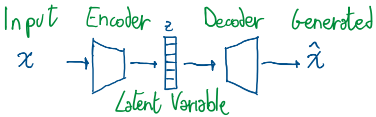
Friday, 29 November 2024 17:51

Variational Autoencoder

Proposed by Kingma & Welling in 2014.

Encoder and Decoder realized by two neural networks



The latent varible z encapsulates the information used to desribe the input.
At the same time, z, is the "seed" used by the decoder to generate $\hat{x}$.
Two different z's should give us two different generated outputs.

Definition: Latent Variables

In a probabilistic model, Latent variables (z) are unobserved variables that are not directly available in the training dataset but are included as a part of the model's structure to capture hidden or underlying factors influencing the observed data.

A bunch of new distributions.

$p(x)$: The true distribution of x. Unknown. if we knew $p(x)$ in order to generate we could draw a sample x to maximize $\log p(x)$.

Self information: $I(x) = -\log p(x)$

Low probability → High Information. rare events are more surprising
High probability → Low Information. Common events are predictable
↓ unsurprising

Entropy: $H(p) = \mathbb{E}_{x\sim p(x)}\left[-\log p(x)\right]$ Is the average amount of information (also the overall uncertainty) of $p(x)$

$= -\sum_x p(x)\log p(x)$

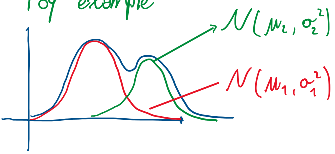
$p(z)$ Distribution of the latent variable. Usually $\mathcal{N}(0, I)$. Any distribution can be generated by mapping a Gaussian through a sufficiently complicated function (e.g. neural network)

$p(z|x)$ Conditional distribution associated to the encoder. We do not have access to it an it is not the encoder itself. Likelihood of z when given x.

$p(x|z)$ Conditional distribution associated to the decoder. Posterior probability of getting x given z.

Deep Latent Variables: $p(z)$ $p(z|x)$ $p(x|z)$ parametrized by neural networks

Toy example



X random variable distributed according to a gaussian Mixture model with latent variable $k \in \{1,2\}$ denoting the cluster identity $p_z(k) = P[z=k] = \pi_k$ for k in $\{1,2\}$.

Assuming $\sum_{k=1}^{2} \pi_k = 1$ if we need to look only in cluster 1 we look at the conditional distribution

$$p_{x|z}(x|1) = \mathcal{N}(x|\mu_1, \sigma_1^2 I)$$

The total probability instead:

$$p_x(x) = \sum_{k=1}^{2} p_{x|z}(x|k)\, p_z(k) = \sum_{k=1}^{2} \pi_k \mathcal{N}(x|\mu_k, \sigma_k^2 I)$$

In this scenario, given $p_x(x)$ we want to learn an encoder such that for every sample $x \sim p_x(x)$ the latent code $z \in \{1,2\}$ with a distribution $z \sim p_z(k)$.

Encoder How do we obtain z from x? $z = \begin{cases} 1 & \text{if } p_{z|x}(1|x) > p_{z|x}(2|x) \\ 2 & \text{otherwise} \end{cases}$

Decoder Since we know means and stds we just need to sample from the conditional distribution

$$p_{x|z}(x|k) = \mathcal{N}(x|\mu_k, \sigma_k^2 I).$$

A different z gives one of the K mixture components

We use an encoder parametrized by ϕ as a proxy for $p(z|x)$.

$$(\mu, \sigma^2) = \text{Encoder}_\phi(x) \to q_\phi(z|x) = \mathcal{N}(z|\mu, \underset{\substack{\downarrow \\ \sigma^2 I}}{\text{diag}(\sigma^2)})$$

We use a decoder parametrized by θ as a proxy for $p(x|z)$

$f_\theta(z) = \text{Decoder}_\theta(z) \longrightarrow p_\theta(x|z) = \mathcal{N}(x \mid f_\theta(z), \sigma_g^2 I)$

↘ hyperparameter

Trivial and linear cases.

$x \in \mathbb{R}^d \quad z \in \mathbb{R}^d \qquad x \sim p(x) = \mathcal{N}(x \mid \mu, \sigma^2 I)$

$\qquad\qquad\qquad\qquad z \sim p(z) = \mathcal{N}(z \mid 0, I)$

with p(x) known $\qquad z = \frac{(x-\mu)}{\sigma} \qquad \hat{x} = \mu + \sigma z$

we can use the delta function to express the conditionals

$$p(x|z) = \delta(x - (\mu + \sigma z))$$

$$p(z|x) = \delta\left(z - \frac{(x-\mu)}{\sigma}\right)$$

If p(x) is unknow we could use a liner layer for the mean and a scalar for the variance.

$$q_\phi(z|x) = \mathcal{N}(z \mid ax+b, t^2 I)$$

$$p_\theta(x|z) = \mathcal{N}(x \mid cz+v, s^2 I).$$

But how do we obtain the best parameters?

