Procesamiento del lenguaje natural

Aprendizaje profundo

Departamento de Sistemas Informáticos

E.T.S.I. de Sistemas Informáticos - UPM

Introducción (I)

Dentro del aprendizaje automático existen tres técnicas/esquemas de entrenamiento

- **Supervisado**: Se presentan ejemplos y respuestas y el modelo aprende a inferir
- **No supervisado**: Se presentan datos y el modelo aprende patrones
- **Por refuerzo**: Se presenta un entorno y el modelo aprende a desenvolverse en él

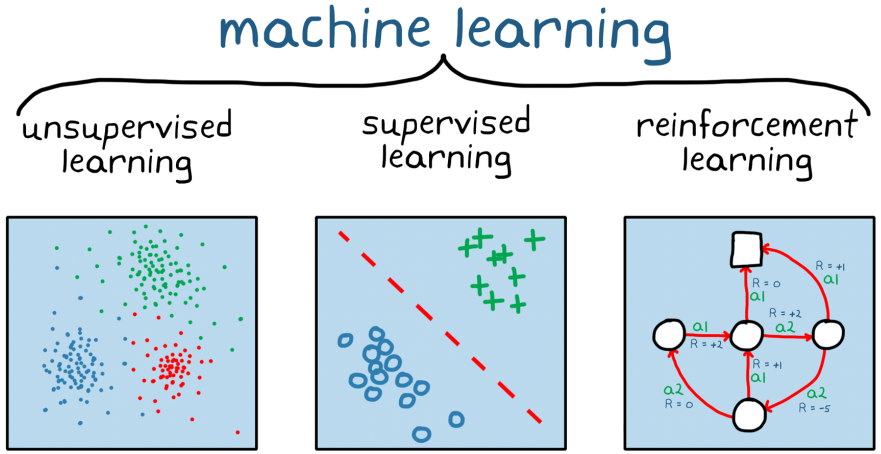

Figura 1. Diferentes esquemas de aprendizaje. Fuente: MathWorks

Introducción (II)

Otro punto de vista permite describir los modelos de ML en dos categorías:

- **Discriminativos**: Predicen la probabilidad de pertenecer a una clase según los datos de entrada
- **Generativos**: Buscan modelar cómo se generan los datos observados y pueden generar nuevos datos similares

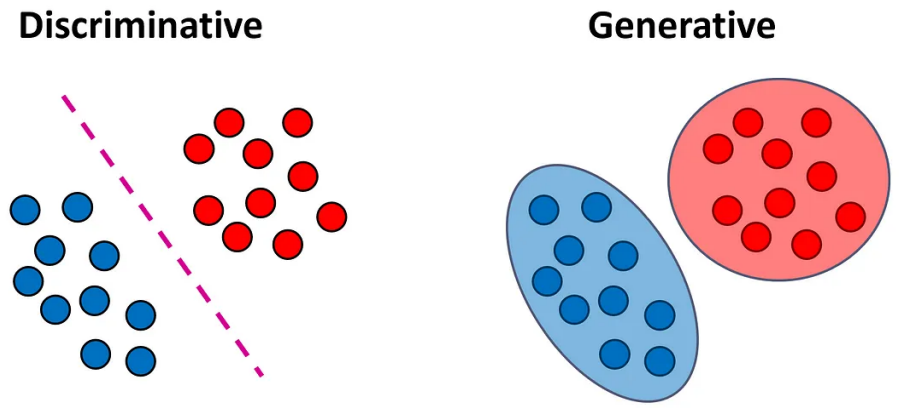

Figura 2. Modelos discriminativos frente a generativos. Fuente: Medium

Un poquito de historia

Primeros fundamentos

- **1960-1990**: Modelos probabilísticos clásicos. Se utilizan métodos estadísticos para modelar distribuciones de datos
  - Modelos de Márkov y modelos ocultos de Márkov (HMM, *hidden márkov models*) se aplican en el modelado de secuencias (e.g., reconocimiento del habla)[1]
  - Modelos de mezcla gaussiana (GMM, *gaussian mixture models*)[2] permiten representar distribuciones complejas mediante la combinación de múltiples distribuciones simples
- **1980-1990**: Máquinas de Boltzmann y redes neuronales. Se introducen modelos generativos basados en redes neuronales[3]
- **2006**: Redes de creencia profundas (DBN, *deep belief networks*). Enfoque donde se apilan capas de modelos probabilísticos para aprender representaciones jerárquicas de datos de forma generativa [4].

