

- $D(q||p)$
  - approximate distribution defines support
  - tends to give overconfident results
  - will be tractable
Example 1

- $p = 2D$ Gaussian with arbitrary co-variance ($\Sigma$)
- $q = 2D$ Gaussian with isotropic co-variance ($\sigma^2 I$)

$\text{argmin}_q \text{KL} (p \parallel q)$

$\text{argmin}_q \text{KL} (q \parallel p)$

$p = \text{Green}; q = \text{Red}$
Example 2

- $p = \text{Mixture of Two Gaussians}$
- $q = \text{Single Gaussian}$

$\arg\min_q \; KL(p \| q)$

$\arg\min_q \; KL(q \| p)$

$p = \text{Blue}; \; q = \text{Red}$

(C) Dhruv Batra
Plan for Today

• VAEs
  – Variational Inference → Evidence Based Lower Bound
  – Putting it all together

• Next time:
  – Reparameterization trick for optimizing VAEs
The general learning problem with missing data

- Marginal likelihood – \( \mathbf{x} \) is observed, \( \mathbf{z} \) is missing:

\[
ll(\theta : \mathcal{D}) = \log \prod_{i=1}^{N} P(\mathbf{x}_i | \theta) \\
= \sum_{i=1}^{N} \log P(\mathbf{x}_i | \theta) \\
= \sum_{i=1}^{N} \log \sum_{\mathbf{z}} P(\mathbf{x}_i, \mathbf{z} | \theta)
\]
Applying Jensen’s inequality

- Use: $\log \sum_z P(z) \, g(z) \geq \sum_z P(z) \, \log g(z)$
Applying Jensen’s inequality

• Use: \( \log \sum_z P(z) g(z) \geq \sum_z P(z) \log g(z) \)

\[ \begin{align*}
\log \mathbb{E}_{P(x_i, z | \theta)} \frac{P(x_i, z | \theta)}{Q_i(z)} &= \log \mathbb{E}_{P(x_i, z | \theta)} \frac{P(x_i, z | \theta)}{Q_i(z)} \\
\log \mathbb{E}_{P(x_i, z | \theta)} \frac{P(x_i, z | \theta)}{Q_i(z)} &\geq \sum_{i=1}^{N} Q_i(z_i) \log \mathbb{E}_{P(x_i, z | \theta)} \frac{P(x_i, z | \theta)}{Q_i(z_i)} \\
\end{align*} \]

“Free Energy” \( F(\theta, \psi_i) \)

Variational Lower Bound
Evidence Lower Bound (ELBO)
Evidence Lower Bound

- Define potential function $F(\theta, Q)$:

$$
l(\theta : D) \geq F(\theta, Q_i) = \sum_{i=1}^{N} \sum_{z} Q_i(z) \log \frac{P(x_i, z | \theta)}{Q_i(z)}$$
ELBO: Factorization #1 (GMMs)

\[ ll(\theta : D) \geq F(\theta, Q_i) = \sum_{i=1}^{N} \sum_{z} Q_i(z) \log \frac{P(\tilde{z}_i; \theta) P(z_i | \tilde{z}_i, \theta)}{Q_i(z)} \]

\[ = \left[ \sum_{z} Q_i(z) \right] \log \left[ P(\tilde{z}_i; \theta) \right] + \left[ \sum_{z} Q_i(z) \right] \log \left[ \frac{P(z_i | \tilde{z}_i, \theta)}{Q_i(z)} \right] \]

\[ F(\theta, Q_i) = \left[ \log P(\tilde{z}_i; \theta) \right] - \left[ \sum_{z} Q_i(z) \right] \log \left[ \frac{P(z_i | \tilde{z}_i, \theta)}{Q_i(z)} \right] \]

\[ ll(\theta) \geq F(\theta, Q_i) \]

\[ ll(\theta) = F(\theta, Q_i) + KL(Q_i || P(z_i | \tilde{z}_i, \theta)) \]