Our goal is to maximize log p(x) which is unknown. We need to decompose it in tractable terms.

$$\log p(x) = \log p(x) \underbrace{\int q_\phi(z|x)\,dz}_{=1} \qquad \text{Multiply by } 1$$

$$= \int \underbrace{\log p(x)}_{\text{constant wrt } z} \underbrace{q_\phi(z|x)}_{\text{distribution in } z}\,dz \qquad \int a\, p_z(z)\,dz = E[a] = a$$

$$= E_{q_\phi(z|x)}[\log p(x)] \qquad \text{we introduced the encoder!}$$

Now we use Bayes to introduce $p(x|z)$

$$p(x,z) = p(z|x)\,p(x)$$

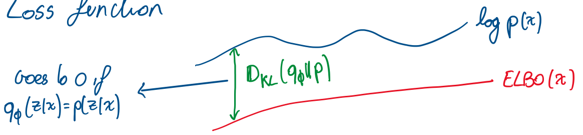
$$= E_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{p(z|x)}\right]$$

Lets Multiply by 1 again

$$= E_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{p(z|x)} \frac{q_\phi(z|x)}{q_\phi(z|x)}\right]$$

$$= \underbrace{E_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]}_{\text{ELBO}} + \underbrace{E_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p(z|x)}\right]}_{D_{KL}(q_\phi(z|x) \,\|\, p(z|x))}$$

The Evidence Lower BOund will be our Loss function



$$ELBO(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]$$

Let's use Bayes again

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x|z)\,p(z)}{q_\phi(z|x)}\right]$$

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log p(x|z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z|x)}\right]$$

we need to replace the intractable $p(x|z)$ with $p_\theta(x|z)$ our decoder

$$= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(z|x) \| p(z)\right)$$

Reconstruction

Decoder (how good it is)

Maximize the likelihood.

The expectation tells you that z cannot be an arbitrary noise vector but must be sampled from $q_\phi(z|x)$ the learned latent space

Prior Matching (Regularizer)

Encoder (how good it is)

The encoder turns x into z such that z follows our target distribution $\mathcal{N}(0, \sigma^2 I)$

Limitation of ELBO.

Let's use our toy example. $x \in \mathbb{R}^d$ $z \in \mathbb{R}^d$. Let's assume that we know the unbiased mean but not the variance. The mean is consistent with the delta function of $p(x|z)$ and $p(z|x)$

$$q_\phi(z|x) = \mathcal{N}\left(z \,\middle|\, \frac{x-\mu}{\sigma}, t^2 I\right)$$

$$p_\theta(x|z) = \mathcal{N}\left(x \,|\, \sigma z + \mu, s^2 I\right).$$

To Maximize the ELBO we compute the derivatives respect to t and s.

The derivative of the $D_{KL}(q_\phi(z|x) \| p(z))$ has closed form solution, because we have two multivariate Gaussian.

We find that $t = \frac{1}{\sqrt{d}}$

Similarly taking the derivative with respect to s and setting it to zero gives us $s = \frac{\sigma}{\sqrt{d}}$.

Therefore

$$q_\phi(z|x) = \mathcal{N}\left(z \,\middle|\, \frac{x-\mu}{\sigma}, \frac{1}{d} I\right)$$

$$p_\theta(x|z) = \mathcal{N}\left(x \,\middle|\, \sigma z + \mu, \underline{\frac{\sigma^2}{s}} I\right)$$

The variance is finite but non zero.
This because the ELBO is not exactly $\log p(x)$.

These finite variances add randomness to the process.
Interestingly it scales linearly with $\frac{1}{d}$ where d is the dimension of the input. Therefore when d is large q_ϕ and p_θ will asymptotically approach the true distributions

Optimization in VAE

Goal : approximate the true distribution $p(x)$

How : For a VAE parametrized by $(\phi\ \theta)$ we optimize the lower bound of $\log p(x)$

$$(\phi\ \theta) = \operatorname{argmax}_{\phi\theta} \sum_{x\sim\mathcal{X}} ELBO(x)$$

$\mathcal{X}$ → Training Dataset

However the gradient $\nabla_{\phi\theta}$ is intractable !! ... Let's see why

$$\nabla_{\phi\theta} ELBO(x) = \nabla_{\phi\theta}\left\{ E_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]\right\}$$

$$= \nabla_{\phi\theta}\left\{E_{q_\phi}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]\right\}$$

$\log\frac{x}{y} = \log x - \log y$

Let's derive for θ

$$\nabla_\theta ELBO = \nabla_\theta\left\{E_{q_\phi}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]\right\}$$

