Auto-encoders y variational auto-encoders

Auto-encoders y variational auto-encoders

Introducción

Hasta ahora hemos estado hablando siempre de **aprendizaje profundo supervisado**, pero también podemos resolver problemas **no supervisados**.

Objetivo:

- Extraer patrones directamente de los datos "sin etiquetar", solo tenemos x y no y.

Tareas comunes:

- Modelos generativos: Entender la distribución de x y generar nuevas muestras.
- Autoencoders: "Comprimir" x proyectándolo en un espacio de menor dimensión.

The brain has about 10^{14} *synapses and we only live for about* 10^{9} *seconds. So we have a lot more parameters than data. This motivates the idea that we must do a lot of* **unsupervised learning** *since the perceptual input (including proprioception) is the only place we can get* 10^{5} *dimensions of constraint per second.*

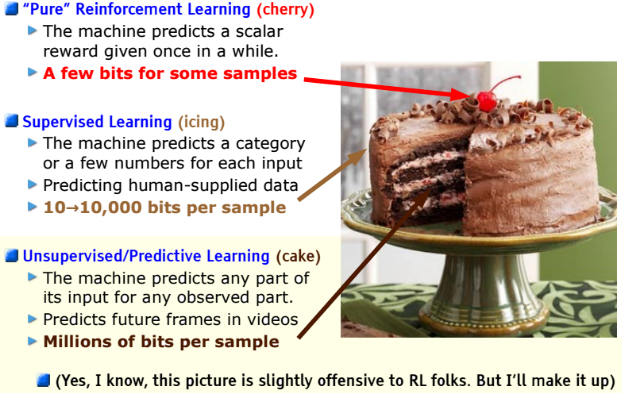
Geoffrey Hinton, 2014

Introducción

Aprendizaje no supervisado

We need tremendous amount of information to build machines that have common sense and generalize.

Yann LeCun, 2016



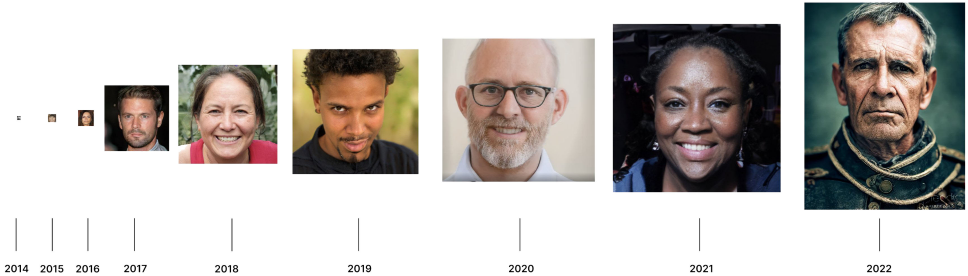
Un **modelo generativo** es un modelo probabilístico p que puede ser utilizado como un "simulador de datos".

Su propósito es generar datos sintéticos pero realistas de alta dimensión

$$\mathbf{x} \sim p_\theta(\mathbf{x}),$$

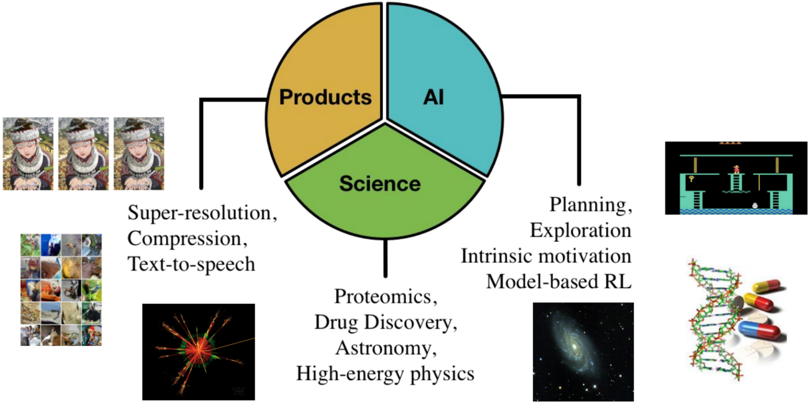
que se asemejen lo más posible a la distribución desconocida de datos $p(\mathbf{x})$.

Introducción
Modelos generativos



Ley de Moore de los modelos generativos de imágenes

Algunas aplicaciones:



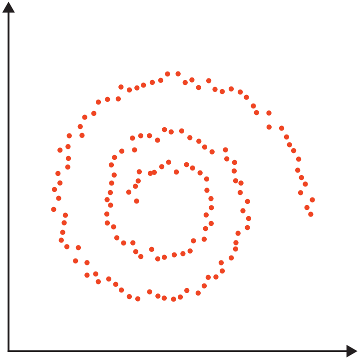
Los modelos generativos tienen un rol muy importante en muchos problemas actuales

