Deep Learning en Imágenes con GANs y Modelos de Difusión - Parte I

Prof. Peter Montalvo

Agenda

- Introducción
- Autoencoders y sus tipos
- Autoencoders Variacionales

Agenda

- Introducción
- Autoencoders y sus tipos
- Autoencoders Variacionales

Recordando…

Si tenemos una imagen de 256x256 con 1 canal de 8 bits, ¿cuántas posibles imágenes tenemos?

A) 65536

B) 16777216

Recordando…

Si tenemos una imagen de 256x256 con 1 canal de 8 bits, ¿cuántas posibles imágenes tenemos?

256x256x256 = 16777216

A) 65536

B) 16777216

Recordando…

¿Y si la imagen tuviera 3 canales de 8 bits?

A) 1677721600

B) 1.0995116e+12

Recordando…

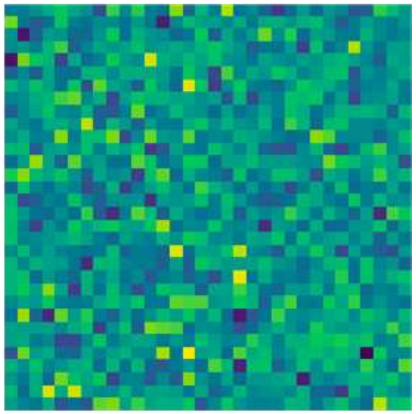
¿Y si la imagen tuviera 3 canales de 8 bits?

256x256x256x256x256 = 16777216

A) 1677721600

B) 1.0995116e+12

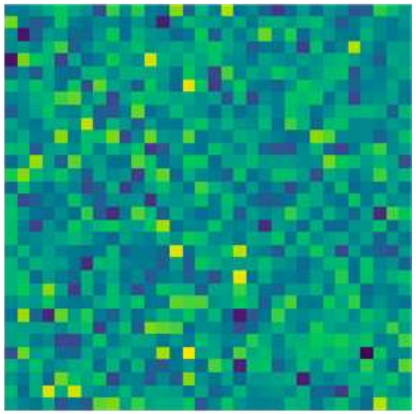
En resumen



En tan solo 256x256 píxeles podemos tener muchísimas posibilidades de imágenes.

Incluso podríamos tardar vidas en recorrer todas las posibles imágenes si viéramos cada una por un lapso de 1 segundo.

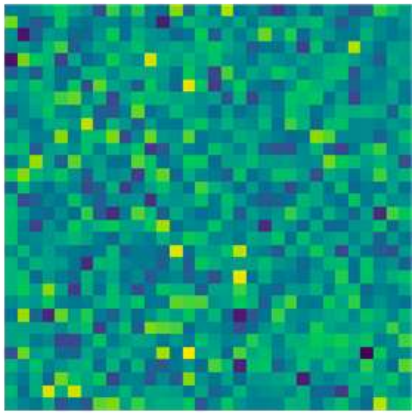
En resumen



En tan solo 256x256 píxeles podemos tener muchísimas posibilidades de imágenes.

Aunque muchas de las imágenes será ruido…

En resumen



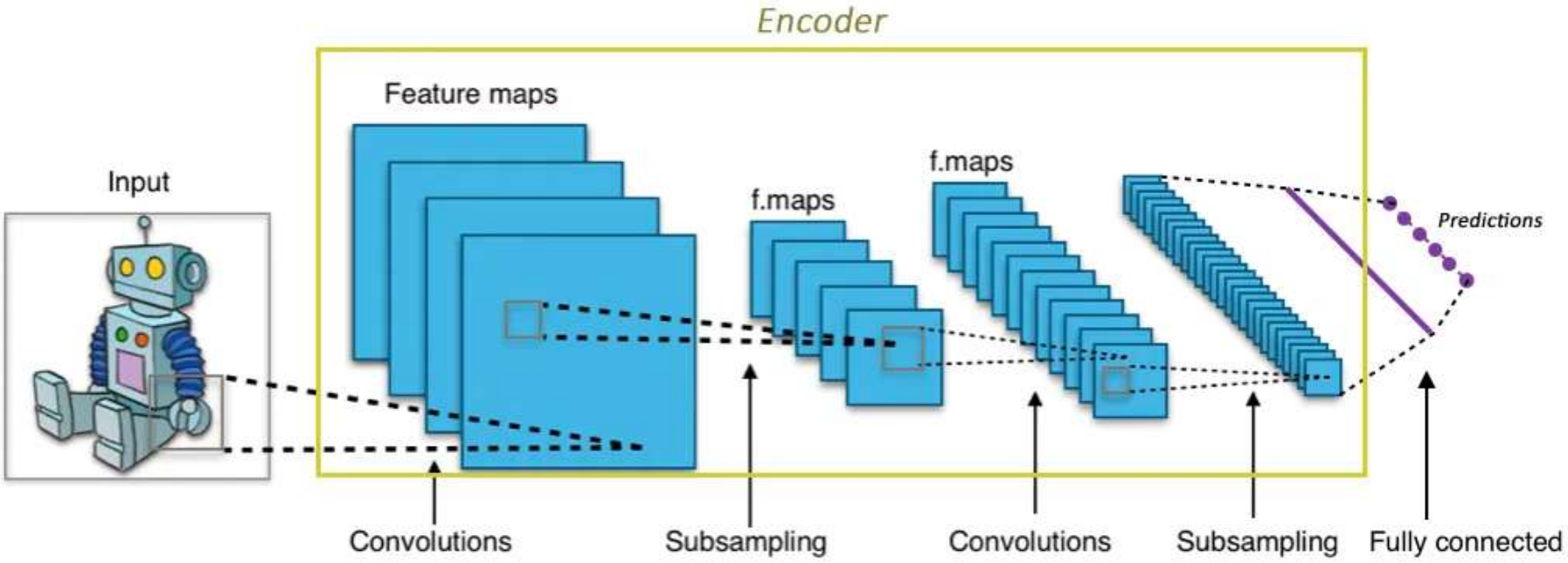
Ejemplo:
https://onlinetools.com/image/generate-random-image

En resumen

En tan solo 256x256 píxeles podemos tener muchísimas posibilidades de imágenes.

En algunos casos nos pueden dar sentido **acorde a nuestra percepción**

Recordando las arquitecturas de las CNN



Agenda

- Introducción
- Autoencoders y sus tipos
- Autoencoders Variacionales

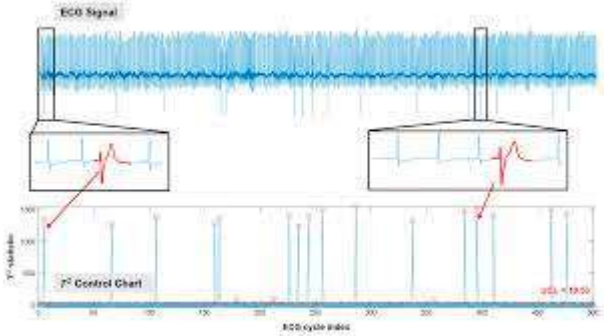
Autoencoders

Los autoencoders son una técnica de aprendizaje no supervisada en la que se aprovechan las redes neuronales para la tarea de aprendizaje de representación.

