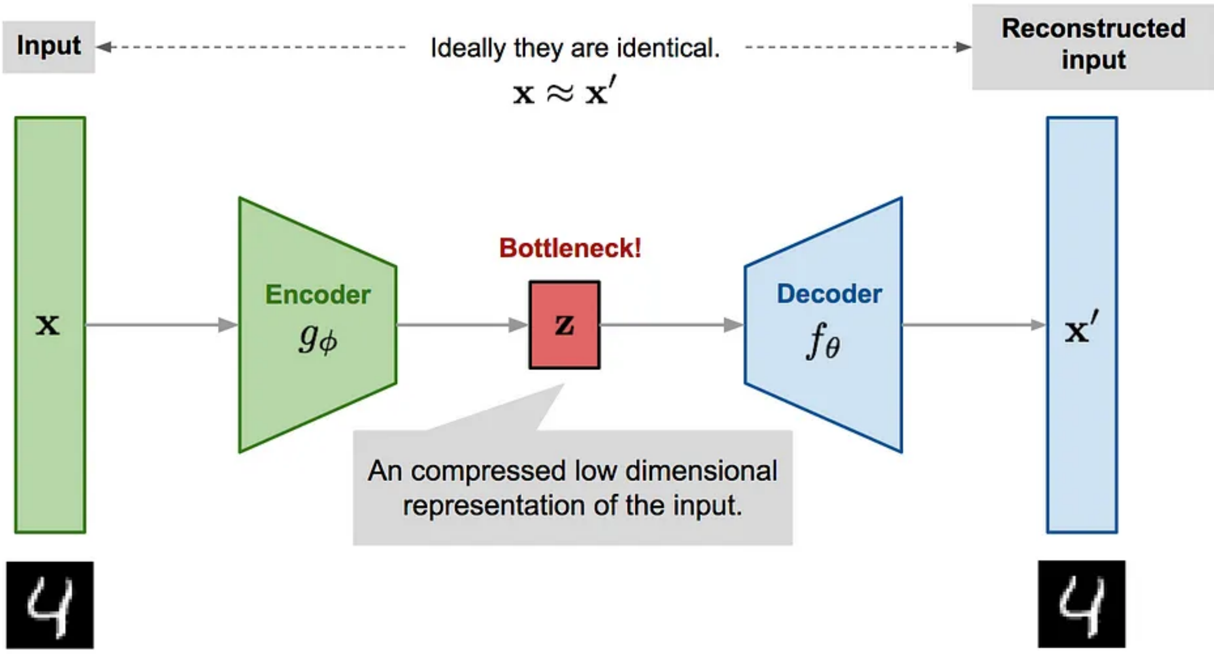
Variational Auto Encoder

Auto Encoder

- 将x压缩成低维的隐变量z，再重新生成x'，用$||x-x'||^2$来更新模型
  - 但对于一个生成模型，采样新的z并不是一件容易的事情，因为z的分布难以得到



VAE

- Given a set of training samples $D=\{x_1,\dots,x_n\}$, we want to generate new data similar to D. In general, sampling from a distribution $x\sim P(x)$ is difficult! Only some distributions are easy to sample from: $U[a,b]$, $N(\mu,\Sigma)$
  What if we sample z from a simple distribution, then use deterministic $f(z;\theta)$ to map z to x (z, x are random variables)?

$$P(x)=\int p(x|z)p(z)dz$$
$$z\sim N(0,I)\quad x\sim N(x|f(z;\theta),\sigma^2 I)$$

  - f is some neural network
- **Objective**: MLE of $P(x)$ on D to learn θ
  - Can we sample a large amount of z from $P(z)$ to approximate $P(x)$?

$$P(x)\approx\frac{1}{N}\sum_{z_i\sim P(z)}P(x|z_i)$$

    - It doesn't work because $P(z)$ is still high-dimensional, need a super large amount of samples z_i to accurately approximate $P(x)$
  - Can we use MLE of $\log P(x_i)$ with hidden variables z_i, which remind us of EM

$$\text{E-step:}\quad P(z_i|x_i;\theta^{old})=\frac{P(x_i|z_i;\theta^{old})P(z_i)}{\int_z P(x_i|z_i;\theta^{old})P(z_i)dz}$$
$$\text{M-step:}\quad \theta=\arg\max_\theta\int_z \log P(x_i,z_i;\theta)P(z_i|x_i;\theta^{old})dz$$

    - EM doesn't work because $P(z|x;\theta)$ is **intractable**

VAE loss

- Find a tractable variational distribution $q(z|x;\theta')$ to approximate $P(z|x;\theta)$

$$\begin{aligned}\log P(x;\theta) &= \int_z q(z|x;\theta')\log\frac{P(x,z;\theta)}{q(z|x;\theta')}dz - \int_z q(z|x;\theta')\log\frac{P(z|x;\theta)}{q(z|x;\theta')}dz \\ &= \int_z q(z|x;\theta')\log\frac{P(x,z;\theta)}{q(z|x;\theta')}dz + \mathrm{KL}\big(q(z|x;\theta')\|P(z|x;\theta)\big)\end{aligned}$$