Auto-encoders y variational auto-encoders

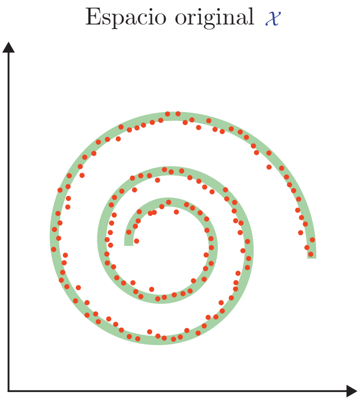
Auto-encoders

Auto-encoders

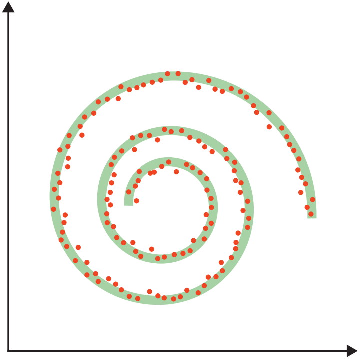
Espacio original $\mathcal{X}$



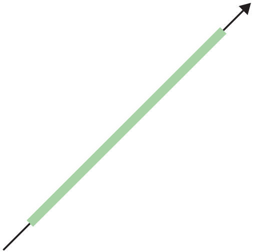
Auto-encoders



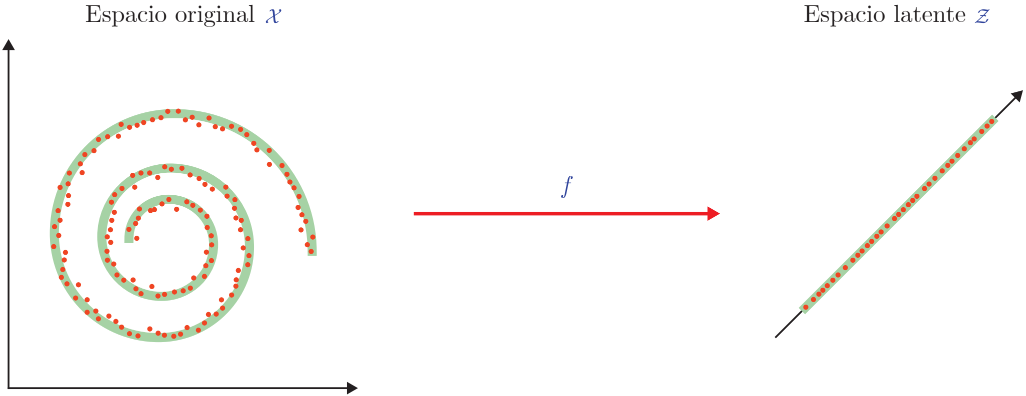
Espacio original $\mathcal{X}$



Espacio latente $\mathcal{Z}$

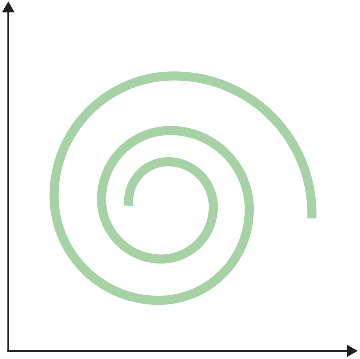


Auto-encoders

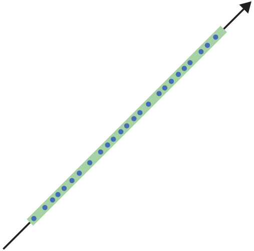


Auto-encoders

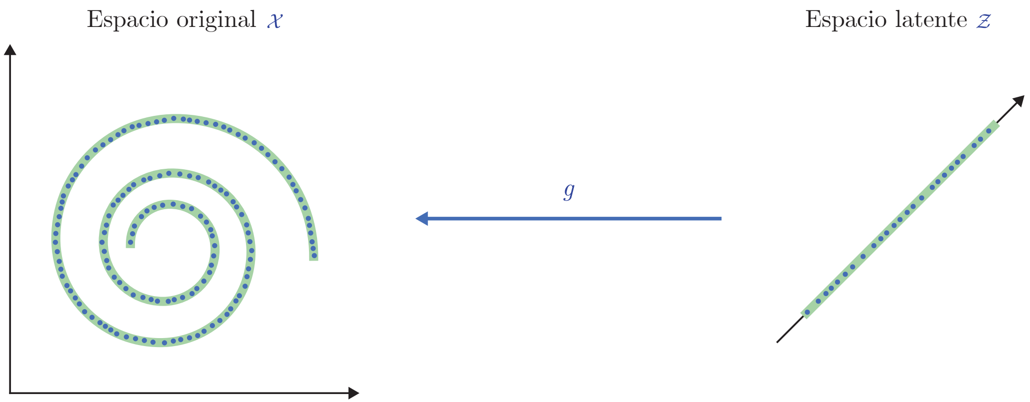
Espacio original $\mathcal{X}$



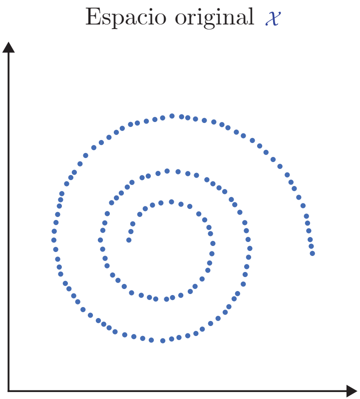
Espacio latente $\mathcal{Z}$



Auto-encoders



Auto-encoders

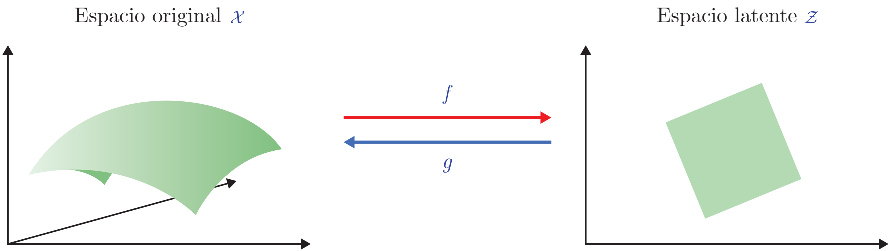


Auto-encoders

Un **auto-encoder** es una función compuesta a partir de:

- Un **encoder** f que proyecta del espacio original $\mathcal{X}$ al espacio latente $\mathcal{Z}$.
- Un **decoder** g que proyecta de vuelta al espacio original.

El objetivo es que $g \circ f$, es decir, que la composición de funciones se aproxime lo máximo posible a los datos originales o función identidad.



Auto-encoders

Siendo $p(\mathbf{x})$ la distribución de los datos en $\mathcal{X}$, un buen auto-encoder puede caracterizarse con la *reconstruction loss*:

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[||\mathbf{x} - g \circ f(\mathbf{x})||^2\right] \approx 0.$$

Esta función de pérdida mide como de bien el auto-encoder puede reconstruir los datos originales.

Dadas dos funciones de proyección con parámetros $f(\cdot;\theta_f)$ and $g(\cdot;\theta_g)$, el entrenamiento consiste aprender los parámetros que minimicen dicha loss:

$$\theta_f, \theta_g = \arg\min_{\theta_f,\theta_g} \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}_i - g(f(\mathbf{x}_i, \theta_f), \theta_g)||^2.$$

Imaginemos, por ejemplo, un auto-encoder lineal con

$$f : \mathbf{z} = \mathbf{U}^T \mathbf{x}$$
$$g : \hat{\mathbf{x}} = \mathbf{U}\mathbf{z},$$

con $\mathbf{U} \in \mathbb{R}^{p \times d}$, el *reconstruction loss* se reduce a

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[||\mathbf{x} - \mathbf{U}\mathbf{U}^T \mathbf{x}||^2 \right].$$

Auto-encoders y variational auto-encoders

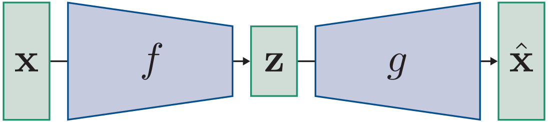
Deep Auto-encoders

Mayor profundidad

Para obtener mejores resultados, en vez de proyecciones lineales se suelen utilizar redes neuronales profundas en f y g.

Algunos ejemplos:

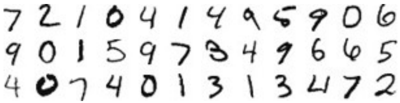
- Combinando un MLP encoder $f : \mathbb{R}^p \to \mathbb{R}^d$ con un MLP decoder $g : \mathbb{R}^d \to \mathbb{R}^p$.
- Combinando un convolutional network encoder $f : \mathbb{R}^{w\times h\times c} \to \mathbb{R}^d$ con un decoder decoder $g : \mathbb{R}^d \to \mathbb{R}^{w\times h\times c}$ compuesto de capas convolucionales reciprocas.



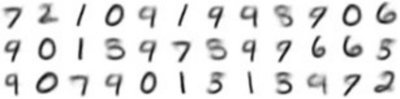
Deep Auto-encoders

Ejemplo MNIST

Datos originales **x** con $d = 784$.

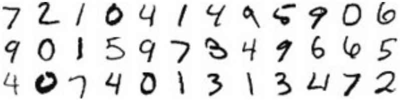


Resultado de auto-encoder $g \circ f$ creado a partir de CNN con $d = 2$.

