Introduction to Information Theory

Lecture 12

Lecturer: Haim Permuter *Scribe: Tom Galili*

I. Variation Inference

The objective of Variation Inference is to estimate $P(z^m|x^n)$,

$$P(z^m|x^n) = \frac{P(z^m, x^n)}{P(x^n)} = \frac{P(z^m)P(x^n|z^m)}{\int P(z^m)P(x^n|z^m)\, dz^m} \tag{1}$$

- z^m - latent (hidden)
- x^n - observation evidence

Notice that the estimation of $P(z^m|x^n)$ is not trivial, therefore we simplify the term

$$\begin{aligned}
\arg\min_{q(z^m)\in\mathcal{Q}} D((q_{z^m})||P(z^m|x^n)) &\overset{(a)}{=} \arg\min \mathbb{E}_{q(z^m)}[\log q(z^m)] - \mathbb{E}_q\left[\log\frac{P(z^m, x^n)}{P(x^n)}\right] \\
&\overset{(b)}{=} \arg\min \mathbb{E}_q[\log q(z^m)] - \mathbb{E}_q[\log P(z^m, x^n)] + \log P(x^n) \\
&\overset{(c)}{=} \arg\min(-ELBO + \log P(x^n)) \qquad (2)
\end{aligned}$$

$$\begin{aligned}
-ELBO &= \mathbb{E}_q[\log q(z^m)] - \mathbb{E}_q[\log P(z^m, x^n)] \\
&\overset{(d)}{=} \mathbb{E}_q[\log q(z^m)] - \mathbb{E}_q[\log p(z^m)] - \mathbb{E}_q[\log P(x^n|z^m)] \\
&= \mathbb{E}_q[-\log P(x^n|z^m)] + D(q(z^m)||P(z^m)) \qquad (3)
\end{aligned}$$

where

(a) follows from the definition of divergence.

(b) follows from the logarithm rules.

(c) follows from the definition of evidence Lower Bound (ELBO) as defined in the previous lectures.

(d) follows from conditional probability.

Note that $P(z^m)$ is the prior probability and $q(z^m) \approx P(z^m|x^n)$ is the posterior probability (we want to estimate) of the latent space given the evidence. First interpretation: find maximum, we want to get as close as possible to the prior and on the other hand the probability of $q(z^m)$ will be greater as z^m gives more information about x^n:

$$\max(\mathbb{E}[\log(P(x^n|z^m))] - D(q(z^m)||P(z^m))) \tag{4}$$

Second interpretation: MLD - minimum description length: Description of x^n using z^m with as few as possible bits.

II. Auto Encoder (AE)

AutoEncoders are unsupervised learning models. The general idea of Auto Encoders consists of setting an encoder and a decoder as neural networks and learning the best encoding-decoding scheme using an iterative optimization process.[1]

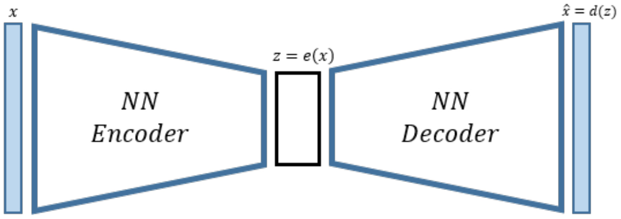

Fig. 1. Illustration of an Auto Encoder

In this way, the architecture creates an information bottleneck for the data that ensures only the main structured part of the information, with which it can be restored exactly well, can go through and be reconstructed. Therefore, we would like to use *dimension* *reduction* (feature reduction). In many cases, the data you want to analyze has a high dimension, which means that each sample has a large number of features. For the most part, not all characteristics are equally significant. Because it is difficult to analyze data from a high dimension and build models for such data, in many cases we will try to reduce the dimension of the data with as little information loss as possible. As illustrated in Fig. 1, after the encoder part of the neural network we get $z = e(x)$ which is the latent vector of the input, characterized by a lower dimension than the data, represented by the important features to be reconstructed in the encoder.

The AE model objective is a minimization of the recovery error between the input data and the reconstructed output data to be as small as possible,

$$Loss = ||x - \hat{x}||^2 = ||x - d(z)||^2 = ||x - d(e(x))||^2 \tag{5}$$

If the equality $x = d(e(x))$ holds then no information was lost in the encoder-decoder process. On the other hand, if $x \neq d(e(x))$ then some information is lost due to the dimension reduction and the complete reconstruction of the encoded information is not possible in the decoder.