1 Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286.
2 Reynolds, D. A. (2009). *Gaussian mixture models*. Encyclopedia of biometrics, 741(659-663), 3.
3 Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). *A learning algorithm for Boltzmann machines*. Cognitive Science, 9(1), 147-169.
4 Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). *A fast learning algorithm for deep belief nets*. Neural Computation, 18(7), 1527-1554.

La transformación a modelos generativos

- **2013**: Autoencoders variacionales (VAE, *variational adversarial networks*): Permiten aprender una representación latente continua y generan nuevos datos a partir de ella, combinando ideas de autoencoders y métodos bayesianos [5]
- **2014**: Redes generativas antagónicas (GAN, _generative adversarial *networks*): Proponen un marco en el que dos redes neuronales (generadora y discriminadora) se entrenan de forma competitiva, logrando resultados impactantes en la generación de imágenes y otros dominios [6]

[5] Kingma, D. P., & Welling, M. (2013). *Auto-encoding variational bayes*. arXiv preprint arXiv:1312.6114.
[6] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). *Generative adversarial nets*. Advances in Neural Information Processing Systems, 27.

Evolución y especialización

- **2015-2016**: Redes GAN especializadas en imágenes.
  - Modelos autoregresivos para imágenes: Se presentan PixelRNN y PixelCNN, que generan imágenes modelando la distribución de píxeles de manera secuencial[7]
  - DCGAN (*deep Convolutional GAN*): Arquitectura basada en CNN que facilita la generación de imágenes de alta calidad y que establece un nuevo estándar en la comunidad[8]
- **2017-2018**: Modelos autoregresivos basados en Transformers. Son una revolución en la generación de texto (gran capacidad para aprender y sintetizar el lenguaje[9])
- **2020** en adelante: Introducción de los *denoising diffusion probabilistic models* (DDPM), que abren una nueva vía para la generación de **muy** alta calidad de imágenes[10] y otros tipos de datos[11][12].

[7] Van den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016). *Pixel recurrent neural networks*. arXiv preprint arXiv:1601.06759.
[8] Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.
[9] Radford, A., et al. (2018). *Improving language understanding by generative pre-training*.
[10] Ho, J., Jain, A., & Abbeel, P. (2020). *Denoising diffusion probabilistic models*. Advances in Neural Information Processing Systems, 33.
[11] Rombach, R., et al. (2022). *High-resolution image synthesis with latent diffusion models*. arXiv preprint arXiv:2112.10752.
[12] OpenAI. (2020). *Jukebox: A Generative Model for Music*. OpenAI Blog.

¿Qué aportan estos modelos?

No solo aprenden a diferenciar, sino que **aprenden la estructura de los datos**

- Y por tanto a generar datos similares que sigan esa estructura
- Nos permiten **interpolar** datos de forma «inteligente»
- También nos permiten **manipular** características de los datos generados
- Y **inferir** representaciones latentes de los datos
- Además, son útiles en la **detección de anomalías**

Sin embargo, tienen un coste computacional mayor que los modelos discriminativos

- Y en ocasiones son **mucho más** difíciles de entrenar y evaluar

Autoencoders (AE)

¿Qué son los Autoencoders? (I)

Tipo de red neuronal que puede aprender a **comprimir y reconstruir** datos

- Se utiliza en tareas de **aprendizaje no supervisado**
- Buscan aprender una **representación compacta** de los datos de entrada
- Teinen como objetivo minimizar la diferencia entre los datos de entrada y los reconstruidos por el decodificador

Se componen de dos componentes que se entrenan al mismo tiempo:

- **Codificador**: Transforma los datos de entrada en una representación de menor dimensión.
- **Decodificador**: Toma esta representación y reconstruye los datos originales.

