Aprendizado Profundo

Autoencoders e Variational Autoencoders

Professor: Lucas Silveira Kupssinskü

Variational Autoencoders

- São modelos geradores probabilísticos cujo objetivo é aprender uma distribuição $p(x)$ sobre os dados
- É um modelo não-linear de variável latente
  - Depois do treinamento deixamos de lado tanto a parte variacional quanto o autoencoder

Modelos de variáveis latentes

- Abordagem indireta para modelar uma distribuição de probabilidades $p(x)$ sobre uma variável multidimensional x
  - Marginalização de uma probabilidade conjunta $p(x, z)$

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

- Motivação: Distribuições relativamente simples $p(x|z)$ e $p(z)$ podem definir distribuições $p(x)$ complexas

Exemplo: Mistura de Gaussianas 1D

- z é uma variável latente discreta, e a distribuição a priori $p(z)$ é uma distribuição categórica com uma probabilidade λ_n para cada possível valor de z
  - $p(z = n) = \lambda_n$
- A verossimilhança segue uma normal $p(x|z = n)$
  - $p(x|z = n) = \mathcal{N}(\mu_n, \sigma_n^2)$
- $p(x) = \sum \lambda_n \mathcal{N}(\mu_n, \sigma_n^2)$

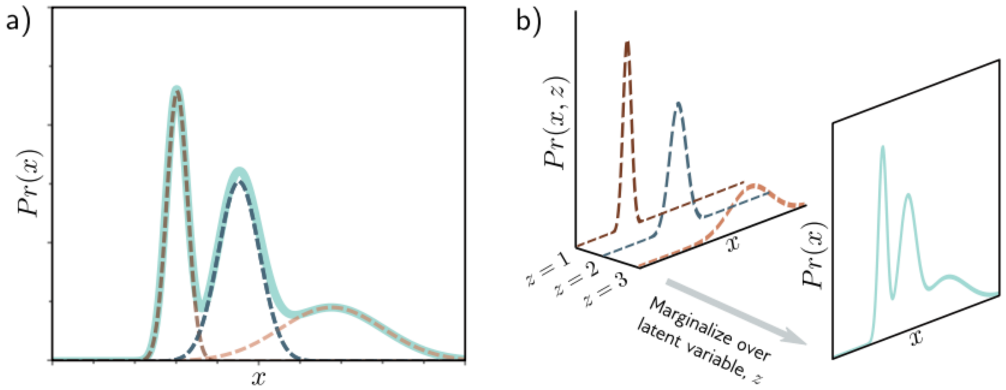


Figure 17.1 Mixture of Gaussians (MoG). a) The MoG describes a complex probability distribution (cyan curve) as a weighted sum of Gaussian components (dashed curves). b) This sum is the marginalization of the joint density $Pr(x, z)$ between the continuous observed data x and a discrete latent variable z.

Modelos não lineares de variáveis latentes

- Em modelos não lineares de variáveis latentes, tanto z quanto x são contínuos e multivariados
- A distribuição a priori é a normal padrão

$$p(z) = \mathcal{N}_z(0, \mathrm{I})$$

- A verossimilhança segue uma normal cuja média é uma função não linear $f(z, \theta)$ da variável latente com covariância esférica

$$p(x|z, \theta) = \mathcal{N}_x(f(z, \theta), \sigma^2 \mathrm{I})$$

- A distribuição de probabilidade que estamos interessados pode ser obtida via marginalização

$$p(x|\theta) = \int \mathcal{N}_x(f(z, \theta), \sigma^2 \mathrm{I})\, \mathcal{N}_z(0, \mathrm{I}) dz$$

Gerando novas instâncias x^*

- Para gerar uma nova instância x^* realizamos um processo conhecido com amostragem ancestral
  - Primeiro amostramos um latente z^* da distribuição a priori $p(z)$
  - Passamos z^* pela rede $f(z^*, \theta)$ para computar o valor da média da verossimilhança $p(x|z^*, \theta)$
  - Usamos $p(x|z^*, \theta)$ para amostrar uma nova instância x^*

