Generative Models - DRE7053
Lecture 2

NORA Summer School 2024

Rogelio A Mancisidor
Assistant Professor
Department of Data Science and Analytics
BI Norwegian Business School

June 10-14, 2024

Outline

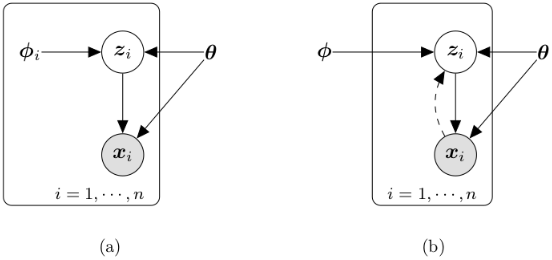
Outline

Variational Autoencoder - I

- The Variational Autoencoder (VAE) is an example of LVM where the posterior distribution is approximated using the variational inference principle
- Assume we observe $\boldsymbol{X} = \{\boldsymbol{x}_i\}_i^n$, and for each $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ we have one latent variable $\boldsymbol{z} \in \mathbb{R}^{d_z}$. Hence, $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_i^n$.
- VAE assumes a *mean-field* factorization

$$q(\boldsymbol{Z}|\boldsymbol{X};\phi) = \prod_i^n q_i(\boldsymbol{z}_i|\boldsymbol{x}_i;\phi), \tag{1}$$

Amortized Inference



- a) panel mean-field approximation, and b) panel VAE
- Note that ϕ does not depend on the i-th latent variable
- Amortized variational inference shares ϕ across all data points!

Variational Autoencoder - II

- Generative model
  - $p(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z})$
- Inference (recognition) model
  - $q(\boldsymbol{z}|\boldsymbol{x})$
- VAE assumes the following distributions:

$$\begin{aligned} p(\boldsymbol{z}) &\sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\ p(\boldsymbol{x}|\boldsymbol{z}) &\sim f(\cdot) \\ q(\boldsymbol{z}|\boldsymbol{x}) &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \end{aligned}$$

  where $\boldsymbol{\Sigma}$ is a diagonal matrix with main diagonal $\boldsymbol{\sigma}^2 = (\sigma_1^2, \cdots, \sigma_{d_z}^2)$
- $p(\boldsymbol{x}|\boldsymbol{z})$ can take different distributions depending on the data, e.g. Gaussian, Bernoulli, Laplace, etc.

Another way to derive the ELBO

ELBO - Closed Form

- Note that ELBO $= \mathbb{E}_q[\log p(\boldsymbol{x}|\boldsymbol{z}) + \log p(\boldsymbol{z}) - \log q(\boldsymbol{z}|\boldsymbol{x})]$ is composed by

$$\int q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} = \int \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}) \log \mathcal{N}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} \tag{2}$$

$$\int q(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{z}) d\boldsymbol{z} = \int \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}) \log \mathcal{N}(\boldsymbol{z}) d\boldsymbol{z} \tag{3}$$

$$-\int q(\boldsymbol{z}|\boldsymbol{x}) \log q(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} = -\int \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}) \log \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \tag{4}$$

ELBO - Closed Form

- According to Lemma 1 in [11] (page 48)

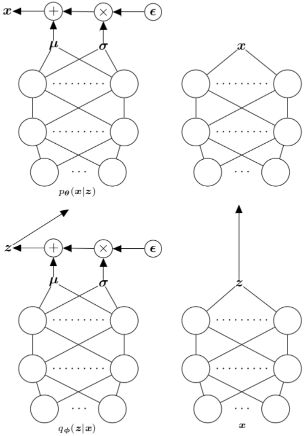
$$\int q(\boldsymbol{x}) \log p(\boldsymbol{z}) d\boldsymbol{x} = \sum_{j=1}^{d_x} -\frac{1}{2} \log(2\pi\sigma_{1,j}^2) - \frac{\sigma_{2,j}^2}{2\sigma_{1,j}^2} - \frac{(\mu_{2,j} - \mu_{1,j})^2}{2\sigma_{1,j}^2} \tag{5}$$

where $\sigma_{i,j}^2$ and $\mu_{i,j}$ are the j-th element of their respective $\boldsymbol{\mu}_1$ and $\boldsymbol{\sigma}_1^2$ parameters of $p(\boldsymbol{x})$ or $\boldsymbol{\mu}_2$ and $\boldsymbol{\sigma}_2^2$ of $q(\boldsymbol{x})$.

