Variational Autoencoders – Theory

Thomas Marchioro

April 2023

Disclaimer These notes are just a summary of what we covered during the tutorial on variational autoencoders. For more details, refer to the original paper [1].

1 General pipeline

Figure 1 depicts the general pipeline of a variational autoencoder (VAE), which includes:

- An encoder neural network (Enc), which estimates the mean and log-standard deviation[1] (or log-variance) of the latent variable z for a given input x.
- A random number generator (RNG), which produces normally-distributed samples $\epsilon \sim \mathcal{N}(0, I)$. The values sampled by the RNG are reparametrized using the estimated means and standard deviations $z = \hat{\sigma} \odot \epsilon + \hat{\mu}$, so that $z \sim \mathcal{N}(\hat{\mu}, \text{diag}(\hat{\sigma}^2))$.
- A decoder neural network (Dec), which reconstructs $\hat{x}$ using z as input.

While both the encoder and the decoder neural networks are deterministic, the overall procedure is stochastic, since the latent variable is sampled using the RNG.

Training During training, each data point x of the dataset is first mapped to a random z using the encoder and the RNG, and then z is mapped to the reconstructed data point $\hat{x}$. The optimizer minimizes the loss function described in the next section w.r.t. the encoder and decoder parameters.

Generation A trained VAE can generate new data points using only the decoder. In order to do so, one must: (i) sample from an isotropic Gaussian $z \sim \mathcal{N}(0, I)$, and (ii) map z to a data point $\hat{x} = \text{Dec}(z)$.

[1] This is simply for convenience, since the $\log \hat{\sigma}$ can be both positive and negative, so it is easier to predict that and then take the exponential to obtain the actual standard deviation.

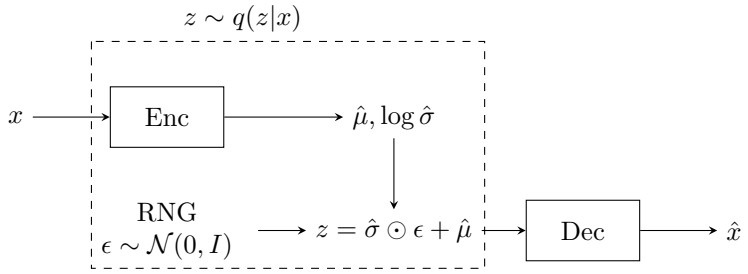

Figure 1: Pipeline of a variational autoencoder. The encoder estimates the mean and standard deviation of the latent variable $z \sim \mathcal{N}(\hat{\mu}, \text{diag}(\hat{\sigma}^2))$ for a given input x. The latent variable z is sampled using the reparametrization trick, sampling $\epsilon \sim \mathcal{N}(0, I)$ and computing $z = \hat{\sigma} \odot \epsilon + \hat{\mu}$. Finally, the decoder produces the reconstructed input $\hat{x}$.

2 Loss function

The loss function consists in two main components the reconstruction loss $\mathcal{L}_{\text{Rec}}$ and the Kullback-Leibler divergence (KLD) for the latent variable $\mathcal{L}_{\text{KLD}}$, which are summed to obtain the overall loss

$$\mathcal{L} = \mathcal{L}_{\text{Rec}} + \mathcal{L}_{\text{KLD}}. \tag{1}$$

The reconstruction loss can be computed assuming a specific distribution of $p(x|z)$. If we assume that $p(x|z)$ follows a Bernoulli distribution, this becomes the binary crossentropy loss (BCE)

$$\mathcal{L}_{\text{Rec}} = \underbrace{-x \log \hat{x} - (1-x) \log(1-\hat{x})}_{\text{BCE}}. \tag{2}$$

