

# Introducción al Aprendizaje Probabilístico Profundo. VAEs

Rafael Molina

D.P. Kingma, M. Welling, **Auto-Encoding Variational Bayes**, The International Conference on Learning Representations (ICLR), 2014

# Contenido

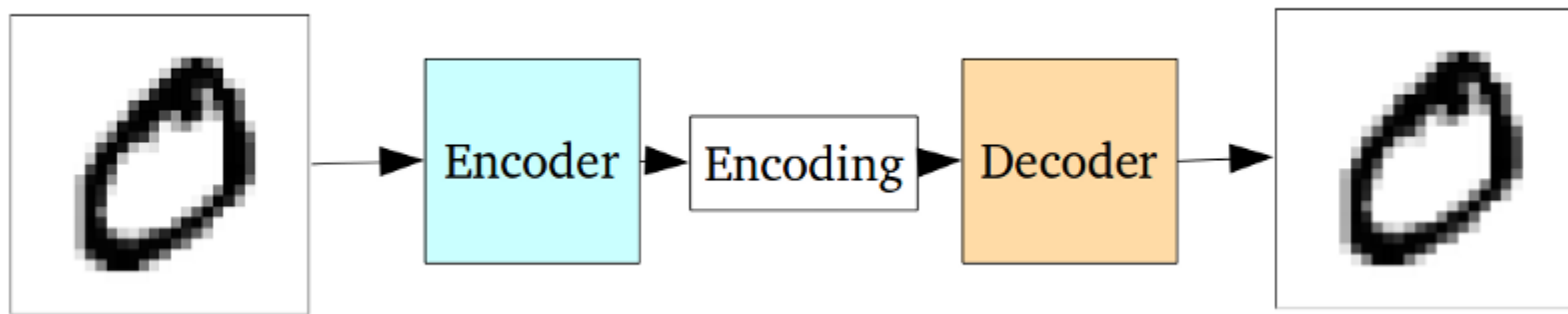
1. Introducción
2. Método
3. Ejemplo: Autoencoder Variacional
4. Resumen

# I. Introducción

## Intuitively Understanding Variational Autoencoders

<https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>

Un autoencoder es un par de redes conectadas: un codificador (encoder) y un decodificador (decoder). El codificador acepta una entrada y la convierte en una representación más pequeña y densa que el decodificador puede convertir de nuevo en la entrada original

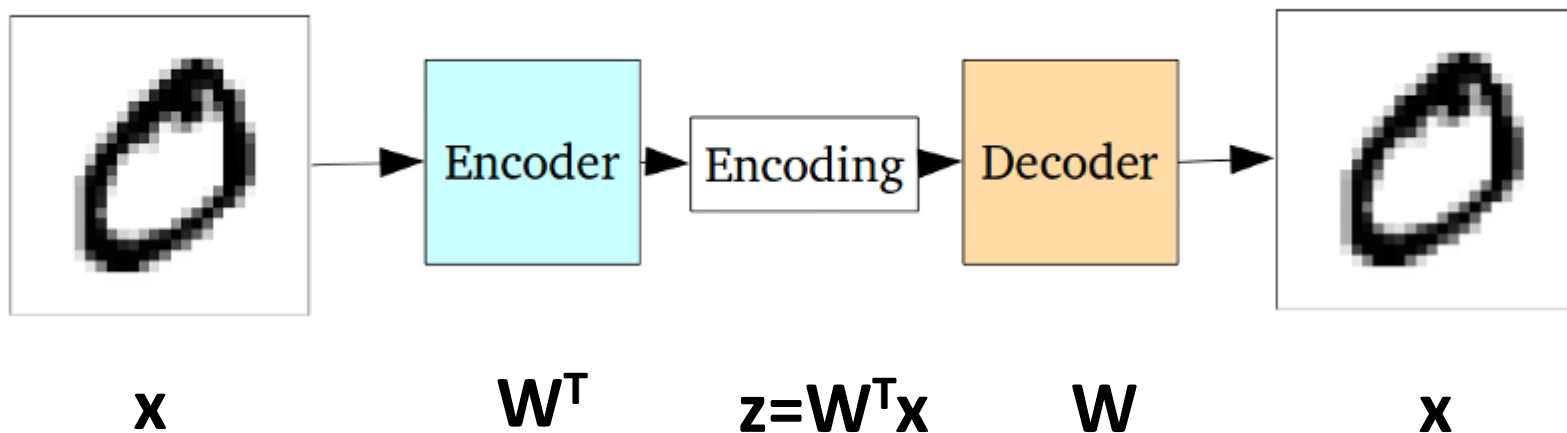


# I. Introducción

## Intuitively Understanding Variational Autoencoders

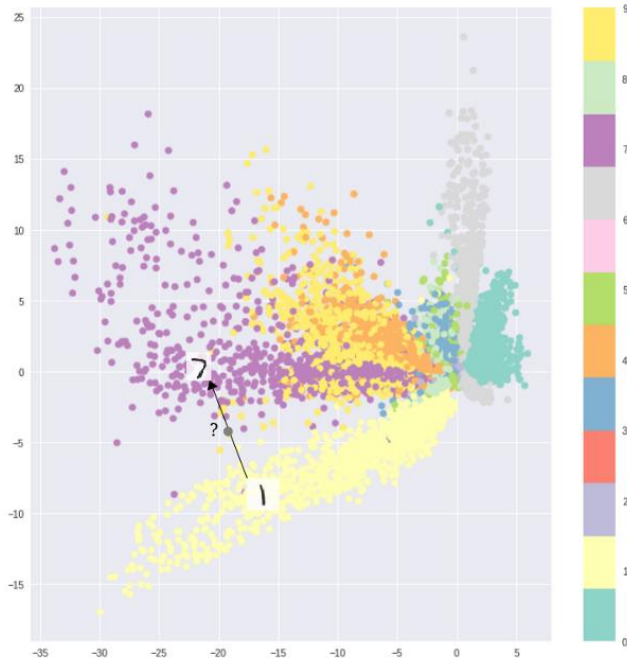
<https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>

