Variational Autoencoders

Foundations of Data Analysis

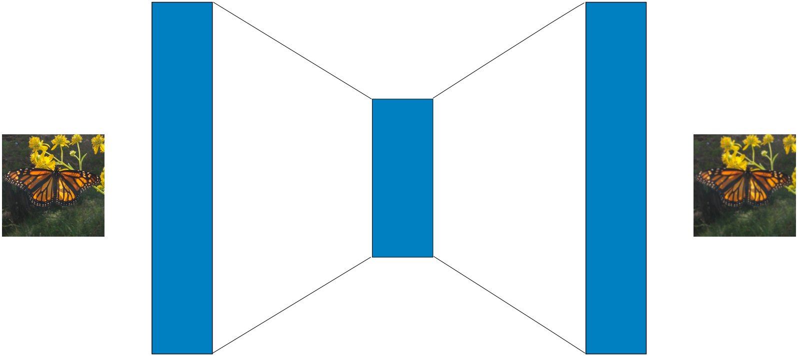
May 6, 2021

Talking about this paper:

Diederik Kingma and Max Welling, Auto-Encoding Variational Bayes, In *International Conference on Learning Representation (ICLR)*, 2014.

Autoencoders

Input Latent Space Output



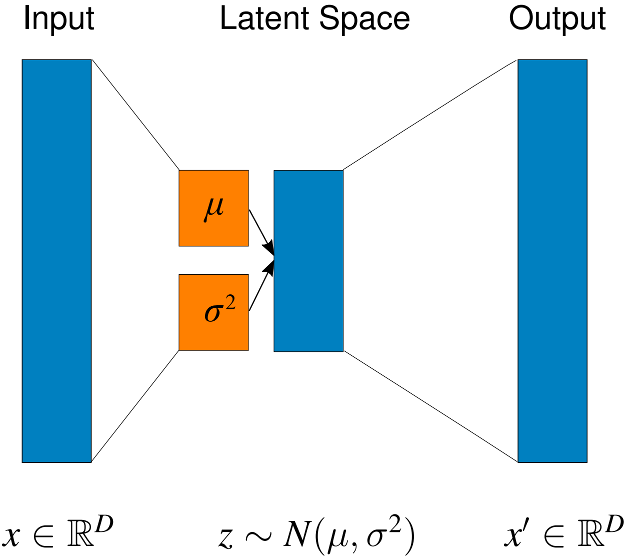
$x \in \mathbb{R}^D$ $z \in \mathbb{R}^d$ $x' \in \mathbb{R}^D$

$$d << D$$

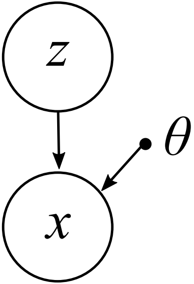
Autoencoders

- Linear activation functions give you PCA
- Training:
  1. Given data x, feedforward to x' output
  2. Compute loss, e.g., $L(x, x') = \|x - x'\|^2$
  3. Backpropagate loss gradient to update weights
- **Not** a generative model!

Variational Autoencoders



Generative Models



Sample a new x in two steps:

Prior: $p(z)$

Generator: $p_\theta(x \mid z)$

Now the analogy to the "encoder" is:

Posterior: $p(z \mid x)$

Bayesian Inference

Posterior via Bayes' Rule:

$$
\begin{aligned}
p(z \mid x) &= \frac{p_\theta(x \mid z)p(z)}{p(x)} \\
&= \frac{p_\theta(x \mid z)p(z)}{\int p_\theta(x \mid z)p(z)dz}
\end{aligned}
$$

Integral in denominator is (usually) intractable!

Kullback-Leibler Divergence

$$\begin{aligned} D_{\mathrm{KL}}(q \| p) &= -\int q(z) \log\left(\frac{p(z)}{q(z)}\right) dz \\ &= E_q\left[-\log\left(\frac{p}{q}\right)\right] \end{aligned}$$

The average *information gained* from moving from q to p

Variational Inference

Approximate intractable posterior $p(z \mid x)$ with a manageable distribution $q(z)$

Minimize the KL divergence: $D_{\mathrm{KL}}(q(z) \| p(z \mid x))$

Evidence Lower Bound (ELBO)

$$
\begin{aligned}
D_{\mathrm{KL}}(q(z) \| p(z \mid x)) \\
&= E_q\left[-\log\left(\frac{p(z \mid x)}{q(z)}\right)\right] \\
&= E_q\left[-\log\frac{p(z, x)}{q(z)p(x)}\right] \\
&= E_q[-\log p(z, x) + \log q(z) + \log p(x)] \\
&= -E_q[\log p(z, x)] + E_q[\log q(z)] + \log p(x)
\end{aligned}
$$