Autoencoders

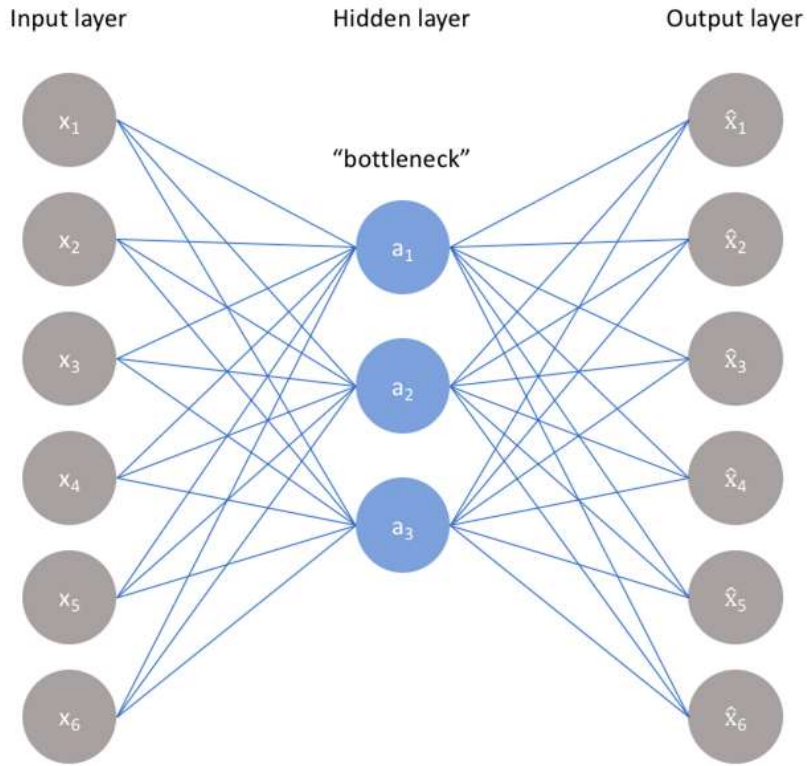
Las aplicaciones usuales son:

- Detección de anomalías (ECG) - Anomaly detection
- Eliminación de ruido (daños) Data denoising (ex. images, audio)
- Reconstrucción de Imágenes - Image inpainting
- Encontrar información - Information retrieval



Autoencoders

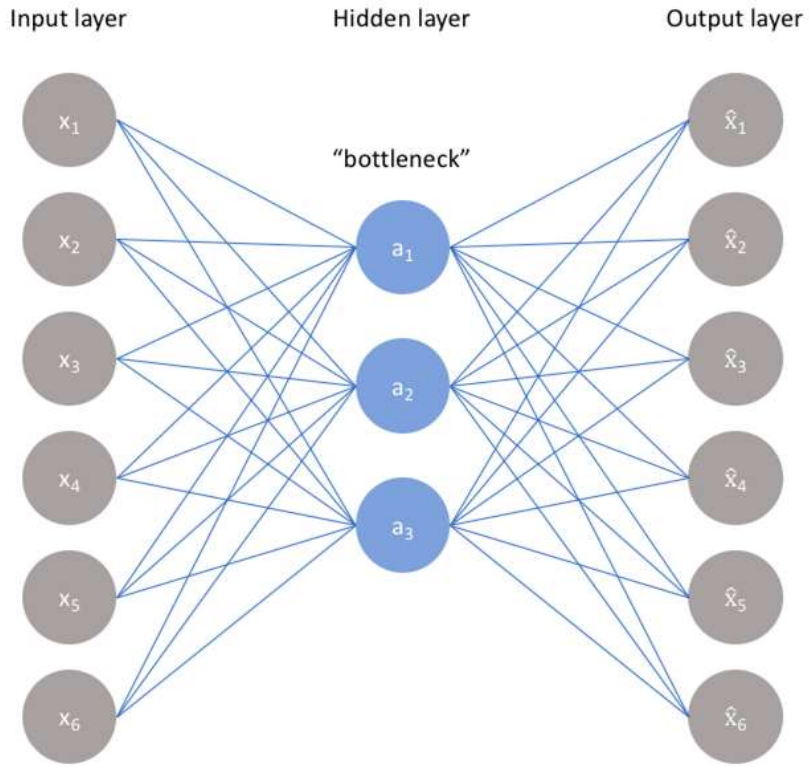
Colocar un cuello de botella en la red obliga a un comprimir la representación de la imagen (o conocimiento).



Autoencoders

Colocar un cuello de botella en la red obliga a un comprimir la representación de la imagen (o conocimiento).

Si las características de entrada fueran cada independientes unos de otros, esta compresión y posterior reconstrucción sería una tarea muy difícil.

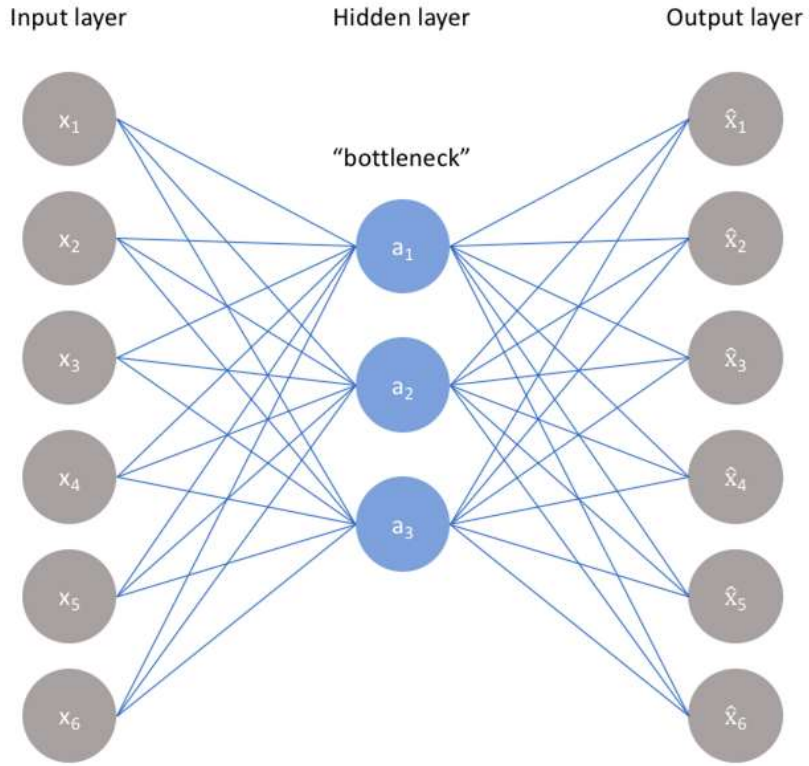


Autoencoders

Colocar un cuello de botella en la red obliga a un comprimir la representación de la imagen (o conocimiento).

Si las características de entrada fueran cada independientes unos de otros, esta compresión y posterior reconstrucción sería una tarea muy difícil.

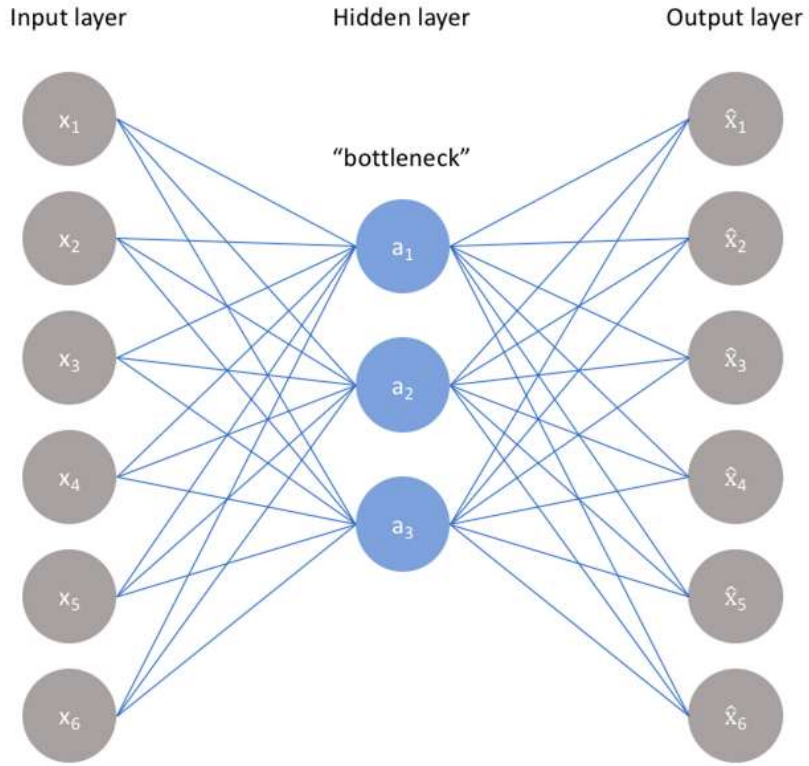
Sin embargo, si existe algún tipo de estructura en los datos (es decir, correlaciones entre datos), la red puede aprender y, en consecuencia, aprovechar el cuello de botella de la red.