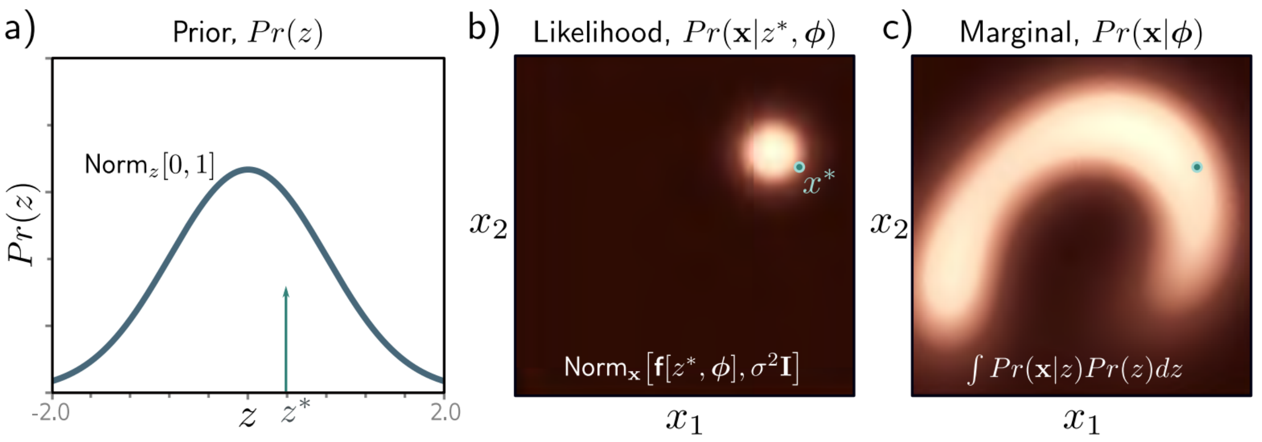


Figure 17.3 Generation from nonlinear latent variable model. a) We draw a sample z^* from the prior probability $Pr(z)$ over the latent variable. b) A sample $\mathbf{x}^*$ is then drawn from $Pr(\mathbf{x}|z^*, \phi)$. This is a spherical Gaussian with a mean that is a nonlinear function $\mathbf{f}[\bullet, \phi]$ of z^* and a fixed variance $\sigma^2\mathbf{I}$. c) If we repeat this process many times, we recover the density $Pr(\mathbf{x}|\phi)$.

Treinamento

- Maximizar a log verossimilhança dos dados com os parâmetros do modelo diretamente é inviável pois não existe uma expressão fechada para a integral, nem uma maneira fácil de avaliar o valor para uma instância de treinamento $x^{(i)}$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \left[\sum_{i=1}^{N} \int \mathcal{N}_{x^{(i)}}(f(z,\theta), \sigma^2 \mathrm{I})\, \mathcal{N}_z(0, \mathrm{I}) dz \right]$$

Treinamento

- Como não conseguimos maximizar diretamente o log da verossimilhança, vamos criar uma expressão para descrever o limite inferior da log verossimilhança e tentar otimizar esse limite
- O limite inferior vai ser uma função que garantidamente é menor ou igual a log verossimilhança
    - Para implementar esse limite inferior, vamos precisar adicionar uma segunda rede com parâmetros treináveis ϕ distintos de θ

Desigualdade de Jensen

- Para qualquer função côncava $g[.]$
  - $g(\mathbb{E}[x]) \geq \mathbb{E}[g(x)]$
- Na imagem ao lado vemos um exemplo para a função log