III. Variational Auto Encoder (VAE)

Unlike AE which takes data and performs dimension reduction, VAE [3] determines a prior distribution to the latent space z, for example, Gaussian distribution $z \sim \mathcal{N}(0,\, I)$, when I - Identity covariance matrix. The encoder network is trained to receive data x and output $\mu(x), \sigma(x)$ parameters of z ($z \sim \mathcal{N}(\mu_x,\, \sigma_x)$), in order to minimizing as much as possible the distance between $P(z)$ and $P(z|x)$. Then sample vectors from $z|x$ (given by the parameters calculated in the encoder) and pass them through the decoder to produce parameters of the $P(x|z)$ [1] [2].

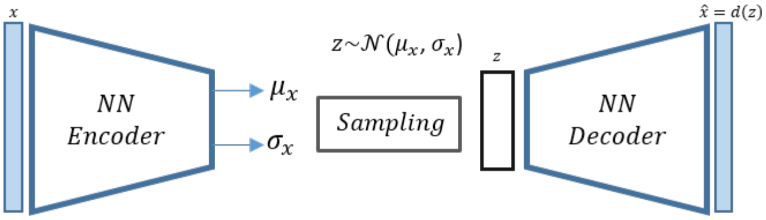

Fig. 2. Illustration of Variational Auto Encoder

It is important to mention that in comparison to the AE decoder part which uses for the training process only, the VAE decoder is important as the encoder since it uses to generate new data at inference time and to make the whole Variational Auto Encoder model to a generative model.

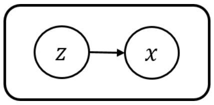

Fig. 3. Graphical model of the data generation process

Lets derived the loss function of VAE, first define the link between the encoder and the decoder as

$$P(z|x) \overset{(a)}{=} \frac{P(x|z)P(z)}{P(x)} \tag{6}$$

Where

(a) follows from the Bayes theorem.

$P(z|x)$ describes the distribution of the encoded variable given the input data.

$P(x|z)$ describes the distribution of the decoded variable given the encoded one.

The objective is to approximate $P(z|x)$ by a Gaussian distribution $q_x(z)$ whose mean and covariance are defined by two functions, g and h, of the parameter x.

$$\begin{aligned} P(x|z) &\sim \mathcal{N}(f(z),\, cI) \quad , \quad c > 0 \\ P(z) &\sim \mathcal{N}(0,\, I) \\ q_x(z) &\sim \mathcal{N}(g(x),\, h(x)) \end{aligned} \tag{7}$$

We are looking for the optimal g^* and h^* such that

$$\begin{aligned} (g^*, h^*) &= \arg\min_{g,h} D((q_x(z))||P(z|x)) \\ &\overset{(a)}{=} \arg\min_{g,h}\ [E_{z\sim q_x(z)}(\log q_x(z)) - E_{z\sim q_x(z)}\left(\log\frac{P(x|z)P(z)}{P(x)}\right)] \\ &\overset{(b)}{=} \arg\min_{g,h}\ [E_{z\sim q_x(z)}(\log q_x(z)) - E_{z\sim q_x(z)}(\log P(x,z))] \\ &= \arg\min_{g,h}\ [E_{z\sim q_x(z)}(\log q_x(z)) - E_q(\log P(z)) - E_q(\log P(x|z))] \\ &= \arg\min_{g,h}\ D(q||p) - E_q(\log P(x|z)) = \arg\min_{g,h}\ (-ELBO) \end{aligned} \tag{8}$$

Where

(a) follows from the definition of divergence and Bayes theorem.

(b) follows from the fact that $P(x)$ and q are independents.Thus, $P(x)$ Considered a constant and doesn't affect.

We know that

$$P(x|z) \sim \mathcal{N}(f(z),\, cI) \tag{9}$$

$$P(x|z) = \frac{1}{\sqrt{2\pi c}} e^{\frac{-(x-f(z))^2}{2c}} \quad , \quad P(z) \sim \mathcal{N}(0,\, I) \tag{10}$$

Thus,

$$(g^*, h^*) = \arg\min_{g,h}\; E_{z\sim q_x(z)} \left[\frac{(x-f(z))^2}{2c} \right] + D(\mathcal{N}(g(x),\, h(x))\,, \mathcal{N}(0,\, I)\,) \tag{11}$$

Now, by the Maximum likelihood principle,

$$\begin{aligned} max_f(E_{z\sim q_x(z)}(\log(P(x|z))) &= max_f(E_{z\sim q_x(z)}[\log(\mathcal{N}(f(z,\, cI))\,]) \\ &= max_f(E_{z\sim q_x(z)} \left[-\frac{(x-f(z))^2}{2c} \right]) \end{aligned} \tag{12}$$

