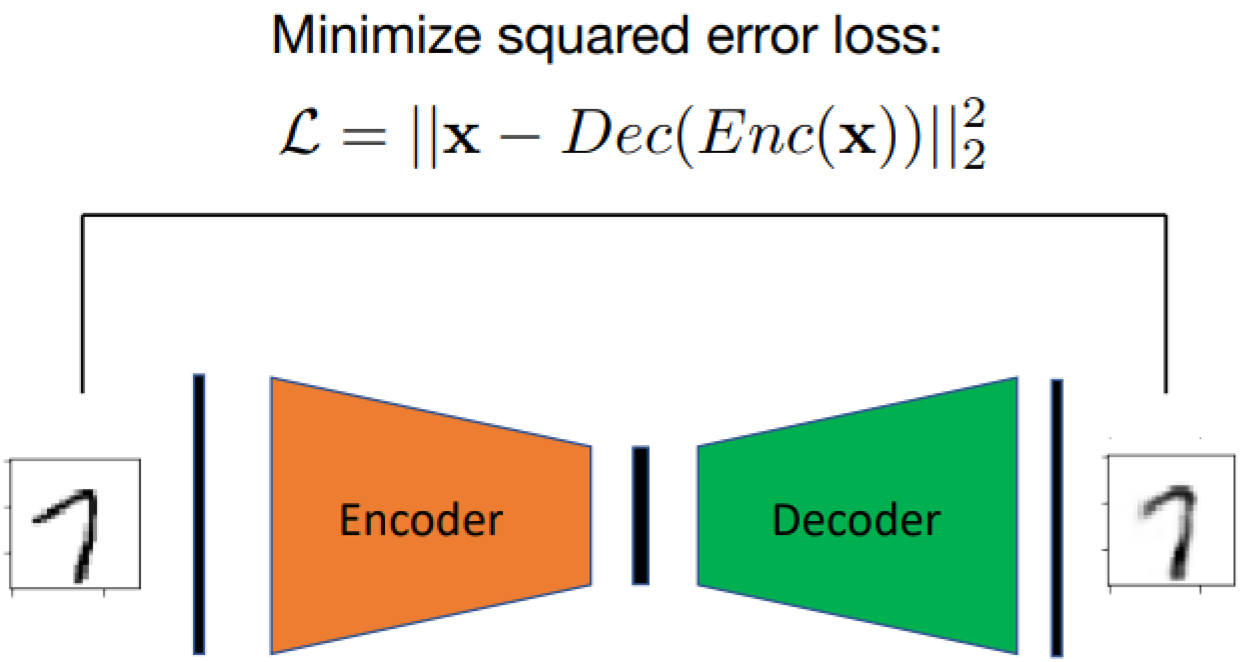
Lecture 07b
Introducción a los Variational Autoencoders (VAEs)

UNIVERSIDAD EAFIT
Escuela de Ciencias Aplicadas e Ingeniería

Codificadores automáticos vs codificadores automáticos variacionales

1. **Descripción general del VAE**
2. Muestreo de un VAE
3. Truco de Log-Var
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Resumen: autoencoders