PCA es un autoencoder muy simple



# I. Introducción

El problema fundamental con los autoencoders, cuando queremos generar muestras con ellos, es que el espacio latente, en quien convertimos los inputs y donde están los vectores codificados, puede no ser continuo y no permitir una interpolación fácil.

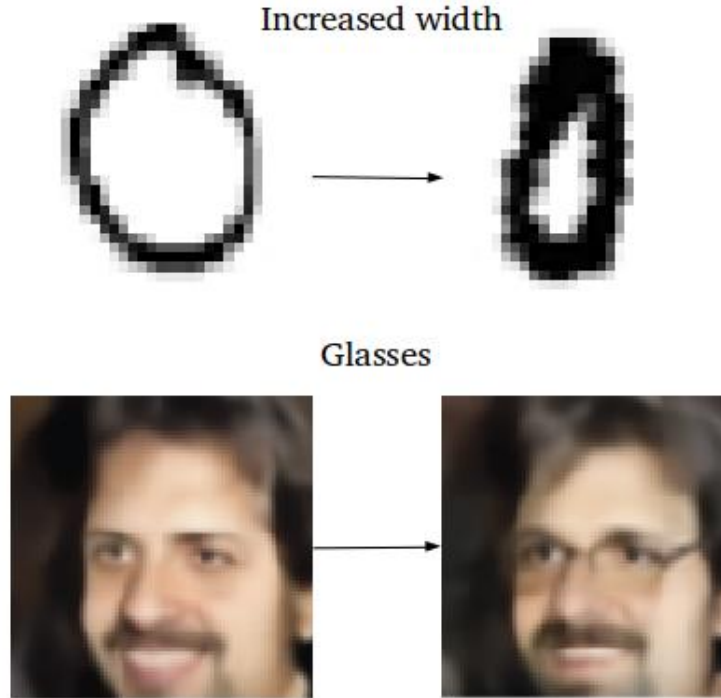


Por ejemplo, si entrenamos un autoencoder en MNIST usando un espacio latente 2D, obtenemos el gráfico de la izquierda.

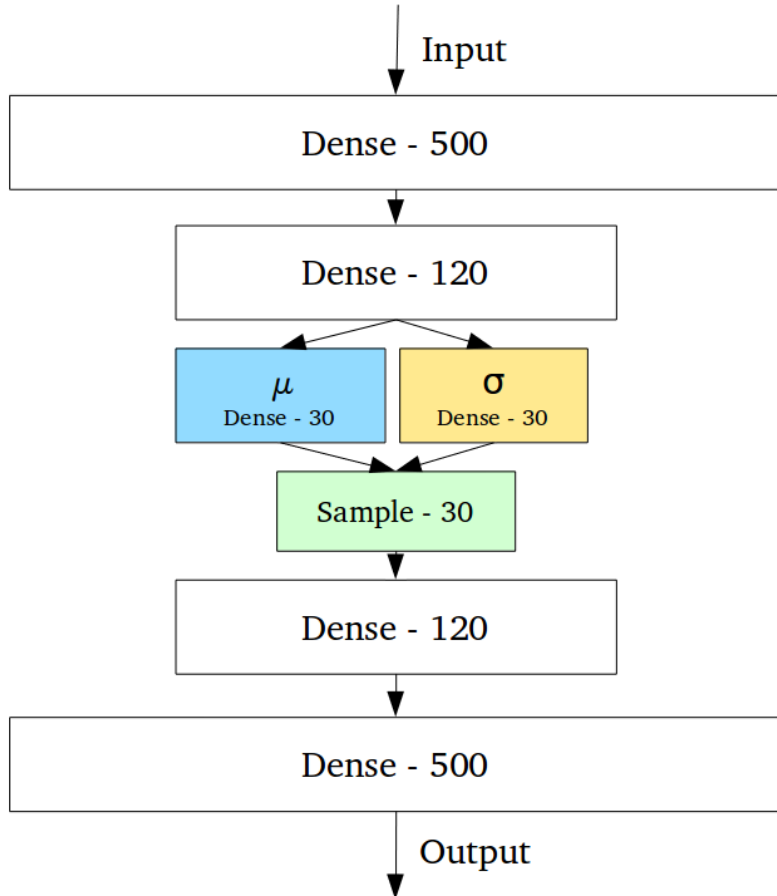
Sin embargo, si queremos muestrear el espacio latente o generar variaciones en los datos observados (en nuestro caso una imagen) el modelo no es muy útil.

# I. Introducción

Obtener imágenes de la forma siguiente no es posible con los autoencoders



# I. Introducción



Los VAEs tienen una propiedad que los hace distintos de los autoencoders clásicos. El espacio latente es, por diseño, continuo y permite (y tiene sentido) su interpolación

Convertiremos cada entrada en una distribución de probabilidad en el espacio de variables latentes.