Closed form

Variational Autoencoder - III

- $q(\boldsymbol{z}|\boldsymbol{x})$ is often referred to as *probabilistic encoder*
- $p(\boldsymbol{z}|\boldsymbol{x})$ is often referred to as *probabilistic decoder*
- The reason is its similarity with autoencoders



Algorithm 1 Generative Modeling with VAEs

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Optimized trainable parameters
repeat for $i = 1, \cdots, N$
  $\boldsymbol{x}^i \leftarrow$ Random sample from test set
  $\boldsymbol{\epsilon}^i \leftarrow$ Random samples from $\mathcal{N}(\mathbf{0}, \mathbf{1})$
  $\boldsymbol{\mu}^i_{\boldsymbol{\phi}}, \boldsymbol{\sigma}^i_{\boldsymbol{\phi}} = f_{\boldsymbol{\phi}}(\boldsymbol{x}^i)$
  $\boldsymbol{z}^i = \boldsymbol{\mu}^i_{\boldsymbol{\phi}} + \boldsymbol{\sigma}^i_{\boldsymbol{\phi}} \boldsymbol{\epsilon}^i \leftarrow$ latent from posterior
  $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^i) \leftarrow$ Generate from likelihood
until
return $\boldsymbol{x}$

Generative modeling with the prior

Algorithm 2 Generative Modeling with VAEs

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Optimized trainable parameters
repeat for $i = 1, \cdots, N$
  $\boldsymbol{z}^i \sim p(\boldsymbol{z}) \leftarrow$ latent from prior
  $\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^i) \leftarrow$ Generate from likelihood
until
return $\boldsymbol{x}$

- Point-estimate: $\boldsymbol{\mu}_{\boldsymbol{\theta}}$. The mode or median is also possible.
- Random samples: $\boldsymbol{x}^i = \boldsymbol{\mu}^i_{\boldsymbol{\theta}} + \boldsymbol{\sigma}^i_{\boldsymbol{\theta}} \boldsymbol{\epsilon}^i$

Arithmetic Operations on the Latent Space

```python
# pseudo code
def interpolate(start, end, steps):
    interpolation = tf.zeros([start.shape[0], steps+2])
    for dim, (s, e) in enumerate(zip(start, end)):
        interpolation[dim] = tf.linspace(s, e, steps+2)
    return interpolation.T

z1 = tf.random.normal(shape=[N,latent_size])
z2 = tf.random.normal(shape=[N,latent_size])
z  = interpolate(start=z1, end=z2, steps=4)
```

- You can use any arithmetic operation on the latent vectors $\boldsymbol{z}$!

Reparameterization Trick

- The VAE parameterize the distribution parameters with neural networks, i.e.,

$$p(\boldsymbol{x}|\boldsymbol{z}) \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{x}|\boldsymbol{z}} = f_{\boldsymbol{\theta}}(\boldsymbol{z}), \boldsymbol{\sigma}^2_{\boldsymbol{x}|\boldsymbol{z}} = f_{\boldsymbol{\theta}}(\boldsymbol{z})), \tag{6}$$

  and

$$q(\boldsymbol{z}|\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\mu}_{\boldsymbol{z}|\boldsymbol{x}} = f_{\phi}(\boldsymbol{x}), \boldsymbol{\sigma}^2_{\boldsymbol{z}|\boldsymbol{x}} = f_{\phi}(\boldsymbol{x})), \tag{7}$$

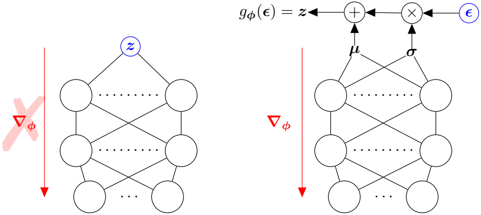
  where $f_{\phi}(\boldsymbol{x})$ and $f_{\boldsymbol{\theta}}(\boldsymbol{z})$ are neural networks with trainable parameters ϕ and $\boldsymbol{\theta}$
- Note that

$$\begin{aligned} \text{ELBO} = \mathbb{E}_q\Big[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\Big] &= \mathbb{E}_q[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - KL[q_{\phi}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \\ &= \mathbb{E}_q[f_{\boldsymbol{\theta}}(\boldsymbol{z})] - KL[q_{\phi}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \end{aligned}$$

- Remember, the closed form solution includes the parameters of $q(\boldsymbol{z}|\boldsymbol{x})$!

Reparameterization Trick - II

- To backpropagate $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ we adopt the following architecture



- But the reparameterization trick is more than that...

