Basics VAE

March 20, 2022

Useful papers - [Doe16]; [KW13]

1 Introduction

Let $x_1, ..., x_n$ - independent and identically distributed (i.i.d) random variables from distribution $P_{data}(X)$ (let's say this would be training dataset). Goal of any generative model is to reproduce the distribution $P_{data}(X)$ (or at least be able to make samples from it). VAE works with latent variables Z from some distribution $P(Z)$. The main idea behind latent distribution is to reproduce initial object (X variable) using hidden description and variable Z can be easy sampled from $P(Z)$. Then we need to build parameterized function $f : Z \times \Theta \to X$ to display hidden variable to initial. In this way we create samples from conditional distribution $P(X|Z, \Theta)$. We can obtain marginal distribution as

$$\begin{aligned} P(X|\Theta) = \int\limits_Z P(X, Z|\Theta)dz = [P(X, Z|\Theta) = P(Z, X|\Theta) = P(X|Z, \Theta)P(Z|\Theta), P(Z|\Theta) = P(Z)] = \\ = \int\limits_Z P(X|Z, \Theta)P(Z)dz = \mathbb{E}_{z\sim P}\, P(X|Z, \Theta) \approx \frac{1}{m}\sum_{i=1}^{m} P(X|z_i, \Theta) \end{aligned} \tag{1}$$

This marginal will be an estimate of our data distribution $P_{data}(X)$. VAE using following assumptions: $P(X|Z, \Theta) = \mathbf{N}(X|f_\Theta(Z), \sigma I)$, $P(Z) = \mathbf{N}(Z|0, I)$ .We can use maximum log likelihood estimation for approximation data distribution:

$$\begin{aligned} \Theta^* = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{z\sim P}\, P(X|Z, \Theta) \approx \underset{\Theta}{\operatorname{argmax}} \frac{1}{m}\sum_{i=1}^{m}\prod_{j=1}^{n} P(x_j|z_{j_i}, \Theta) \\ P_{data}(X) \approx P(X|\Theta^*) \end{aligned} \tag{2}$$

The main problem consist in product in 2. It is so hard to take gradients from product, this procedure is ineffective and resource-consuming. We need to take logarithm for avoid product, but can do it directly for equation 2. For this, a new arbitrary distribution is introduced - $Q(Z)$. Now we need to connect $Q(Z)$ and $P(X|\Theta)$.

$$\begin{aligned} \log P(X|\Theta) = \int\limits_Z Q(Z)\log P(X|\Theta)dZ = [P(X|\Theta) = \frac{P(X, Z|\Theta)}{P(Z|X, \Theta)}] = \\ = \int\limits_Z Q(Z)\log\frac{P(X, Z|\Theta)}{P(Z|X, \Theta)}dZ = \int\limits_Z Q(Z)\log\frac{P(X, Z|\Theta)Q(Z)}{P(Z|X, \Theta)Q(Z)}dZ = \\ = \int\limits_Z Q(Z)\log\frac{P(X, Z|\Theta)}{Q(Z)}dZ + \int\limits_Z Q(Z)\log\frac{Q(Z)}{P(Z|X, \Theta)}dZ = \\ = \int\limits_Z Q(Z)\log P(X|Z, \Theta)dZ + \int\limits_Z Q(Z)\log\frac{P(Z)}{Q(Z)}dZ + \int\limits_Z Q(Z)\log\frac{Q(Z)}{P(Z|X, \Theta)}dZ \end{aligned} \tag{3}$$

Finally, we have

$$
\begin{gathered}
\log P(X|\Theta) - \mathbf{KL}(Q(Z)||P(Z|X)) = \mathbb{E}_{z\sim Q}\log P(X|Z,\Theta) - \mathbf{KL}(Q(Z)||P(Z)) = \mathcal{L}(\Theta, Q) \\
\log P(X|\Theta) = \mathcal{L}(\Theta, Q) + \mathbf{KL}(Q(Z)||P(Z|X,\Theta))
\end{gathered}
\tag{4}
$$

$\mathcal{L}(\Theta, Q)$ calls evidence lower bound.

Definition 1 *Variational lower bound*

Function g(x, y(x)) calls lower bound for function f(x) if and only if
1. $\forall x, y \quad g(x, y(x)) \leq f(x)$
2. $\exists x_0 : g(x_0, y(x_0) = f(x_0)$

As can be seen $\mathcal{L}(\Theta, Q)$ satisfy conditionals of definition:

$$
\begin{gathered}
1. \log P(X|\Theta) \leq \mathcal{L}(\Theta, Q); \\
\mathbf{KL}(Q(Z)||P(Z|X)) \geq 0, \forall \Theta, \Phi \\
\\
2. \; \exists Q : Q(Z) = P(Z|X,\Theta) \Rightarrow \log P(X|\Theta) = \mathcal{L}(\Theta, Q)
\end{gathered}
\tag{5}
$$