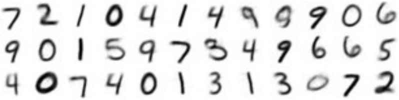


Resultado de auto-encoder $g \circ f$ creado a partir de PCA con $d = 2$.

Datos originales **x** con $d = 784$.

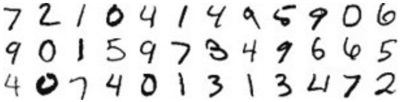


Resultado de auto-encoder $g \circ f$ creado a partir de CNN con $d = 4$.

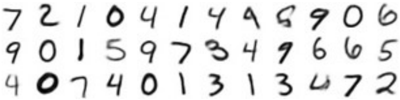


Resultado de auto-encoder $g \circ f$ creado a partir de PCA con $d = 4$.

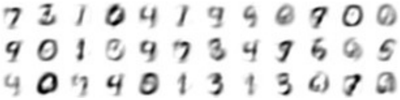
Datos originales $\mathbf{x}$ con $d = 784$.



Resultado de auto-encoder $g \circ f$ creado a partir de CNN con $d = 8$.



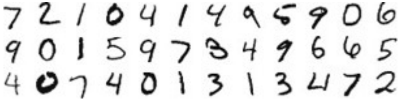
Resultado de auto-encoder $g \circ f$ creado a partir de PCA con $d = 8$.



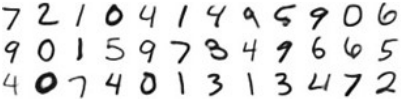
Deep Auto-encoders

Ejemplo MNIST

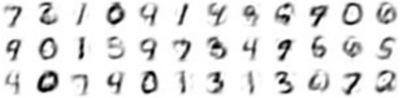
Datos originales $\mathbf{x}$ con $d = 784$.



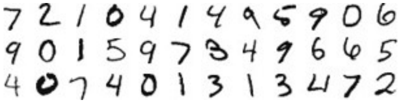
Resultado de auto-encoder $g \circ f$ creado a partir de CNN con $d = 16$.



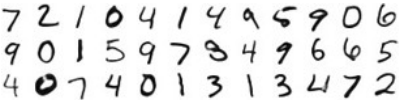
Resultado de auto-encoder $g \circ f$ creado a partir de PCA con $d = 16$.



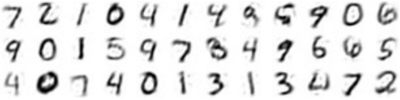
Datos originales **x** con $d = 784$.



Resultado de auto-encoder $g \circ f$ creado a partir de CNN con $d = 32$.



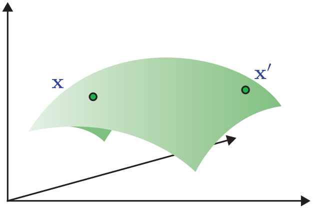
Resultado de auto-encoder $g \circ f$ creado a partir de PCA con $d = 32$.



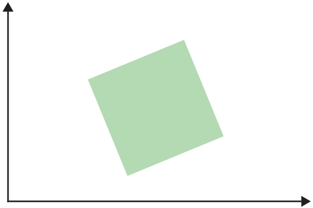
Interpolar puntos

Para comprender la representación latente aprendida, podemos elegir dos muestras $\mathbf{x}$ y $\mathbf{x}'$ al azar e interpolar muestras a lo largo de la línea en el espacio latente.

Espacio original $\mathcal{X}$

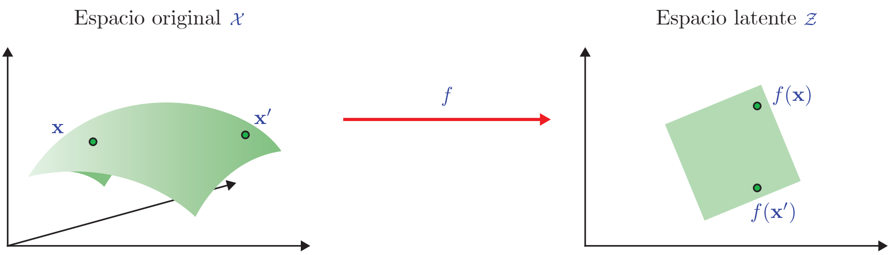


Espacio latente $\mathcal{Z}$



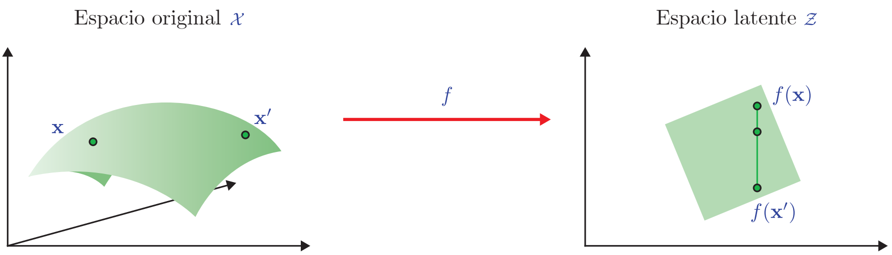
Interpolar puntos

Para comprender la representación latente aprendida, podemos elegir dos muestras $\mathbf{x}$ y $\mathbf{x}'$ al azar e interpolar muestras a lo largo de la línea en el espacio latente.



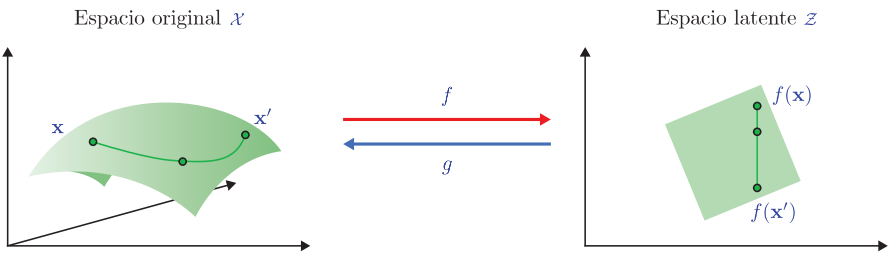
Interpolar puntos

Para comprender la representación latente aprendida, podemos elegir dos muestras **x** y **x**′ al azar e interpolar muestras a lo largo de la línea en el espacio latente.



Interpolar puntos

Para comprender la representación latente aprendida, podemos elegir dos muestras $\mathbf{x}$ y $\mathbf{x}'$ al azar e interpolar muestras a lo largo de la línea en el espacio latente.



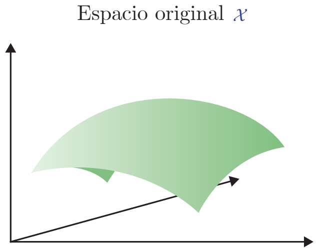
Interpolación de PCA ($d = 32$).

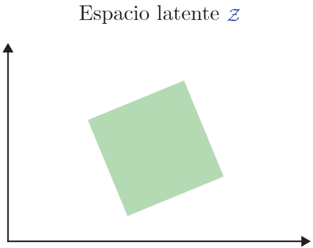
Interpolación de Auto-encoder ($d = 32$).

