# Variational Autoencoder

2019年5月8日 22:48

**Variational Autoencoder(VAEs)**

Good essays about about Variational Autoencoder:
Pixel Art generation using VAE: https://mlexplained.wordpress.com/category/generative-models/vae/
This Bolg is Good: Tutorial - What is a variational autoencoder?: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/
Paper "Tutorial on Variational Autoencoders": https://arxiv.org/pdf/1606.05908.pdf
Paper "Auto-Encoding Variational Bayes": https://arxiv.org/pdf/1312.6114.pdf

**Latent Variable**
Intuitively, it helps if the model first decides which character to generate before it assigns a value to any specific pixel. This kind of decision if formally called a **latent variable**.

**Variational lower bound**

To train this model, we have to approximate $P_\theta(X)$. One way to do that is by finding a feasible lower bound $L_\theta(X)$ for $P_\theta(X)$. Thus, by maximizing the lower bound, we hope to maximize the likelihood.

One way to derive this lower bound is to notice that the joint distribution $P_\theta(X, Z)$ can be written in the following two ways:

$$P_\theta(X)P_\theta(Z|X) = P_\theta(Z)P_\theta(X|Z)$$

Notice if we believed that $P_\theta(X)$ is computationally infeasible, then it implies $P_\theta(Z|X)$ should also be computationally infeasible. The obvious way of computing $P_\theta(Z|X)$ requires the computation of $P_\theta(X)$.

The second key idea is to approximate $P_\theta(Z|X)$ by some feasible distribution $Q_\phi(Z|X)$

$$P_\theta(X)Q_\phi(Z|X)\frac{P_\theta(Z|X)}{Q_\phi(Z|X)} = P_\theta(Z)P_\theta(X|Z)$$

The term $\frac{P_\theta(Z|X)}{Q_\phi(Z|X)}$ represents the approximation error.

By taking the log of the last formula, and rearranging the terms, we have:

$$\log P_\theta(X) + \log \frac{P_\theta(Z|X)}{Q_\phi(Z|X)} = \log \frac{P_\theta(Z)}{Q_\phi(Z|X)} + \log P_\theta(X|Z)$$

This formula holds for any $X, Z$ such that $P(X, Z) > 0$. Thus, we can take the expectation of the two sides with respect to an appropriate distribution of $Z$. One natural way is to use $Q_\phi(Z|X)$. This is so because, given our formulation, it represents the closest distribution we know to the infeasible $P_\theta(Z|X)$. Thus, it results in the tightest bound:

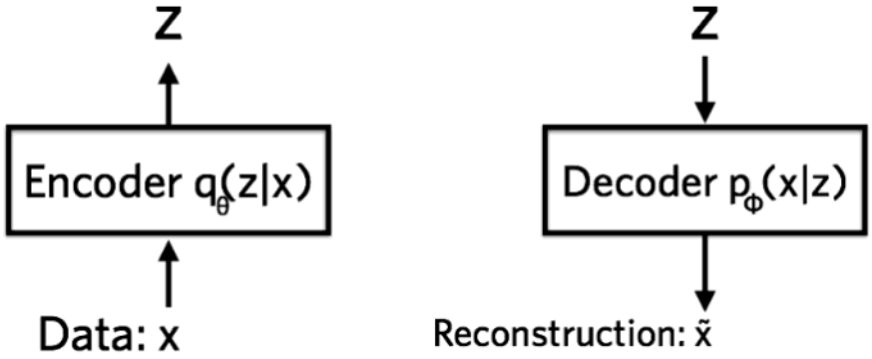
$$\log P_\theta(X) - KL(Q_\phi(Z|X)||P_\theta(Z|X)) = -KL(Q_\phi(Z|X)||P_\theta(Z)) + \mathbb{E}_{Q_\phi(Z|X)}\{\log P_\theta(X|Z)\}$$

We can see from the last formula that its right hand side is a lower bound for the marginal $\log P_\theta(X)$. Let's denote it by $L(\theta, \phi)$

$$L(\theta, \phi) = -KL(Q_\phi(Z|X)||P_\theta(Z)) + \mathbb{E}_{Q_\phi(Z|X)}\{\log P_\theta(X|Z)\}$$

**The neural net perspective**

Variational autoencoders can design complex generative models of data, and fit them to large datasets. They can generate images of fictional celebrity faces and high-resolution digital artwork. These models also yield state-of-the-art machine learning results in **image generation** and **reinforcement learning**.



In neural net language, a variational autoencoder consists of an encoder, a decoder, and a loss function. The encoder is a neural network. Its input is a data point x, its output is a hidden representation z, and it has weights and biases θ. The decoder is another neural net. Its input is the representation z, it outputs the parameters to the probability distribution of the data, and has weights and biases φ. The loss function of the variational autoencoder is the negative log-likelihood with a regularizer. The loss function li for data point xi:

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

The first term is the reconstruction loss, or expected negative log-likelihood of the i-th data point. The expectation is taken with respect to the encoder's distribution over the representations. This term encourages the decoder to learn to reconstruct the data. If the decoder's output does not reconstruct the data well, statistically we say that the decoder parameterizes a likelihood distribution that does not place much probability mass on the true data. For example, if our goal is to model black and white images and our model places high probability on there being black spots where there are actually white spots, this will yield the worst possible reconstruction. Poor reconstruction will incur a large cost in this loss function.