Autoencoders - Propiedades

- Sensible a las entradas lo suficiente como para construir una reconstrucción con precisión.

- Lo suficientemente insensible a las entradas como para que el modelo no simplemente memorice o sobreajuste a los datos de entrenamiento.



Autoencoders - Función de Pérdida

Esta compensación obliga al modelo a mantener sólo las variaciones en los datos necesarios para reconstruir la entrada sin aferrarse a redundancias dentro de la entrada.

Autoencoders - Función de Pérdida

Esta compensación obliga al modelo a mantener sólo las variaciones en los datos necesarios para reconstruir la entrada sin aferrarse a redundancias dentro de la entrada.

Esto implica construir una función de pérdida donde un término restringe al modelo a ser sensible a las entradas (es decir, pérdida de reconstrucción) y un segundo término restringe la memorización/sobreajuste (regularizador).

Autoencoders - Función de Pérdida

Esta compensación obliga al modelo a mantener sólo las variaciones en los datos necesarios para reconstruir la entrada sin aferrarse a redundancias dentro de la entrada.

Esto implica construir una función de pérdida donde un término restringe al modelo a ser sensible a las entradas (es decir, pérdida de reconstrucción) y un segundo término restringe la memorización/sobreajuste (regularizador).

$$\mathcal{L}\left(x, \hat{x}\right) + regularizer$$

Tipos de autoencoders

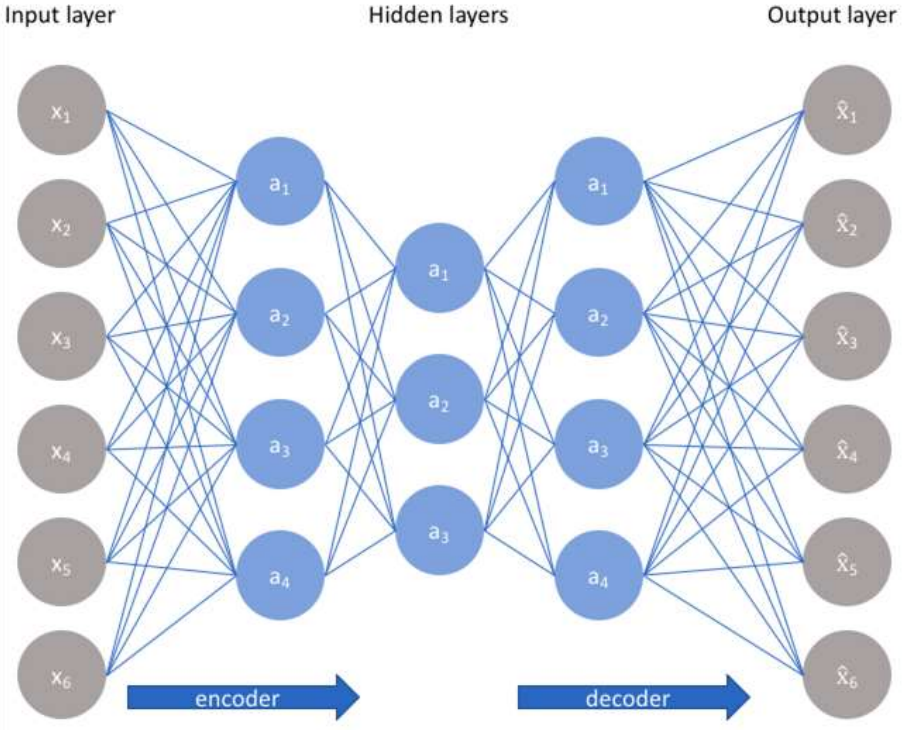
- Incompleto (Undercomplete)
- Dispersos (Sparse)
- Sin ruido (Denoising)

Autoencoders - Incompleto

La arquitectura más simple para construir un autoencoder es restringir la cantidad de nodos presentes en el oculto (n capas), lo que limita la cantidad de información que puede fluir.

Al penalizar la red según el error de reconstrucción, nuestro modelo puede aprender los atributos más importantes de la imagen y cómo reconstruir mejor la imagen original a partir de un estado "codificado".

Idealmente, esta aquitectura aprenderá y describirá los atributos latentes de la imagen.

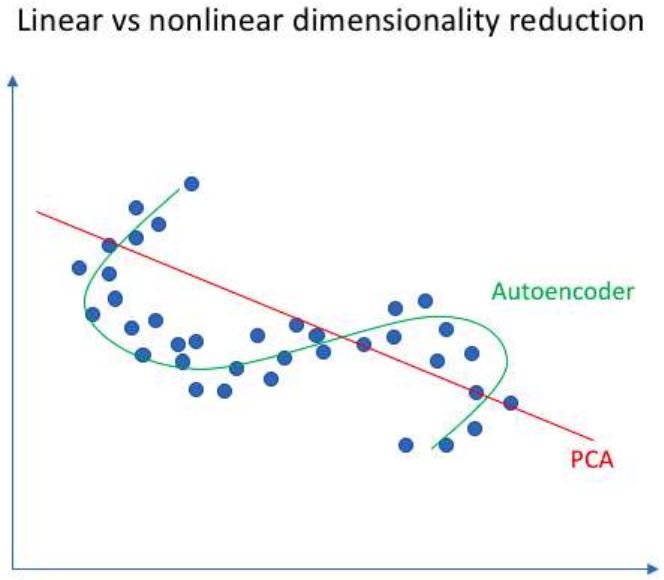


Autoencoders - Incompleto

Debido a que las redes neuronales son capaces de aprender relaciones no lineales, esto puede considerarse como una generalización (no lineal) más compleja que PCA .

Mientras que PCA intenta descubrir un hiperplano de dimensiones inferiores que describa los datos originales, los autoencoders son capaces de aprender variedades no lineales (una variedad se define en términos simples como una variedad continua, no lineal).

La diferencia entre estos dos enfoques se visualiza a continuación.



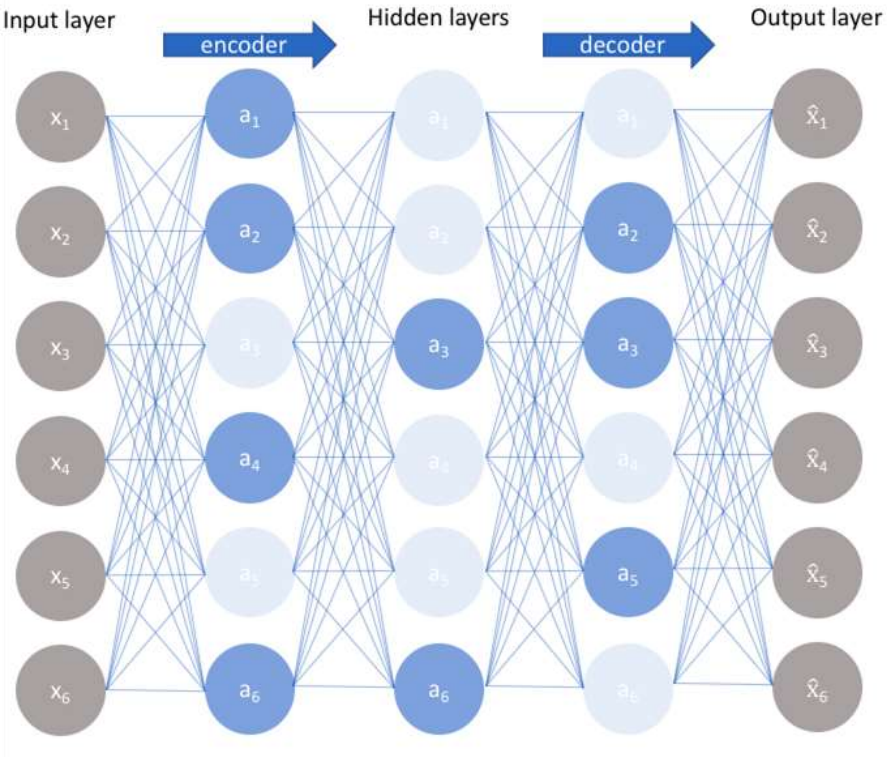
Autoencoders - Disperso

Ofrecen un método alternativo sin requerir una reducción en la cantidad de nodos en nuestras capas ocultas.