Frecuentemente, sus componentes serán independientes. Cada una de ellas una normal con una media y una desviación típica (también podemos correlarlas)

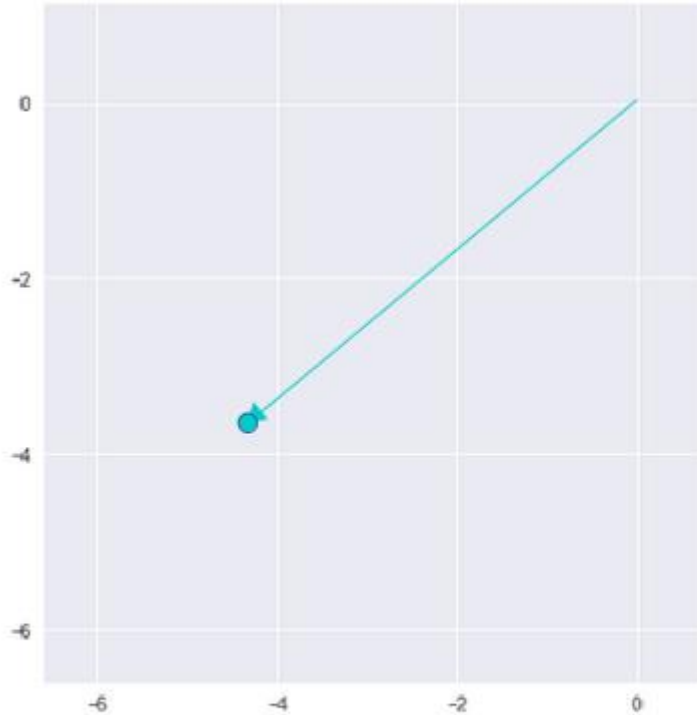
# I. Introducción

el encoder se llama a veces  
modelo de reconocimiento y el  
decoder modelo generativo

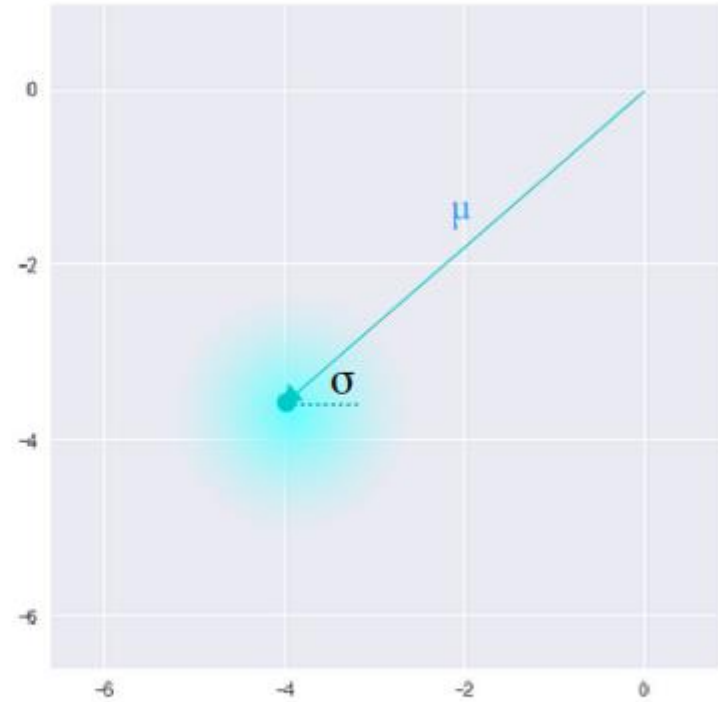


# I. Introducción

Intuitivamente



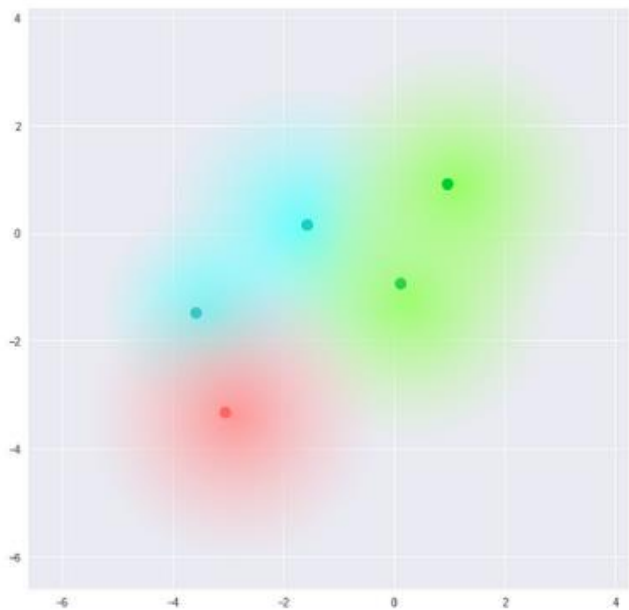
Standard Autoencoder  
(direct encoding coordinates)