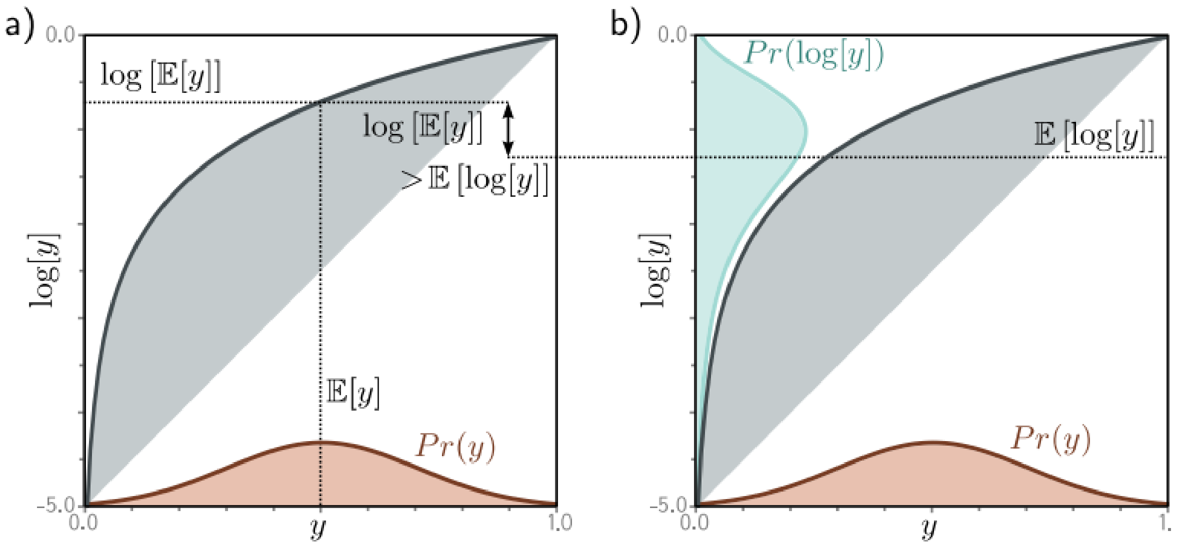


Figure 17.5 Jensen's inequality (continuous case). For a concave function, computing the expectation of a distribution $Pr(y)$ and passing it throu[illegible] Gerando novas instâncias x^* gives a result greater than or equal to transforming the variable y by the function and then computing the expectation of the new variable. In the case of the logarithm, we have $\log[\mathbb{E}[y]] \geq \mathbb{E}[\log[y]]$. The left-hand side of the figure corresponds to the left-hand side of this inequality and the right-hand side of the figure to the right-hand side. One way of thinking about this is to consider that we are taking a convex combination of the points in the orange distribution defined over $y \in [0, 1]$. By the logic of figure 17.4, this must lie under the curve. Alternatively, we can think about the concave function as compressing the high values of y relative to the low values, so the expected value is lower when we pass y through the function first.

Limite Inferior

- Vamos usar a desigualdade de Jensen para derivar o limite inferior da log verossimilhança

$$\log[p(x|\theta)] = \log\left[\int p(x,z|\theta)dz\right]$$

- Multiplicamos e dividimos por uma distribuição arbitrária $q(z)$ sobre a variável latente

$$\log[p(x|\theta)] = \log\left[\int q(z)\frac{p(x,z|\theta)}{q(z)}dz\right]$$

Limite Inferior

- Aplicando a Desigualdade de Jensen

$$\log[p(x|\theta)] = \log\left[\int q(z)\frac{p(x,z|\theta)}{q(z)}dz\right]$$

$$\log\left[\int q(z)\frac{p(x,z|\theta)}{q(z)}dz\right] \geq \boxed{\int q(z)\log\left(\frac{p(x,z|\theta)}{q(z)}\right)dz}$$

- Esse termo é conhecido com *evidence lower bound (ELBO)*
  - Na prática a distribuição $q(z)$ também é parametrizada, então escrevemos $q(z|\phi)$

ELBO

$$ELBO[\theta, \phi] = \int q(z|\phi) \log\left(\frac{p(x, z|\theta)}{q(z|\phi)}\right) dz$$

Podemos reorganizar essa expressão para melhor compreender o significado dessa expressão e facilitar a construção de uma rede neural para estimar os parâmetros da função

ELBO

$$ELBO[\theta, \phi] = \int q(z|\phi) \log\left(\frac{p(x, z|\theta)}{q(z|\phi)}\right) dz$$

$$= \int q(z|\phi) \log\left(\frac{p(x|z, \theta)p(z)}{q(z|\phi)}\right) dz$$

Aplicando a regra de manipulação de probabilidades conjuntas

$$= \int q(z|\phi) \log\big(p(x|z, \theta)\big)\, dz + \int q(z|\phi) \log\left(\frac{p(z)}{q(z|\phi)}\right) dz$$

$$= \int q(z|\phi) \log\big(p(x|z, \theta)\big)\, dz - D_{KL}[q(z|\phi)|p(z)]$$

ELBO

$$ELBO[\theta, \phi] = \boxed{\int q(z|\phi) \log\big(p(x|z, \theta)\big)\, dz} - \boxed{D_{KL}[q(z|\phi)|p(z)]}$$

- O primeiro termo avalia a concordância média entre a variável latente e os dados. Também chamado de *Reconstruction Loss*
- O segundo termo avalia o quanto a distribuição auxiliar $q(z)$ coincide com a distribuição a priori