(C) Dhruv Batra
Recall: Intuition of VI

\[ \min_{q \in \mathcal{Y}} d(p, q) \]

\[ \min_{q \in \mathcal{Y}} \text{KL}(p(2) \| q(2)) \]

\[ \text{KL}(q(2) \| p(2)) \]
ELBO: Factorization #1 (GMMs)

\[
\max_{\theta, Q_i} \ln(\theta : D) \geq \mathbb{E}_{Q_i} \left[ \mathbb{E}_{Q_i} \left[ F(\theta, Q_i) = \sum_{i=1}^{N} \sum_z Q_i(z) \log \frac{P(x_i, z | \theta)}{Q_i(z)} \right] \right]
\]

- **EM** corresponds to coordinate ascent on \( F \) – Thus, maximizes lower bound on marginal log likelihood

- **E-step**: Fix \( \theta^{(t)} \), maximize \( F \) over \( Q_i \)
- **M-step**: Fix \( Q_i^{(t)} \), maximize \( F \) over \( \theta \)
EM for Learning GMMs

• Simple Update Rules

• **E-step:** Fix $\theta^{(t)}$, maximize $F$ over $Q_i$

$$Q_i^{(t)}(z) = P(z | x_i, \theta^{(t)})$$

• **M-step:** Fix $Q_i^{(t)}$, maximize $F$ over $\theta$
  
  – maximize expected likelihood under $Q_i(z)$
  
  – Corresponds to weighted dataset:

  • $<x_1, z=1>$ with weight $Q^{(t+1)}(z=1 | x_1)$
  • $<x_1, z=2>$ with weight $Q^{(t+1)}(z=2 | x_1)$
  • $<x_1, z=3>$ with weight $Q^{(t+1)}(z=3 | x_1)$
  • $<x_2, z=1>$ with weight $Q^{(t+1)}(z=1 | x_2)$
  • $<x_2, z=2>$ with weight $Q^{(t+1)}(z=2 | x_2)$
  • $<x_2, z=3>$ with weight $Q^{(t+1)}(z=3 | x_2)$
Gaussian Mixture Example: Start

\( \Theta = \{ \pi_i, \mu_i, \Sigma_i \} \)  \( k = 3 \)

\( q(\mathbf{z} | x_i) = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix} \)
After 1st iteration
After 2nd iteration
After 3rd iteration
After 4th iteration
After 5th iteration
After 6th iteration
After 20th iteration
\[ \begin{align*} \ell_2(\theta : D) & \geq \max_{\theta, Q_i} F(\theta, Q_i) = \sum_{i=1}^{N} \sum_{z} Q_i(z) \log \frac{P(x_i, z | \theta)}{Q_i(z)} \\ & = \sum_{z} \sum_{i=1}^{N} Q_i(z) \log \frac{P(x_i, z | \theta)}{Q_i(z)} = \sum_{i=1}^{N} \mathbb{E}_{Q_i(z)} \left[ \log \frac{P(x_i, z | \theta)}{Q_i(z)} \right] \\ & = \sum_{i=1}^{N} \mathbb{E}_{Q_i(z)} \left[ \log \frac{P(z_i, \theta)}{Q_i(z)} \right] = \mathcal{L}(Q_i, \theta) \end{align*} \]

"Explain the data" and "Be simple"
Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

2. Variational Approximation
   - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

4. “Reparameterization” Trick
Amortized Inference Neural Networks

\[ Q_i(z) = \begin{pmatrix} 0.3 \\ 0.3 \\ 0.3 \end{pmatrix} \]

\[ x_i \rightarrow \text{NN} \rightarrow q(\theta_i | x_i) \]

\[ z \sim \text{Cat}(\cdot) \]

\[ z \sim \mathcal{N}(\mu_{\text{encoder}}, \Sigma_{\text{encoder}}) \]

\[ z \sim \mathcal{N}(\mu_{x_i}, \Sigma_{x_i}) \]

\[ \pi \sim \mathcal{N}(\pi_{x_i}) \]
Variational Autoencoders

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

Input Image → Inference → Generative → Reconstructed Image

\[ \sqrt{\sum (x - \tilde{x})^2} \]