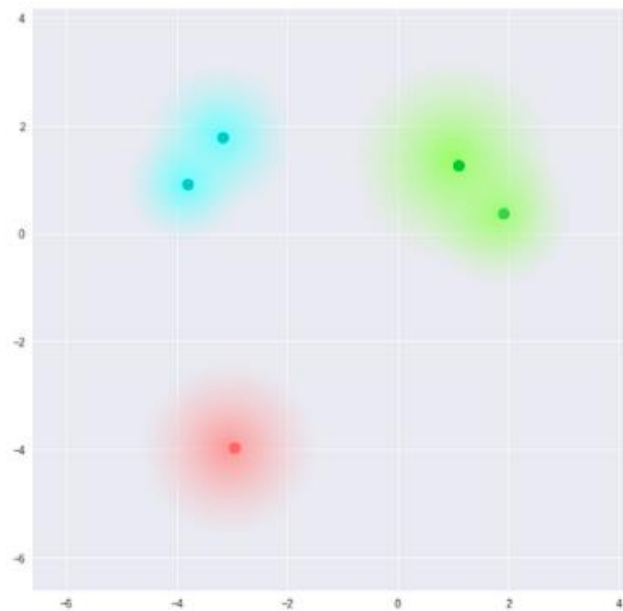
Variational Autoencoder  
( $\mu$  and  $\sigma$  initialize a probability distribution)

# I. Introducción

Si usamos un autoencoder clásico para aprender las medias y varianzas (variables latentes) y luego muestreamos para decodificar, en lugar de obtener la imagen de la izquierda (que es lo que queremos), obtendremos la de la derecha

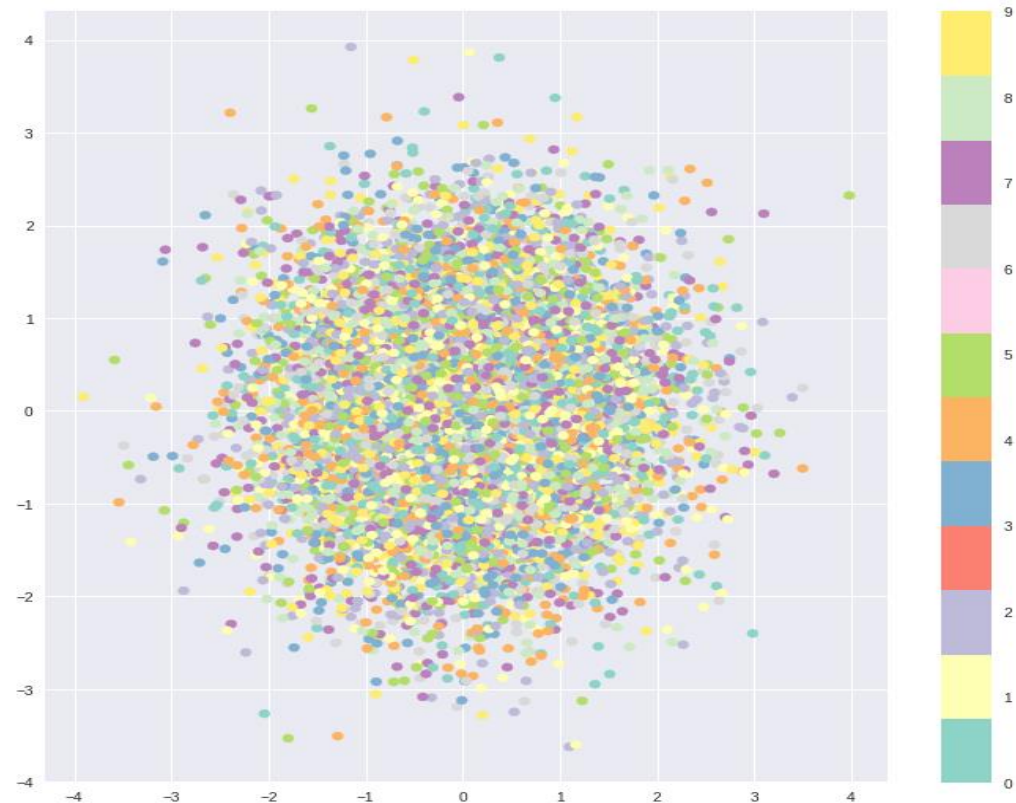


What we require



What we may inadvertently end up with

# I. Introducción



Si no pedimos fidelidad a los datos y sólo le indicamos que nuestras distribuciones Normales deben estar cercanas (en sentido KL) a la  $N(0, I)$  obtenemos la siguiente representación.

Esta representación tampoco nos es útil. No podemos decodificar nada.

# I. Introducción

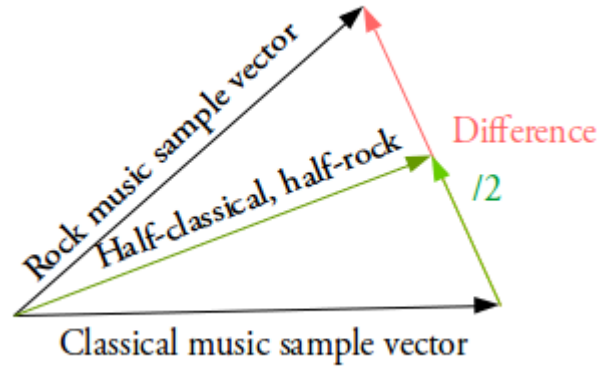


Finalmente, si combinamos los dos criterios obtendremos la solución buscada

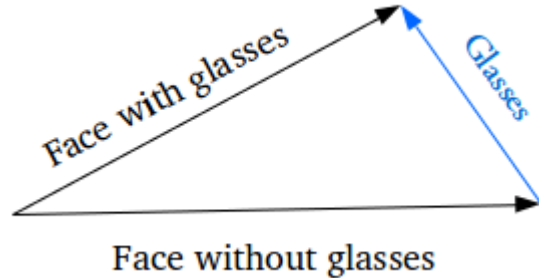
Observa que ahora cuando interpolemos no habrá gaps en los clusters y sí una mezcla de rasgos que el decodificador puede entender

# I. Introducción

¿Cómo obtenemos esas interpolaciones de las que hemos hablado?. Por simple aritmética de vectores en el espacio latente



Si queremos generar una nueva muestra en el punto medio de dos muestras: calcula la diferencia de las medias y añade la mitad de la diferencia al original.