Let's rewrite our likelihood

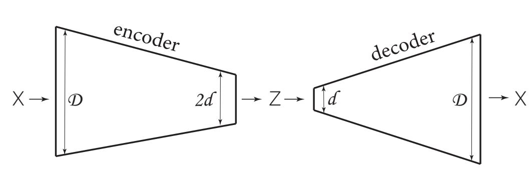
$$
\log P(X|\Theta) = \sum_{i=1}^{n} \log p(x_i, \Theta) \geq \sum_{i=1}^{n} \left[\int_{z_i} Q(z_i) \log p(x_i|z_i,\Theta) dz_i - \mathbf{KL}(Q(z_i)||p(z_i)) \right] \tag{6}
$$

Where n is the number of training data. So, the main idea of VAE is to maximize ELBO instead of maximzing likelihood directly, because $\mathbf{KL}(Q(Z)||P(Z|X,\Theta))$ is intractable, because we dont know $P(Z|X,\Theta)$. Also, it is hard to optimizing by function. Thus, lets constrain Q by parameterical class $Q(z_i|\phi_i) = \mathbf{N}(z_i|\mu(\phi_i), \sigma(\phi_i))$ and consider $P(z_i) = \mathbf{N}(z_i|0, I)$. Here ϕ_i depends on the number of object, because for every object we want to get distribution of latent variable. We have two more problems here: 1) Using stochastic gradient descent, we will not update all parameters at the end of the epoch, thus it leads to slow convergence; 2) If we want to calculate latent distribution for a new object, we have to retrain our network, because we have no parameters ϕ_i for a new object. To solve both problems we can make dependencies on X and no dependecies on i, i.e. $Q(z_i|x_i, \phi)$. So, now ELBO looks like

$$
\mathcal{L}(\Theta, \phi) = \sum_{i=1}^{n} \left[\int_{z_i} Q(z_i|x_i,\phi) \log p(x_i|z_i,\Theta) dz_i - \mathbf{KL}(Q(z_i|x_i,\phi)||p(z_i)) \right] \tag{7}
$$

With assumptions:

$$
\begin{gathered}
Q(z_i|x_i,\phi) = \mathbf{N}(z_i|\mu(\phi, x_i), \sigma(\phi, x_i)) - \text{Encoder} \\
p(z_i) = \mathbf{N}(z_i|0, I) - \text{Prior latent distribution} \\
p(x_i|z_i,\Theta) = \mathbf{N}(x_i|f(z_i,\Theta), s^2 I) - \text{Decoder} \\
x_i, f(z_i,\Theta) \in \mathbb{R}^D; \; \mu, \sigma, z_i \in \mathbb{R}^d \\
\phi, \Theta - \text{Parameters of neural network} \\
(x_1 ..., x_n) - \text{Training data}
\end{gathered}
\tag{8}
$$



Lets calculate the second term of ELBO - $\mathbf{KL}(Q(z_i|x_i,\phi)||P(z_i))$

$$\begin{aligned}\mathbf{KL}(\mathbf{N}(z_i|\mu(x_i,\phi),\sigma(x_i,\phi))||\mathbf{N}(z_i|0,I)) = \int\limits_{z_i} \mathbf{N}(z_i|\mu,\sigma)\log\frac{\mathbf{N}(z_i|\mu,\sigma)}{\mathbf{N}(z_i|0,I)}dz_i = \\ = \int\limits_{z_i} \mathbf{N}(z_i|\mu,\sigma)\log\mathbf{N}(z_i|\mu,\sigma)dz_i - \int\limits_{z_i} \mathbf{N}(z_i|\mu,\sigma)\log\mathbf{N}(z_i|0,I)dz_i = \mathbf{I}_1 - \mathbf{I}_2\end{aligned} \tag{9}$$

Note that $\mathbf{cov}(z_i) = diag(\sigma)$

$$\begin{aligned}\mathbf{N}(z_i|\mu,\sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\mathbf{cov}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(z_i-\mu)^T\mathbf{cov}^{-1}(z_i-\mu)\right) = \\ = \frac{1}{(2\pi)^{\frac{d}{2}}\prod_{k=1}^d \sigma_k}\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right) \\ \mathbf{N}(z_i|0,I) = \frac{1}{(2\pi)^{\frac{d}{2}}}\exp\left(-\frac{1}{2}z_i^T z_i\right) = \frac{1}{(2\pi)^{\frac{d}{2}}}\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j)^2\right)\end{aligned} \tag{10}$$