¿Qué son los Autoencoders? (II)

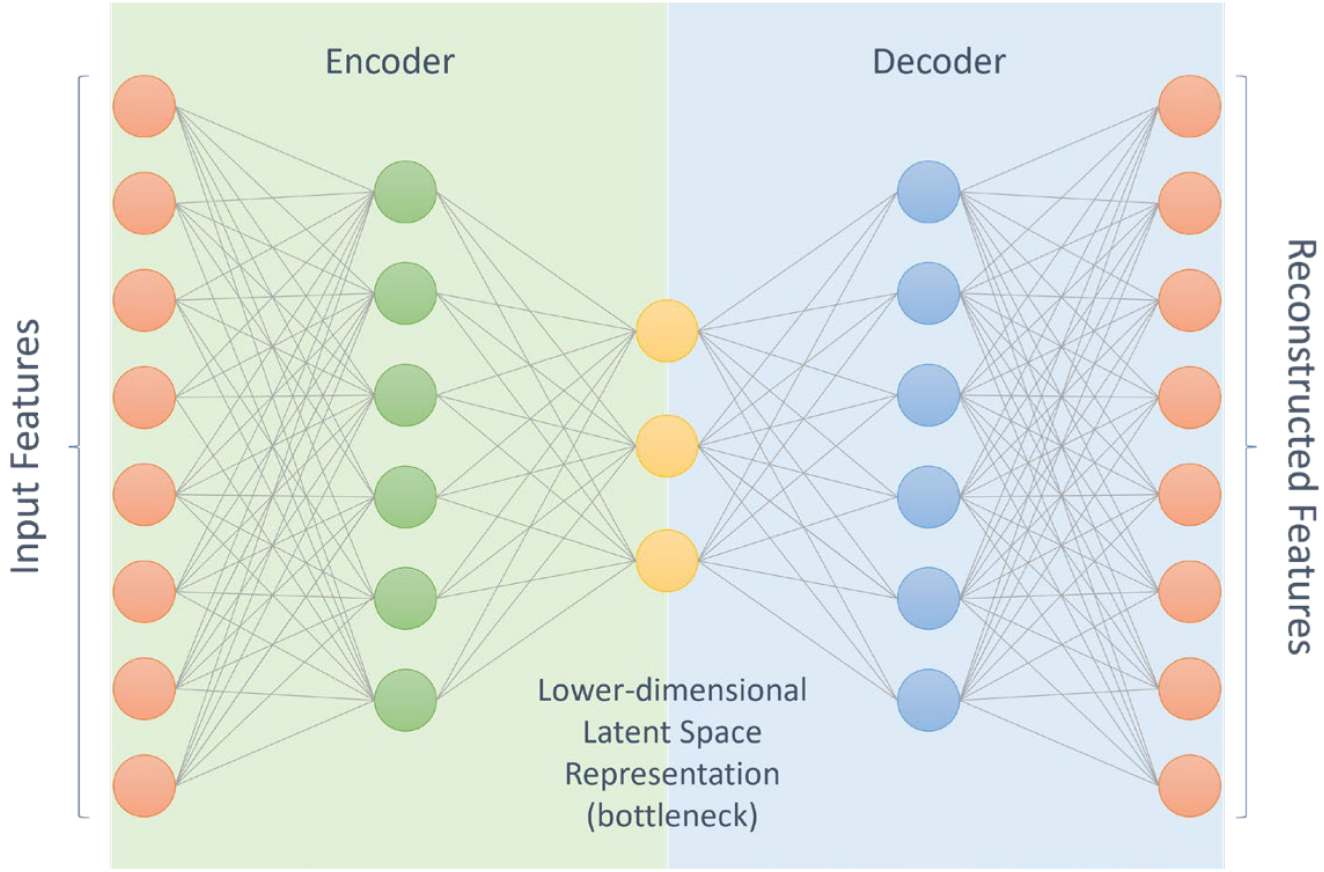

Figura 3. Autoencoder con perceptrones multicapa en la entrada y la salida del modelo

¿Qué son los Autoencoders? (III)