Si queremos generar gafas en una cara, encuentra dos muestras una con gafas, la otra sin ellas y réstales, Añade esta diferencia a cualquier otra cara.

Para ..... (y lo entenderás más claramente después): el encoder se llama a veces modelo de reconocimiento y el decoder modelo generativo.

## II. Método

Consideremos una base de datos  $X=\{x^{(i)}\}_{i=1,\dots,N}$  formada por  $N$  muestras i.i.d de una variable discreta o continua.

Suponemos que los datos son generados por un proceso aleatorio que tiene asociado una variable continua no observable  $z$ . El proceso tiene dos pasos:

- $z^{(i)}$  es generado por una distribución a priori  $p_{\theta^*}(z)$
- $x^{(i)}$  es generado de la distribución condicionada  $p_{\theta^*}(x|z)$

Por simplicidad vamos a suponer que  $p_{\theta^*}(z)$  no tiene realmente parámetros a estimar, es decir, escribimos  $p(z)$ .

## II. Método

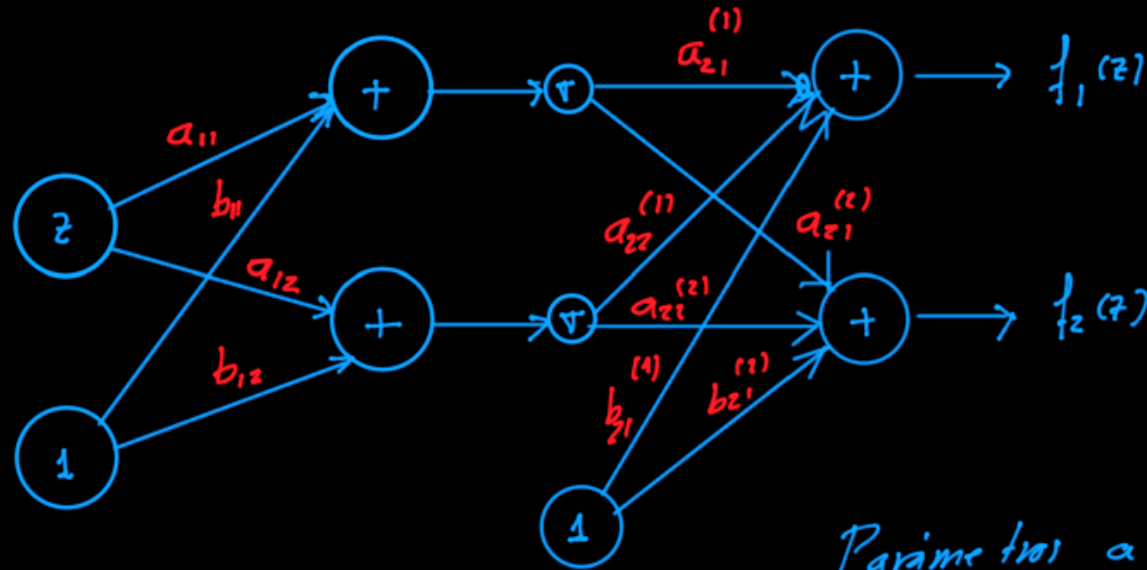
Supondremos que la a priori y el modelo de observación ( $p(z)$  y  $p_{\theta^*}(x|z)$ , respectivamente) vienen de una familia de distribuciones paramétricas ( $p(z)$  y  $p_{\theta}(x|z)$ , respectivamente). Es decir, nuestros datos se obtienen con  $\theta = \theta^*$ .

- Un caso típico es

$$p(z) = N(z|0, I) \quad p_{\theta}(x|z) = N(x|f_{\theta}(z), \sigma^2 I)$$

CNN, DNN, ...

Criterio de error  $\frac{1}{2} \left( (x_1 - f_1(z))^2 + (x_2 - f_2(z))^2 \right)$



$z$  tiene una única componente  
 $\begin{pmatrix} f_1(z) \\ f_2(z) \end{pmatrix}$  es un vector bidimensional

Parámetros a  
 estimar  
 Todos los  $a$   
 y todos los  $b$



## II. Método

Queremos abordar problemas como:

- Intratabilidad de la verosimilitud marginal  $p_{\theta}(x)$ .
  - Estimación de máxima verosimilitud de  $\theta$ .
  - Obtener buenas aproximaciones de la marginal de las observaciones.
- Obtener buenas aproximaciones de la distribución de  $z$  dado  $x$ .
- Ser capaces de trabajar con grandes bases de datos.