Utilizan una función de pérdida de manera que penaliza las activaciones dentro de una capa.

Se solicita al modelo a aprender una codificación y decodificación que solo se basa en la activación de una pequeña cantidad de neuronas.

Este es un enfoque diferente hacia la regularización, ya que normalmente regularizamos los pesos de una red, no las activaciones.



Autoencoders - Disperso

Hay dos formas principales de imponer esta restricción de escasez; ambos implican medir las activaciones de la capa oculta para cada lote de entrenamiento y agregar algún término a la función de pérdida para penalizar las activaciones excesivas.

Estos términos son:

Regularización L1

$$\mathcal{L}\left(x, \hat{x}\right) + \lambda \sum_{i} \left| a_i^{(h)} \right|$$

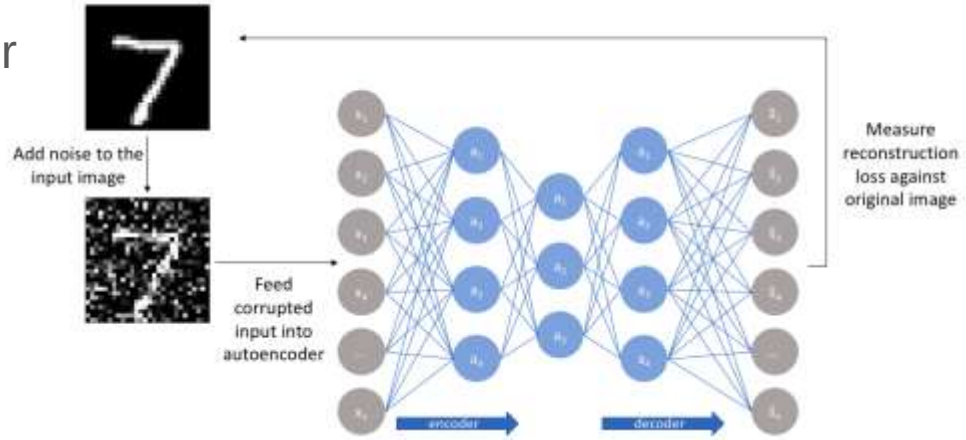
Distancia de KL

$$\mathcal{L}\left(x, \hat{x}\right) + \sum_{j} KL\left(\rho || \hat{\rho}_j\right)$$

Autoencoders - Denoising

Otro enfoque es corromper ligeramente los datos de entrada pero aún mantener los datos no dañados como nuestro objetivo.

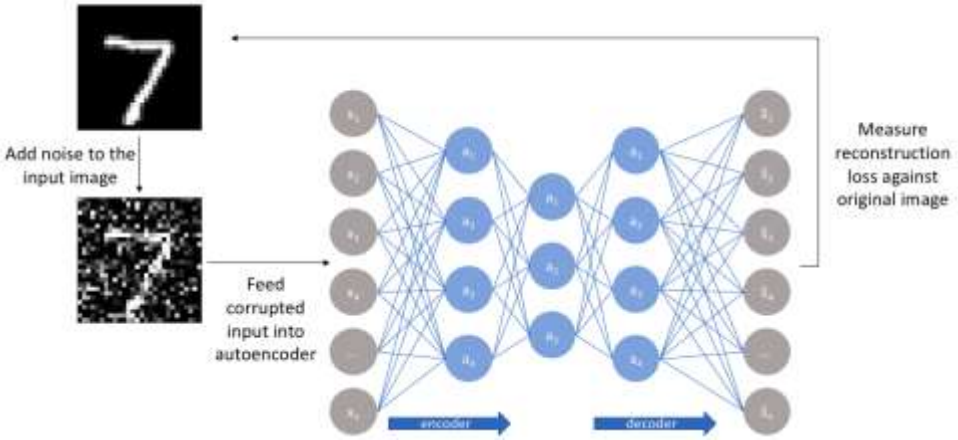
Con este enfoque, el modelo no puede simplemente desarrollar un mapeo que memorice los datos de entrenamiento porque la entrada y la salida objetivo ya no son las mismas.



Autoencoders - Denoising

Más bien, el modelo aprende un campo vectorial para mapear los datos de entrada hacia una variedad de dimensiones inferiores.

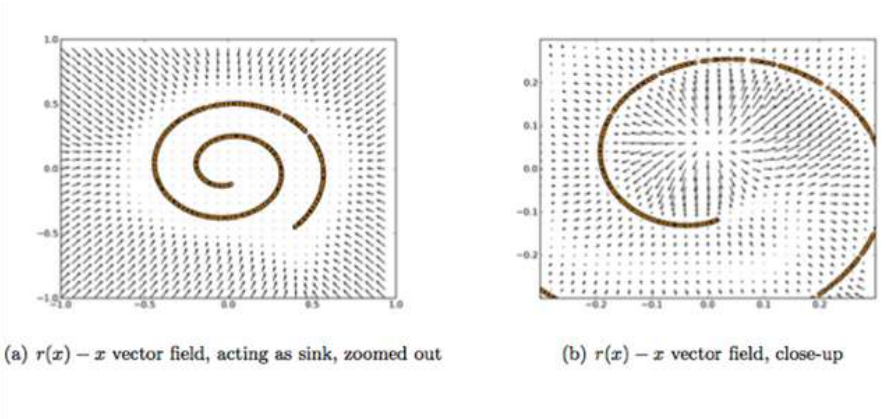
Si esta variedad describe con precisión los datos naturales, efectivamente habremos "cancelado" el ruido agregado.



Autoencoders - Denoising

La figura anterior visualiza el campo vectorial descrito comparando la reconstrucción de x con el valor original de

Los puntos amarillos representan ejemplos de entrenamiento antes de agregar ruido. El modelo ha aprendido a ajustar la entrada corrupta hacia la variedad aprendida.



(a) $r(x) - x$ vector field, acting as sink, zoomed out

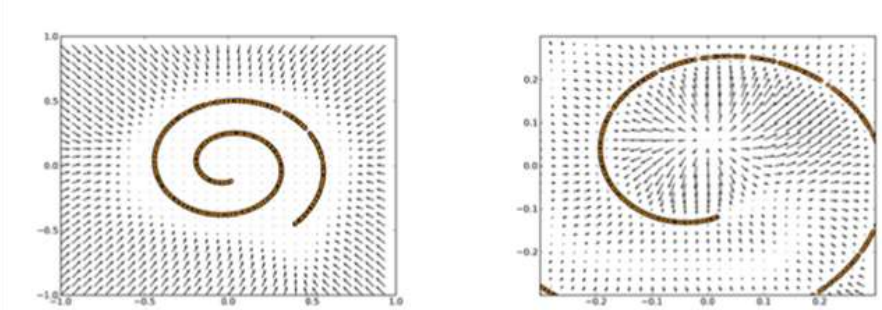
(b) $r(x) - x$ vector field, close-up

https://arxiv.org/abs/1211.4246

Autoencoders - Denoising

El campo vectorial generalmente solo se comporta bien en las regiones donde el modelo ha observado durante el entrenamiento.

En áreas alejadas de la distribución natural de los datos, el error de reconstrucción es grande y no siempre apunta en la dirección de la distribución verdadera.