If instead we assume that $p(x|z)$ is Gaussian with fixed variance, we obtain a sum of squared error (SSE) loss

$$\mathcal{L}_{\text{Rec}} = \underbrace{\|x - \hat{x}\|^2}_{\text{SSE}}, \tag{3}$$

or, analogously, a mean squared error (MSE), it just changes by a multiplicative constant. The KLD for the latent variable, instead, is obtained assuming that the prior $p(z)$ of the latent variable is a standard isotropic Gaussian $\mathcal{N}(0, I)$, while the approximate posterior distribution $q(z|x)$ is $\mathcal{N}(\hat{\mu}, \text{diag}(\hat{\sigma}^2))$, with $\hat{\mu}, \hat{\sigma}^2$ being the parameters estimated by the encoder. This has a known expression, which is[2]

$$\mathcal{L}_{\text{KLD}} = D_{\text{KL}}(\mathcal{N}(0,1) \| \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)) = \underbrace{\frac{1}{2}\left(1 + \log \hat{\sigma}^2 - \hat{\mu} - \hat{\sigma}^2\right)}_{\text{Gaussian KLD}}. \tag{4}$$

Why the KLD loss? While the idea behind the reconstruction loss is intuitive (we want to make x and $\hat{x}$ as similar as possible), one may wonder what is the purpose of the KLD loss. The core idea is that minimizing this component of the loss brings the produced latent variables "closer" to a common distribution. Every time we reparametrize ϵ using the estimated $\hat{\mu}$ and $\hat{\sigma}$, we obtain that the latent variable z is sampled by the distribution $\mathcal{N}(\hat{\mu}, \text{diag}(\hat{\sigma}^2))$. This distribution varies for each input x, since the produced parameters are depend on the input. Minimizing the KLD loss ensures that the produced samples get gradually closer to the distribution $\mathcal{N}(0, I)$. This is necessary so that during the generation part, in which the latent variable z is sampled from $\mathcal{N}(0, I)$, z is mapped to a sample that some from the input distribution $p(x)$. Summing up, the goals of a VAE are:

- Correctly reconstruct the input x (i.e., minimize the difference with $\hat{x}$).
- Bring the latent variable close to a common distribution $\mathcal{N}(0, I)$.

2.1 Loss function derivation via ELBO

The analytical derivation of the loss function is based on the so-called evidence lower bound (ELBO). This is a lower bound to the log-likelihood $p(x)$ of the input data, which is obtained by introducing a latent variable z. The ELBO formula can be written as follows[3]:

$$\log p(x) \geq \underbrace{\int_{\mathcal{Z}} q(z|x) \log \frac{p(x,z)}{q(z|x)} dz}_{\text{ELBO}} \tag{5}$$

This expression can be further manipulated using the definition of conditional probability $p(x, z) =$

[2] I write the univariate expression for the sake of simplicity. In reality, you should sum this expression over all the components of z.

[3] This can be easily proven by introducing the latent variable via the law of total probability $\int_Z p(x,z)dz = p(x)$ and applying Jensen's inequality to find that $\log p(x) = \log[\int_Z p(x,z)\frac{q(z|x)}{q(z|x)}]dz = \log[\int_Z q(x|z)\frac{p(x,z)}{q(z|x)}]dz \geq \int_Z q(x|z) \log[\frac{p(x,z)}{q(z|x)}]dz$.

$p(x|z)p(z)$ and the properties of logarithms

$$\log p(x) \geq \int_{\mathcal{Z}} q(z|x) \log \frac{p(x,z)}{q(z|x)} dz \tag{6}$$

$$= \int_{\mathcal{Z}} q(z|x) \log \frac{p(x|z)p(z)}{q(z|x)} dz \tag{7}$$

$$= \int_{\mathcal{Z}} q(z|x) \left(\log p(x|z) - \log \frac{q(z|x)}{p(z)} \right) dz \tag{8}$$

$$= \underbrace{\int_{\mathcal{Z}} q(z|x) \log p(x|z) dz}_{-\mathcal{L}_{\text{Rec}}} - \underbrace{\int_{\mathcal{Z}} q(z|x) \log \frac{q(z|x)}{p(z)} dz}_{\mathcal{L}_{\text{KLD}} = D_{\text{KL}}(q(z|x) \| p(z))}. \tag{9}$$

This lower bound to the log-likelihood $p(x)$ implies that the negative log-likelihood is upper-bounded by

$$-\log p(x) = \mathcal{L}_{\text{Rec}} + \mathcal{L}_{\text{KLD}}, \tag{10}$$

which is our loss function. In other words, this is yet another maximum likelihood estimation approach.

What are $p(z)$, $q(z|x)$, and $p(x|z)$?