## II. Método

- Para alcanzar estos objetivos, vamos a introducir un **modelo de reconocimiento**  $q_{\phi}(z|x)$  que aproximará la verdadera a posteriori  $p_{\theta}(z|x)$ .
- Desde el punto de vista de la teoría de la codificación,  $z$  tiene interpretación como variable latente o código. Por ello, **el modelo de reconocimiento recibirá también el nombre de codificador**.
- De igual manera  $p_{\theta}(x|z)$  recibirá el nombre de **decodificador** y el modelo  $p(z)p_{\theta}(x|z)$  será un **modelo generativo**.

## II. Método

La verosimilitud marginal es la suma de las verosimilitudes marginales de los puntos individuales:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \sum_{i=1}^N \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$$

Usando la desigualdad de Jensen tendremos

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} [-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})]$$



Recuerda: puede ser cualquier distribución en  $\mathbf{z}$

## II. Método

La cota inferior puede reescribirse como

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right]$$

Queremos diferenciar y optimizar la cota inferior

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

con respecto a  $\boldsymbol{\phi}$  y  $\boldsymbol{\theta}$  para aproximar el MLE de  $\boldsymbol{\theta}$  y encontrar el mejor modelo de reconocimiento, es decir, los mejores parámetros  $\boldsymbol{\phi}$ .

**Lo que queremos hacer es problemático porque las dos integrales de arriba podrían no ser calculables analíticamente.**

## II. Método

Ten claro que:

- estamos optimizando una cota inferior de la verosimilitud,
- no estamos optimizando la verosimilitud,
- que la cota sea buena o mala depende de lo bueno que sea la arquitectura para representar la distribución a posteriori.

## II. Método

¿Cómo calculamos o aproximamos

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log p_{\theta}(\mathbf{x}^{(i)}|\mathbf{z}) \right] \quad ?$$

### La trampa de la reparametrización

Sea  $\mathbf{z}$  una variable continua y

$$\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$$

frecuentemente es posible escribir

$$\mathbf{z} = g_{\phi}(\epsilon, \mathbf{x})$$

donde  $\epsilon$  es una variable auxiliar con marginal independiente  $p(\epsilon)$  y  $g_{\phi}(\cdot)$  es una función vector-valuada parametrizada por  $\phi$ .

## II. Método

Si tenemos una función  $f(\mathbf{z})$  y queremos calcular su media con respecto a  $q_\phi(\mathbf{z})$  podemos aproximarla de la forma siguiente

$$\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^L f(g_\phi(\mathbf{x}, \epsilon^{(l)}))$$

## II. Método

Por ejemplo, si

$$z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$$

puedo escribir

$$z = \mu + \sigma\epsilon \qquad \epsilon \sim \mathcal{N}(0, 1)$$

y tendría

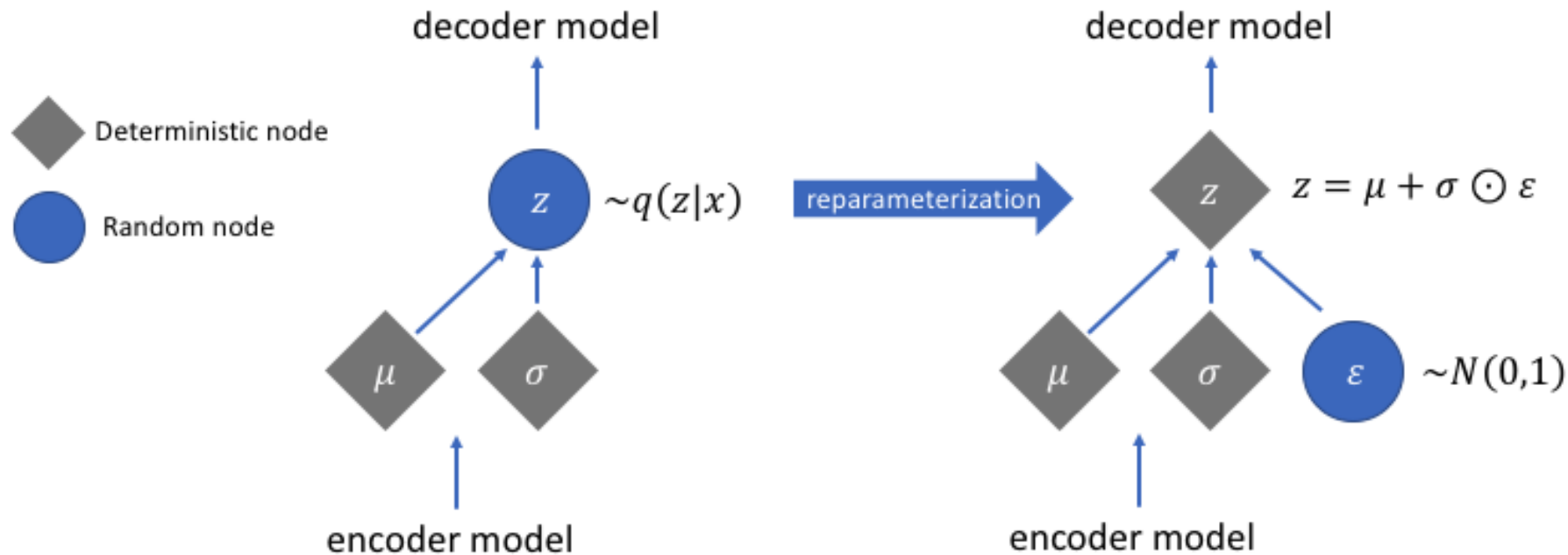
$$\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)} [f(z)] = \mathbb{E}_{\mathcal{N}(\epsilon;0,1)} [f(\mu + \sigma\epsilon)] \simeq \frac{1}{L} \sum_{l=1}^L f(\mu + \sigma\epsilon^{(l)})$$