Remember $\int_z a\, p(z)\, dz = E[a] = a$

$$\nabla_\theta\left\{\int \left(\log p_\theta(x,z) - \log q_\phi(z|x)\right) \times q_\phi(z|x)\, dz\right\}$$

$$= \int \nabla_\theta\left\{\left(\log p_\theta(x,z) - \log q_\phi(z|x)\right)\right\} q_\phi(z|x)\, dx$$

let's use it again

$$= E_{q_\phi(z|x)}\left[\nabla_\theta\left\{\log p_\theta(x,z) - \log q_\phi(z|x)\right\}\right]$$

does not depend on θ

$$= E_{q_\phi(z|x)}\left[\nabla_\theta\left\{\log p_\theta(x,z)\right\}\right]$$

$$\simeq \frac{1}{N}\sum_{i}^{N} \nabla_\theta\left\{\log p_\theta(x,z^i)\right\} \quad \text{where } z^i \sim q_\phi(z|x)$$

If p_θ is a neural network we can backpropagate this gradient without issues

Now we know that the issue is with ϕ

$$\nabla_\theta ELBO = \nabla_\phi \left\{ E_{q_\phi}\left[\log p_\theta(x,z) - \log q_\phi(z|x)\right]\right\}$$

Let's expand the gradient of the expectation

$$= \nabla_\phi \left\{ \int \left(\log p_\theta(x,z) - \log q_\phi(z|x)\right) \times q_\phi(z|x)\, dz \right\}$$

$$= \int \nabla_\phi \left\{ \left(\log p_\theta(x,z) - \log q_\phi(z|x)\right) q_\phi(z|x)\, dx \right.$$

$$\boxed{\neq}\; E_{q_\phi(z|x)}\left[\nabla_\phi \left\{\log p_\theta(x,z) - \log q_\phi(z|x)\right\}\right]$$

$$= E_{q_\phi(z|x)}\left[\nabla_\phi \left\{-\log q_\phi(z|x)\right\}\right] \approx \frac{1}{N}\sum^{N} \nabla_\phi \left\{-\log q_\phi(z^i|x)\right\}$$

where $z^i \sim q_\phi(z|x)$

Interchangeability of operators

$$E_{q_\phi(z|x)}[f(z)] = \int f(z)\, q_\phi(z|x)\, dz$$

only if 1) the gradient of $f(z)$ and $q_\phi(z|x)$ wrt ϕ exists and continuous
2) $|\nabla_\phi (f(z)\, q_\phi(z|x))|$ is integrable over z

now since $q_\phi(z|x)$ depends on ϕ and also $f(z)$ depends on it the gradient acts on the function inside the expectation and the expectation violating the continuity of the gradient

Reparametrization Trick

The intractability of the gradients derives from the fact that we draw samples $z \sim q_\phi(z|x)$ which itself is a function of ϕ.
The reparametrization trick express z as a transformation of another random variable ε independent from x and ϕ:

$$z = g(\varepsilon, \phi, x) = \varepsilon\sigma + \mu \quad \text{where } \varepsilon \sim N(0, I)$$
$$z \sim N(z|\mu, \sigma^2 I)$$

which leads to $q_\phi(z|x) \cdot \left|\det\left(\frac{\partial z}{\partial \varepsilon}\right)\right| = p(\varepsilon)$

Let's rewrite the expectation in terms of $p(\varepsilon)$

$$E_{q_\phi(z|x)}\left[-\log q_\phi(z|x)\right] = \int -\log q_\phi(z|x) \cdot q_\phi(z|x)\, dz$$
$$= \int -\log q_\phi(g(\varepsilon)|x) \cdot q_\phi(g(\varepsilon)|x)\, d\,g(\varepsilon)$$
$$= \int -\log q_\phi(g(\varepsilon)|x) \cdot q_\phi(g(\varepsilon)|x) \cdot \left|\det\left(\frac{\partial g(\varepsilon)}{\partial \varepsilon}\right)\right| d\varepsilon$$
$$= \int -\log q_\phi(g(\varepsilon)|x)\, p(\varepsilon)\, d\varepsilon$$
$$= E_{p(\varepsilon)}\left[-\log q_\phi(g(\varepsilon)|x)\right]$$

Jacobian introduced by the change of variables

Nice! now let's compute the gradient wrt ϕ

$$\nabla_\phi E_{q_\phi}\left[-\log q_\phi(z|x)\right] = \nabla_\phi E_{p(\varepsilon)}\left[-\log q_\phi(g(\varepsilon)|x)\right]$$

$$= \nabla_\phi \int -\log q_\phi(g(\varepsilon)|x)\, p(\varepsilon)\, d\varepsilon$$

$$= \int \nabla_\phi \{-\log q_\phi(g(\varepsilon)|x)\, p(\varepsilon)\}\, d\varepsilon$$

$$= \int \nabla_\phi \{-\log q_\phi(g(\varepsilon)|x)\}\, p(\varepsilon)\, d\varepsilon$$