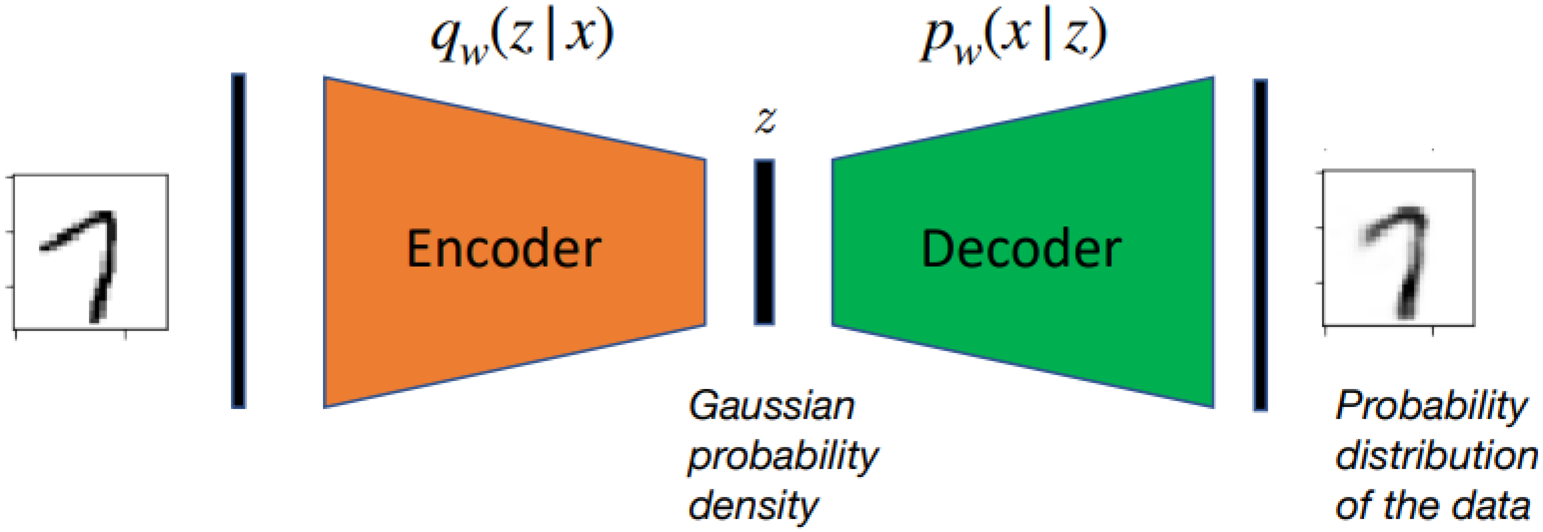


Variational Autoencoder

$$\mathcal{L} = -\mathbb{E}_{z \sim q_w(z \mid x^{[i]})} \left[\log p_w\left(x^{[i]} \mid z\right)\right] + \mathbf{KL}\left(q_w\left(z \mid x^{[i]}\right) \| p(z)\right)$$

Expected neg. log likelihood term; wrt to encoder distribution

Kullback-Leibler divergence term where $p(z) = \mathcal{N}\left(\mu = 0, \sigma^2 = 1\right)$



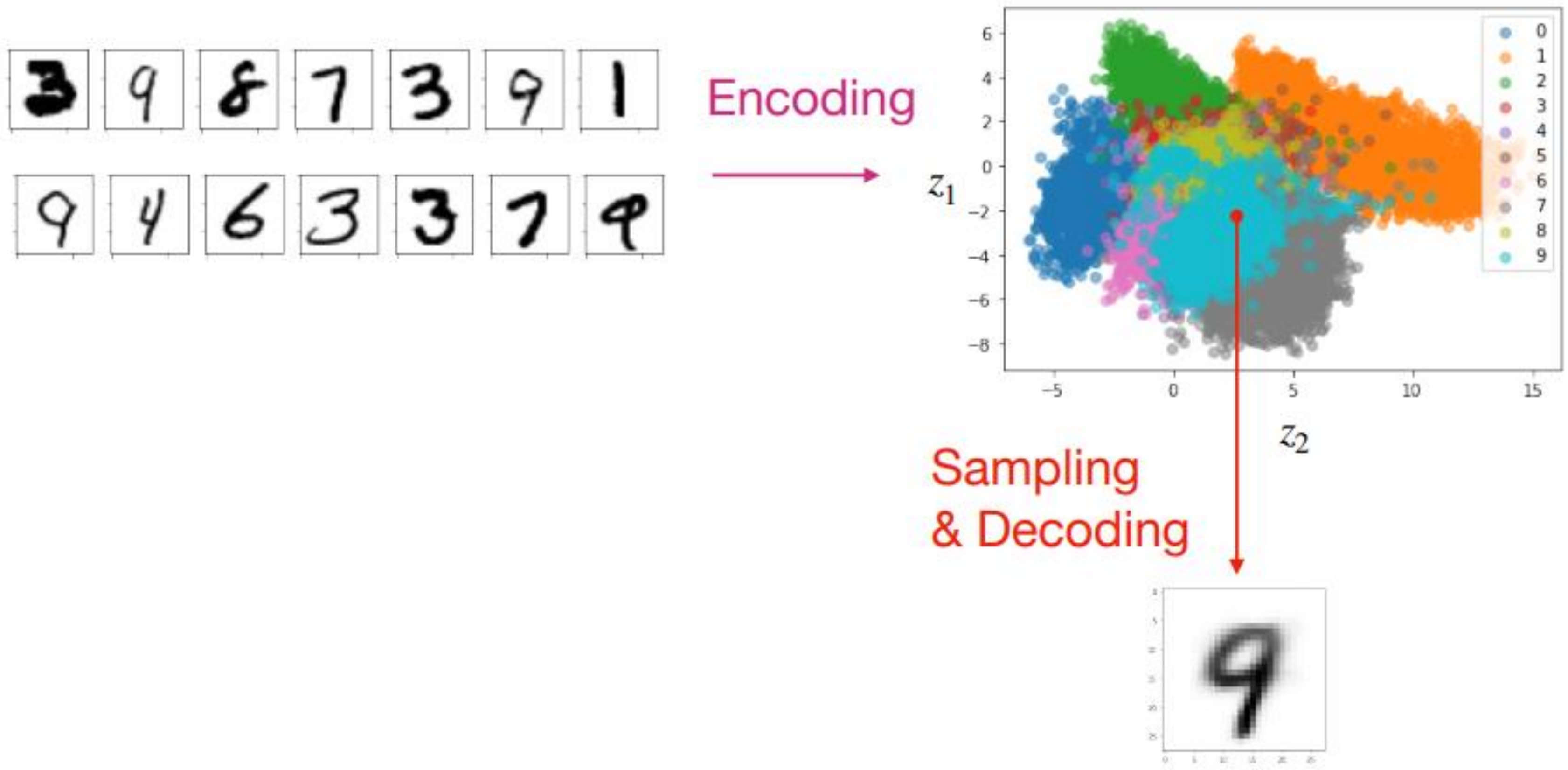
Kingma, D. P., & Welling, M. (2013). Auto-encoding Variational Bayes. arXiv preprint arXiv:1312.6114. https://arxiv.org/abs/1312.6114