donde L es el número de muestras que hemos tomado de la  $\mathcal{N}(0,1)$



# II. Método

## Gráficamente



## II. Método

Si podemos aplicar la trampa de la reparametrización a nuestro problema. Es decir, si para  $\tilde{\mathbf{z}} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$  podemos escribir

$$\tilde{\mathbf{z}} = g_{\phi}(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{con} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

tendremos

$$\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} [f(\mathbf{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[ f(g_{\phi}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)})) \right] \simeq \frac{1}{L} \sum_{l=1}^L f(g_{\phi}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)}))$$

donde

$$\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

## II. Método

Aplicando esta técnica, que se llama **Stochastic Gradient Variational Bayes (SGVB)**, a nuestro problema podemos aproximar

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right]$$

mediante

$$\tilde{\mathcal{L}}^A(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = \frac{1}{L} \sum_{l=1}^L (\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_{\phi}(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)}))$$

donde

$$\mathbf{z}^{(i,l)} = g_{\phi}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

## II. Método

Frecuentemente la divergencia en la ecuación

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right]$$

puede ser calculada analíticamente (lo veremos en la próxima sección) y sólo tendremos que estimar por muestreo

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})} \left[ \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) \right]$$

# III. Ejemplo: Autoencoder Variacional

Consideremos el siguiente modelo

- **Modelo a priori**

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$$

- **Modelo de observación:** una Gaussiana Multivariante

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

Los parámetros de esta distribución corresponden a los de un Perceptrón Multicapa (MLP). Es decir,

# III. Ejemplo: Autoencoder Variacional

$$\log p(\mathbf{x}|\mathbf{z}) = \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \sigma^2 \mathbf{I})$$

$$\text{donde } \boldsymbol{\mu} = \mathbf{W}_4 \mathbf{h} + \mathbf{b}_4$$

$$\log \sigma^2 = \mathbf{W}_5 \mathbf{h} + \mathbf{b}_5$$

$$\mathbf{h} = \tanh(\mathbf{W}_3 \mathbf{z} + \mathbf{b}_3)$$

$$\theta = \{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$$

# III. Ejemplo: Autoencoder Variacional

**Modelo variacional que aproxima la distribución a posteriori**

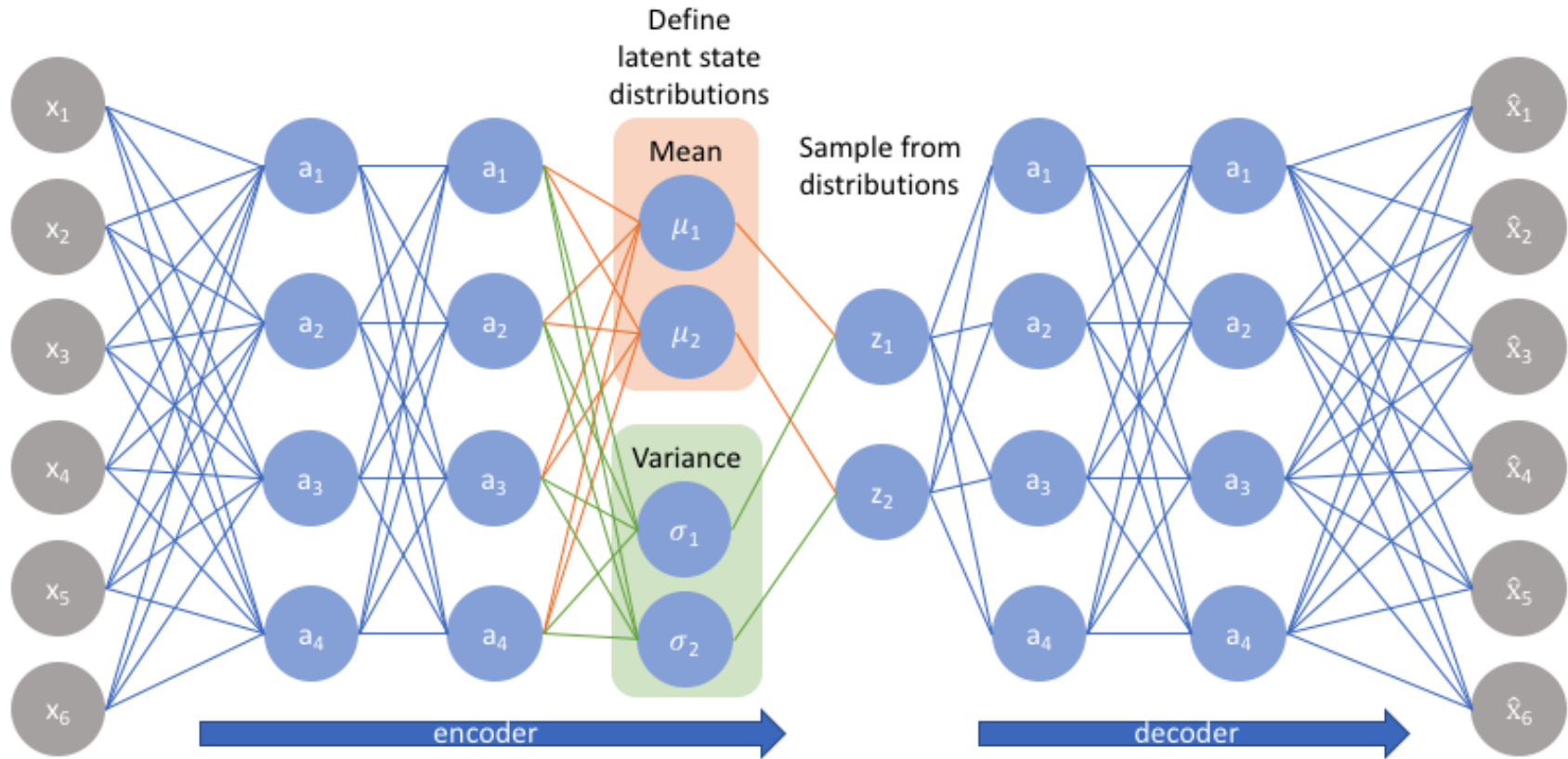
$$\log q_{\phi}(\mathbf{z} | \mathbf{x}^{(i)}) = \log N(\mathbf{z}, \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)})$$