Generar nuevos datos

Además de generar valores intermedios entre dos datos existentes, también podemos utilizar el decoder para obtener la representación en $\mathcal{X}$ de valores desconocidos en $\mathcal{Z}$.

Se introduce un modelo de densidad q en $\mathcal{Z}$ y se utiliza g para mapear en $\mathcal{X}$.

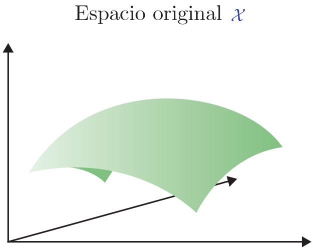


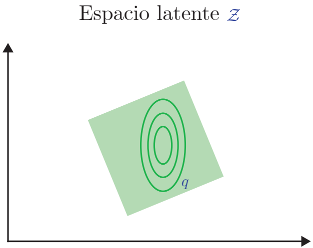


Generar nuevos datos

Además de generar valores intermedios entre dos datos existentes, también podemos utilizar el decoder para obtener la representación en $\mathcal{X}$ de valores desconocidos en $\mathcal{Z}$.

Se introduce un modelo de densidad q en $\mathcal{Z}$ y se utiliza g para mapear en $\mathcal{X}$.





Generar nuevos datos

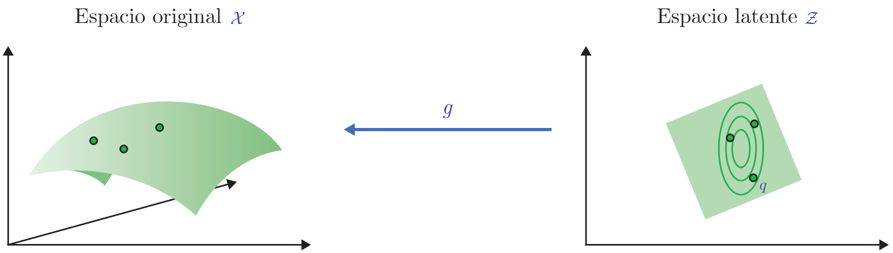
Además de generar valores intermedios entre dos datos existentes, también podemos utilizar el decoder para obtener la representación en $\mathcal{X}$ de valores desconocidos en $\mathcal{Z}$.

Se introduce un modelo de densidad q en $\mathcal{Z}$ y se utiliza g para mapear en $\mathcal{X}$.

Generar nuevos datos

Además de generar valores intermedios entre dos datos existentes, también podemos utilizar el decoder para obtener la representación en $\mathcal{X}$ de valores desconocidos en $\mathcal{Z}$.

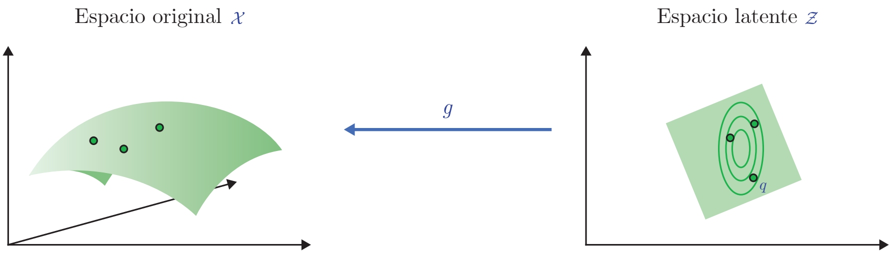
Se introduce un modelo de densidad q en $\mathcal{Z}$ y se utiliza g para mapear en $\mathcal{X}$.



Generar nuevos datos

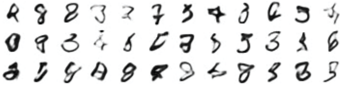
Además de generar valores intermedios entre dos datos existentes, también podemos utilizar el decoder para obtener la representación en $\mathcal{X}$ de valores desconocidos en $\mathcal{Z}$.

Se introduce un modelo de densidad q en $\mathcal{Z}$ y se utiliza g para mapear en $\mathcal{X}$.

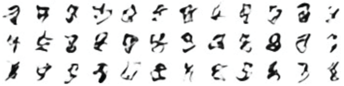


Por ejemplo, si utilizamos una distribución gaussiana $q(\mathbf{z}) = \mathcal{N}(\hat{\mu}, \hat{\Sigma})$, donde $\hat{\mu}$ y $\hat{\Sigma}$ se estiman a partir de los datos de entrenamiento:

Generación de datos con auto-encoder ($d = 16$).



Generación de datos con auto-encoder ($d = 32$).



Los resultados son malos porque la distribución seleccionada es **simple e inadecuada**.

Auto-encoders y variational auto-encoders

Denoising Auto-encoders

Denoising Auto-encoders

Eliminar ruido

Además de la ya mencionada reducción de dimensión, los auto-encoders también son capaces de **restaurar datos dañados o con ruido**.

En este caso particular podemos ignorar la estructura encoder/decoder, lo que nos dejaría con:

$$h : \mathcal{X} \to \mathcal{X}$$

Esta expresión hace referencia a un **denoising auto-encoder**.

El objetivo es optimizar h de tal forma que, cualquier perturbación $\tilde{\mathbf{x}}$ de la señal $\mathbf{x}$ sea restaurada a $\mathbf{x}$ de nuevo, por tanto

$$h(\tilde{\mathbf{x}}) \approx \mathbf{x}.$$

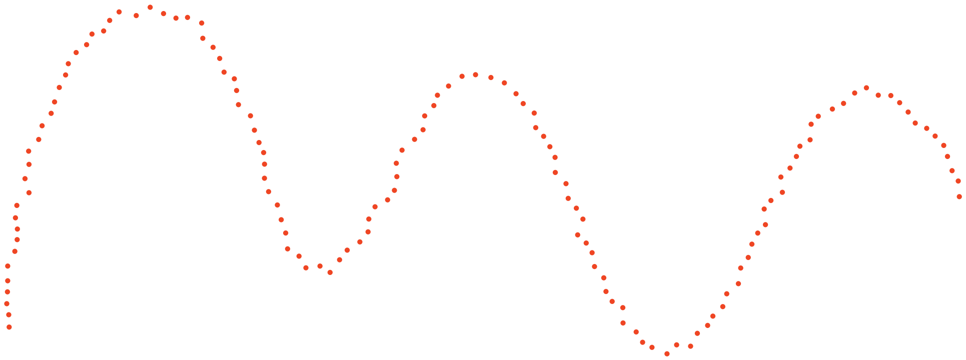
Podemos ilustrar este comportamiento con datos en 2D, añadiendo ruido Gaussiano y utilizando la MSE:

$$\hat{\theta} = \underset{\theta}{argmin} \frac{1}{N} \sum_{n=1}^{N} \|x_n - h(x_n + \epsilon_n; \theta)\|^2$$

donde x_n son los datos originales y ϵ_n el ruido Gaussiano.