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. **Muestreo de un VAE**
3. Truco de Log-Var
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Uso de autoencoders para el muestreo

En la clase anterior:



Uso de autoencoders para el muestreo

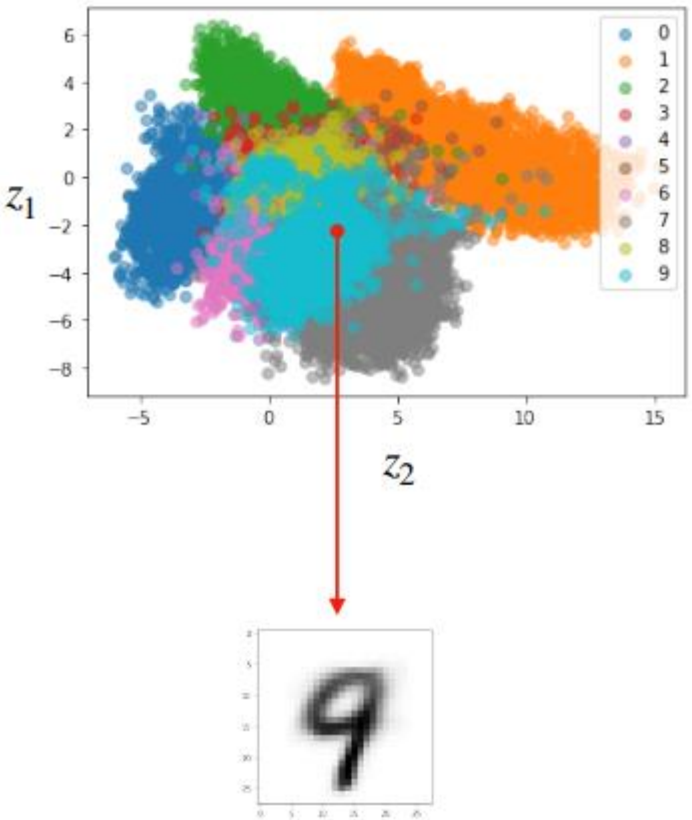
Desafío: los codificadores automáticos normales son difíciles de muestrear, porque:

1. distribución de forma extraña, difícil de muestrear de forma equilibrada

2. distribución no centrada en (0, 0)