The second term is a regularizer that we throw in (we'll see how it's derived later). This is the Kullback-Leibler divergence between the encoder's distribution q(z|x) and p(z). This divergence measures how much information is lost when using qq to represent pp. It is one measure of how close qq is to pp.

In the variational autoencoder, p is specified as a standard Normal distribution with mean zero and variance one, or p(z) = Normal(0, 1). If the encoder outputs representations, it will receive a penalty in the loss. This regularizer term means 'keep the representations z of each digit sufficiently diverse'. If we didn't include the regularizer, the encoder could learn to cheat and give each data point a representation in a different region of Euclidean space. We want the representation space of z to be meaningful, so we penalize this behavior. This has the effect of keeping similar numbers' representations close together.

We train the variational autoencoder using gradient descent to optimize the loss with respect to the parameters of the encoder and decoder θ and φ. For stochastic gradient descent with step size ρ, the encoder parameters are updated using θ←θ - ρ * ∂θ/∂l and the decoder is updated similarly.

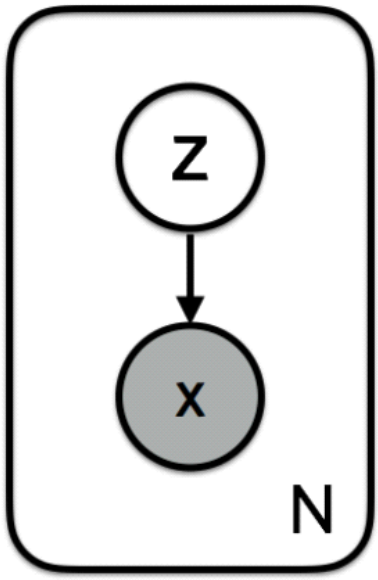
The probability model perspective

In the probability model framework, a variational autoencoder contains a specific probability model of data x and latent variable z. We can write the joint probability of the model as p(x , z) = p(x | z)p(z).

For each datapoint $i$:

- Draw latent variables $z_i \sim p(z)$
- Draw datapoint $x_i \sim p(x \mid z)$

We can represent this as a graphical model:




For black and white digits, the likelihood is **Bernoulli distributed**. Bayes says:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Examine the denominator $p(x)$. This is called the evidence, and we can calculate it by marginalizing out the latent variables: $p(x) = \int p(x \mid z)p(z)dz$. Unfortunately, this integral requires exponential time to compute as it needs to be evaluated over all configurations of latent variables. We therefore need to approximate this posterior distribution.

Variational inference approximates the posterior with a family of distributions $q_\lambda(z \mid x)$. The variational parameter $\lambda$ indexes the family of distributions. For example, if $q$ were Gaussian, it would be the mean and variance of the latent variables for each datapoint $\lambda_{x_i} = (\mu_{x_i}, \sigma^2_{x_i}))$.

How can we know how well our variational posterior $q(z \mid x)$ approximates the true posterior $p(z \mid x)$? We can use the Kullback-Leibler divergence, which measures the information lost when using $q$ to approximate $p$ (in units of nats):

$$\mathbb{KL}(q_\lambda(z \mid x) \,||\, p(z \mid x)) =$$
$$\mathbf{E}_q[\log q_\lambda(z \mid x)] - \mathbf{E}_q[\log p(x, z)] + \log p(x)$$

Our goal is to find the variational parameters $\lambda$ that minimize this divergence. The optimal approximate posterior is thus

$$q_\lambda^*(z \mid x) = \arg\min_\lambda \mathbb{KL}(q_\lambda(z \mid x) \mid\mid p(z \mid x)).$$

Why is this impossible to compute directly? The pesky evidence $p(x)$ appears in the divergence. This is intractable as discussed above. We need one more ingredient for tractable variational inference. Consider the following function:

$$ELBO(\lambda) = \mathbf{E}_q[\log p(x, z)] - \mathbf{E}_q[\log q_\lambda(z \mid x)].$$

Notice that we can combine this with the Kullback-Leibler divergence and rewrite the evidence as

$$\log p(x) = ELBO(\lambda) + \mathbb{KL}(q_\lambda(z \mid x) \mid\mid p(z \mid x))$$

By **Jensen's inequality**, the Kullback-Leibler divergence is always greater than or equal to zero. This means that minimizing the Kullback-Leibler divergence is equivalent to maximizing the **ELBO**. The abbreviation is revealed: **the Evidence Lower BOund allows** us to do approximate posterior inference. We are saved from having to compute and minimize the Kullback-Leibler divergence between the approximate and exact posteriors. Instead, we can maximize the ELBO which is equivalent (but computationally tractable).

In the variational autoencoder model, there are only local latent variables (no datapoint shares its latent zz with the latent variable of another datapoint). So we can decompose the ELBO into a sum where each term depends on a single datapoint. This allows us to use stochastic gradient descent with respect to the parameters λ(**important: the variational parameters are shared across datapoints**). The ELBO for a single datapoint in the variational autoencoder is:

$$ELBO_i(\lambda) = \mathbb{E}q_\lambda(z \mid x_i)[\log p(x_i \mid z)] - \mathbb{KL}(q_\lambda(z \mid x_i) \mid\mid p(z)).$$