Recap of last lecture

1. Autoregressive models:
   - Chain rule based factorization is fully general
   - Compact representation via *conditional independence* and/or *neural parameterizations*

2. Autoregressive models Pros:
   - Easy to evaluate likelihoods
   - Easy to train

3. Autoregressive models Cons:
   - Requires an ordering
   - Generation is sequential
   - Cannot learn features in an unsupervised way
Plan for today

1. Latent Variable Models
   - Mixture models
   - Variational autoencoder
   - Variational inference and learning
Lots of variability in images $x$ due to gender, eye color, hair color, pose, etc. However, unless images are annotated, these factors of variation are not explicitly available (latent).

Idea: explicitly model these factors using latent variables $z$.
1. Only shaded variables $x$ are observed in the data (pixel values)
2. Latent variables $z$ correspond to high level features
   - If $z$ chosen properly, $p(x|z)$ could be much simpler than $p(x)$
   - If we had trained this model, then we could identify features via $p(z|x)$, e.g., $p(\text{EyeColor} = \text{Blue}|x)$
3. Challenge: Very difficult to specify these conditionals by hand
Deep Latent Variable Models

1. \( z \sim \mathcal{N}(0, I) \)
2. \( p(x \mid z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z)) \) where \( \mu_\theta, \Sigma_\theta \) are neural networks
3. Hope that after training, \( z \) will correspond to meaningful latent factors of variation (features). Unsupervised representation learning.
4. As before, features can be computed via \( p(z \mid x) \)
Mixture of Gaussians. Bayes net: \( z \rightarrow x \).

1. \( z \sim \text{Categorical}(1, \cdots, K) \)
2. \( p(x \mid z = k) = \mathcal{N}(\mu_k, \Sigma_k) \)

Generative process

1. Pick a mixture component \( k \) by sampling \( z \)
2. Generate a data point by sampling from that Gaussian
Mixture of Gaussians: a Shallow Latent Variable Model

1. \( z \sim \text{Categorical}(1, \cdots, K) \)
2. \( p(x \mid z = k) = \mathcal{N}(\mu_k, \Sigma_k) \)

Clustering: The posterior \( p(z \mid x) \) identifies the mixture component

Unsupervised learning: We are hoping to learn from unlabeled data (ill-posed problem)
Unsupervised learning
Shown is the posterior probability that a data point was generated by the $i$-th mixture component, $P(z = i| x)$
Unsupervised clustering of handwritten digits.
Mixture models

Combine simple models into a more complex and expressive one

\[
p(x) = \sum_z p(x, z) = \sum_z p(z)p(x | z) = \sum_{k=1}^K p(z = k) \mathcal{N}(x; \mu_k, \Sigma_k)
\]
A mixture of an infinite number of Gaussians:

1. $z \sim \mathcal{N}(0, I)$
2. $p(x \mid z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta, \Sigma_\theta$ are neural networks
   - $\mu_\theta(z) = \sigma(Az + c) = (\sigma(a_1z + c_1), \sigma(a_2z + c_2)) = (\mu_1(z), \mu_2(z))$
   - $\Sigma_\theta(z) = \text{diag}(\exp(\sigma(Bz + d))) = \begin{pmatrix} \exp(\sigma(b_1z+d_1)) & 0 \\ 0 & \exp(\sigma(b_2z+d_2)) \end{pmatrix}$
   - $\theta = (A, B, c, d)$
3. Even though $p(x \mid z)$ is simple, the marginal $p(x)$ is very complex/flexible
Latent Variable Models

- Allow us to define complex models $p(x)$ in terms of simple building blocks $p(x \mid z)$
- Natural for unsupervised learning tasks (clustering, unsupervised representation learning, etc.)
- No free lunch: much more difficult to learn compared to fully observed, autoregressive models
Marginal Likelihood

- Suppose some pixel values are missing at train time (e.g., top half)
- Let \( X \) denote observed random variables, and \( Z \) the unobserved ones (also called hidden or latent)
- Suppose we have a model for the joint distribution (e.g., PixelCNN)

\[
p(X, Z; \theta)
\]