(a) $r(x) - x$ vector field, acting as sink, zoomed out

(b) $r(x) - x$ vector field, close-up

https://arxiv.org/abs/1211.4246

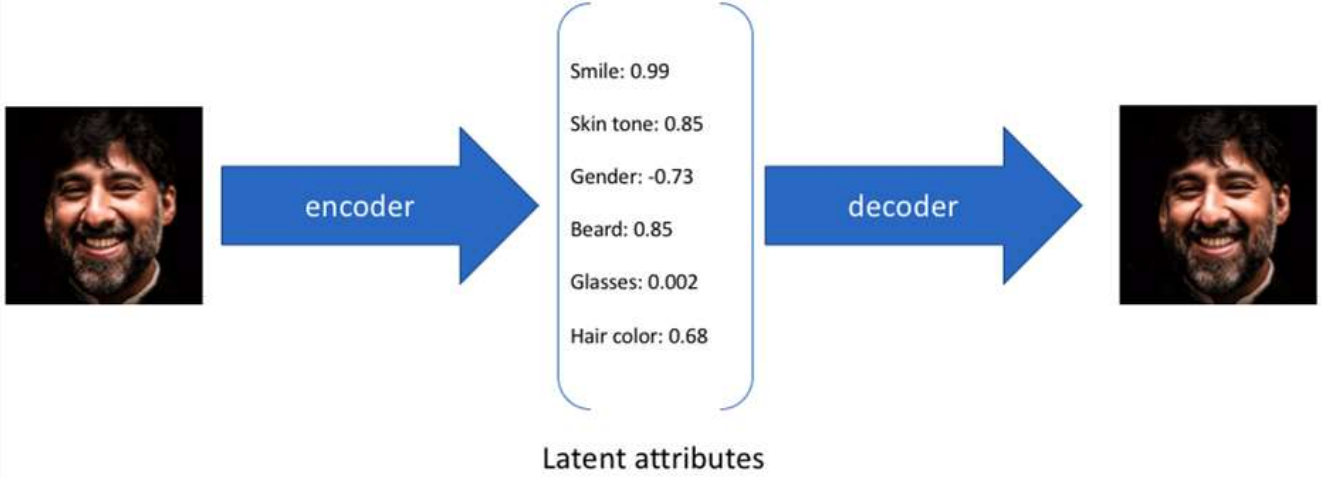
Agenda

- Introducción
- Autoencoders y sus tipos
- Autoencoders Variacionales

Autoencoders Variacionales

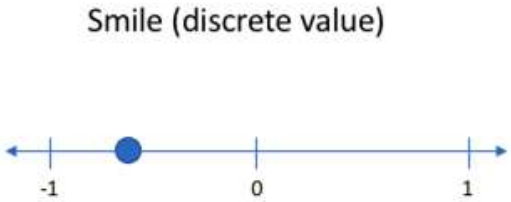
Recordemos que los datos de entrada se convierten en un vector de codificación donde cada dimensión representa algún atributo aprendido sobre los datos.

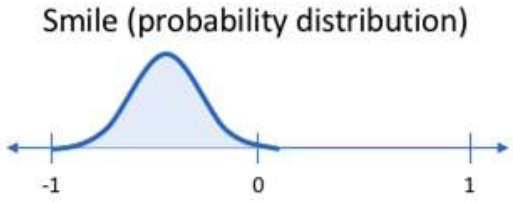
La red genera un valor único para cada dimensión de codificación. Posteriormente, la red toma estos valores e intenta recrear la entrada original.



Autoencoders Variacionales

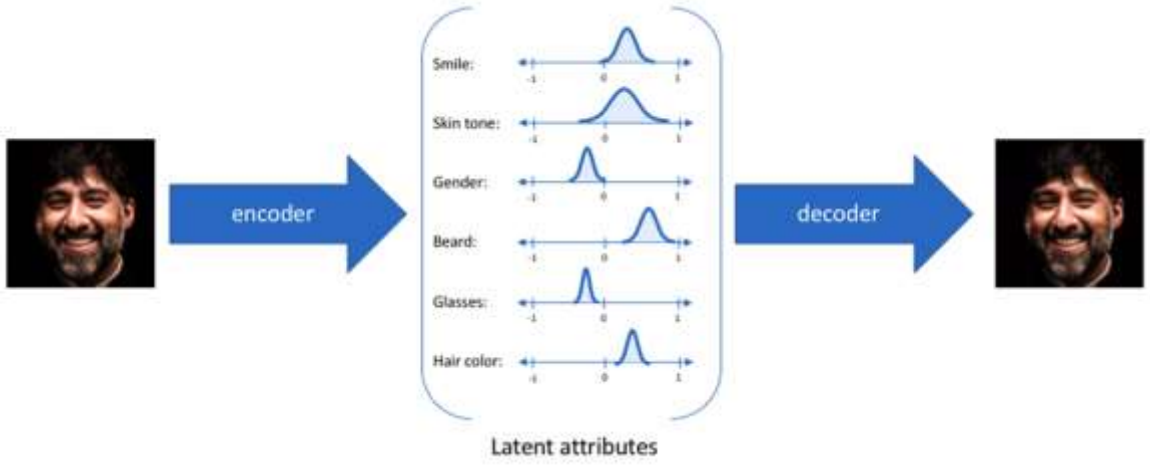
Un autoencoder variacional (VAE) proporciona una forma probabilística de describir una observación en el espacio latente. Por lo tanto, en lugar de crear un codificador que genere un valor único para describir cada atributo de estado latente, **el codificador describe una distribución de probabilidad para cada atributo latente.**





Autoencoders Variacionales

Con este enfoque, se representa cada atributo latente para una entrada determinada como una distribución de probabilidad. Al decodificar desde el estado latente, tomaremos muestras aleatorias de cada distribución de estado latente para generar un vector como entrada para nuestro modelo de decodificador.

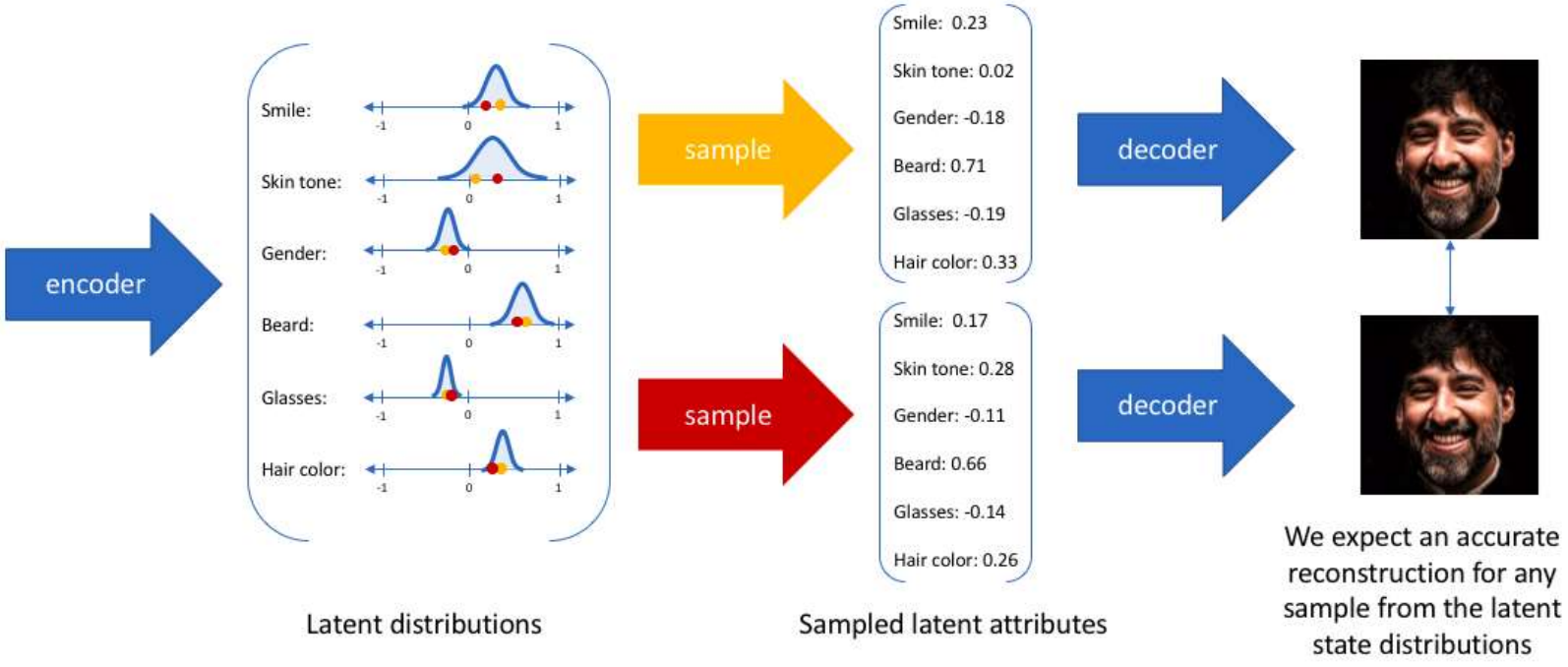


Autoencoders Variacionales - denominaciones

En el caso de los VAE, el modelo codificador a veces se denomina modelo de reconocimiento , mientras que el modelo decodificador a veces se denomina modelo generativo.