Latent Distribution

\[ q_\phi(z|x) \rightarrow \mu, \sigma \rightarrow [P_\theta(x|z) \rightarrow \tilde{x}] \]

(C) Dhruv Batra  Image Credit: https://www.kaggle.com/rvislaywade/visualizing-mnist-using-a-variational-autoencoder
Putting it all together: maximizing the likelihood lower bound

\[ \mathbb{E}_z \left[ \log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) \| p_{\theta}(z)) \]

\[ \mathcal{L}(x^{(i)}, \theta, \phi) \]
Putting it all together: maximizing the likelihood lower bound

\[
\mathbb{E}_z \left[ \log p_\theta(x^{(i)} | z) \right] - D_{KL}(q_\phi(z | x^{(i)}) \| p_\theta(z))
\]

\[
\mathcal{L}(x^{(i)}, \theta, \phi)
\]

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

Input Data $\mathbf{x}$
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

\[ \mathbb{E}_z \left[ \log p_{\theta}(x^{(i)} \mid z) \right] - D_{KL}(q_{\phi}(z \mid x^{(i)}) \mid \mid p(z)) \]

\[ \mathcal{L}(x^{(i)}, \theta, \phi) \]

Encoder network

Input Data

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\mathbb{E}_z \left[ \log p_\theta(x^{(i)} | z) \right] - D_{KL}(q_\phi(z \mid x^{(i)}) \parallel p_\theta(z))$$

Make approximate posterior distribution close to prior

Encoder network

Input Data

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\mathbb{E}_z \left[ \log p_{\theta}(x^{(i)} \mid z) \right] - D_{KL}(q_{\phi}(z \mid x^{(i)}) \mid \mid p_{\theta}(z))$$

Make approximate posterior distribution close to prior

Encoder network $q_{\phi}(z \mid x)$

Sample $z$ from $z \mid x \sim \mathcal{N}(\mu_z \mid x, \Sigma_z \mid x)$

Input Data $x$
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\mathbb{E}_z \left[ \log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) \parallel p_{\theta}(z))$$

Make approximate posterior distribution close to prior

Encoder network

Decoder network

Input Data

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

\[
\mathbb{E}_z \left[ \log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z))
\]

Maximize likelihood of original input being reconstructed

Sample \( x|z \) from \( x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z}) \)

Sample \( z \) from \( z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x}) \)

Make approximate posterior distribution close to prior

Encoder network

Encoder network

Decoder network

Input Data
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

Maximize likelihood of original input being reconstructed

Sample $x|z$ from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$

Make approximate posterior distribution close to prior

Sample $z$ from $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$

For every minibatch of input data: compute this forward pass, and then backprop!

Encoder network

$q_\phi(z | x)$

Decoder network

$p_\theta(x | z)$

Input Data

$\hat{x}$

$\mu_{x|z}$

$\Sigma_{x|z}$

$\mu_{z|x}$

$\Sigma_{z|x}$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Use decoder network. Now sample z from prior!

Sample $x|z$ from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$

Sample $z$ from $z \sim \mathcal{N}(0, I)$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014
Use decoder network. Now sample z from prior!

Decoder network
$p_\theta(x|z)$

Sample z from $z \sim \mathcal{N}(0, I)$

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

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Auto Encoders: Generating Data

Use decoder network. Now sample $z$ from prior!

$$p_\theta(x|z)$$

$z \sim \mathcal{N}(0, I)$

Variational Auto Encoders: Generating Data

Data manifold for 2-d $z$

Vary $z_1$

Vary $z_2$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014
Diagonal prior on \( z \) => independent latent variables

Different dimensions of \( z \) encode interpretable factors of variation

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

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Auto Encoders: Generating Data

32x32 CIFAR-10

Labeled Faces in the Wild


Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n
Variational Autoencoders

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 $\text{q}(z|x)$, 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)

Active areas of research:
- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian
- Incorporating structure in latent variables

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n