Denoising Auto-encoders

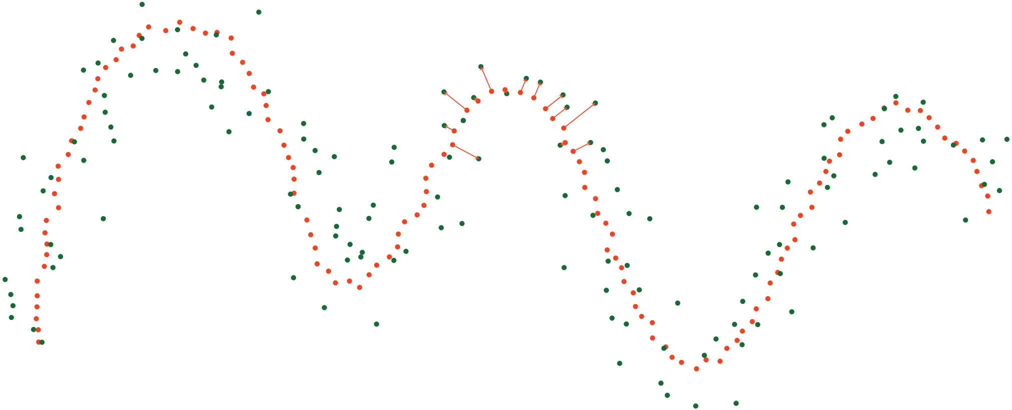
Ejemplo 2D



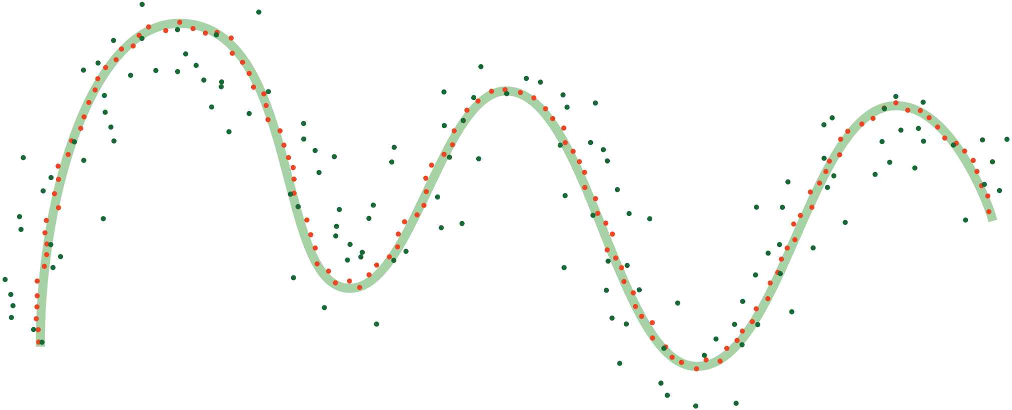
Conjunto de datos original **x**.

Denoising Auto-encoders

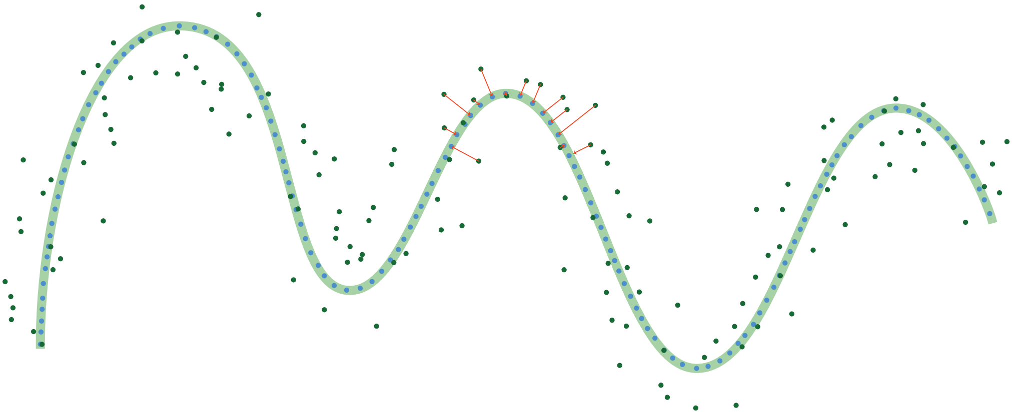
Ejemplo 2D



Datos con ruido adicional $\tilde{\mathbf{x}} = x + \epsilon$.



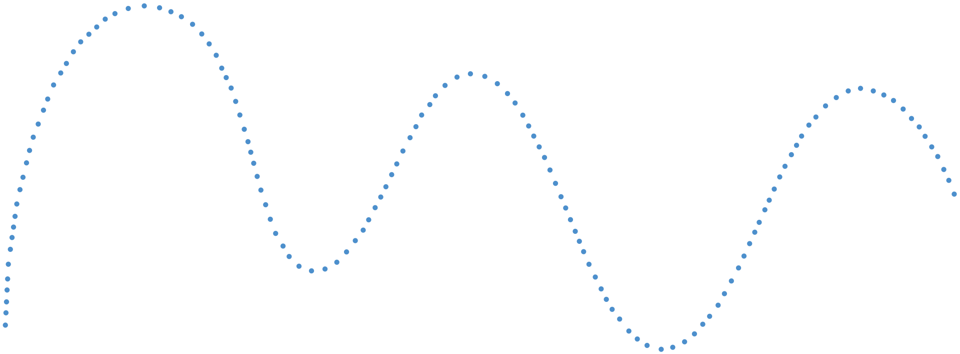
Distribución aprendida por el denoising auto-encoder.



Resultado $h(\tilde{\mathbf{x}})$ del decoder.

Denoising Auto-encoders

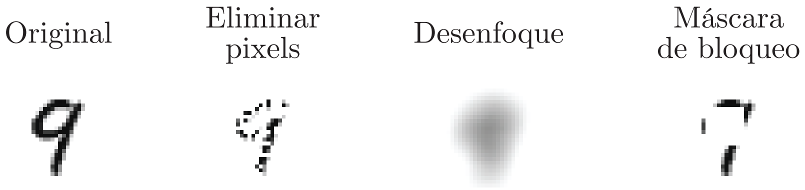
Ejemplo 2D



Resultado $h(\tilde{\mathbf{x}})$ del decoder.

Podemos hacer lo mismo con datos más complejos, por ejemplo el conjunto MNIST.

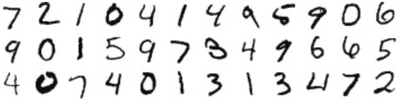
Cuando trabajamos con imágenes, los tipos de *ruido* que podemos aplicar son más variados.



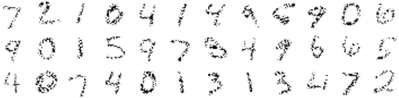
Denoising Auto-encoders

Ejemplo MNIST

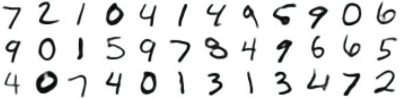
Datos originales.