diagonal

En este caso  $\boldsymbol{\mu}^{(i)}$  y los elementos diagonales de  $\boldsymbol{\Sigma}^{(i)}$  son salidas del codificador, es decir, funciones no lineales de los datos  $\mathbf{x}^{(i)}$  y los parámetros variacionales.

Importante: Cuando el MLP se usa como codificador,  $q_{\phi}(\mathbf{z} | \mathbf{x})$ , los papeles de  $\mathbf{z}$  y  $\mathbf{x}$  se intercambian. Los pesos y sesgos son los parámetros variacionales  $\phi$ . Mira el espejo de las arquitecturas a continuación

# III. Ejemplo: Autoencoder Variacional





# III. Ejemplo: Autoencoder Variacional

Calculemos

$$-D_{KL}(q_{\phi}(\mathbf{z})||p_{\theta}(\mathbf{z}))$$

para distribuciones Gaussianas. Tenemos

$$\int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) d\mathbf{z} = -\frac{J}{2} \log(2\pi) - \frac{1}{2} \sum_{j=1}^J (\mu_j^2 + \sigma_j^2)$$

$$\int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) d\mathbf{z} = -\frac{J}{2} \log(2\pi) - \frac{1}{2} \sum_{j=1}^J (1 + \log \sigma_j^2)$$

y, por tanto,

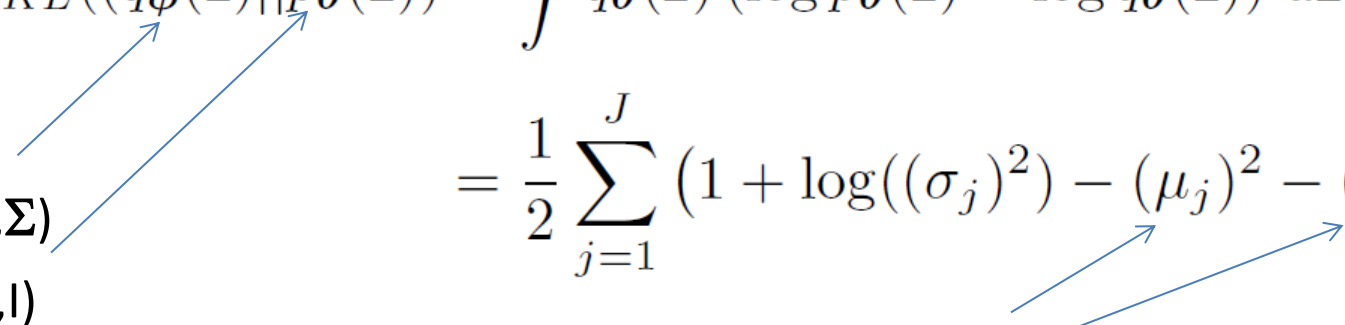
Observa que estamos usando  $\boldsymbol{\sigma}^2$  para denotar la matriz de covarianza diagonal  $\boldsymbol{\Sigma}$  y que sus elementos diagonales los notamos  $\sigma_j^2$

# III. Ejemplo: Autoencoder Variacional

Por tanto

$$\begin{aligned} -D_{KL}((q_{\phi}(\mathbf{z})||p_{\theta}(\mathbf{z}))) &= \int q_{\theta}(\mathbf{z}) (\log p_{\theta}(\mathbf{z}) - \log q_{\theta}(\mathbf{z})) d\mathbf{z} \\ &= \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2) \end{aligned}$$

$N(\mu, \Sigma)$   
 $N(0, I)$



Componentes de  $N(\mu, \Sigma)$ , la matriz de covarianzas se supone diagonal

# III. Ejemplo: Autoencoder Variacional

Obtenemos, por tanto,

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^J \left( 1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right) + \frac{1}{L} \sum_{l=1}^L \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)})$$

con

$$\mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\Sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \quad \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0, \mathbf{I})$$

# IV. Resumen

¿Qué hemos obtenido?

1. Una aproximación de la distribución a posteriori notada  $q_{\phi}(z|x)$  que aproxima  $p_{\theta}(z|x)$
2. Un modelo generativo  $p_{\theta}(z,x)$  que puede usarse de la misma forma que el modelo obtenido por una GAN (una lectura recomendada).

Por último, piensa en el camino recorrido desde las PCAs a las VAEs. Intenta resumir lo que has aprendido.