Let's consider $\mathbf{I}_2$

$$\begin{aligned}\mathbf{I}_2 = \frac{1}{(2\pi)^{\frac{d}{2}}\prod_{k=1}^d \sigma_k}\int\limits_{z_i}\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)\log\left(\frac{1}{(2\pi)^{\frac{d}{2}}}\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j)^2\right)\right)dz_i = \\ = \frac{1}{(2\pi)^{\frac{d}{2}}\prod_{k=1}^d \sigma_k}\left[\int\limits_{z_i}\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)\log\left(\frac{1}{(2\pi)^{\frac{d}{2}}}\right)dz_i - \frac{1}{2}\int\limits_{z_i}\sum_{j=1}^d (z_i^j)^2\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)dz_i\right] = \\ = \log\left(\frac{1}{(2\pi)^{\frac{d}{2}}}\right) - \frac{1}{2(2\pi)^{\frac{d}{2}}\prod_{k=1}^d \sigma_k}\mathbf{J}\end{aligned} \tag{11}$$

$$\begin{aligned}\mathbf{J} = \int\limits_{z_i}\sum_{j=1}^d (z_i^j)^2\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)dz_i = \int\limits_{z_i}(z_i^1)^2\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)dz_i + ... + \\ + \int\limits_{z_i}(z_i^d)^2\exp\left(-\frac{1}{2}\sum_{j=1}^d (z_i^j-\mu_j)^2\frac{1}{\sigma_j^2}\right)dz_i = \mathbf{J_1} + ... + \mathbf{J_n}\end{aligned} \tag{12}$$

$$\begin{aligned}\mathbf{J_1} = \int\limits_{-\infty}^{\infty}(z_i^1)^2\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1 ... \int\limits_{-\infty}^{\infty}\exp\left(-\frac{1}{2}(z_i^d-\mu_d)^2\frac{1}{\sigma_d^2}\right)dz_i^d = \\ = \int\limits_{-\infty}^{\infty}(z_i^1)^2\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1(2\pi)^{\frac{d-1}{2}}\prod_{k=2}^d \sigma_k = \\ = (2\pi)^{\frac{d-1}{2}}\prod_{k=2}^d \sigma_k\int\limits_{-\infty}^{\infty}((z_i^1-\mu_1)^2 + 2\mu_1 z_i^1 - \mu_1^2)\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1 = \\ = (2\pi)^{\frac{d-1}{2}}\prod_{k=2}^d \sigma_k\left[\int\limits_{-\infty}^{\infty}(z_i^1-\mu_1)^2\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1 + 2\mu_1\int\limits_{-\infty}^{\infty}z_i^1\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1\right] - \\ -(2\pi)^{\frac{d-1}{2}}\prod_{k=2}^d \sigma_k\mu_1^2\int\limits_{-\infty}^{\infty}\exp\left(-\frac{1}{2}(z_i^1-\mu_1)^2\frac{1}{\sigma_1^2}\right)dz_i^1 = (2\pi)^{\frac{d}{2}}\prod_{k=1}^d \sigma_k(\sigma_1^2+\mu_1^2)\end{aligned} \tag{13}$$

$$
\begin{aligned}
\mathbf{J} &= (2\pi)^{\frac{d}{2}} \prod_{k=1}^{d} \sigma_k \sum_{j=1}^{d} (\sigma_j^2 + \mu_j^2) \\
\mathbf{I}_2 &= \log\left(\frac{1}{(2\pi)^{\frac{d}{2}}}\right) - \frac{1}{2}\sum_{j=1}^{d}(\sigma_j^2 + \mu_j^2)
\end{aligned}
\tag{14}
$$

Analogue

$$
\mathbf{I}_1 = \log\left(\frac{1}{(2\pi)^{\frac{d}{2}}}\right) - \frac{1}{2}\sum_{j=1}^{d}(1 + \log \sigma_j^2)
\tag{15}
$$

Finally

$$
\mathbf{KL}(\mathbf{N}(z_i|\mu(x_i,\phi),\sigma(x_i,\phi))||\mathbf{N}(z_i|0,I)) = -\frac{1}{2}\sum_{j=1}^{d}(1 + \log \sigma_j(x_i,\phi)^2 - \sigma_j(x_i,\phi)^2 - \mu_j(x_i,\phi)^2)
\tag{16}
$$

Our goal is to find a gradient for maximizing ELBO

$$
\begin{aligned}
\nabla_{\Theta,\phi}\mathcal{L}(\Theta,\phi) = \nabla_{\Theta,\phi}\sum_{i=1}^{n}\left[\int\limits_{z_i} Q(z_i|x_i,\phi)\log p(x_i|z_i,\Theta)dz_i + \frac{1}{2}\sum_{j=1}^{d}(1 + \log \sigma_j(x_i,\phi)^2 - \sigma_j(x_i,\phi)^2 - \mu_j(x_i,\phi)^2)\right] = \\
= \sum_{i=1}^{n}\nabla_{\Theta,\phi}\mathcal{L}_i(\Theta,\phi) - ?
\end{aligned}
\tag{17}
$$