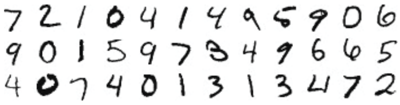
Datos con el 50% de los pixels eliminados.



Datos reconstruidos.

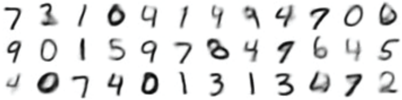


Datos originales.



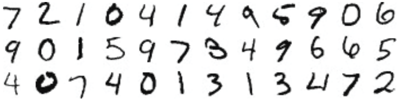
Datos con el 90% de los pixels eliminados.

Datos reconstruidos.



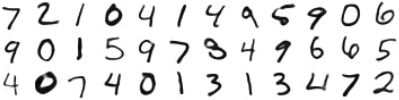
Denoising Auto-encoders
Ejemplo MNIST

Datos originales.



Datos con desenfoque ($\sigma = 4$).

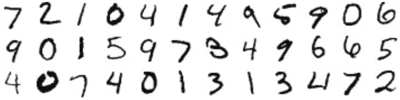
Datos reconstruidos.



Denoising Auto-encoders

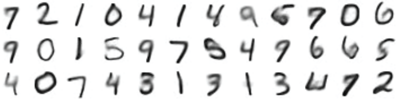
Ejemplo MNIST

Datos originales.



Datos con máscara de bloqueo de tamaño 16×16 pixels.

Datos reconstruidos.



Multiples soluciones

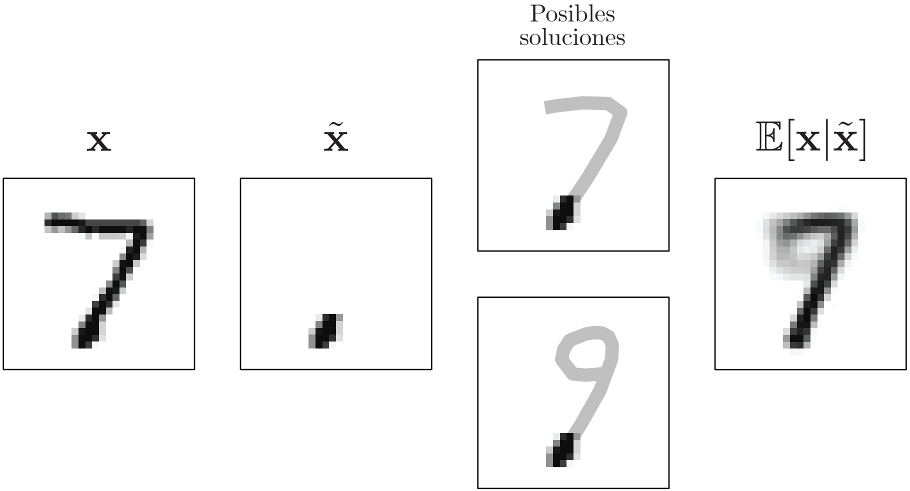
La distribución previa $p(\mathbf{x}|\tilde{\mathbf{x}})$ (probabilidad de obtener $\mathbf{x}$ dado $\tilde{\mathbf{x}}$), **puede ser multimodal**, es decir, puede tener **múltiples soluciones posibles**.

Si entrenamos un autoencoder utilizando MSE entonces la mejor reconstrucción que podemos esperar es el "promedio" de las posibles soluciones

$$h(\tilde{\mathbf{x}}) = \mathbb{E}[\mathbf{x}|\tilde{\mathbf{x}}],$$

lo que es improbable que coincida con $p(\mathbf{x}|\tilde{\mathbf{x}})$.

Denoising Auto-encoders
Ejemplo MNIST



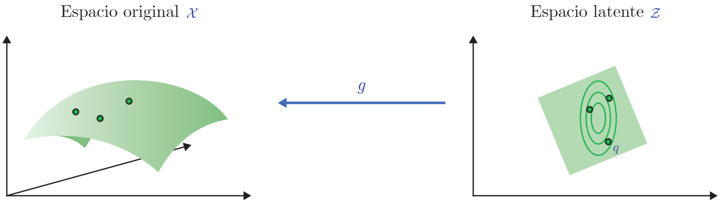
Auto-encoders y variational auto-encoders

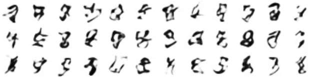
Inferencia variacional

Inferencia variacional

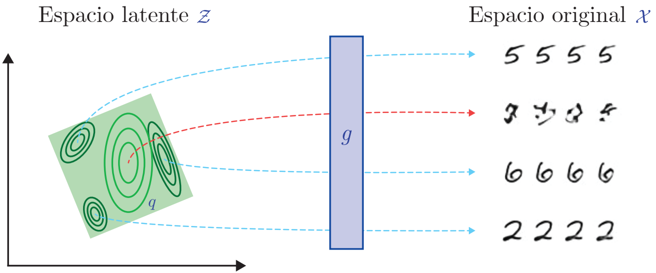
Motivación

Como veíamos anteriormente (página 38), si intentamos **generar nuevos datos** en $\mathcal{X}$ a partir de unos puntos en $\mathcal{Z}$ asumiendo que siguen una distribución q, los resultados no son prometedores.





Este problema proviene de la distribución q seleccionada. Esta es **muy simple e inadecuada**, los datos en $\mathcal{Z}$ se comportan de manera diferente.



El espacio latente tiene discontinuidades y al tomar una muestra en estas zonas intermedias el decoder produce salidas poco realistas.

Imponer distribución

No conocemos la distribución en $\mathcal{Z}$ para poder generar ejemplos con sentido, pero podemos **forzar al auto-encoder para que aprenda una específica** durante el entrenamiento.

En vez de entrenar el auto-encoder y adivinar la distribución de $\mathcal{Z}$ vamos a:

1. Imponer una distribución para $\mathcal{Z}$.
2. Entrenar el decoder g de tal forma que $g(\mathcal{Z})$ coincida con los datos de entrenamiento.

Inferencia variacional

Consideremos un modelo que relaciona un conjunto de variables observables $\mathbf{x} \in \mathcal{X}$ con un conjunto de variables latentes $\mathbf{z} \in \mathcal{Z}$.

Asumimos que podemos generar datos en $\mathcal{X}$ siguiendo un proceso aleatorio controlado por las variables latentes $\mathbf{z}$.

Inferencia variacional

Consideremos un modelo que relaciona un conjunto de variables observables $\mathbf{x} \in \mathcal{X}$ con un conjunto de variables latentes $\mathbf{z} \in \mathcal{Z}$.

$$\mathbf{z} \longrightarrow \mathbf{x}$$

Asumimos que podemos generar datos en $\mathcal{X}$ siguiendo un proceso aleatorio controlado por las variables latentes $\mathbf{z}$.

$$\mathbf{z} \qquad \qquad \mathbf{x}$$

$$\begin{array}{r} \textbf{Barba} = \\ \textbf{Rubio} = \end{array} \begin{bmatrix} \mathbf{0.0} \\ \mathbf{0.0} \end{bmatrix} \longrightarrow$$

Inferencia variacional

Consideremos un modelo que relaciona un conjunto de variables observables $\mathbf{x} \in \mathcal{X}$ con un conjunto de variables latentes $\mathbf{z} \in \mathcal{Z}$.

$$\mathbf{z} \longrightarrow \mathbf{x}$$

Asumimos que podemos generar datos en $\mathcal{X}$ siguiendo un proceso aleatorio controlado por las variables latentes $\mathbf{z}$.

