Variational Encoder

It consists an encoder,a decoder and a loss function.

Basically, it compresses data into a latent space, then decoder reconstructs the data given the hidden representation.

Encoder

For example , the input is a 28x28 pixel image, which is a 784-dimensional vector, then encoder compresses it into a lower dimension vector,typically referred to as a 'bottleneck'. So lets denote the encoder as

$$q_\theta(z|x)$$

the lower-dimensional space is stochastic,which is a Gaussian probability density.

So we can sample from this distribution to get noisy values of the representations z.

Decoder

The decoder is denoted by

$$p_\phi(x|z)$$

Let take MNIST for example,the photo are black and white and each pixel is represented as 0 or 1. **Then the probability distribution of a single pixel can be represented using a Bernoulli distribution.**

The decoder decodes the z back into a 784 dimension vector.Note that Information is lost.

But how to measure the loss of the information? We can use the reconstruction log-likelihood

$$log_{p_\phi}(x|z)$$

Loss Function

The loss function of the VAE is the negative log-likelihood with a regularizer. Since there are no global representations that are shared by all datapoints, we can decompose the loss function into terms that depend on a single datapoint li .So the total loss is then $\sum_{i=1}^{N} l_i$ for N total datapoints.

The loss function l_i for datapoint x_i is

$$l_i(\theta,\phi) = -E_{z-q_\theta(z|x_i)}[log_{p_\phi}(x_i|z)] + KL(q_\theta(z|x_i)||p(z))$$

The first term is the reconstruction loss, this term encourages the decoder to learn how to reconstruct data.

The second term is called Kullback-Leibler divergence, which measures how much information is lost when using q to represent p.

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 z that are different than those from a standard normal distribution, it will receive a penalty in the loss.This regularizer term means 'keeps the representations z of each digit sufficiently diverse'

If we did not include the regularizer,the encoder could learn to cheat and give each datapoint a representation in a different region of Euclidean space.

We want the representation space of z to be meaningful, so we penalize this behaviour. This equals to keeping similar number's representations close together.

Training Perspective

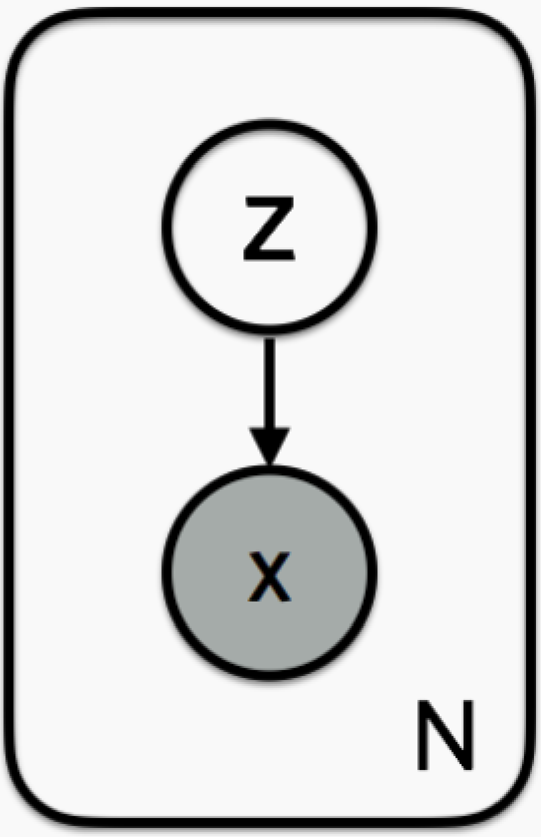
Use gradient descent to optimize the loss with respect to the parameters of the encoder and decoder θ and ϕ

probability model perspective

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:



THE GRAPHICAL MODEL REPRESENTATION OF THE MODEL IN THE VARIATIONAL AUTOENCODER. THE LATENT VARIABLE Z IS A STANDARD NORMAL, AND THE DATA ARE DRAWN FROM P(X|Z). THE SHADED NODE FOR X DENOTES OBSERVED DATA. FOR BLACK AND WHITE IMAGES OF HANDWRITTEN DIGITS, THIS DATA LIKELIHOOD IS BERNOULLI DISTRIBUTED.

The joint probability of the model can be written as $p(x,z) = p(x|z)p(z)$

here $p(z)$ is a prior.