We will use stochastic gradient for gradient estimation

$$
\nabla_{\Theta,\phi}\mathcal{L}(\Theta,\phi) \approx \frac{n}{m}\sum_{i=1}^{m}\nabla_{\Theta,\phi}\mathcal{L}_i(\Theta,\phi),\ \ i \sim \mathbf{U}(1,...,n)
\tag{18}
$$

We need to take gradient from intergral in ELBO, we cant just estimate this integral by Monte-Carlo. Lets make reparameterization trick

$$
\begin{aligned}
\int\limits_{z_i} Q(z_i|x_i,\phi)\log p(x_i|z_i,\Theta)dz_i = \int\limits_{\epsilon} p(\epsilon)\log p(x_i|\epsilon\sigma(x_i,\phi)+\mu(x_i,\phi),\Theta)d\epsilon \approx \\
\approx \frac{1}{V}\sum_{v=1}^{V}\log p(x_i|\epsilon_v\sigma(x_i,\phi)+\mu(x_i,\phi),\Theta),\ \ \epsilon_v \sim \mathbf{N}(\epsilon_v|0,I) \\
p(x_i|\epsilon_v\sigma(x_i,\phi)+\mu(x_i,\phi),\Theta) = \mathbf{N}(x_i|f_\Theta(\epsilon_v\sigma(x_i,\phi)+\mu(x_i,\phi)),s^2I)
\end{aligned}
\tag{19}
$$

Now everything is done, we are ready to write final optimization problem

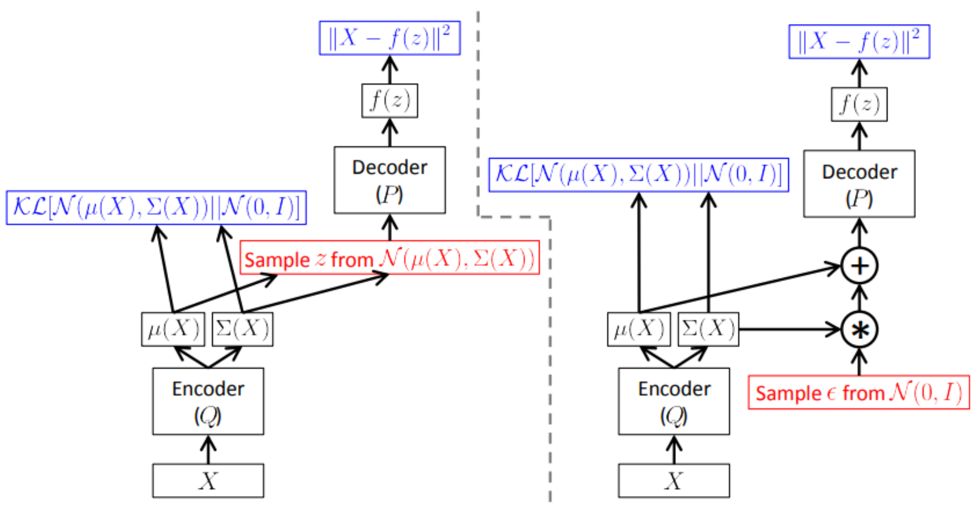
$$
\begin{aligned}
\nabla_{\Theta,\phi}\mathcal{L}(\Theta,\phi) \approx \frac{n}{m}\sum_{i=1}^{m}\nabla_{\Theta,\phi}\left[-\frac{1}{V}\sum_{v=1}^{V}||f_\Theta(\epsilon_v\sigma(x_i,\phi)+\mu(x_i,\phi)) - x_i||^2 + \frac{1}{2}\sum_{j=1}^{d}(1 + \log \sigma_j(x_i,\phi)^2 - \sigma_j(x_i,\phi)^2 - \mu_j(x_i,\phi)^2\right. \\
V - \text{number of sampling for Monte-Carlo estimation}; m - \text{batch size}; n - \text{training size}; \Theta,\phi - \text{parameters of NN}; \\
f_\Theta, x_i \in \mathbb{R}^D\ (D - \text{data dimension});\ \ \sigma_\phi,\mu_\phi \in \mathbb{R}^d\ (d - \text{latent dimension})
\end{aligned}
\tag{20}
$$

This procedure calls double stochastic derivation or variational Bayesian derivation. Schematically it's looks like Figure 1

References

[Doe16] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.


$\|X - f(z)\|^2$
$f(z)$
Decoder (P)
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$
Sample z from $\mathcal{N}(\mu(X), \Sigma(X))$
$\mu(X)$
$\Sigma(X)$
Encoder (Q)
X
$\|X - f(z)\|^2$
$f(z)$
$\mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)]$
Decoder (P)
$\mu(X)$
$\Sigma(X)$
Sample ϵ from $\mathcal{N}(0, I)$
Encoder (Q)
X