Auto-Encoding Variational Bayes

Xinghu Yao

September 26, 2018

1 Problem Establishment

Using auto-encoding variational Bayes we can perform efficient approximate inference and learning with directed probabilistic models whose continuous latent variables and/or parameters have intractable posterior distributions.

Let us consider some dataset $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ consisting of N i.i.d samples of some continuous or discrete variable $\mathbf{x}$. We assume that the data are generated by some random process, involving an unobserved continuous random variable $\mathbf{z}$. The process consists of two steps: (1) a value $\mathbf{z}^{(i)}$ is generated from some prior distribution $p_{\boldsymbol{\theta}}(\mathbf{z})$. (2) a value $\mathbf{x}^{(i)}$ is generated from some conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

We can use two networks to train a generative model and the two parts of our networks have the following relationship.

$$\mathbf{x} \xrightarrow{F_{\boldsymbol{\theta}}} (\mathbf{z}|\mathbf{x}) \xrightarrow{G_{\boldsymbol{\theta}}} \hat{\mathbf{x}} \tag{1}$$

where $F_{\boldsymbol{\theta}}$ is a network to cover the latent hidden variable $\mathbf{z}$ and $G_{\boldsymbol{\theta}}$ is another network to decode $\mathbf{x}$ using the hidden variable $\mathbf{z}$. Thus, the marginal likelihood of this structure can be written as:

$$\begin{aligned}
\ln p(\mathbf{x}) &= \ln \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \\
&= \ln \int \frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}p(\mathbf{z})q(\mathbf{z}|\mathbf{x})\mathrm{d}\mathbf{z} \\
&= \ln \mathop{\mathbb{E}}_{\mathbf{z}\backsim(\mathbf{z}|\mathbf{x})} \left(\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}p(\mathbf{z})\right) \\
&\geq \mathop{\mathbb{E}}_{\mathbf{z}\backsim q(\mathbf{z}|\mathbf{x})} \ln \left(\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}p(\mathbf{z})\right) && \text{(Jensen Inequality)} \\
&= \mathop{\mathbb{E}}_{\mathbf{z}\backsim q(\mathbf{z}|\mathbf{x})} \ln \frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}|\mathbf{x})} + \mathop{\mathbb{E}}_{\mathbf{z}\backsim q(\mathbf{z}|\mathbf{x})} \ln p(\mathbf{z}) \\
&= \int q(\mathbf{z}|\mathbf{x})\ln p(\mathbf{x}|\mathbf{z})\mathrm{d}\mathbf{z} - \mathrm{KL}\left[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right] && (2)
\end{aligned}$$

where $q(\mathbf{z}|\mathbf{x})$ is the probabilistic encoder, since given a datapoint $\mathbf{x}$ it produce a distribution over the possible values of the code $\mathbf{z}$ from which the datapoint $\mathbf{x}$ could have been generated. In a similar vein we will refer to $p(\mathbf{x}|\mathbf{z})$ as a probabilistic decoder, since given a code $\mathbf{z}$ it produce a distribution over the possible corresponding values of $\mathbf{x}$. Thus, we can get the following optimization problem

$$\max_{q(\mathbf{z}|\mathbf{x})} \mathop{\mathbb{E}}_{\mathbf{z}\backsim q(\mathbf{z}|\mathbf{x})} \left[\ln p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}\left[q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right] \tag{3}$$

2 The reparameterization trick

In order to solve our problem we invoke an alternative method for generating samples from $q(\mathbf{z}|\mathbf{x})$. The essential parameterization trick is quite simple. Let $z \backsim p(z|x) = \mathcal{N}(\mu, \sigma^2)$. In this case, a valid reparameterization is $z = \mu + \sigma\epsilon$, where ϵ is an auxiliary noise variable $\epsilon \backsim \mathcal{N}(0, 1)$