As for the inference state of the model: the goal is to infer good values of the latent variables given observed data, or to calculate the posterior $p(z|x)$

Bayes says:

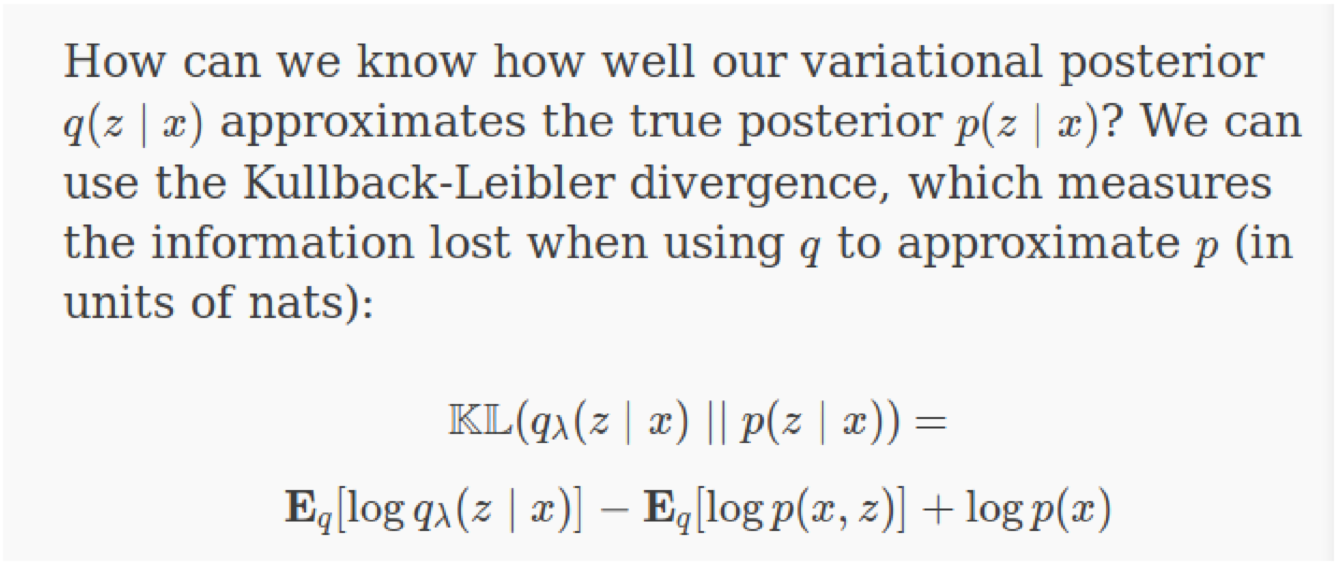
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

The denominator p(x) is called the evidence, we can calculate it by marginalizing out the latent variables:p(x) = $\int p(x|z)p(z)dz$

But the integral requires exponential time to compute as it needs to be evaluated over all configurations of latent variables.

Variational inference approximates the posterior with a family of distributions

$q_\lambda(z|x)$ The variational parameter λ indexes the family of distributions. For example,if q is Gaussian,it would be the mean and variance of the latent variables for each datapoint $\lambda_{xi} =$ （$\mu_{xi}, \sigma^2_{x_i}$）



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):

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

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

Our goal is to find the variational parameters λ that minimize this divergence.

The optimal approximate posterior is thus

$$q^*_\lambda(z|x) = argmin_\lambda KL(q_\lambda(z|x)||p(z|x))$$

But the above is impossible to compute directly, as mentioned above,p(z|x) is intractable.

So we need one more ingredient for tractable variational inference. Consider the following:

$$ELBO(\lambda) = E_q[logp(x,z)] - E_q[log_{q\lambda}(z|x)]$$

we can combine this with KL divergence and rewrite the evidence as

$$logp(x) = ELBO(\lambda) + KL(q_\lambda(z|x)||p(z|x))$$

The we can decompose the ELBO into a sum where each term depends on a single datapoint.

$$ELBO(\lambda) = Eq_\lambda(z|x_i)[logp(x_i|z)] - KL(q_\lambda(z|x)||p(z|x))$$

We parametrize the approximate posterior $q_\theta(z|x)$ with an inference network.

and parametrize the likelihood $p(x|z)$ with a generative network.

The inference and generative networks have parameters θ and ϕ respectively, which typically are weights and biases of the neural nets.

we then optimize these to maximize the ELBO using stochastic gradient descent.

$$ELBO(\lambda) = Eq_\lambda(z|x_i)[logp(x_i|z)] - KL(q_\lambda(z|x)||p(z|x))$$

This evidence lower bound is the negative of the loss function for VAE

which means

$$ELBO_i(\theta, \phi) = -l_i(\theta, \phi)$$

Above the KL term can still be seen as a regularizer, which minimize the divergence of

approximate q_λ and model posterior $p(z|x)$

总结一下，因为求**p(z|x)**的时候遇到了一些小问题，具体来讲就是因为其引入了积分，见上，时间复杂度指数级增长，为了对付这个问题，使用了一个$q_\lambda(z|x)$来模拟这个**inference**分布，那么该怎么衡量这个模拟的分布与实际分布的误差呢？于是引入了**KL**散度，用来衡量两个分布的差异，这里的**KL**散度可以看作是一个正则化项，用来减少两者之间的误差。

Ref

https://jaan.io/what-is-variational-autoencoder-vae-tutorial/