Latent Space Oddity On The Curvature Of Deep Generative Models

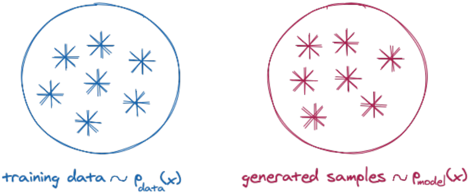
Clément GRISI, Timothée DARCET

6 janvier 2020

Generative Models

Goal

Given training data, generate new samples from the same distribution



learn $p_{\text{model}}(x)$ close to $p_{\text{data}}(x)$

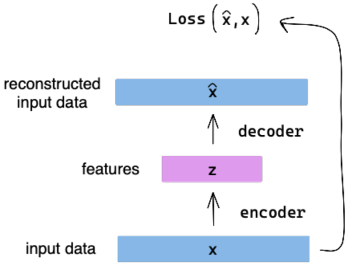
Density estimation

- core problem in the **unsupervised** learning setting
- explicit : explicitly define and solve $p_{\text{model}}(x)$
- implicit : learn a model that can sample from $p_{\text{model}}(x)$ without explicitly defining it

Variational Auto Encoders

Auto encoders

unsupervised approach for learning a lower dimensional feature representation **z** from unlabeled training data x



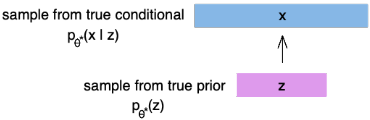
z captures meaningful factors of variation in the data

Question : can we generate new images from an auto encoder ?

Variational auto encoders

VAEs are a probabilistic spin on auto encoders that will let us sample from the model to generate data

Assumption : training data $\{x_i\}_{i \in [1,N]}$ is generated from underlying (latent) **unobserved** representation **z**



Goal : estimate the true parameters θ^* of this generative model

Question : how do we train the model ?

Intractability

Choose simple prior $p(z)$ (e.g. Gaussian)
Conditional $p(x|z)$ is **complex** : represent it with a neural network

Natural strategy : learn model parameters to maximize the likelihood of the training data

$$p_\theta(x) = \int p_\theta(x|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$$

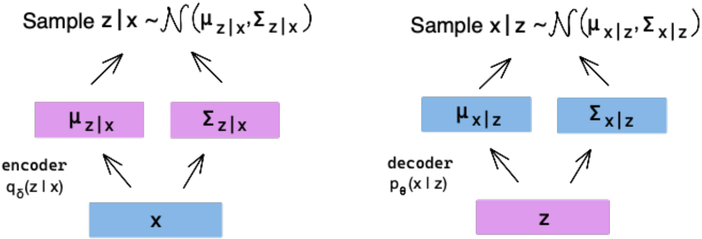
Though we know $p_\theta(x|\mathbf{z})$ and $p_\theta(\mathbf{z})$, it is **intractable** to compute $p_\theta(x|\mathbf{z})$ for every z !

Posterior density is also **intractable** :

$$p_\theta(\mathbf{z}|x) = \frac{p_\theta(x|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(x)}$$

Solution

In addition to the **decoder** network modeling $p_\theta(x|z)$, define an additional **encoder** network $q_\delta(z|x)$ that approximates $p_\theta(z|x)$.



This allows us to derive a **tractable** lower bound on the data likelihood

Tractable lower bound

$$
\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{z\sim q_\delta(z|x)}\left[\log p_\theta(x)\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(z|x)}\frac{q_\delta(z|x)}{q_\delta(z|x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x|z)\right] - \mathbb{E}_z\left[\log \frac{q_\delta(z|x)}{p_\theta(z)}\right] + \mathbb{E}_z\left[\log \frac{q_\delta(z|x)}{p_\theta(z|x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x|z)\right] - d_{\mathrm{KL}}\left(q_\delta(z|x)||p_\theta(z)\right) + d_{\mathrm{KL}}\left(q_\delta(z|x)||p_\theta(z|x)\right)
\end{aligned}
$$