VAE

$$ELBO[\theta, \phi] = \boxed{\int q(z|\phi) \log\big(p(x|z,\theta)\big)\, dz} - D_{KL}[q(z|\phi)|p(z)]$$

- O primeiro termo ainda é uma integral intratável, contudo como ele assume a forma de uma esperança, podemos aproximá-lo com uma estimativa *Monte Carlo*
  - Uma estimativa grosseira mas eficaz seria aproximar esse termo a partir de uma única instância z^* de $q(z|x,\phi)$

$$ELBO[\theta, \phi] \approx \log\big(p(x|z^*,\theta)\big) - D_{KL}[q(z|\phi)|p(z)]$$

VAE

Adicionamos mais um condicionante para termos um limite inferior justo

$$ELBO[\theta, \phi] = \int q(z|\phi) \log\big(p(x|z,\theta)\big)\, dz - \boxed{D_{KL}[q(z|\boxed{x,}\phi)|p(z)]}$$

- O segundo termo é uma divergência de KL
  - Existe uma forma fechada para a divergência de KL entre duas distribuições normais (sendo que uma delas é uma normal padrão e a outra possui média μ e desvio padrão Σ)

$$D_{KL}[q(z|x,\phi)|p(z)] = \frac{1}{2}(Tr[\Sigma] + \mu^T\mu - D_z - \log[\det[\Sigma]])$$

