IFT6135 – Homework #4

Generative Models

Jean-Philippe Reid
701300

April 14, 2018

1 Reparameterization trick of variational autoencoder

(a)

$$\begin{aligned}\mathbb{E}[z] &= \mathbb{E}[\mu(x) + \sigma(x) \odot \epsilon] \\ &= \mathbb{E}[\mu(x)] + \mathbb{E}[\sigma(x) \odot \epsilon] \\ &= \mu(x) + \mathbb{E}[\sigma(x) \odot \mathcal{N}(\epsilon; 0, \mathbf{I})] \\ &= \mu(x) \end{aligned} \tag{1.1}$$

$$\begin{aligned}\mathrm{Var}[\mathrm{z}] &= \mathrm{Var}[\mu(\mathrm{x}) + \sigma(\mathrm{x}) \odot \epsilon] \\ &= \mathrm{Var}[\mu(\mathrm{x})] + \mathrm{Var}[\sigma(\mathrm{x}) \odot \epsilon] \\ &= \mathrm{Var}[\sigma(\mathrm{x}) \odot \mathcal{N}(\epsilon; 0, \mathbf{I})] \\ &= \sigma(x)^2 \end{aligned} \tag{1.2}$$

Let's suppose that $z = \mu(x) + S(x)\epsilon$, where $S(x) \in \mathbb{R}^{k \times k}$. $\mu(x)$ remains unchanged. The variance is, however, given by

$$\begin{aligned}\mathrm{Var}[\mathrm{S}(\mathrm{x})\epsilon] &= S(x)\,\mathrm{Var}[\epsilon]\,\mathrm{S}^{\intercal}(\mathrm{x}) \\ &= S(x)\,\mathrm{Var}[\mathcal{N}(\epsilon; 0, \mathbf{I})]\,\mathrm{S}^{\intercal}(\mathrm{x}) \\ &= S(x)\,\mathbf{I}S^{\intercal}(x) \\ &= S(x)\,S^{\intercal}(x)\,. \end{aligned} \tag{1.3}$$

Hence, in this specific case, $z \sim \mathcal{N}(\mu(x), S(x)\,S^{\intercal}(x))$, *i.e.* a normal distribution with full covariance matrix given by $S(x)\,S^{\intercal}(x)$.

(b) Let's first state the main objective of generative models by considering the directed latent-variable model

$$p(x, z \mid \theta) = p(x \mid z, \theta)\, p(z, \theta) = p(z \mid x, \theta)\, p(x, \theta). \tag{1.4}$$

The main objectives is to learn the parameters θ of p in order to approximate the posterior inference over z given an input x ($p(z \mid x, \theta)$), but also the marginal inference over x given a latent variable z ($p(x \mid z, \theta)$). There is, however, a problem with this approach as $p(z \mid x, \theta)$ is intractable.

The Bayes variational auto-encoder can efficiently solve this problem by maximizing the evidence lower bound (ELBO) defined as

$$\mathcal{L}(p_\theta, q_\phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x, z) - \log q_\phi(z \mid x)\right]. \tag{1.5}$$

The main idea is to find a distribution function $q_\phi(z \mid x)$ that is the closest to $p_\theta(z \mid x)$. Then, we will be able to sample directly from the new approximated function $q_\phi(z \mid x)$ to encode the inputs.

It is important to note that the Eq. 1.5 is directly related to the log likelihood $p_\theta(x)$,

$$\log p_\theta(x) = \mathbb{KL}\left[q_\phi(z \mid x) \,||\, p_\theta(z \mid x)\right] + \mathcal{L}(p_\theta, q_\phi). \tag{1.6}$$

Since x is fixed, we can replace the prior by $q(z \mid x)$ in Eq. 1.5. This means that $q(z)$ will be different for each input x. Thus, the model will approximate a better posterior $q(z)$.

The optimization problem is twofold:

(a) optimize over ϕ to keep tight the ELBO around $\log p(x)$

(b) optimize over θ to increase the lower bound.

Of course, the global effect will be to increase the likelihood $p(x)$.

It is possible to rewrite Eq. 1.6 as

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - \mathbb{KL}\left[q_\phi(z \mid x) \,||\, p_\theta(z)\right] \tag{1.7}$$

This equation is the central piece of the variational auto-encoder. Within this framework, $q_\phi(z \mid x) \sim \mathcal{N}(\mu(x), \mathrm{diag}(\sigma^2(x)))$.

Here are few observations for the variational auto-encoder:

(a) Both terms on the right hand side of Eq. 1.7 has the term $q_\phi(z \mid x)$, which can be interpreted as a encoder that describes x.

(b) In the first term, $\log p_\theta(x \mid z)$ is the log-likelihood of an input given the latent variable z. In short, this term helps to reconstruct, or decode x given the latent variable z.

(c) The second term is the $\mathbb{KL}$ divergence between $q_\phi(z \mid x)$ and the prior $p(z)$, usually fixed to a Gaussian. It prevents $q_\phi(z \mid x)$ from simply encoding an identity mapping. It forces the model to learn more generalized representations of the data. Hence its appellation as regularization term.