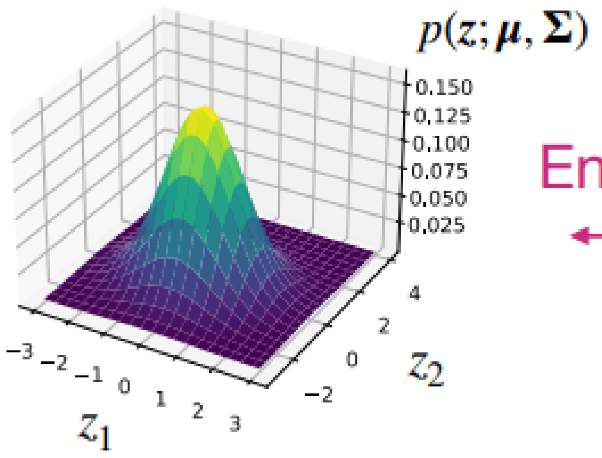
3. distribución no necesariamente continua (difícil de ver en 2D, pero un gran problema en espacios latentes de dimensiones superiores)

En la clase anterior:

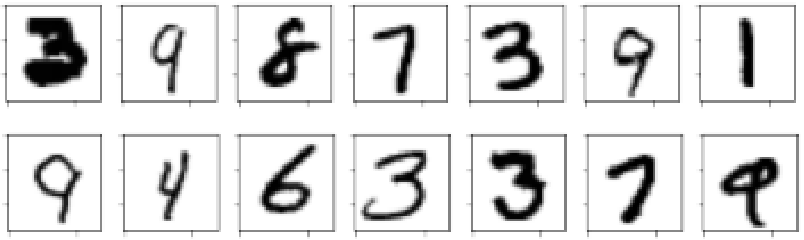


Uso de VAEs para el muestreo

Esta clase:



Encoding



d-dimensional probability density for multivariate Gaussian

$$p(z;\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(z-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(z-\boldsymbol{\mu})\right)$$

$$Z \sim \mathcal{N}(0, \boldsymbol{I})$$

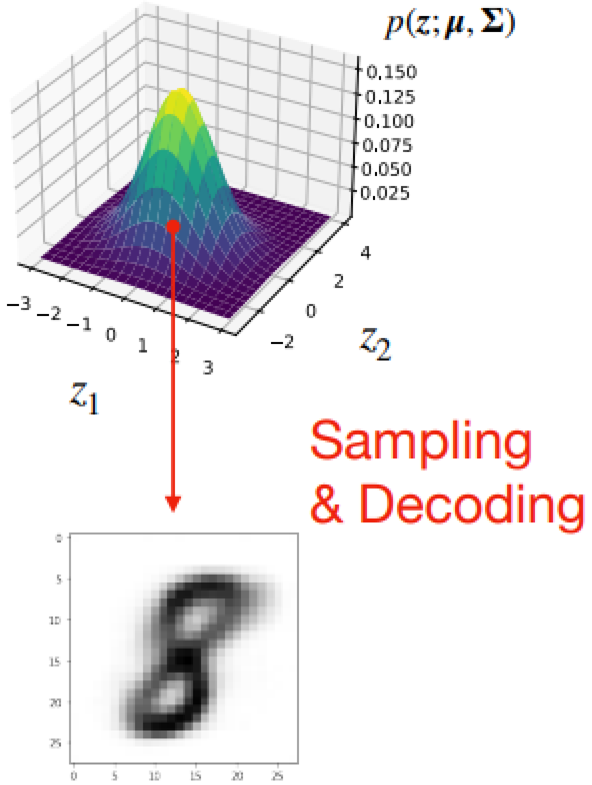
with $\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$

Muestreo desde un VAE

$$z = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$$

$$\text{Where} \quad \boldsymbol{\sigma}^2 = \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}$$

$$\epsilon_1, \epsilon_2 \sim N(0,1)$$



- Los VAE asumen una matriz de covarianza diagonal (sin interacción entre las características).