Sendo D_z a dimensionalidade de Z

VAE

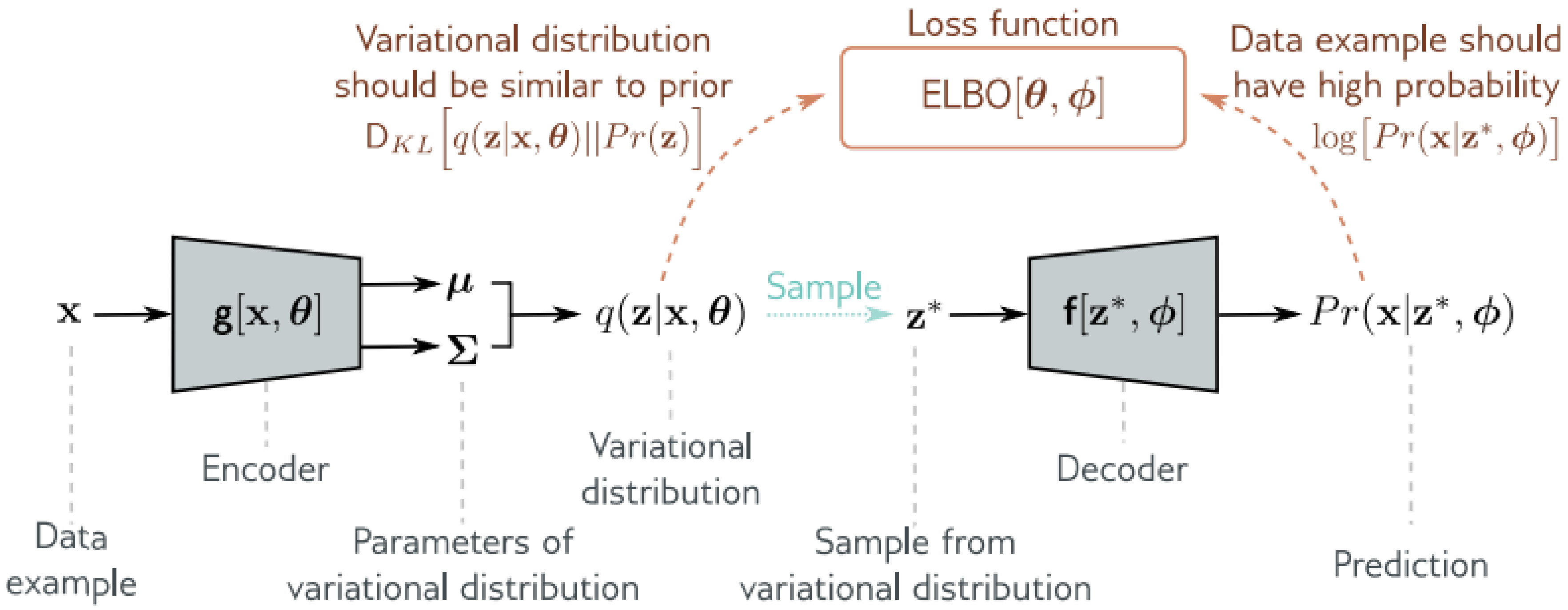


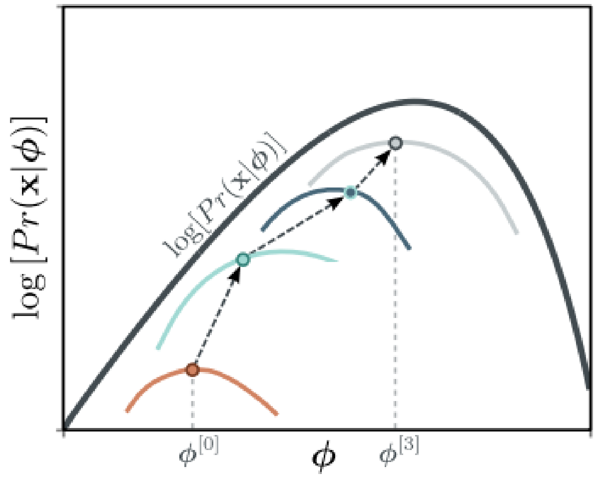
Figure 17.9 Variational autoencoder. The encoder $\mathbf{g}[\mathbf{x},\boldsymbol{\theta}]$ takes a training example $\mathbf{x}$ and predicts the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ of the variational distribution $q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})$. We sample from this distribution and then use the decoder $\mathbf{f}[\mathbf{z},\phi]$ to predict the data $\mathbf{x}$. The loss function is the negative ELBO, which depends on how accurate this prediction is and how similar the variational distribution $q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})$ is to the prior $Pr(\mathbf{z})$ (equation 17.21).

Reparametrization Trick

- Como temos um termo estocástico na nossa rede neural é difícil/inviável de computar backpropagation
- Para evitar esse problema vamos reparametrizar a saída do *encoder* da seguinte forma
- $z^* = \mu + \Sigma^{1/2}\epsilon^*$
- ϵ^* é amostrado de $\mathcal{N}(0, I)$

VAE

Figure 17.10 The VAE updates both factors that determine the lower bound at each iteration. Both the parameters ϕ of the decoder and the parameters θ of the encoder are manipulated to increase this lower bound.



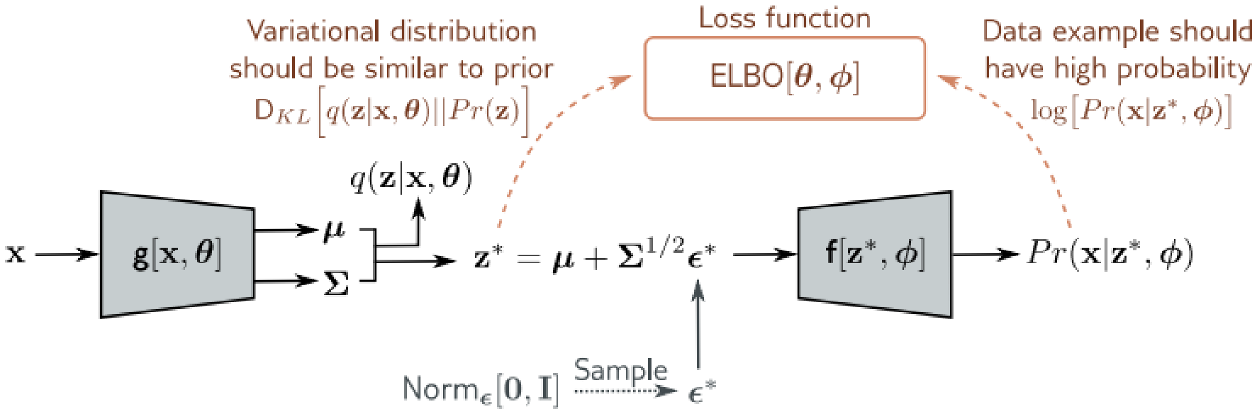


Figure 17.11 Reparameterization trick. With the original architecture (figure 17.9), we cannot easily backpropagate through the sampling step. The reparameterization trick removes the sampling step from the main pipeline; we draw from a standard normal and combine this with the predicted mean and covariance to get a sample from the variational distribution.

Referências

- Capítulo 17 – Understanding Deep Learning – Simon J. Prince