(d) This task is embedded into a pair of neural networks that are trained to minimize the reconstruction error $|\tilde{x} - x|$ where $\tilde{x}$ is the reconstructed input. The implementation of the reparameterization trick is essential in order to train the model end-to-end.

An alternative approach is to use mean-field inference to resolve this problem. This approach consist on factorizing $q(z \mid x)$ as

$$q(z \mid x) = \prod_{j=1}^{n} q(z_j \mid x) \tag{1.8}$$

Such representation is based on an important assumption that the latent variables are independent. It helps to approximate all the possible distributions from which we want to inference from.

By putting together the Eqs. 1.5 and 1.8, it is possible to obtain an optimal solution given by

$$q_k(z)^\star = \exp \mathbb{E}_{-k}[\log p(z, x)] + \text{cst.} \tag{1.9}$$

The optimal solution is given by the expected value of the joint distribution with respect to all the latent variables except the one we optimized. The usual method to optimize such problem is called *coordinate descent* and consists of optimizing Eqs. 1.5 for the problem in hand for each q_j individually while keeping the other coordinates constant.

Here are few observations for the mean-field inference approach :

(a) The conditional probability distribution defined in Eq. 1.8 does not contain the *true* posterior since, in reality, the latent variables z_i are dependent with each other. This is also true for the VAE.

(b) The assumption that $q(z \mid x)$ is fully factored is too strong considering the rigidity of coordinate descent optimization algorithm. This is in contrast with the variational auto-encoder where all the variables are optimized in a jointly manner through a neural net, $q_\phi(z \mid x)$. That is the fundamental difference between the two approaches.

In short, the inference variational auto-encoder generally the mean-field method. It can be explained by the capacity of both models to represent the true distribution $q(z \mid x)$. For the VAE, $q_\phi(z \mid x)$ which is a function of ϕ. Hence, the the model has

more flexibility to generalize. This is in contrast with the mean field method where $q^{mf}(z \mid x)$ depends on all other latent variable which are set to be constant.

References:

- Lecture notes of CS228, Sections on *The variational auto-encoder* and *Variational inference* – https://ermongroup.github.io/cs228-notes
- C.M Bishop, chapter 10.

2 Importance weighted autoencoder

(a) *Show that IWLB is a lower bound on the log likelihood* $\log p(x)$

$$
\begin{aligned}
L_k &= \mathbb{E}_{z_{1:k}\sim q(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{k}\sum_{j=1}^{k}\frac{p(\mathbf{x}, z_j)}{q(z_j \mid \mathbf{x})}\right] \\
&\leq \log\left(\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\frac{1}{k}\sum_{j=1}^{k}\frac{p(\mathbf{x}, z_j)}{q(z_j \mid \mathbf{x})}\right]\right) \qquad (2.1) \\
&= \log\left(\frac{1}{k}\sum_{j=1}^{k}\mathbb{E}_{q(\mathbf{x}|z_j)}\left[\frac{p(\mathbf{x}, z_j)}{q(z_j \mid \mathbf{x})}\right]\right) \\
&= \log\left(\frac{1}{k}\sum_{j=1}^{k}\int q(z_j \mid \mathbf{x})\,\frac{p(\mathbf{x}, z_j)}{q(z_j \mid \mathbf{x})}\,\mathrm{d}z_j\right) \qquad (2.2) \\
&= \log\left(\frac{1}{k}\sum_{j=1}^{k}\int p(\mathbf{x}, z_j)\;\mathrm{d}z_j\right) \\
&= \log\left(\frac{1}{k}\sum_{j=1}^{k} p(\mathbf{x})\right) \\
&= \log\left(p(\mathbf{x})\right) \qquad (2.3)
\end{aligned}
$$

Eq 2.1 : we used the Jensen's Equality.
Eq 2.2 : The integral can be distributed over each z_j since they are sampled independently.

Hence, more samples can only improve the thightness of the bound!

Reference: Y. Burda *et al.*, Importance weighted autoencoders.
https://arxiv.org/abs/1509.00519

(b) L_2 is defined as,

$$
L_2 = \mathbb{E}_{z_{1:k}\sim q(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{2}\sum_{j=1}^{2}\frac{p(\mathbf{x}, z_j)}{q(z_j \mid \mathbf{x})}\right]. \qquad (2.4)
$$

- $L_2 \leq \log p(x)$

  This is a consequence of the Jensen Inequality. See Eq. 2.1 for more details.