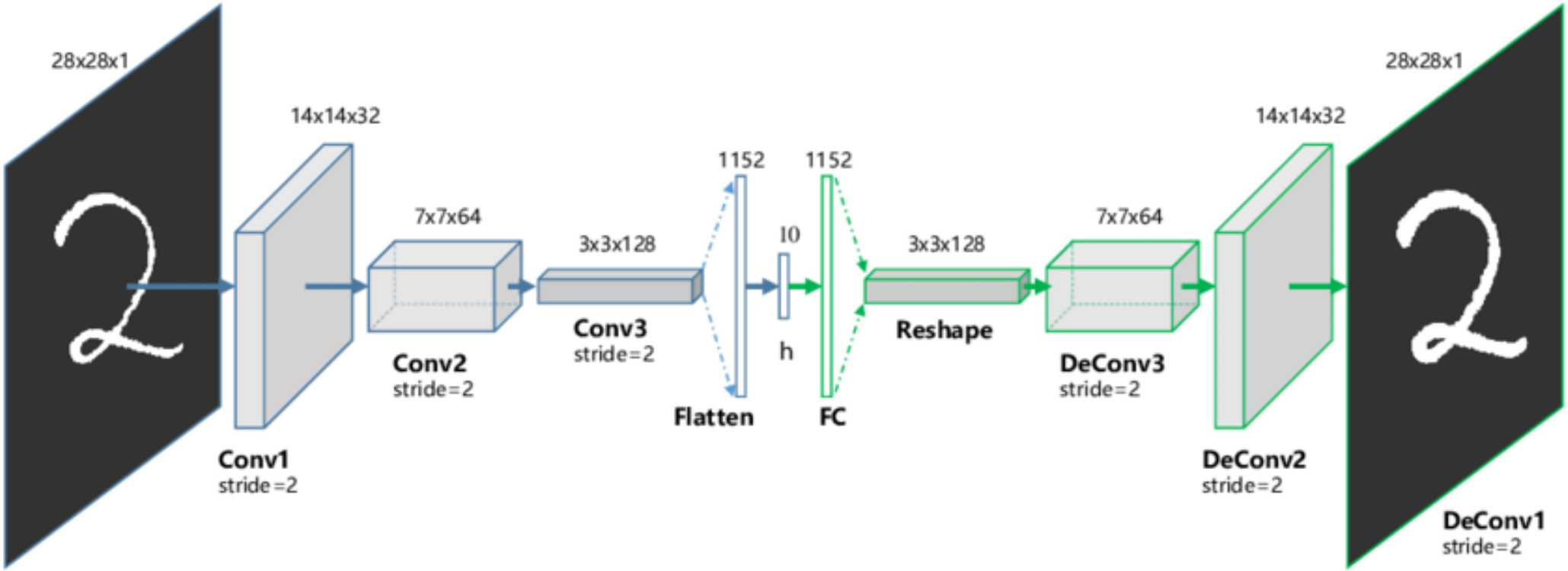

Figura 4. Auteoencoder con redes convolucionales a la entrada y la salida del modelo. (autoencoders convolucionales). Fuente _Deep Clustering with Convolutional Autoencoders

Áreas de aplicación

Este tipo de arquitectura es **muy** util en una amplia variedad de aplicaciones:

- **Entrenamiento de modelos con muchas capas**: Una de sus primeras aplicaciones fue en la pre-entrenamiento de redes neuronales profundas
- **Reducción de dimensionalidad**: Al aprender una representación compacta de los datos, se pueden reducir las dimensiones de los mismos
- **Eliminación de ruido**: Al aprender a reconstruir los datos, se pueden eliminar ruidos de los mismos
- **Detección de anomalías**: Modelar la distribución de los datos normales y detectar desviaciones significativas como anomalías
- **Generación de datos**: Al muestrear del espacio latente, los autoencoders pueden generar nuevas muestras de **datos similares** a los ejemplos de entrenamiento

¿Aprendizaje supervisado o no supervisado?