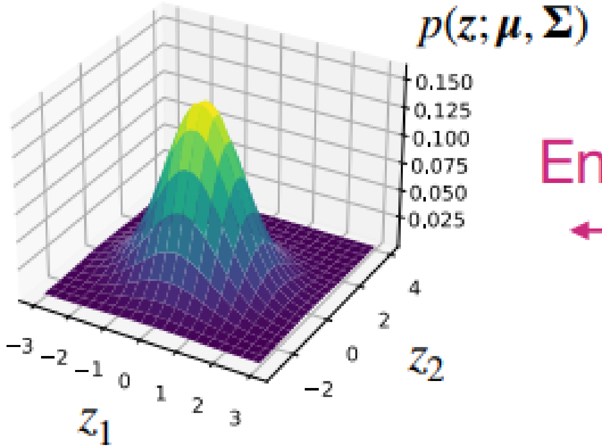
- Por lo tanto, solo necesitamos una media y un vector de varianza, no una matriz de covarianza

Codificadores automáticos vs codificadores automáticos variacionales

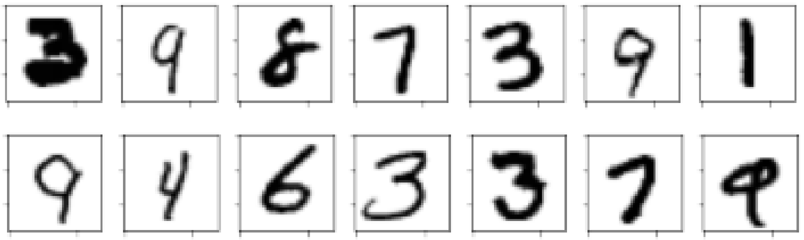
1. Descripción general del VAE
2. Muestreo de un VAE
3. **Truco de Log-Var**
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Uso de VAEs para el muestreo

Esta clase:



Encoding



d-dimensional probability density for multivariate Gaussian

$$p(z;\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(z-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(z-\boldsymbol{\mu})\right)$$

$$Z \sim \mathcal{N}(0,\boldsymbol{I})$$

with $\boldsymbol{z} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$

Muestreo desde un VAE

$$z = \mu + \sigma \cdot \epsilon$$

Muestreado de la distribución normal multivariante estándar en cada paso hacia adelante

Where $\sigma^2 = \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}$

$$\epsilon_1, \epsilon_2 \sim N(0,1)$$

Pero ¿por qué ε? Distribución continua; VAE debe asegurarse de que los puntos en el vecindario codifiquen la misma imagen para que al decodificarlos produzcan la misma imagen

Piense en estos como vectores de parámetros incluidos en el entrenamiento y la retropropagación.

Muestreo desde un VAE - El truco de Log-Var

En lugar de utilizar un vector de varianza, $\boldsymbol{\sigma}^2 = \begin{pmatrix} \sigma_1^2 \\ \sigma_2^2 \end{pmatrix}$

Se usa el vector log-var para permitir valores positivos y negativos: $\log(\boldsymbol{\sigma}^2)$

¿Por qué podemos hacer esto?

$$\log(\sigma^2) = 2 \cdot \log(\sigma)$$
$$\log(\sigma^2)/2 = \log(\sigma)$$
$$\sigma = e^{\log(\sigma^2)/2}$$

Entonces, cuando tomamos muestras de los puntos, podemos hacer

$$z = \mu + e^{\log(\sigma^2)/2} \cdot \epsilon$$

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. Muestreo de un VAE
3. Truco de Log-Var
4. **Función de pérdida del VAE**
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Función de costo del VAE

Minimiza ELBO (Evidence lower bound - límite inferior de evidencia), que consiste en pérdida de reconstrucción y término KL

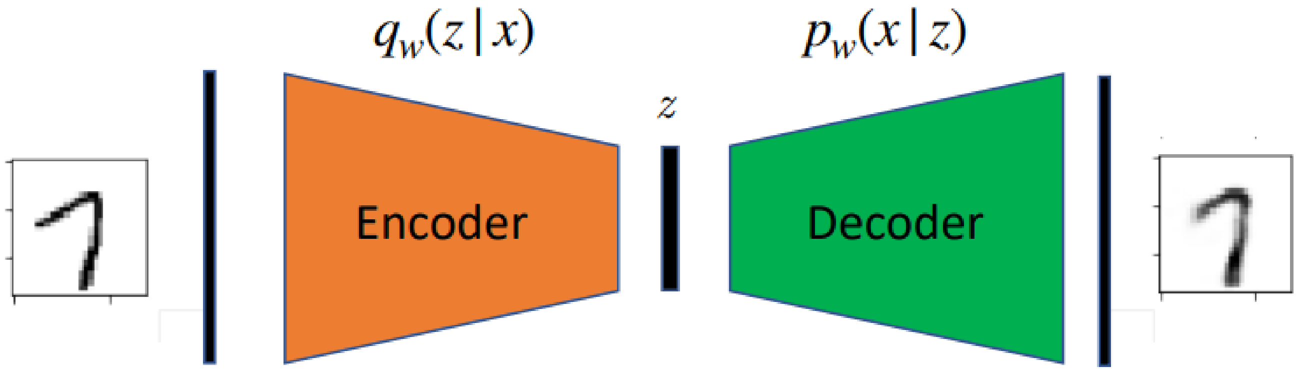
Si asume $p_w(x|z)$ que sigue Bernoulli multivariante, use la entropía cruzada; si asume que sigue la distribución normal, use MSE