Reparameterized Gradients

- Use an invertible function, e.g.

$$z = g_\phi(\epsilon) = \boldsymbol{\mu} + \boldsymbol{\sigma}\epsilon, \tag{8}$$

where $\epsilon \sim N(0, 1)$.

- Use the *change of variable* result (see Lemma 2 in [11]) that says

$$\begin{aligned}\int q(\boldsymbol{z}|\boldsymbol{x}) f(\boldsymbol{z}) dz &= \int p(\boldsymbol{\epsilon}) f(\boldsymbol{z}) d\boldsymbol{\epsilon} \\ &= \int p(\boldsymbol{\epsilon}) f(g_\phi(\boldsymbol{\epsilon})) d\boldsymbol{\epsilon} \\ \mathbb{E}_q[f(\boldsymbol{z})] =& \mathbb{E}_p[f(g_\phi(\boldsymbol{\epsilon}))] \end{aligned} \tag{9}$$

- Therefore, the Monte Carlo estimate

$$\frac{1}{L} \sum_{l=1}^{L} \log p(\boldsymbol{x}_i | \boldsymbol{z}_i = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \boldsymbol{\epsilon}_i)$$

is an expectation over $p(\epsilon)$!

Variance of Reparameterized and Score Gradients

- Assume that $p(x) \sim \mathcal{N}(\theta, 1)$ and we want to minimize

$$\arg\min_{\theta} \mathbb{E}_p[x^2].$$

- The score derivative is given by

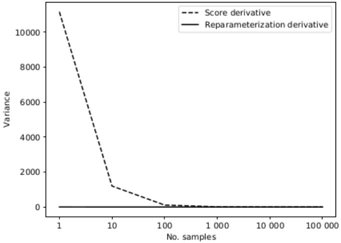
$$(10)$$

- Use the reparameterization $x = \theta + \epsilon$ where $q(\epsilon) \sim \mathcal{N}(0, 1)$. Therefore, $\mathbb{E}_p[x^2] = \mathbb{E}_q[(\theta + \epsilon)^2]$ and its derivative is

$$(11)$$

Simulation

- We simulate $N = [1, 10, 100, 1000, 10000, 100000]$ samples from $p(x) \sim \mathcal{N}(\theta, 1)$, where $\theta = 10$, and $q(\epsilon) \sim \mathcal{N}(0, 1)$ to estimate the variance of 100 Monte Carlo estimates of Equation 10 and 11.



Outline

Bound on Mutual Information

- Let's define the following distributions

$$
\begin{aligned}
q_\phi(\boldsymbol{x}, \boldsymbol{z}) =& p(\boldsymbol{x}) q_\phi(\boldsymbol{z}|\boldsymbol{x}) && (12)\\
q_\phi(\boldsymbol{z}) =& \mathbb{E}_{p(\boldsymbol{x})}[q_\phi(\boldsymbol{z}|\boldsymbol{x})] \approx \frac{1}{N}\sum_n q(\boldsymbol{z}|\boldsymbol{x}_n) && (13)\\
q_\phi(\boldsymbol{x}, \boldsymbol{z}) =& p_\theta(\boldsymbol{x}|\boldsymbol{z}) q_\phi(\boldsymbol{z}) && (14)
\end{aligned}
$$

- Note

$$
I(\boldsymbol{z}, \boldsymbol{x}) = \mathbb{E}_{q_\phi(\boldsymbol{z}, \boldsymbol{x})}\Big[\log \frac{q_\phi(\boldsymbol{z}, \boldsymbol{x})}{p(\boldsymbol{x}) q_\phi(\boldsymbol{z})}\Big]
$$

- Meaning that

(15)

- What does that mean?

Posterior Collapse

- In practice, we maximize

$$\text{ELBO} = \mathbb{E}_{p(\boldsymbol{x})}\left[\mathbb{E}_q[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] - KL[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]\right] \tag{16}$$

- We know that $\mathbb{E}_{p(\boldsymbol{x})} KL[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \geq I(\boldsymbol{z}, \boldsymbol{x})$
- Minimizing the average KL makes the posterior *collapse* into the prior, i.e.

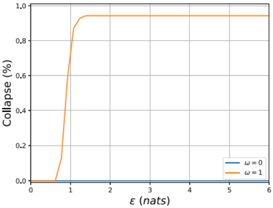
$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) \approx p(\boldsymbol{z})$$

meaning $\boldsymbol{z}$ is independent of $\boldsymbol{x}$!!!