Tradicionalmente, se han clasificado como **aprendizaje no supervisado**

- Después de todo, no trabajan con datos *etiquetados*

Pero sí tienen una salida a ajustar, ¿no? como en **aprendizaje supervisado**

- Se aprende con retroalimentación de los datos, intentando minimizar el error al comparar la salida con la entrada

Yann LeCun inventó el término **aprendizaje auto-supervisado** para hablar sobre estos modelos

Variational autoencoders (VAE)

Motivación

Los *autoencoders* nos permiten generar datos, pero no demasiado buenos

- A cada ejemplo se le asigna un punto en el espacio latente proyectado
- Lo malo $\rightarrow$ No hay garantía de que puntos cercanos generen datos similares

Veamos un ejemplo sencillo: la reconstrucción de imágenes del dataset MNIST

- ¿Cómo sería el espacio latente?

Motivación

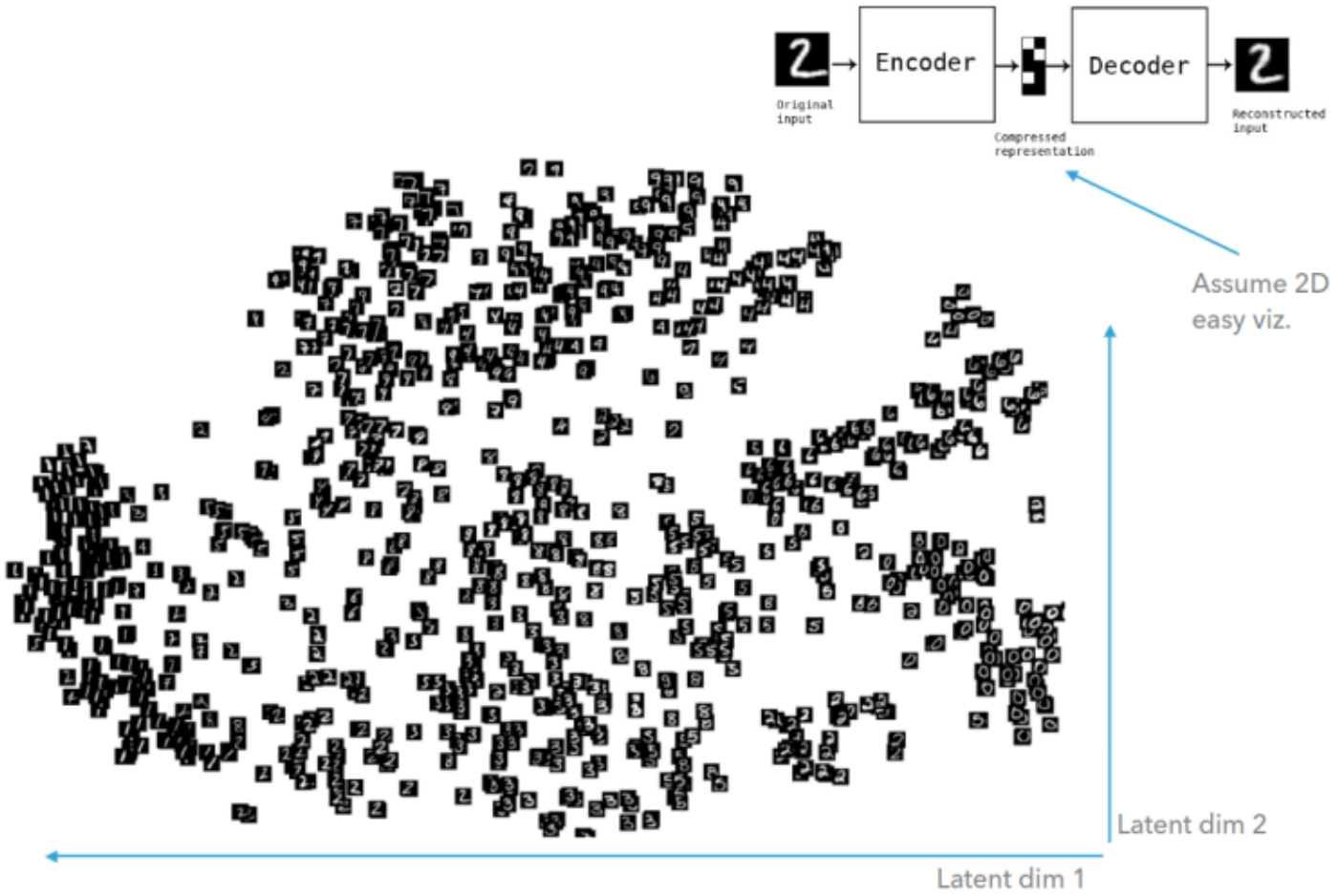

Figura 5. Espacio latente de dos dimensiones para un autoencoder entrenado con MNIST