MSE es lo mismo que la entropía cruzada entre la distribución empírica y un modelo gaussiano(Referencia: Deep Learning book by Goodfellow et al., pg. 132)

$$\mathcal{L} = -\mathbb{E}_{z\sim q_w(z|x^{[i]})}\left[\log p_w\left(x^{[i]}|z\right)\right] + \mathbf{KL}\left(q_w\left(z|x^{[i]}\right) \| p(z)\right)$$

Expected neg. log likelihood term; wrt to encoder distribution

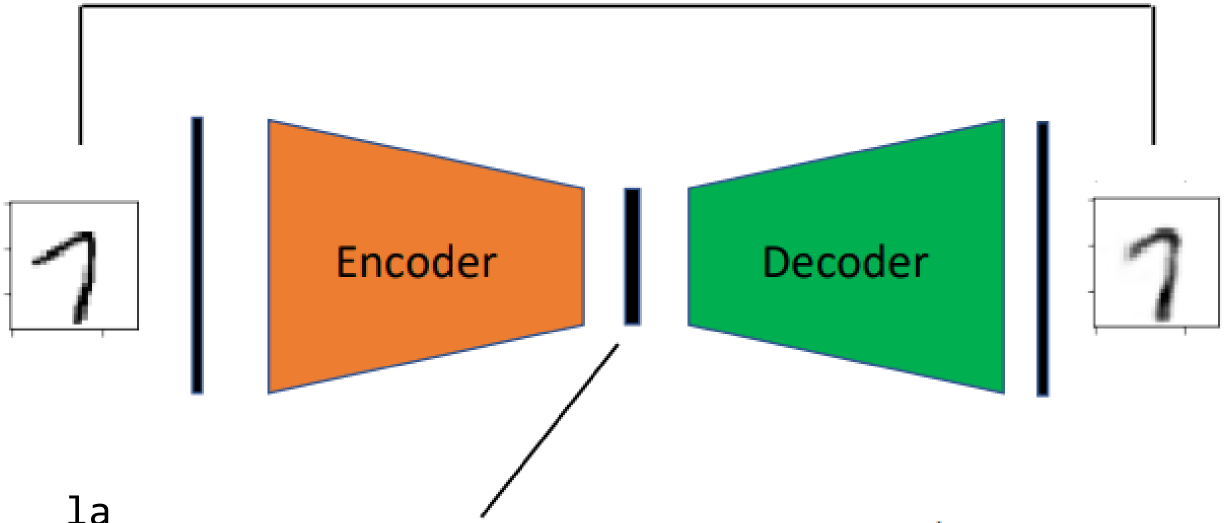
Kullback-Leibler divergence term where $p(z) = \mathcal{N}\left(\mu = 0, \sigma^2 = 1\right)$



Función de costo del VAE

1) Minimizar la pérdida con MSE: (asegura una buena reconstrucción):

$$\mathcal{L}_1 = ||\mathbf{x} - Dec(Enc(\mathbf{x}))||_2^2 = \sum_{i=1}^{d} (x_i - x_i')^2$$



2) Minimizar la divergencia KL:

$$\mathcal{L}_2 = D_{KL}\left[N(\mu,\sigma) \| N(0,1)\right] = -\frac{1}{2}\sum\left(1 + \log\left(\sigma^2\right) - \mu^2 - \sigma^2\right)$$

(asegura que el espacio latente sea continuo y sea una distribución normal estándar)