$$= \mathbb{E}_{p(\varepsilon)}\left[\nabla_\phi - \log q_\phi(g(\varepsilon)|x)\right]$$

$$\approx -\frac{1}{N}\sum^{N} \nabla_\phi \log q_\phi(g(\varepsilon^i)|x)\Big]$$

$$= -\frac{1}{N}\sum^{N} \nabla_\phi \left[\log p(\varepsilon^i) - \log\left|\det \frac{\partial z^i}{\partial \varepsilon^i}\right|\right]$$

$$= \frac{1}{N}\sum^{N} \nabla_\phi \left[\underline{\log\left|\det \frac{\partial z^i}{\partial \varepsilon^i}\right|}\right]$$

Let's simplify that $\quad q_\phi(z|x) = \mathcal{N}(z|\mu, \sigma^2 I) \qquad z = \mu + \sigma \odot \varepsilon$

with $\varepsilon \sim \mathcal{N}(0, I)$

$$\log\left|\det \frac{\partial z}{\partial \varepsilon}\right| = \log\left|\det \frac{\partial(\mu + \sigma \odot \varepsilon)}{\partial \varepsilon}\right|$$

$$= \log\left|\det \sigma I\right|$$

$$= \log \prod_i^d \sigma_i = \sum_i^d \log \sigma^i$$

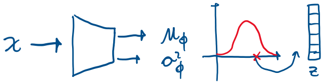
the gradient now becomes $\nabla_\phi \sum_i^d \log \sigma^i = \frac{1}{\sigma} \odot \nabla_\phi \{\underset{\downarrow}{\sigma_\phi}(x)\}$

std output of encoder

infact given a training image k

$$z^k \sim \mathcal{N}(z|\mu_\phi(x^k), \sigma_\phi^2(x^k) I)$$

The network estimates the mean and variance for the input image and then draws a sample z from the distribution



With the reparametrization trick this becomes

$$z^k \sim \mathcal{N}(z|\mu_\phi(x^k), \sigma_\phi^2(x^k)\varepsilon) \quad \text{with } \varepsilon \sim \mathcal{N}(0, I)$$

Similarly for the decoder we had $\hat{x} \sim p_\theta(x|z) = \mathcal{N}(x|f_\theta(z), \sigma_{dec}^2 I)$
with the reparametrization trick this becomes

$$\hat{x} = f_\theta(z) + \sigma_{dec}\varepsilon \quad \text{with } \varepsilon \sim \mathcal{N}(0, I)$$

Going Back to the ELBO

$$\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] = \int \log\left[N\left(x \mid f_\theta(z), \sigma^2_{dec} I\right)\right] N\left(z \mid \mu_\phi(x), \sigma^2_\phi(x)\right) dz$$

$$= -\int \frac{\|x - f_\theta(z)\|^2}{2\sigma^2_{dec}} N\left(z \mid \mu_\phi(x), \sigma^2_\phi(x)\right) dz$$

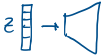
we dropped the constant term c

Now performing the reparametrization

$$= -\int \frac{\|x - f_\theta(\mu_\phi(x) + \sigma_\phi(x)\varepsilon)\|^2}{2\sigma^2_{dec}} N\left(z \mid \mu_\phi(x), \sigma^2_\phi(x)\right) dz$$

$$\approx -\frac{1}{M}\sum_{m}^{M} \frac{\|x - f_\theta(\mu_\phi(x) + \sigma_\phi(x)\varepsilon^m)\|^2}{2\sigma^2_{dec}}$$

Monte carlo over the number of samples ε



$$f_\theta(z) \rightarrow \log p_\theta(x|z) = -\frac{\|x - f_\theta(z)\|^2}{2\sigma^2_{dec}}$$

L^2 Norm between reconstructed image and real image

Training a VAE

$$\underset{\theta\phi}{\operatorname{argmax}} \sum_{x \in X} ELBO_{\phi\theta}(x) \quad \text{where} \quad ELBO_{\phi\theta}(x) = \mathbb{E}_{q_\phi}\left[\log p_\theta(x|z)\right] + D_{KL}\left(q_\phi(z|x) \| p(z)\right)$$

$$-\frac{1}{M}\sum^{M} \frac{\|x - f_\theta(\mu_\phi(x) + \sigma_\phi(x)\varepsilon^m)\|^2}{2\sigma^2_{dec}}$$

Derived from KL Divergence of two Gaussians.

$$\frac{1}{2}\left(\sigma^2_\phi(x)d - d + \|\mu_\phi(x)\|^2 - 2\log\sigma_\phi(x)\right)$$

VAE inference

We sample a latent $z \in \mathbb{R}^d \sim p(z) = N(0, I)$ and we pass it trough the decoder f_θ parametrizing $p_\theta(x|z)$ yielding a new image