Motivación

Al **no ser una distribución de datos continua**, tendremos problemas cuando la entrada sea ligeramente distinta a los datos con los que se entrenó el autoencoder:

¿Qué ocurrirá cuando la entrada sean imágenes que generen espacios latentes entre medio de las muestras de entrenamiento?

Motivación

- **Espacios continuos**: En un espacio continuo, los datos pueden tomar un rango infinito de valores dentro de un intervalo determinado.
- **Espacios discretos**: En un espacio discreto, los datos solo pueden tomar un conjunto finito o contablemente infinito de valores distintos.

Motivación

La mejor situación que buscamos es conseguir:

- Un espacio latente **continuo** y **ordenado**
- En el espacio ordenado permite tener las muestras similares agrupadas
- No se pierde la capacidad de interpolar entre diferentes muestras

Motivación - ¿Cómo lo conseguimos?

- Solo podemos forzar a la propia red a que ordene el espacio latente
- ¿Cómo?
- Durante el entrenamiento, se minimiza una función de perdida
- ¿Y...?
- Pués ahí es donde vamos a trabajar, pero entonces ya no usamos un Autoencoder...

Variational Autoencoders (VAEs)

Son una variante de los autoencoders que buscan la generación de datos sintéticos.

- Combinan redes neuronales con distribuciones de probabilidad.
- Permiten que los datos generados sigan el mismo patrón que los datos de entrada.

Así, la red aprende los parámetros de una distribución de probabilidad.

- Construyen explícitamente un ***espacio latente continuo*** y ***ordenado***.
- No una función arbitraria como en las redes neuronales convencionales.

Variational Autoencoders (VAEs)

El espacio latente está definido por **dos vectores** de tamaño n:

Luego debemos ajustar las funciones de pérdida individualmente de tal manera que:

- Una ***función de pérdida tradicional*** que calcula la diferencia con el objeto generado.
- La ***divergencia KL*** (Kullback-Leibler) entre la distribución latente aprendida y la distribución anterior (prior distribution), que actúa como término de regularización.

KL-divergence

¿Por qué necesitamos las pérdidas de reconstrucción y la divergencia KL?

KL-divergence

La ***funcion KL-divergence*** mide la diferencia entre dos distribuciones de probabilidad.

</p

Por ejemplo, en las distribuciones de la figura tenemos dos distribuciones:

- Una distribución normal y conocida p(x).
- Una distribución normal y desconocida q(x).

Es una divergencia, no una distancia, ya que no es simétrica.

KL-divergence

La fórmula de la divergencia Kullback-Leibler (KL) entre dos distribuciones de probabilidad (P) y (Q) es:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$

Donde:

- (p_i) es la probabilidad de la categoría (i) en la distribución (P).
- (q_i) es la probabilidad de la categoría (i) en la distribución (Q).

Aplicada en el contexto de Variational Autoencoders (VAEs) es:

$$D_{\mathrm{KL}}(P(z) \parallel Q(z)) = \frac{1}{2}\sum_{i=1}^{K}\left(\sigma_i^2 + \mu_i^2 - 1 - \log(\sigma_i^2)\right)$$

Donde:

- (K) es la dimensionalidad del espacio latente.
- (μ_i) y (σ_i) son la media y la desviación estándar de la distribución ($P(z)$) en la dimensión (i) del espacio latente.

I now call it “self-supervised learning”, because “unsupervised” is both a loaded and confusing term. […] Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That’s why calling it “unsupervised” is totally misleading.

Yann LeCun - Recent Advances in Deep Learning (2019)

Recursos didácticos

1. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006
2. Extracting and composing robust features with denoising autoencoders, 2008.
3. Semi-Supervised Recurrent Variational Autoencoder Approach for Visual Diagnosis of Atrial Fibrillation
4. Variational Autoencoders, Radboud University