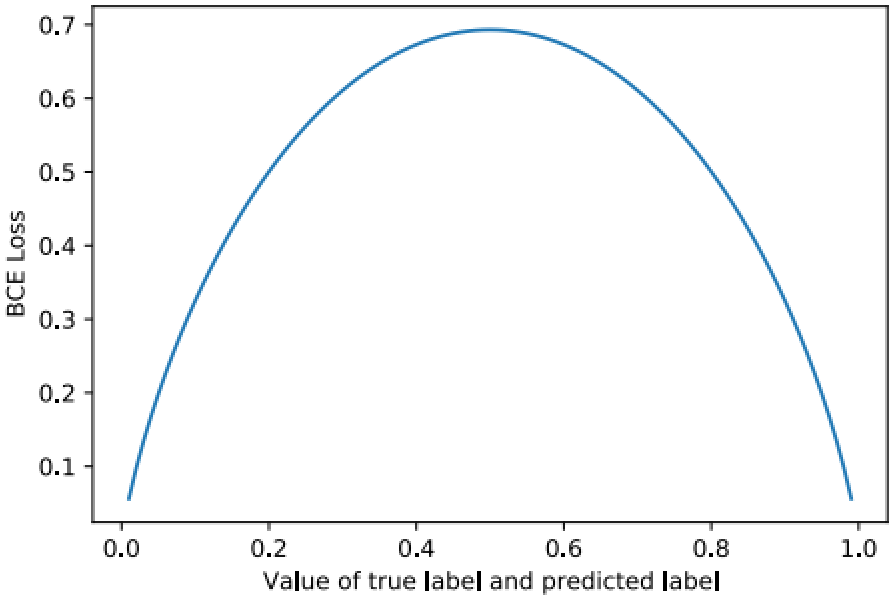
Pérdida general: $\mathcal{L} = \alpha \cdot \mathcal{L}_1 + \mathcal{L}_2$

Binary Cross Entropy vs MSE

La entropía cruzada no es simétrica:

$$H(p,q) = -\sum_{x\in\mathcal{X}} p(x)\cdot\log q(x)$$

pixel in x   pixel in x'



-0.8 * log(0.7) = 0.285340
-0.8 * log(0.9) = 0.0842884

-0.2 * log(0.1) = 0.460517
-0.2 * log(0.3) = 0.240795

Derivación de pérdidas KL

The encoder distribution is $q(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x))$ where $\Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$.

The latent prior is given by $p(z) = \mathcal{N}(0, I)$.

Both are multivariate Gaussians of dimension n, for which in general the KL divergence is:

$$\mathfrak{D}_{\text{KL}}[p_1 \parallel p_2] = \frac{1}{2}\left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - n + \text{tr}\{\Sigma_2^{-1}\Sigma_1\} + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1)\right]$$

where $p_1 = \mathcal{N}(\mu_1, \Sigma_1)$ and $p_2 = \mathcal{N}(\mu_2, \Sigma_2)$.

In the VAE case, $p_1 = q(z|x)$ and $p_2 = p(z)$, so $\mu_1 = \mu$, $\Sigma_1 = \Sigma$, $\mu_2 = \vec{0}$, $\Sigma_2 = I$. Thus:

$$\begin{aligned}
\mathfrak{D}_{\text{KL}}[q(z|x) \parallel p(z)] &= \frac{1}{2}\left[\log \frac{|\Sigma_2|}{|\Sigma_1|} - n + \text{tr}\{\Sigma_2^{-1}\Sigma_1\} + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1)\right] \\
&= \frac{1}{2}\left[\log \frac{|I|}{|\Sigma|} - n + \text{tr}\{I^{-1}\Sigma\} + (\vec{0} - \mu)^T I^{-1}(\vec{0} - \mu)\right] \\
&= \frac{1}{2}\left[-\log|\Sigma| - n + \text{tr}\{\Sigma\} + \mu^T\mu\right] \\
&= \frac{1}{2}\left[-\log \prod_i \sigma_i^2 - n + \sum_i \sigma_i^2 + \sum_i \mu_i^2\right] \\
&= \frac{1}{2}\left[-\sum_i \log \sigma_i^2 - n + \sum_i \sigma_i^2 + \sum_i \mu_i^2\right] \\
&= \frac{1}{2}\left[-\sum_i \left(\log \sigma_i^2 + 1\right) + \sum_i \sigma_i^2 + \sum_i \mu_i^2\right]
\end{aligned}$$