•

$$L_2 = \mathbb{E}_{z_{1,2}\sim q(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{2}\sum_{j=1}^{2}\frac{p\left(\mathbf{x}, z_j\right)}{q\left(z_j \mid \mathbf{x}\right)}\right] \tag{2.5}$$

$$= \mathbb{E}_{z_{1,2}\sim q(\mathbf{z}|\mathbf{x})}\left[\log \mathbb{E}_{z_1\sim q(\mathbf{z}|\mathbf{x})}\frac{p\left(\mathbf{x}, z_1\right)}{q\left(z_1 \mid \mathbf{x}\right)}\right] \tag{2.6}$$

$$\geq \mathbb{E}_{z_{1,2}\sim q(\mathbf{z}|\mathbf{x})}\left[\mathbb{E}_{z_1\sim q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p\left(\mathbf{x}, z_1\right)}{q\left(z_1 \mid \mathbf{x}\right)}\right]\right] \tag{2.7}$$

$$= \mathbb{E}_{z_1\sim q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p\left(\mathbf{x}, z_1\right)}{q\left(z_1 \mid \mathbf{x}\right)}\right]$$

$$= L_1 \tag{2.8}$$

We used the property of the expectation, $\mathbb{E}\left[1/m\sum_j a_j\right] = 1/n\sum_i a_i$ for the Eq. 2.6; and the Jensen's equality for Eq. 2.7.

Hence, we have that

$$L_1 \leq L_2 \leq \log p\left(x\right), \tag{2.9}$$

which shows that L_2 is a tighter bound than ELBO.

3 Maximum likelihood for GANs

We want to find a function $f(\cdot)$ for which the objective function given by

$$J^{(G)} = \max_g \mathbb{E}_{x\sim P_g}\left[f\left(D\left(G\left(z\right)\right)\right)\right] \tag{3.1}$$

is equivalent to maximizing the likelihood.

It is important to note that maximum likelihood estimation can be interpreted as a process to minimize a KL-Divergence between two distributions. Hence, in order for the cost function $J^{(G)}$ corresponds to the maximum likelihood, we need to show that

$$\frac{\partial}{\partial\theta}J^{(g)} = \frac{\partial}{\partial\theta}\mathrm{KL}\left(P_d \,||\, P_g\right) = \mathbb{E}_{P_d(x)}\left[\frac{\partial}{\partial\theta}\log P_g\left(x\right)\right]. \tag{3.2}$$

where $P_g(x)$ represent the probability distribution of the generated fake image, $G(z)$. Let's develop the left and right hand sides separately,

$$\begin{aligned}
\frac{\partial}{\partial\theta}J^{(g)} &= \frac{\partial}{\partial\theta}\left[\mathbb{E}_{x\sim P_g}\left[f\left(D\left(G\left(z\right)\right)\right)\right]\right] \\
&= \frac{\partial}{\partial\theta}\left[\int f\left(D\left(G\left(z\right)\right)\right)P_g\left(x\right)\,\mathrm{d}x\right] \\
&= \int f\left(D\left(G\left(z\right)\right)\right)\frac{\partial}{\partial\theta}P_g\left(x\right)\,\mathrm{d}x \\
&= \int f\left(D\left(G\left(z\right)\right)\right)P_g\left(x\right)\frac{\partial}{\partial\theta}\log P_g\left(x\right)\,\mathrm{d}x
\end{aligned} \tag{3.3}$$

$$\begin{aligned}
\mathrm{KL}\left(P_d\left(x\right) \,||\, P_g\left(x\right)\right) &= \frac{\partial}{\partial\theta}\mathbb{E}_{P_d(x)}\left[\log P_d\left(x\right) - \log P_g\left(x\right)\right] \\
&= -\frac{\partial}{\partial\theta}\int P_d\left(x\right)\log P_g\left(x\right)\,\mathrm{d}x \\
&= -\int P_d\left(x\right)\frac{\partial}{\partial\theta}\log P_g\left(x\right)\,\mathrm{d}x
\end{aligned} \tag{3.4}$$

For Eqs. 3.3 and 3.4 to be equal, we find that

$$f\left(D\left(G\left(z\right)\right)\right) = -\frac{P_d\left(x\right)}{P_g\left(x\right)} \tag{3.5}$$

Then, by assuming that the discriminator is optimal, we find that

$$D^\star = \frac{P_d\left(x\right)}{P_g\left(x\right) + P_d\left(x\right)} = \sigma\left(a\left(x\right)\right) \tag{3.6}$$

$$\frac{P_d(x)}{P_g(x)\left(1+P_d(x)/P_g(x)\right)} = \frac{1}{1+\exp(a(x))}$$
$$\frac{-f(x)}{\left(1-f(x)\right)} = \frac{1}{1+\exp(a(x))}$$
$$f(x) = -\exp(a(x)) \tag{3.7}$$

where we defined $D^\star = \sigma(a(x))$ (see Eq. 3.6). This is a valid assumption as $D(x)$ will always be between 0 and 1. By replacing $a(x) = \sigma^{-1}(D^\star)$ in Eq. 3.7, we have that

$$f(x) = -\exp\left(\sigma^{-1}(D^\star)\right). \tag{3.8}$$

For this function, the objective function will be equivalent to maximizing the likelihood.

Reference: Ian Goodfellow, NIPS 2016 Tutorial: Generative Adversarial Networks. https://arxiv.org/abs/1701.00160