What is the probability \( p(X = \bar{x}; \theta) \) of observing a training data point \( \bar{x} \)?

\[
\sum_z p(X = \bar{x}, Z = z; \theta) = \sum_z p(\bar{x}, z; \theta)
\]

- Need to consider all possible ways to complete the image (fill green part)
Variational Autoencoder Marginal Likelihood

A mixture of an infinite number of Gaussians:

1. \( z \sim \mathcal{N}(0, I) \)
2. \( p(x \mid z) = \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z)) \) where \( \mu_\theta, \Sigma_\theta \) are neural networks
3. \( Z \) are unobserved at train time (also called hidden or latent)
4. Suppose we have a model for the joint distribution. What is the probability \( p(X = \bar{x}; \theta) \) of observing a training data point \( \bar{x} \)?

\[
\int_z p(X = \bar{x}, Z = z; \theta)dz = \int_z p(\bar{x}, z; \theta)dz
\]
## Partially observed data

- Suppose that our joint distribution is
  \[ p(\mathbf{X}, \mathbf{Z}; \theta) \]

- We have a dataset \( \mathcal{D} \), where for each datapoint the \( \mathbf{X} \) variables are observed (e.g., pixel values) and the variables \( \mathbf{Z} \) are never observed (e.g., cluster or class id.). \( \mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\} \).

- Maximum likelihood learning:
  \[
  \log \prod_{\mathbf{x} \in \mathcal{D}} p(\mathbf{x}; \theta) = \sum_{\mathbf{x} \in \mathcal{D}} \log p(\mathbf{x}; \theta) = \sum_{\mathbf{x} \in \mathcal{D}} \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)
  \]

- Evaluating \( \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta) \) can be intractable. Suppose we have 30 binary latent features, \( \mathbf{z} \in \{0, 1\}^{30} \). Evaluating \( \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta) \) involves a sum with \( 2^{30} \) terms. For continuous variables, \( \log \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta) d\mathbf{z} \) is often intractable. Gradients \( \nabla_{\theta} \) also hard to compute.

- Need **approximations**. One gradient evaluation per training data point \( \mathbf{x} \in \mathcal{D} \), so approximation needs to be cheap.
First attempt: Naive Monte Carlo

Likelihood function \( p_\theta(x) \) for Partially Observed Data is hard to compute:

\[
p_\theta(x) = \sum_{\text{All values of } z} p_\theta(x, z) = |Z| \sum_{z \in Z} \frac{1}{|Z|} p_\theta(x, z) = |Z| \mathbb{E}_{z \sim \text{Uniform}(Z)} [p_\theta(x, z)]
\]

We can think of it as an (intractable) expectation. Monte Carlo to the rescue:

1. Sample \( z^{(1)}, \ldots, z^{(k)} \) uniformly at random
2. Approximate expectation with sample average

\[
\sum_{z} p_\theta(x, z) \approx |Z| \frac{1}{k} \sum_{j=1}^{k} p_\theta(x, z^{(j)})
\]

Works in theory but not in practice. For most \( z \), \( p_\theta(x, z) \) is very low (most completions don't make sense). Some are very large but will never "hit" likely completions by uniform random sampling. Need a clever way to select \( z^{(j)} \) to reduce variance of the estimator.
Likelihood function $p_\theta(x)$ for Partially Observed Data is hard to compute:

$$p_\theta(x) = \sum_{\text{All possible values of } z} p_\theta(x, z)$$

Monte Carlo to the rescue:

1. Sample $z^{(1)}, \ldots, z^{(k)}$ from $q(z)$
2. Approximate expectation with sample average

$$p_\theta(x) \approx \frac{1}{k} \sum_{j=1}^{k} \frac{p_\theta(x, z^{(j)})}{q(z^{(j)})}$$

What is a good choice for $q(z)$? Intuitively, choose likely completions. It would then be tempting to estimate the log-likelihood as:

$$\log \left( p_\theta(x) \right) \approx \log \left( \frac{1}{k} \sum_{j=1}^{k} \frac{p_\theta(x, z^{(j)})}{q(z^{(j)})} \right) \approx \log \left( \frac{p_\theta(x, z^{(1)})}{q(z^{(1)})} \right)$$