Fuente: https://stats.stackexchange.com/questions/318748/deriving-the-kl-divergence-loss-for-vaes/370048#370048

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. Muestreo de un VAE
3. Truco de Log-Var
4. Función de pérdida del VAE
5. **VAE para MNIST en PyTorch**
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. Muestreo de un VAE
3. Truco de Log-Var
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. **VAE para rostros PyTorch**
7. VAEs y Aritmética del espacio latente
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Un codificador automático variable para imágenes faciales

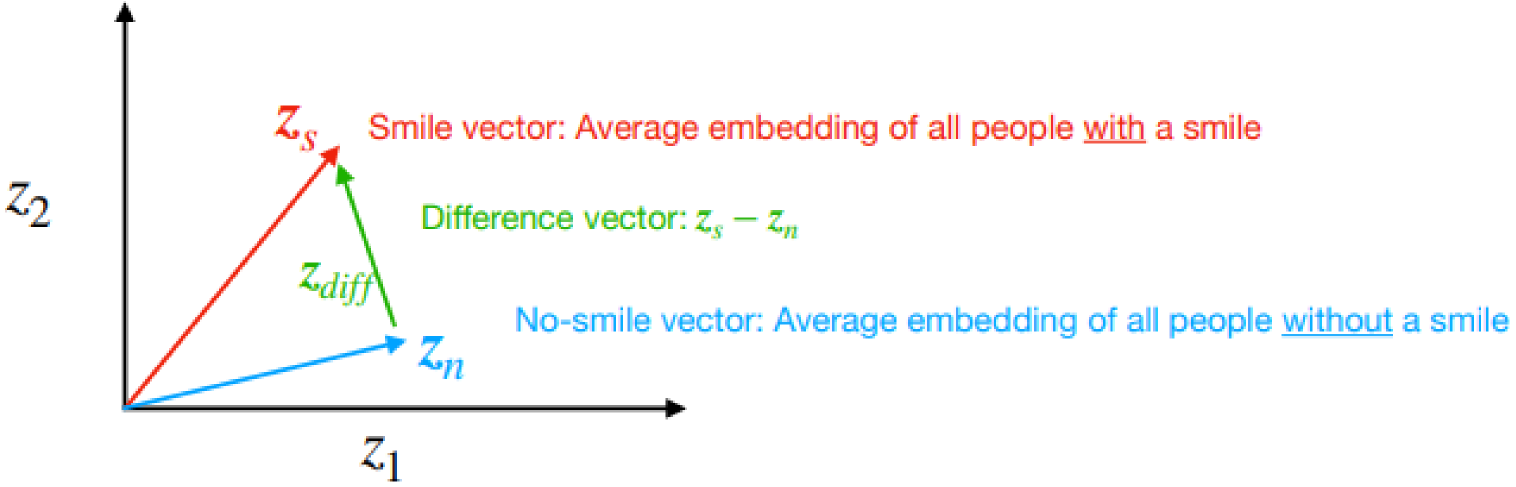
Cambios en la arquitectura en comparación con el ejemplo anterior de MNIST:

- Basado en: Deep Feature Consistent Variational Autoencoder
  httpS://ieeexplore.ieee.org/document/7926714
- 1 -> 3 canales de color
- 2 -> 200 características en el espacio latente
- BatchNorm, Dropout
- Aumentar el coeficiente de pérdida de reconstrucción

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. Muestreo de un VAE
3. Truco de Log-Var
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. **VAEs y Aritmética del espacio latente**
8. Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente

Aritmética del espacio latente



Por ejemplo, podemos darle una sonrisa a una persona triste así:

- $z_{new} = z_{orig} + \alpha \cdot z_{diff}$

Codificadores automáticos vs codificadores automáticos variacionales

1. Descripción general del VAE
2. Muestreo de un VAE
3. Truco de Log-Var
4. Función de pérdida del VAE
5. VAE para MNIST en PyTorch
6. VAE para rostros PyTorch
7. VAEs y Aritmética del espacio latente
8. **Aritmética del espacio latente VAE en PyTorch - Hacer sonreír a la gente**