Autoencoders Variacionales

Bajo esta definición, los auto-encoders con valores de distribución similares, generarán imágenes similares.



Autoencoders - Estadística dentro del modelo…

Supongamos que existe una variable desconocida (o escondida) z tal que genera una observación x.

Autoencoders - Estadística dentro del modelo…

Como z es desconocido, solo podemos inferir sus características con…

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

Autoencoders - Estadística dentro del modelo…

Aunque bajo esta definición, p(x) es costoso computacionalmente.

Sin embargo, se puede utilizar inferencia variacional para estimar el valor.

$$p\left(x\right) = \int p\left(x|z\right)p\left(z\right)dz$$

Autoencoders - Estadística dentro del modelo…

Sea q(z|x) una distribución similara a p(z|x), podemos tomar distancias y optimizarlas para que estas sean más cercanas.

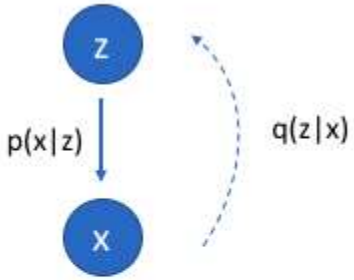
$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right)$$

Divergencia de Kullback Leibler

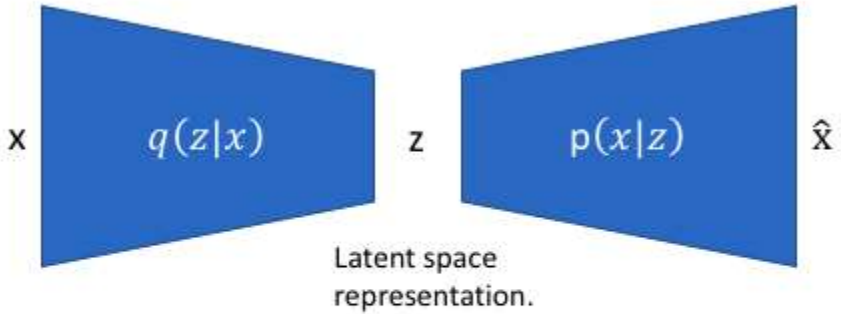
$$E_{q(z|x)}\log p\left(x|z\right) - KL\left(q\left(z|x\right)||p\left(z\right)\right)$$

reducción demostrada aquí

Autoencoders - Estadística dentro del modelo…



We'd like to use our observations to understand the hidden variable.



Latent space representation.

Neural network mapping x to z.

Neural network mapping z to x.

Autoencoders - Estadística dentro del modelo…

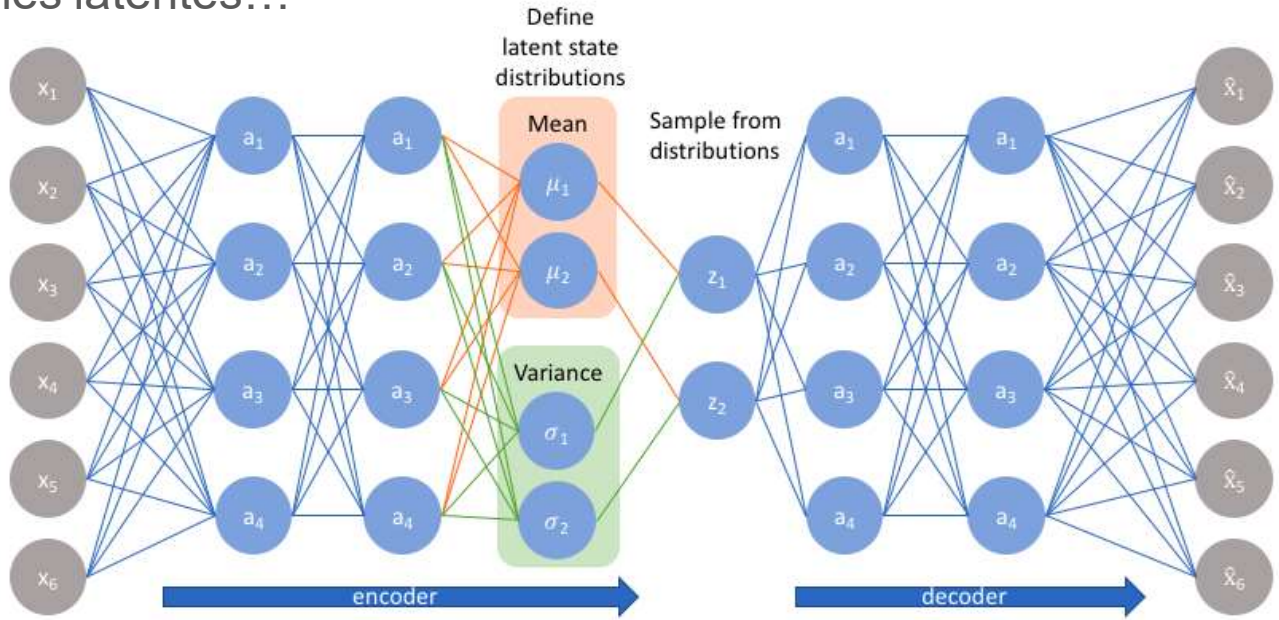
La función de pérdida para esta red tiene dos términos:

- Uno que penaliza el error de reconstrucción
- Un segundo término que fomenta a la distribución aprendida a ser similar a la distribución original.

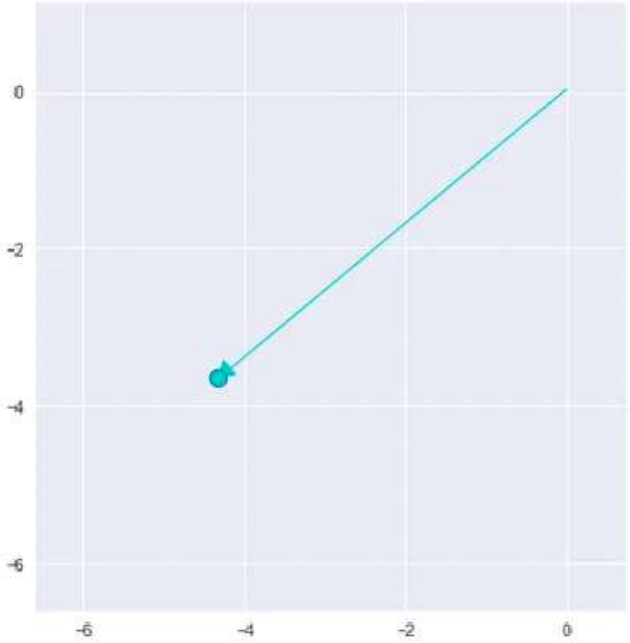
$$\mathcal{L}\left(x, \hat{x}\right) + \sum_{j} KL\left(q_j\left(z|x\right)||p\left(z\right)\right)$$