Inferencia variacional

Consideremos un modelo que relaciona un conjunto de variables observables $\mathbf{x} \in \mathcal{X}$ con un conjunto de variables latentes $\mathbf{z} \in \mathcal{Z}$.

Asumimos que podemos generar datos en $\mathcal{X}$ siguiendo un proceso aleatorio controlado por las variables latentes $\mathbf{z}$.

Inferencia variacional

Consideremos un modelo que relaciona un conjunto de variables observables $\mathbf{x} \in \mathcal{X}$ con un conjunto de variables latentes $\mathbf{z} \in \mathcal{Z}$.

No conocemos $\mathbf{z}$ pero si $\mathbf{x}$, por lo que tratamos de resolver (probabilidad a posteriori):

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}.$$

Obtener $p(\mathbf{x})$ es un **problema intratable** especialmente en modelos complejos dado que requiere una integrar sobre todas las posibles combinaciones de variables latentes

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}.$$

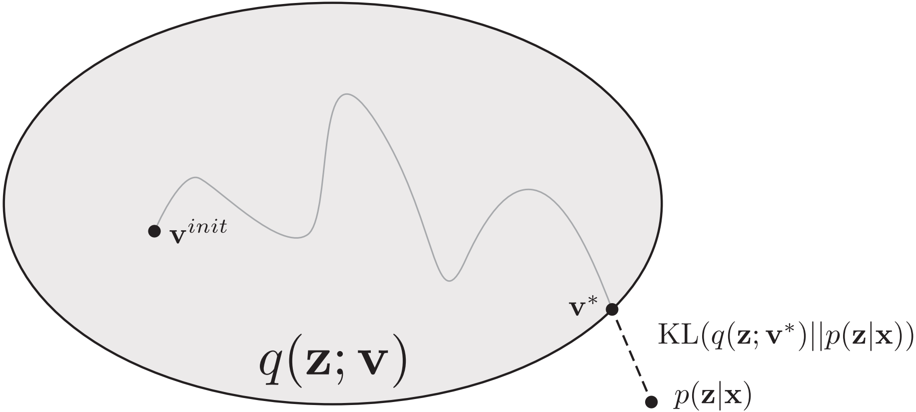
Inferencia variacional

Como no podemos resolver $p(\mathbf{z}|\mathbf{x})$ vamos a intentar aproximarlo mediante un problema de optimización.

- Consideremos una familia de distribuciones $q_v(\mathbf{z}|\mathbf{x})$ que aproximan la probabilidad a posteriori $p(\mathbf{z}|\mathbf{x})$ donde v es el índice de la familia de distribuciones.
- Los parámetros v se aprenden para minimizar la *Kullback Leibeler (KL) divergence* entre la aproximación $q_v(\mathbf{z}|\mathbf{x})$ y la probabilidad a posteriori $p(\mathbf{z}|\mathbf{x})$.
- Queremos resolver $\arg\min_\nu \mathrm{KL}(q_\nu(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$.

En otras palabras, queremos minimizar la diferencia entre la distribución variacional $q_v(\mathbf{z}|\mathbf{x})$ y la verdadera distribución posterior $p(\mathbf{z}|\mathbf{x})$.

Gráficamente:



Auto-encoders y variational auto-encoders

Variational Auto-encoders

Variational Auto-encoders

Un **variational auto-encoder** es un *deep latent variable model* donde:

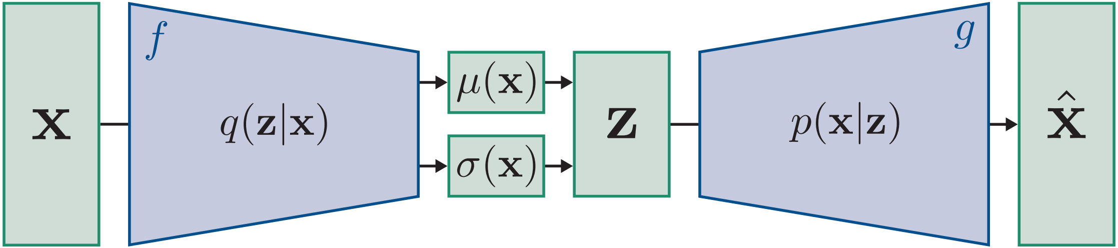
- La probabilidad a posteriori $q_\phi(\mathbf{z}|\mathbf{x})$ aproximada se parametriza con una *red de inferencia* (o encoder) que toma como entrada $\mathbf{x}$ y predice los parámetros para la probabilidad aproximada a posteriori, es decir,

$$\mu, \sigma = \mathrm{NN}_\phi(\mathbf{x})$$
$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \mu, \sigma^2 \mathbf{I})$$

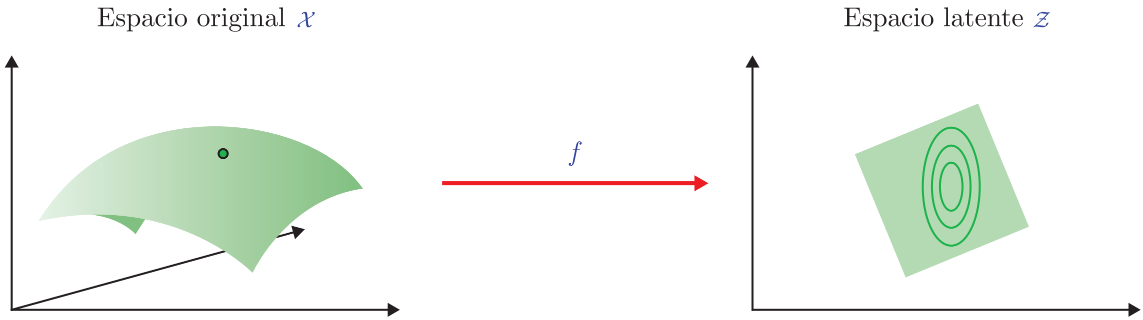
- La probabilidad $p_\theta(\mathbf{x}|\mathbf{z})$ se parametriza con una *red generativa* (o decoder) que toma como entrada $\mathbf{z}$ y predice los valores de la distribución de datos, es decir,

$$\mu, \sigma = \mathrm{NN}_\theta(\mathbf{z})$$
$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mu, \sigma^2 \mathbf{I})$$