Gathering all the pieces together, we are looking for optimal f^*, g^* and h^* such that

$$(g^*, h^*, f^*) = \arg\min_{g,h,f}\; E_{z\sim q_x(z)} \left[\frac{(x-f(z))^2}{2c} \right] + D(\mathcal{N}(g(x),\, h(x))\,, \mathcal{N}(0,\, I)\,) \tag{13}$$

In Eq. (13), we get two terms: The first one for the reconstruction of x using the decoder part, and the second term use the KL divergence to approximating the posterior $P(z|x)$ to be close to the prior probability $P(z)$. The overall architecture is then obtained by concatenating the encoder and the decoder parts and we can use gradient descent optimization to find the optimal parameters of the VAE encoder and decoder and the loss function is well defined as

$$Loss = E_{z\sim q_x(z)} \left[\frac{(x-f(z))^2}{2c} \right] + D(\mathcal{N}(g(x),\, h(x))\,, \mathcal{N}(0,\, I)\,) \tag{14}$$

The second term can also be treated as a regularisation term given by the KL divergence between two Gaussian distributions which helps the VAE model's encoder approximation of the posterior probability to be close to the prior probability (which is a standard Gaussian). We can also notice the constant c that rules the balance between the two previous terms. When c is bigger, we assume a high variance around $f(z)$ for the probabilistic decoder of the VAE, and we are more like to favor the regularisation term over the reconstruction term. Opposite stands if c is low.

A. Reparametrization Trick

We note that we still need to be very careful about the way we sample from the distribution returned by the encoder during the training. The sampling process has to be expressed in a way that allows the error to be backpropagated through the network to compute the gradients for the Gradient Descent process as part of the training. Thus, the *reparametrization trick* [1] is used as illustrated in Fig. 4 to make the gradient descent possible despite the random sampling that occurs halfway through the architecture (after the encoder). Using the fact that z is a random variable following a Gaussian distribution with $g(x)$ (mean) and $h(x)$ (covariance) then it can be expressed as

$$z = g(x) + \zeta h(x) \quad , \quad \zeta \sim \mathcal{N}(0, I) \tag{15}$$

In this approach the whole process becomes deterministic - sample ζ in advance and then only remains to schematically calculate the spread of the value in the network.

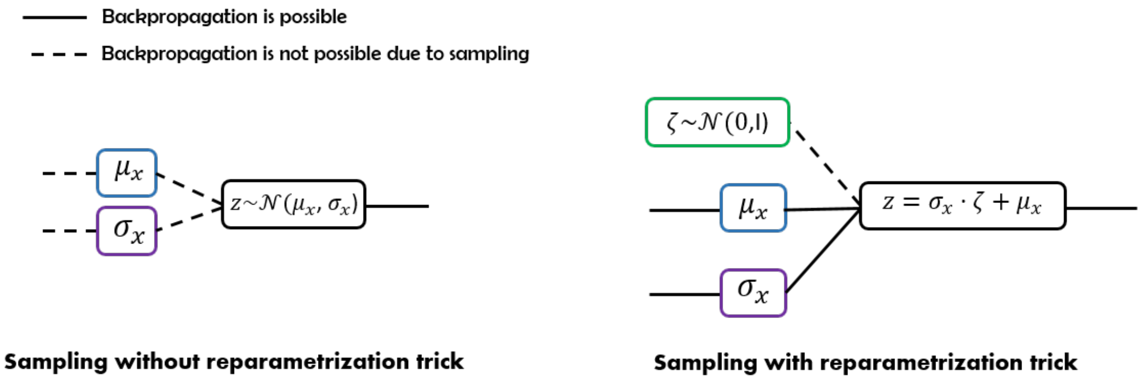
Backpropagation is possible
Backpropagation is not possible due to sampling
μ_x
σ_x
$z \sim \mathcal{N}(\mu_x, \sigma_x)$
Sampling without reparametrization trick
$\zeta \sim \mathcal{N}(0,I)$
μ_x
σ_x
$z = \sigma_x \cdot \zeta + \mu_x$
Sampling with reparametrization trick

Fig. 4. Illustration of reparametrization trick

REFERENCES

[1] Joseph Rocca. *Understanding Variational Autoencoders (VAEs)*. 2019. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[2] Carl Doersch. *Tutorial on Variational Autoencoders*. Carnegie Mellon / UC Berkeley. https://arxiv.org/pdf/1606.05908.pdf.

[3] Pu, Y., Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. *"Variational autoencoder for deep learning of images, labels and captions"*. In: Advances in Neural Information Processing Systems. 2016.