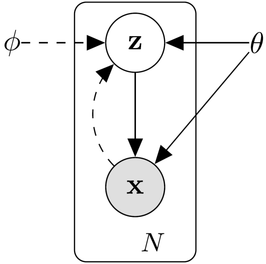
$$\log p(x) = D_{\mathrm{KL}}(q(z) \| p(z \mid x)) + L[q(z)]$$

ELBO: $L[q(z)] = E_q[\log p(z, x)] - E_q[\log q(z)]$

Variational Autoencoder

$$q_\phi(z \mid x)$$

Encoder Network



$$p_\theta(x \mid z)$$

Decoder Network

Maximize ELBO:

$$\mathcal{L}(\theta, \phi, x) = E_{q_\phi}[\log p_\theta(x, z) - \log q_\phi(z \mid x)]$$

VAE ELBO

$$
\begin{aligned}
\mathcal{L}(\theta, \phi, x) &= E_{q_\phi}[\log p_\theta(x, z) - \log q_\phi(z \mid x)] \\
&= E_{q_\phi}[\log p_\theta(z) + \log p_\theta(x \mid z) - \log q_\phi(z \mid x)] \\
&= E_{q_\phi}\left[\log \frac{p_\theta(z)}{q_\phi(z \mid x)} + \log p_\theta(x \mid z)\right] \\
&= -D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z)) + E_{q_\phi}[\log p_\theta(x \mid z)]
\end{aligned}
$$

Problem: Gradient $\nabla_\phi E_{q_\phi}[\log p_\theta(x \mid z)]$ is intractable!

Use Monte Carlo approx., sampling $z^{(s)} \sim q_\phi(z \mid x)$:

$$
\nabla_\phi E_{q_\phi}[\log p_\theta(x \mid z)] \approx \frac{1}{S} \sum_{s=1}^{S} \log p_\theta(x \mid z) \nabla_\phi \log q_\phi(z^{(s)} \mid x)
$$

Reparameterization Trick

What about the other term?

$$-D_{\mathrm{KL}}(q_\phi(z \mid x) \| p_\theta(z))$$

Says encoder, $q_\phi(z \mid x)$, should make code z look like prior distribution

Instead of encoding z, encode parameters for a normal distribution, $N(\mu, \sigma^2)$

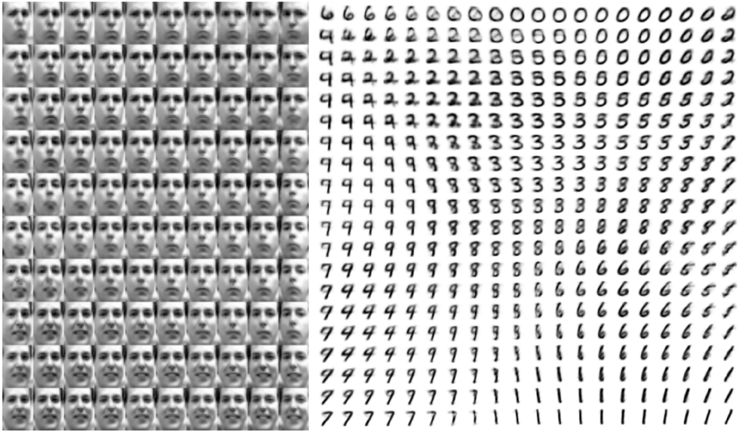
Reparameterization Trick

$$q_\phi(z_j \mid x^{(i)}) = N(\mu_j^{(i)}, \sigma_j^{2(i)})$$
$$p_\theta(z) = N(0, I)$$

KL divergence between these two is:

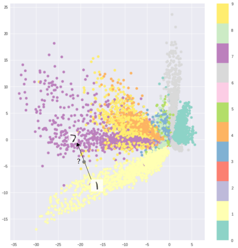
$$D_{\mathrm{KL}}(q_\phi(z \mid x^{(i)}) \| p_\theta(z)) = -\frac{1}{2} \sum_{j=1}^{d} \left(1 + \log(\sigma_j^{2(i)}) - (\mu_j^{(i)})^2 - \sigma_j^{2(i)}\right)$$

Results from Kingma & Welling

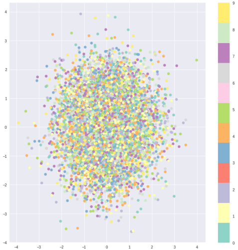


Why Do Variational?

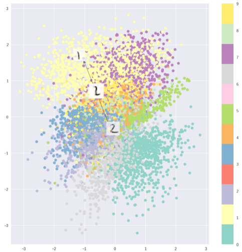
Example trained on MNIST:



Autoencoder
(reconstruction loss)



KL divergence only



VAE
(KL + recon. loss)

From: this webpage