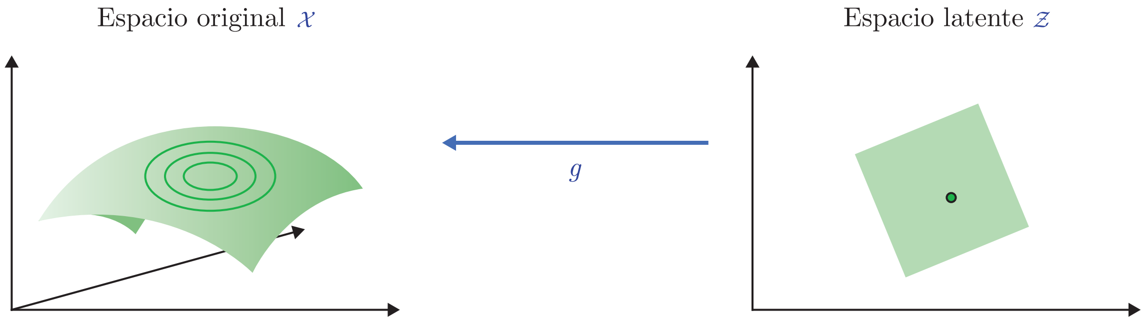
Variational Auto-encoders



Variational Auto-encoders



Variational Auto-encoders



Consideremos como datos **d** el conjunto MNIST:

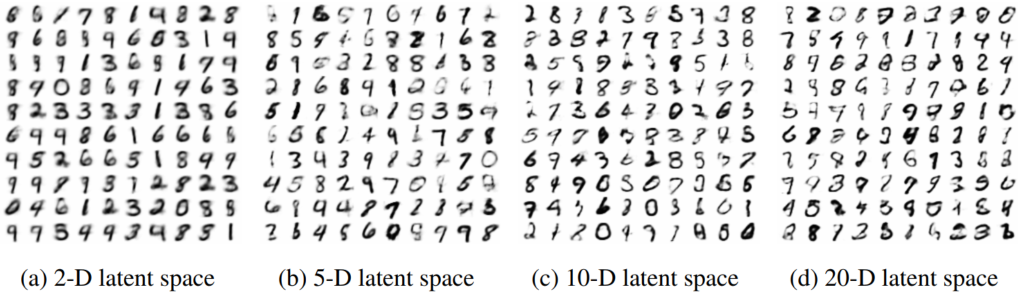
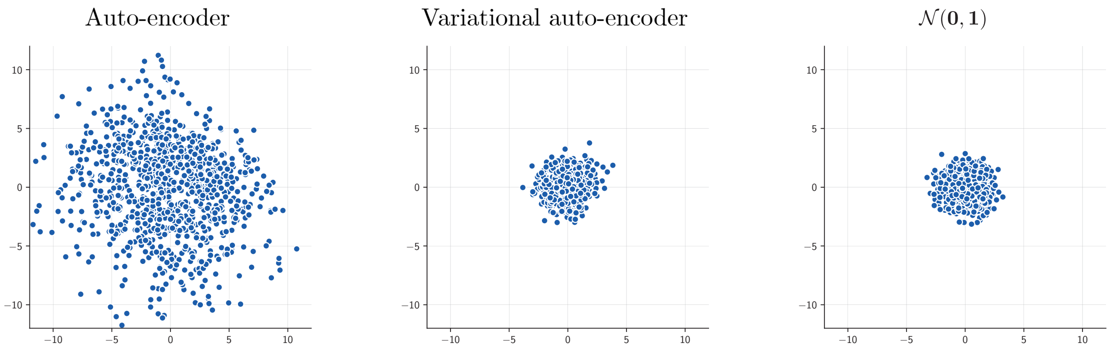
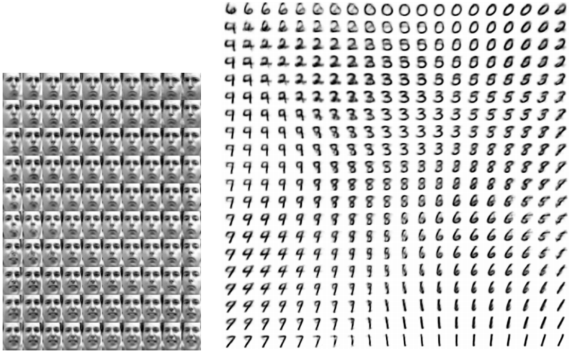


Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

(Kingma and Welling, 2013)


Auto-encoder
Variational auto-encoder
$\mathcal{N}(\mathbf{0}, \mathbf{1})$
10
5
0
−5
−10
−10
−5
0
5
10



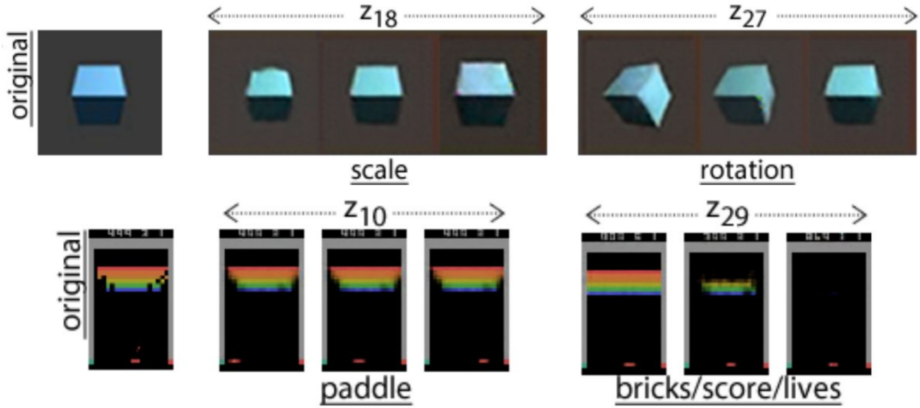
(a) Learned Frey Face manifold

(b) Learned MNIST manifold

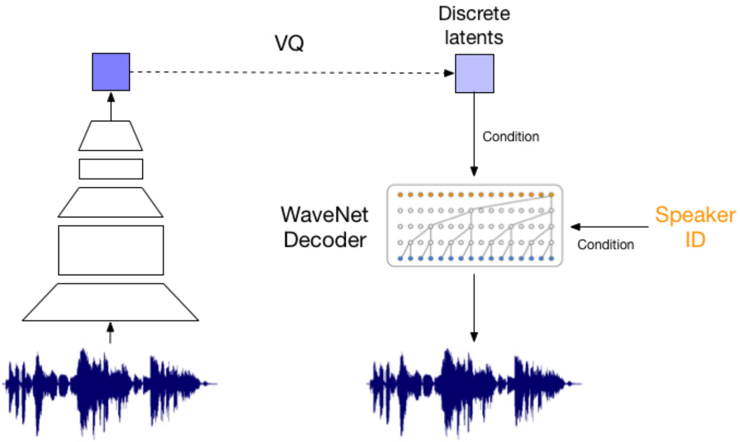
Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

(Kingma and Welling, 2013)

Compresión jerárquica de imágenes y otros datos, por ejemplo, en sistemas de videoconferencia (Gregor et al, 2016).



Comprender los factores de variación y las invarianzas (Higgins et al, 2017).



Transferencia de estilo de voz (van den Oord et al, 2017).

Auto-encoders y variational auto-encoders

Referencias

Referencias

1. **Lecture 11: Auto-encoders and variational auto-encoders**
2. **Deep Learning Course**
3. **Variational autoencoders**