Tractable lower bound

- decoder network gives $p_\theta(x|z)$: we can estimate $\mathbb{E}_z\left[\log p_\theta(x|z)\right]$ through sampling
- $d_{\mathrm{KL}}\left(q_\delta(z|x)||p_\theta(z)\right)$ is the KL-div of two Gaussian distributions : it has a nice closed from solution
- though $p_\theta(z|x)$ is intractable, we know KL-div is always positive : $d_{\mathrm{KL}}\left(q_\delta(z|x)||p_\theta(z|x)\right) \geq 0$

Hence,

$$\begin{aligned} \log p_\theta(x) &\geq \mathbb{E}_z\left[\log p_\theta(x|z)\right] - d_{\mathrm{KL}}\left(q_\delta(z|x)||p_\theta(z)\right) \\ &\geq \mathscr{L}(x,\theta,\delta) \end{aligned}$$

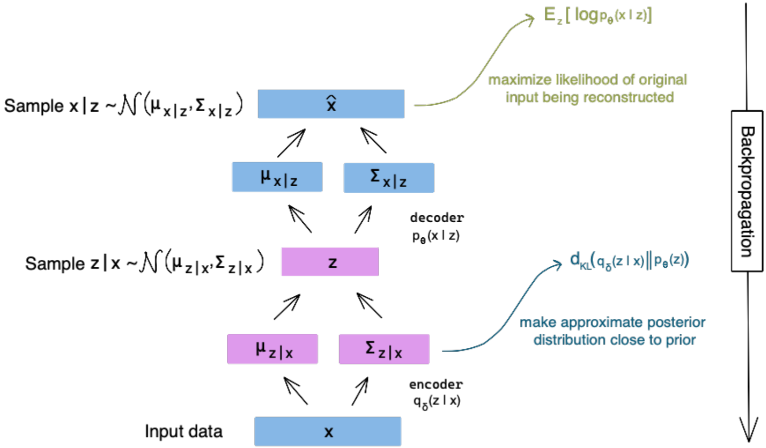
We've identified a **tractable** + **differentiable** lower bound which we can take gradient of and optimize (ELBO, evidence lower bound)

Training

Optimal parameters will be the ones that maximizes this lower bound :

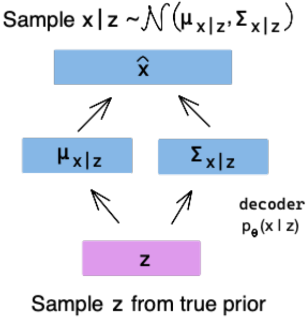
$$\theta^*, \delta^* = \arg\max_{\theta,\delta} \sum_{i=1}^{N} \mathscr{L}(x_i, \theta, \delta)$$

Training


$E_z[\log p_\theta(x \mid z)]$
maximize likelihood of original input being reconstructed
Sample $x \mid z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$
$\hat{x}$
$\mu_{x|z}$
$\Sigma_{x|z}$
decoder
$p_\theta(x \mid z)$
Sample $z \mid x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$
z
$d_{KL}(q_\delta(z \mid x) \| p_\theta(z))$
make approximate posterior distribution close to prior
$\mu_{z|x}$
$\Sigma_{z|x}$
encoder
$q_\delta(z \mid x)$
Input data
x
Backpropagation

Data generation

Once trained, we can sample **z** from prior and use the decoder network to generate new data :



A misinterpretation of the latent space

Starting point

Goal : equiped with a good latent space, points from the same class should be clother to each other than to members of the other classes

Issue : this doesn't *seem* to be the case

But : this *seemed* conclusion is incorrect
→ this is due to a misinterpretation of the latent space : it shouldn't be seen as a linear Euclidian space but rather as a **curved** space

This curvature induces a Riemannian distance which is more meaningful than the usual Eulidian distance : under this new metric, two points from the same class are clother to each other than to members of the other classes

Bibliography :

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg.
Latent space oddity : on the curvature of deep generative models, 2018.