- **VAE ignores KL and only maximize ELBO!**
  - We just hope KL is small
- Rewrite ELBO:

$$\begin{aligned}&\int_z q(z|x;\theta')\log P(x|z;\theta)dz + \int_z q(z|x;\theta')\log\frac{P(z)}{q(z|x;\theta')}dz \\ =&\int_z q(z|x;\theta')\log P(x|z;\theta)dz - \mathrm{KL}\big(q(z|x;\theta')\|P(z)\big)\end{aligned}$$

  - first half is called **Reconstruction quality**

$$\int_z q(z|x;\theta')\log P(x|z;\theta)dz = E_{z\sim q(z|x;\theta')}\log P(x|z;\theta)$$

  - second half regularizes $q(z|x;\theta')$ to approximate prior $P(z)$
- Reconstruction quality:

$$\begin{aligned}P(x|z;\theta) &= N(x|f(z;\theta),\sigma^2 I) \\ \log P(x|z;\theta) &= \log\frac{1}{(2\pi\sigma)^{\frac{d}{2}}}\exp(-\frac{\|x-f(z;\theta)\|^2}{2\sigma^2}) \\ &= C\|x-x'\|^2 \\ \text{Reconstruction quality} &= E_{z\sim q(z|x;\theta')}\|x-x'\|^2 \\ &\approx \|x-x'\|^2\end{aligned}$$

  - x' denotes from one z encode from x
- VAE loss:

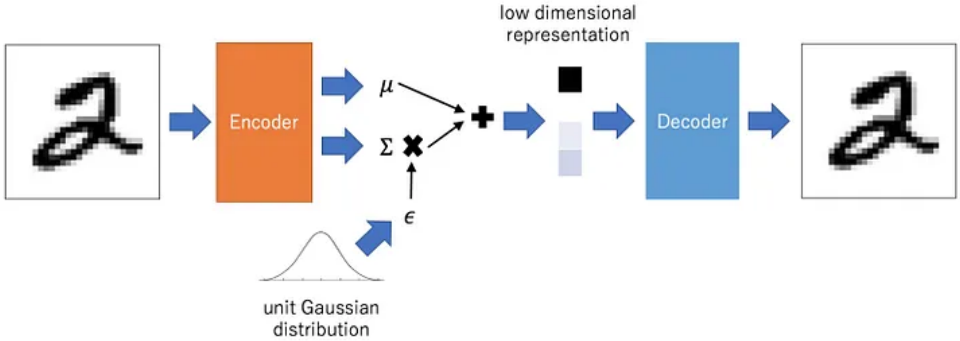
$$\underset{\theta,\theta'}{min}\frac{1}{n}\sum_{i=1}^{n}[\|x_i-x_i'\|^2+\beta\mathrm{KL}\big(q(z|x_i;\theta')\|P(z)\big)]$$

  - $\beta\to 0$, VAE $\to$ AE

Reparameterization Trick

- $q(z|x;\theta') = N(z|\mu(x),\Sigma(x))$
  - μ and Σ are two networks with θ' as parameters
  - z is sampled from $N(\mu,\Sigma)$, so backpropagation cannot be applied
- Now backpropagation can be applied:

$$\begin{aligned}&\epsilon\sim N(0,I) \\ &z=\epsilon\Sigma(x)^{\frac{1}{2}}+\mu(x)\sim N(\mu,\Sigma)\end{aligned}$$



- In practice, we use a diagonal matrix instead Σ to reduce complexity. That is, compute $\sigma(x)\in\mathbb{R}^d$ instead of $\Sigma(x)\in\mathbb{R}^{d\times d}$

Solution of KL

$$
\begin{aligned}
&\mathrm{KL}\big(q(z|x;\theta')\|P(z)\big)\\
=&\mathrm{KL}\big(N(z|\mu(x),\Sigma(x))\|N(0,I)\big)\\
=&\int_z N(z|\mu(x),\Sigma(x))\log\frac{N(z|\mu(x),\Sigma(x))}{N(z|0,I)}dz\\
=&\int_z \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu))\log\big(|\Sigma|^{\frac{1}{2}}\exp(-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)+\frac{1}{2}z^Tz)\big)dz\\
=&\int_z \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}\exp(-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu))(-\frac{1}{2}\log|\Sigma|-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)+\frac{1}{2}z^Tz)dz\\
=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}E_z[(z-\mu)^T\Sigma^{-1}(z-\mu)]+\frac{1}{2}E_z[z^Tz]\\
=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}E_z[\mathrm{tr}(z-\mu)^T\Sigma^{-1}(z-\mu)]+\frac{1}{2}E_z[(z-\mu+\mu)^T(z-\mu+\mu)]\\
=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}E_z[\mathrm{tr}\Sigma^{-1}(z-\mu)(z-\mu)^T]+\frac{1}{2}E_z[(z-\mu+\mu)^T(z-\mu+\mu)]\\
=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}E_z(z-\mu)(z-\mu)^T)+\frac{1}{2}E_z[(z-\mu)^T(z-\mu)]+\frac{1}{2}E_z[\mu^T\mu]+E_z[(z-\mu)^T\mu]\\
=&-\frac{1}{2}\log|\Sigma|-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}\Sigma)+\frac{1}{2}E_z[(z-\mu)^T(z-\mu)]+\frac{1}{2}\mu^T\mu\\
=&\frac{1}{2}(\mathrm{tr}(\Sigma)+\mu^T\mu-d-\log|\Sigma|)
\end{aligned}
$$