Autoencoders - Construcción del modelo

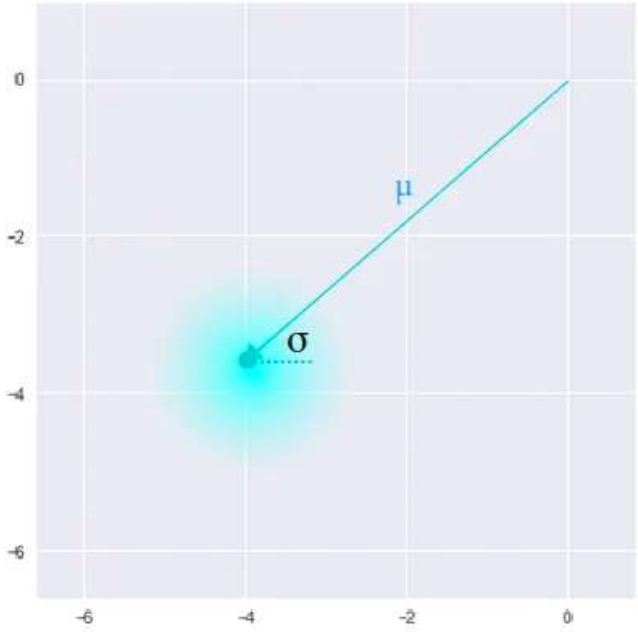
Consideremos que queremos hallar la media y varianza, por lo que se definen sus distribuciones latentes…



Autoencoders - Construcción del modelo



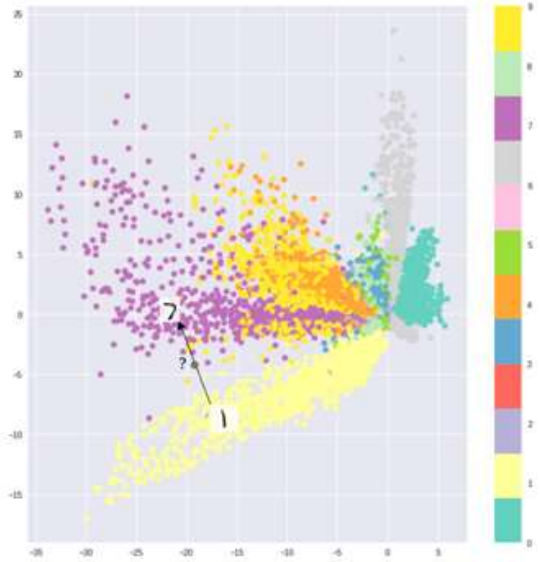
Standard Autoencoder
(direct encoding coordinates)



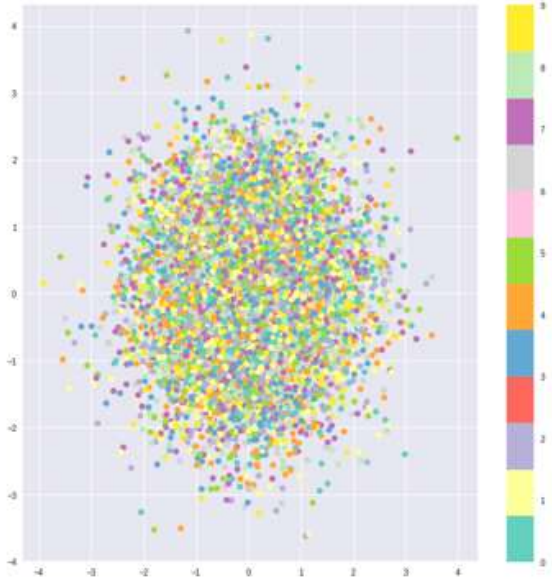
Variational Autoencoder
(μ and σ initialize a probability distribution)

Autoencoders - Construcción del modelo

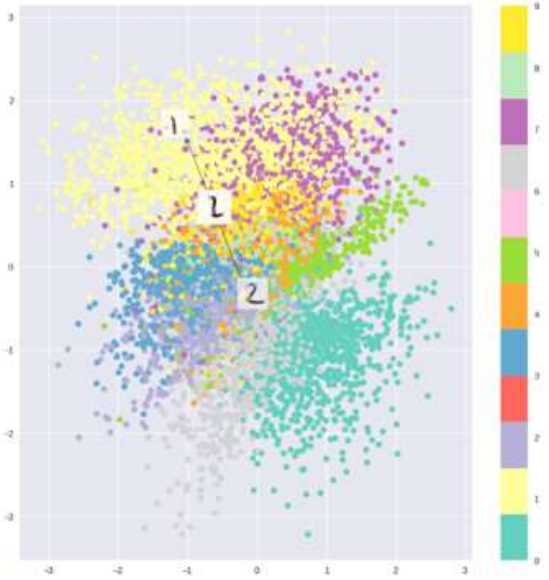
Only reconstruction loss



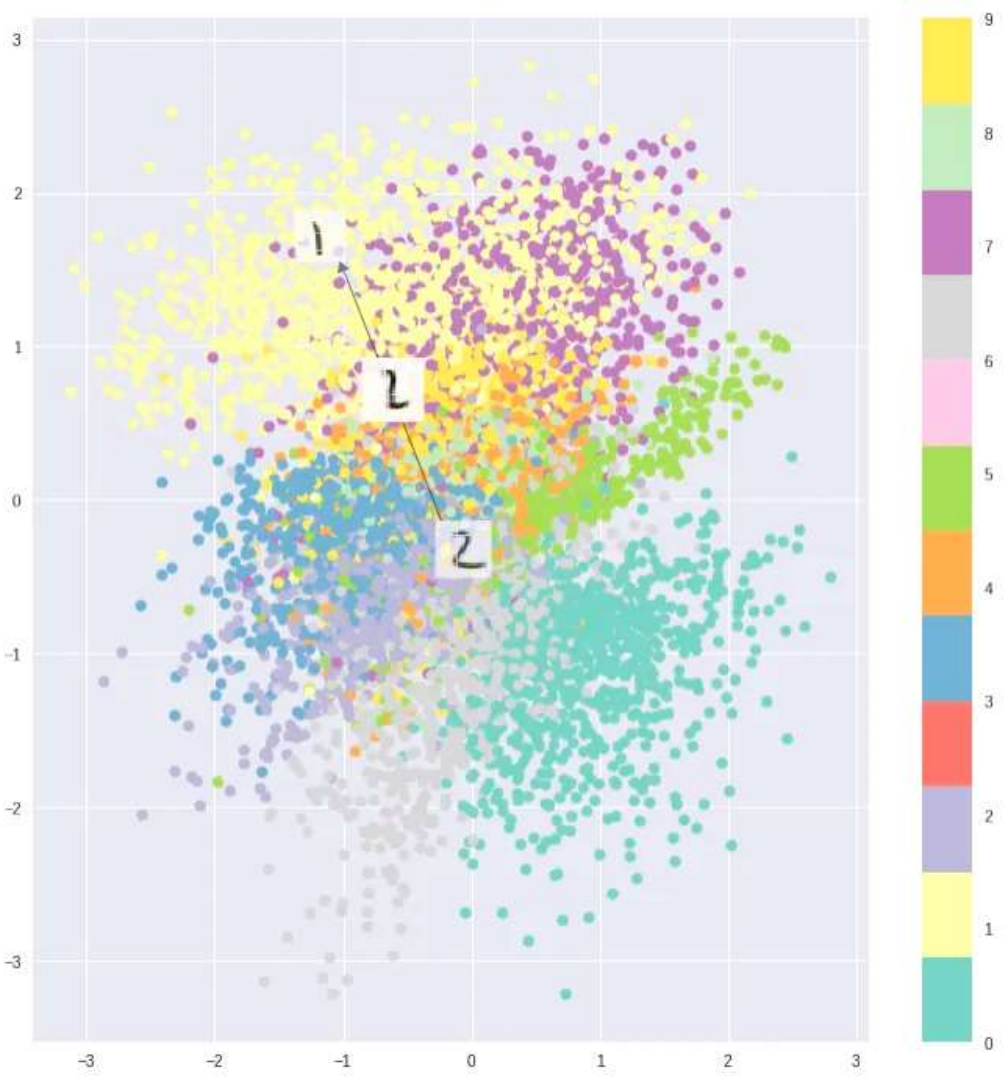
Only KL divergence



Combination

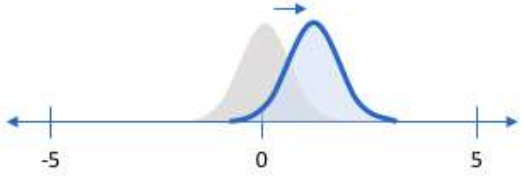


Autoencoders - Construcción del modelo



Autoencoders - Construcción del modelo

Penalizing reconstruction loss encourages the distribution to describe the input



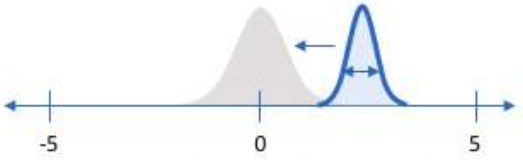
Our distribution deviates from the prior to describe some characteristic of the data