- Ideally VAEs should embed as much information of $\boldsymbol{x}$ into $\boldsymbol{z}$

Measuring Posterior Collapse

- We measure posterior collapse as the proportion of latent dimensions that are within ϵ KL divergence of the prior for at least 99% of the data sample.



Yet another way to derive the ELBO - I

- The expectation in equation 16 is taken with the empirical distribution of the data, i.e.

$$\text{ELBO} = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_q[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n|\boldsymbol{z}_n)] - KL[q_\phi(\boldsymbol{z}_n|\boldsymbol{x}_n)||p(\boldsymbol{z}_n)] \tag{17}$$

- The term-by-term KL is minimized when $q_\phi(\boldsymbol{z}_n|\boldsymbol{x}_n) = p(\boldsymbol{z}_n)$ for all n
- Using the derivation in equation 15 we obtain

$$\frac{1}{N}\sum_{n=1}^{N} KL[q_\phi(\boldsymbol{z}_n|\boldsymbol{x}_n)||p(\boldsymbol{z}_n)] = I(\boldsymbol{x}, \boldsymbol{z}) + KL[q_\phi(\boldsymbol{z}_n)||p(\boldsymbol{z}_n)] \tag{18}$$

Yet another way to derive the ELBO - II

- Therefore

$$
\begin{aligned}
ELBO = & \frac{1}{N}\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{z}_n) \rightarrow ① \\
& -(\log N - \mathbb{E}_{q(\boldsymbol{z})}[\mathbb{H}(p(\boldsymbol{x}|\boldsymbol{z}))]) \rightarrow ② \\
& -KL[q_\phi(\boldsymbol{z})||p(\boldsymbol{z})] \rightarrow ③
\end{aligned} \tag{19}
$$

- ① average reconstruction
- ② mutual information
- ③ marginal (aggregated) KL divergence

Intuition

- ① and ② are in tension with each other. Good reconstructions required $\boldsymbol{z}_n$ to be specific to $\boldsymbol{x}_n$ which corresponds to a low entropy.
- ② is bounded below and above

$$0 \leq \log N - \mathbb{E}_{q(\boldsymbol{z})}[\mathbb{H}(p(\boldsymbol{x}|\boldsymbol{z}))]) \leq \log N$$

- The prior only appears in ③. We could choose the prior to be $q(\boldsymbol{z})$, so the divergence is 0.

Outline

Adding MI to the ELBO

- If the ELBO *encourages* $q(\boldsymbol{z}|\boldsymbol{x}) \to p(\boldsymbol{z})$ that means that $\boldsymbol{z}$ is independent of $\boldsymbol{x}$, what can we do?
- Let's optimize the mutual information $I(\boldsymbol{x}, \boldsymbol{z})$ and add it to the ELBO!
- The objective function is then $\text{ELBO} + (1 - \omega)I(\boldsymbol{x}, \boldsymbol{z})$

Outline

Theoretical Background

- Semi-supervised learning considers the problem of classification when only a subsets of data has class labels
- We observe N pairs of labeled data

$$(\boldsymbol{X}, \boldsymbol{Y}) = \{(\boldsymbol{x}, y)_1, (\boldsymbol{x}, y)_2, \cdots, (\boldsymbol{x}, y)_N\}$$

and M unlabeled observations

$$\boldsymbol{X} = \{\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+2}, \cdots, \boldsymbol{x}_{N+M}\}$$

The M2 model

- We assume the following generative model $p(\boldsymbol{x}, y, \boldsymbol{z}) = p(\boldsymbol{z})p(y)p(\boldsymbol{x}|\boldsymbol{z}, y)$

$$\begin{aligned} p(\boldsymbol{z}) &\sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\ p(y) &\sim \text{Cat}(\boldsymbol{\pi_0}) \\ p(\boldsymbol{x}|\boldsymbol{z}, y) &\sim f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \end{aligned}$$

- The inference model is factorized as $q(\boldsymbol{z}, y|\boldsymbol{x}) = q(\boldsymbol{z}|\boldsymbol{x}, y)q(y|\boldsymbol{x})$

$$\begin{aligned} q(\boldsymbol{z}|\boldsymbol{x}, y) &\sim \mathcal{N}(\boldsymbol{\mu} = f_{\phi}(\boldsymbol{x}, y), \boldsymbol{\sigma}^2 = f_{\phi}(\boldsymbol{x}, y)) \\ q(y|\boldsymbol{x}) &\sim \text{Cat}(\boldsymbol{\pi} = \boldsymbol{f}_{\phi}(\boldsymbol{x})) \end{aligned}$$

ELBO and Training Scheme

- Labeled data

- Unlabeled data