However, it’s clear that $\mathbb{E}_{z^{(1)} \sim q(z)} \left[ \log \left( \frac{p_\theta(x, z^{(1)})}{q(z^{(1)})} \right) \right] \neq \log \left( \mathbb{E}_{z^{(1)} \sim q(z)} \left[ \frac{p_\theta(x, z^{(1)})}{q(z^{(1)})} \right] \right)$
Evidence Lower Bound

Log-Likelihood function for Partially Observed Data is hard to compute:

$$\log \left( \sum_{z \in Z} p_\theta(x, z) \right) = \log \left( \sum_{z \in Z} \frac{q(z)}{q(z)} p_\theta(x, z) \right) = \log \left( \mathbb{E}_{z \sim q(z)} \left[ \frac{p_\theta(x, z)}{q(z)} \right] \right)$$

- log() is a concave function. $\log(px + (1-p)x') \geq p \log(x) + (1-p) \log(x')$.
- Idea: use Jensen Inequality (for concave functions)

$$\log \left( \mathbb{E}_{z \sim q(z)} \left[ f(z) \right] \right) = \log \left( \sum_{z} q(z) f(z) \right) \geq \sum_{z} q(z) \log f(z)$$
Evidence Lower Bound

Log-Likelihood function for Partially Observed Data is hard to compute:

\[
\log \left( \sum_{z \in Z} p_\theta(x, z) \right) = \log \left( \sum_{z \in Z} \frac{q(z)}{q(z)} p_\theta(x, z) \right) = \log \left( \mathbb{E}_{z \sim q(z)} \left[ \frac{p_\theta(x, z)}{q(z)} \right] \right)
\]

- \( \log() \) is a concave function. \( \log(px + (1 - p)x') \geq p \log(x) + (1 - p) \log(x') \).
- Idea: use Jensen Inequality (for concave functions)

\[
\log \left( \mathbb{E}_{z \sim q(z)} [f(z)] \right) = \log \left( \sum_z q(z)f(z) \right) \geq \sum_z q(z) \log f(z)
\]

Choosing \( f(z) = \frac{p_\theta(x, z)}{q(z)} \)

\[
\log \left( \mathbb{E}_{z \sim q(z)} \left[ \frac{p_\theta(x, z)}{q(z)} \right] \right) \geq \mathbb{E}_{z \sim q(z)} \left[ \log \left( \frac{p_\theta(x, z)}{q(z)} \right) \right]
\]

Called Evidence Lower Bound (ELBO).
Variational inference

- Suppose $q(z)$ is any probability distribution over the hidden variables.

- **Evidence lower bound** (ELBO) holds for any $q$

  \[
  \log p(x; \theta) \geq \sum_z q(z) \log \left( \frac{p_\theta(x, z)}{q(z)} \right)
  \]

  \[
  = \sum_z q(z) \log p_\theta(x, z) - \sum_z q(z) \log q(z)
  \]

  \[
  \quad \underbrace{\text{Entropy } H(q) \text{ of } q}_{\text{Entropy}}
  \]

  \[
  = \sum_z q(z) \log p_\theta(x, z) + H(q)
  \]

- Equality holds if $q = p(z|x; \theta)$

  \[
  \log p(x; \theta) = \sum_z q(z) \log p(z, x; \theta) + H(q)
  \]

- (Aside: This is what we compute in the E-step of the EM algorithm)
Why is the bound tight

- We derived this lower bound that holds for any choice of $q(z)$:

$$
\log p(x; \theta) \geq \sum_z q(z) \log \frac{p(x, z; \theta)}{q(z)}
$$

- If $q(z) = p(z|x; \theta)$ the bound becomes:

$$
\sum_z p(z|x; \theta) \log \frac{p(x, z; \theta)}{p(z|x; \theta)} = \sum_z p(z|x; \theta) \log \frac{p(z|x; \theta)p(x; \theta)}{p(z|x; \theta)}
$$

\[
\begin{align*}
&= \sum_z p(z|x; \theta) \log p(x; \theta) \\
&= \log p(x; \theta) \sum_z p(z|x; \theta) \\
&= \log p(x; \theta)
\end{align*}
\]

- Confirms our previous importance sampling intuition: we should choose likely completions.