Without regularization, our network can “cheat” by learning narrow distributions

With a small enough variance, this distribution is effectively only representing a single value

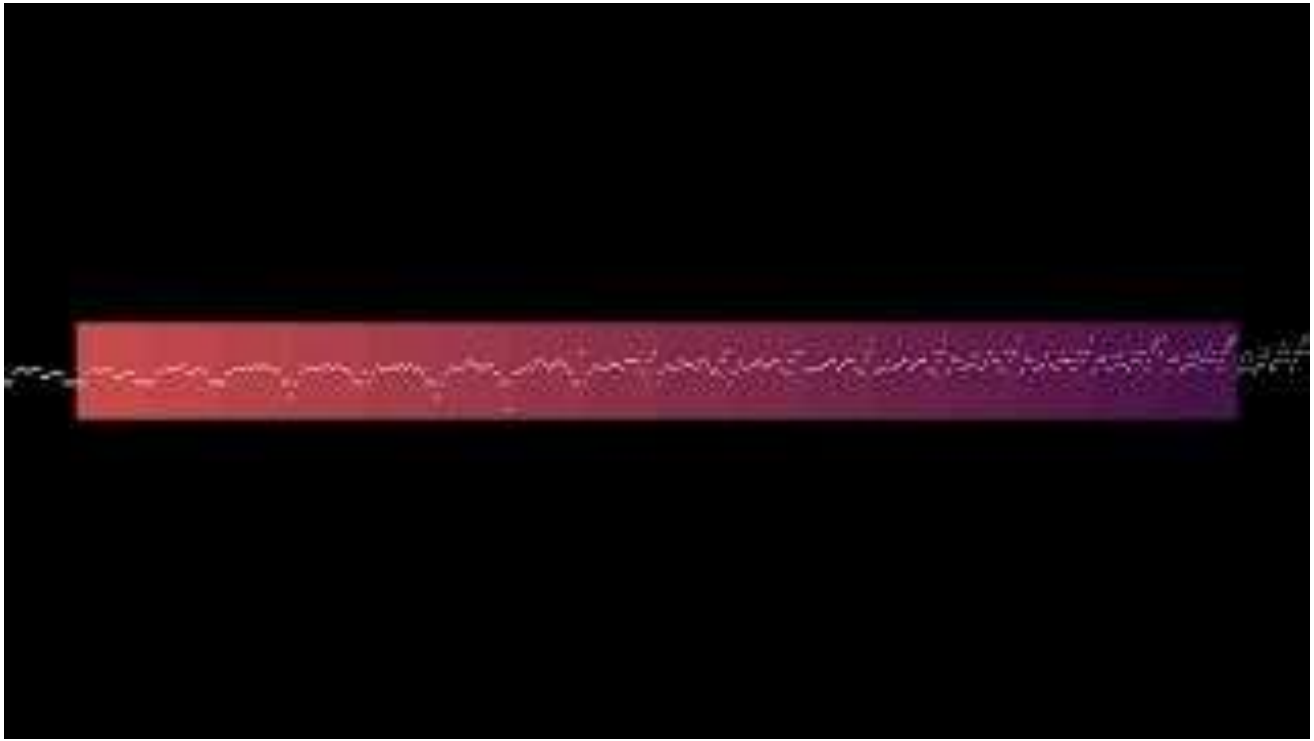
Penalizing KL divergence acts as a regularizing force

Attract distribution to have zero mean



Ensure sufficient variance to yield a smooth latent space

Ejemplo: Music VAE (interpolar 2 pistas de música con VAE)



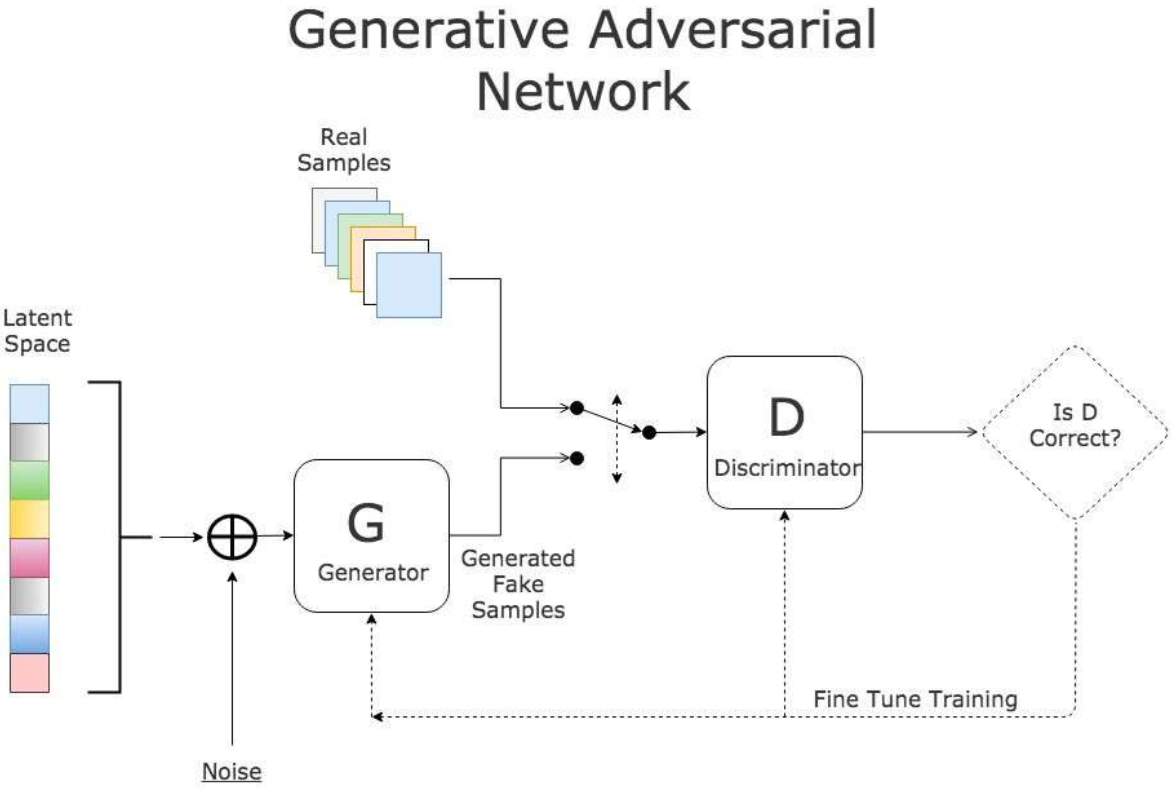
Ver más en:
https://magenta.tensorflow.org/music-vae

Ejemplo

- Interpolación vs Espacios latentes: https://peaceful-stallman-7c072c.netlify.app/

- Notebook: https://drive.google.com/file/d/1bgLZkyme8wxwiTyuzQ4sYU2eyEfGI--V/view?usp=sharing

GAN


Generative Adversarial Network
Real Samples
Latent Space
G
Generator
Generated Fake Samples
D
Discriminator
Is D Correct?
Fine Tune Training
Noise

Fuentes:

https://www.jeremyjordan.me/autoencoders/

https://www.jeremyjordan.me/variational-autoencoders/

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