- $p(z)$ is the prior distribution of the latent variable. We choose this to be $\mathcal{N}(0, I)$.
- $q(z|x)$ is an approximation of the actual conditional distribution $p(z|x)$, which is intractable. Samples of $q(z|x)$ are obtained by computing $(\hat{\mu}, \log \hat{\sigma}) = \text{Enc}(x)$ and using the reparametrization trick on the RNG sample $\epsilon \sim \mathcal{N}(0, I)$.
- $p(x|z)$ is the likelihood of a datapoint x given the latent variable z, which is determined by the decoder. This is maximized during training in the reconstruction loss and it is typically assumed to be a Gaussian or Bernoulli distribution.

Reconstruction loss Above we called the term

$$\mathcal{L}_{\text{Rec}} = \int_{\mathcal{Z}} q(z|x) \log \frac{q(z|x)}{p(z)} dz \tag{11}$$

the "reconstruction loss". But how does minimizing this loss enforce reconstruction? Intuitively, if $\mathcal{L}_{\text{Rec}}$ is minimized that means that $p(x|z)$ is maximized for $z \sim q(z|x)$. In other words, maximizing this reconstruction loss means increasing the likelihood of the latent variable z produced by the stochastic encoder being mapped back into x by the decoder. If $p(x|z)$ is a Bernoulli distribution, that means

$$p(x|z) = \hat{x}^x (1 - \hat{x})^{1-x}, \text{ with } \hat{x} = \text{Dec}(z), z \sim q(z|x), \tag{12}$$

and

$$-\log p(x|z) = \underbrace{-x \log \hat{x} - (1 - x) \log(1 - \hat{x})}_{\text{BCE}}. \tag{13}$$

Likewise, if $p(x|z)$ is Gaussian $\mathcal{N}(\hat{x}, I)$, then

$$p(x|z) = \frac{1}{(\sqrt{2\pi})^d} e^{-\frac{1}{2}\|x - \hat{x}\|^2}, \text{ with } \hat{x} = \text{Dec}(z), z \sim q(z|x), \tag{14}$$

and

$$-\log p(x) = c_1 \underbrace{\|x - \hat{x}\|^2}_{\text{SSE}} + c_2. \tag{15}$$

The constant terms can be neglected in the loss, since it does not depend on the parameters of the encoder or decoder.

Notice that it must be $z \sim q(z|x)$, since in the reconstruction loss we have $-\int_Z q(z|x) \log p(x|z) dz$, meaning that $-\log p(x|z)$ is averaged over $q(x|z)$. For more details, see section C of [1].

3 Conditional VAEs

In conditional VAEs, the objective is to estimate different data distributions depending on a condition y. In the case of the MNIST dataset, for example, y represents the digit that we want to produce. In this case, the likelihood to be maximized is not $p(x)$, but $p(x|y)$ for each y. This can be done by deriving the ELBO for $p(x|y)$, obtaining a similar loss function to the unconditional case. In practice, the condition can be enforced by concatenating the one-hot encoding of y to the input of both the encoder and the decoder. During inference, one can concatenate the one-hot encoding of the desired digit to $z \sim \mathcal{N}(0, I)$.

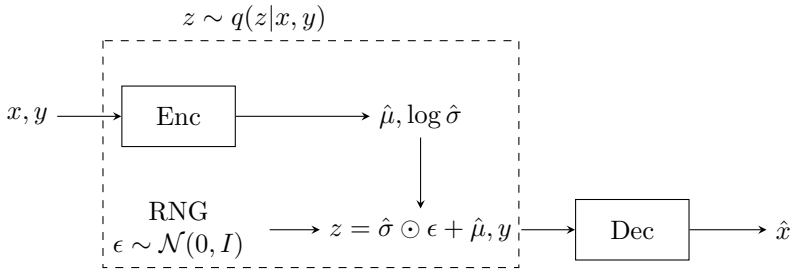

Figure 2: Pipeline of a conditional variational autoencoder. Both the encoder and the decoder receive the condition y as additional input.

References

[1] D. P. Kingma and M. Welling, "Auto-encoding variational bayes." `https://arxiv.org/pdf/1312.6114.pdf`, 2013.