- What if the posterior $p(z|x; \theta)$ is intractable to compute? How loose is the bound?
Variational inference continued

- Suppose \( q(z) \) is any probability distribution over the hidden variables. A little bit of algebra reveals

\[
D_{KL}(q(z)\|p(z|x; \theta)) = -\sum_z q(z) \log p(z, x; \theta) + \log p(x; \theta) - H(q) \geq 0
\]

- Rearranging, we re-derived the **Evidence lower bound** (ELBO)

\[
\log p(x; \theta) \geq \sum_z q(z) \log p(z, x; \theta) + H(q)
\]

- Equality holds if \( q = p(z|x; \theta) \) because \( D_{KL}(q(z)\|p(z|x; \theta)) = 0 \)

\[
\log p(x; \theta) = \sum_z q(z) \log p(z, x; \theta) + H(q)
\]

- In general, \( \log p(x; \theta) = \text{ELBO} + D_{KL}(q(z)\|p(z|x; \theta)) \). The closer \( q(z) \) is to \( p(z|x; \theta) \), the closer the ELBO is to the true log-likelihood
The Evidence Lower bound

What if the posterior $p(z|x; \theta)$ is intractable to compute?

Suppose $q(z; \phi)$ is a (tractable) probability distribution over the hidden variables parameterized by $\phi$ (variational parameters)

- For example, a Gaussian with mean and covariance specified by $\phi$

$$q(z; \phi) = \mathcal{N}(\phi_1, \phi_2)$$

Variational inference: pick $\phi$ so that $q(z; \phi)$ is as close as possible to $p(z|x; \theta)$. In the figure, the posterior $p(z|x; \theta)$ (blue) is better approximated by $\mathcal{N}(2, 2)$ (orange) than $\mathcal{N}(-4, 0.75)$ (green)
A variational approximation to the posterior

- Assume $p(x^{top}, x^{bottom}; \theta)$ assigns high probability to images that look like digits. In this example, we assume $z = x^{top}$ are unobserved (latent).

- Suppose $q(x^{top}; \phi)$ is a (tractable) probability distribution over the hidden variables (missing pixels in this example) $x^{top}$ parameterized by $\phi$ (variational parameters).

$$q(x^{top}; \phi) = \prod_{\text{unobserved variables } x^{top}_i} (\phi_i)^{x^{top}_i} (1 - \phi_i)^{(1-x^{top}_i)}$$

- Is $\phi_i = 0.5 \; \forall i$ a good approximation to the posterior $p(x^{top}|x^{bottom}; \theta)$? No
- Is $\phi_i = 1 \; \forall i$ a good approximation to the posterior $p(x^{top}|x^{bottom}; \theta)$? No
- Is $\phi_i \approx 1$ for pixels $i$ corresponding to the top part of digit 9 a good approximation? Yes
The Evidence Lower bound

\[ \log p(x; \theta) \geq \sum_z q(z; \phi) \log p(z, x; \theta) + H(q(z; \phi)) = \mathcal{L}(x; \theta, \phi) \]

\[ = \mathcal{L}(x; \theta, \phi) + D_{KL}(q(z; \phi)\|p(z|x; \theta)) \]

The better \( q(z; \phi) \) can approximate the posterior \( p(z|x; \theta) \), the smaller \( D_{KL}(q(z; \phi)\|p(z|x; \theta)) \) we can achieve, the closer ELBO will be to \( \log p(x; \theta) \). Next: jointly optimize over \( \theta \) and \( \phi \) to maximize the ELBO over a dataset.
Summary

- **Latent Variable Models Pros:**
  - Easy to build flexible models
  - Suitable for unsupervised learning

- **Latent Variable Models Cons:**
  - Hard to evaluate likelihoods
  - Hard to train via maximum-likelihood
  - Fundamentally, the challenge is that posterior inference $p(z \mid x)$ is hard. Typically requires variational approximations

- **Alternative:** give up on KL-divergence and likelihood (GANs)