

Introducción al Deep Learning

Día 4: Arquitecturas Avanzadas. Transformers

Manuel Germán y David de la Rosa
Universidad de Jaén



Universidad
de Jaén



`(mgerman, drrosa)@ujaen.es`

Cuestiones previas

1. ¿Qué es el *Deep Learning*?
2. ¿Qué es una neurona?
3. ¿Qué es una red neuronal y cómo se programa?
4. ¿Cómo se procesan imágenes con redes neuronales?
5. ¿Cómo se procesan secuencias con redes neuronales?

Expectativas

Tras esta sesión, sabremos:

- No todas las redes neuronales tienen la misma arquitectura.
- ¿Qué es un Autoencoder?.
- ¿Qué arquitectura tiene un Autoencoder?.
- ¿Para qué sirve un Autoencoder?
- Cómo implementar un Autoencoder usando *Pytorch* y *Pytorch Lightning*.

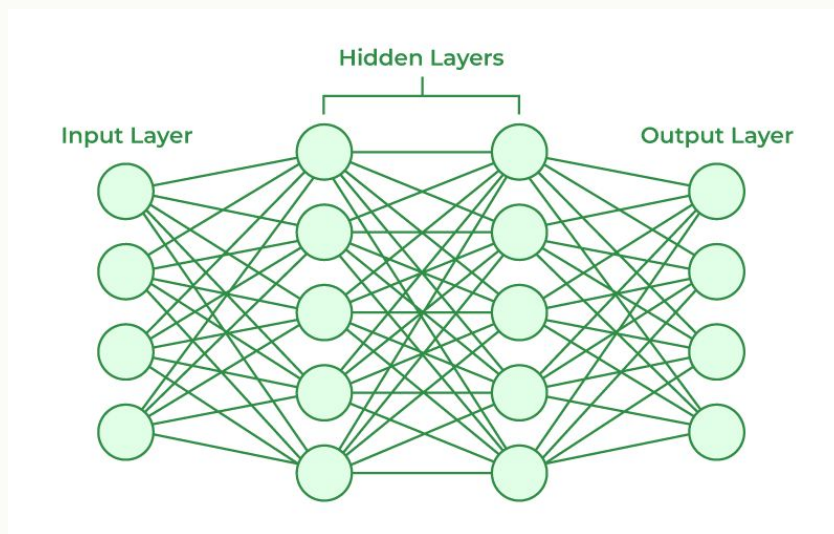
1

Arquitecturas de red neuronal

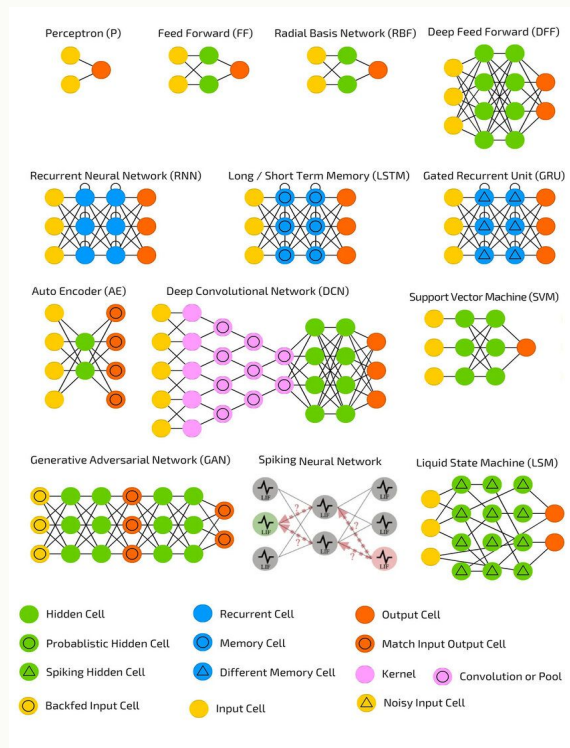
Redes neuronales tradicionales

Hidden Layers grandes para extraer a más características mejor

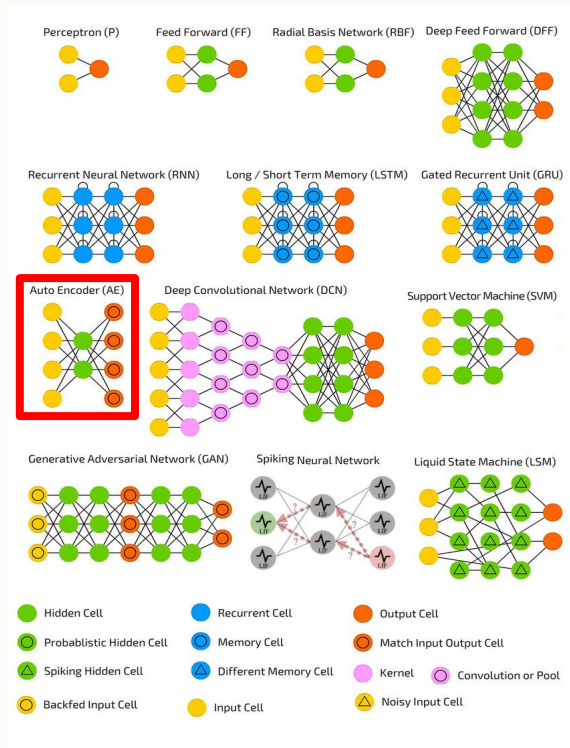
Output layer para clasificación / regresión, con pocas neuronas



Hay más arquitecturas



Hay más arquitecturas

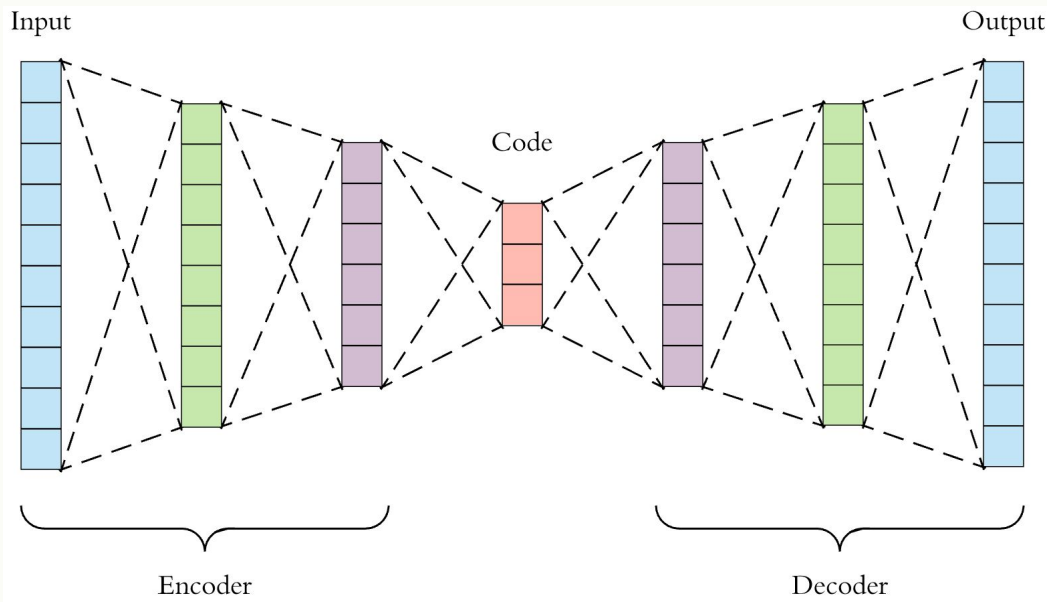


2

Qué es un Autoencoder

¿Qué es un Autoencoder?

Es un tipo de red neuronal entrenada para codificar los datos y reconstruir la entrada a partir de la codificación

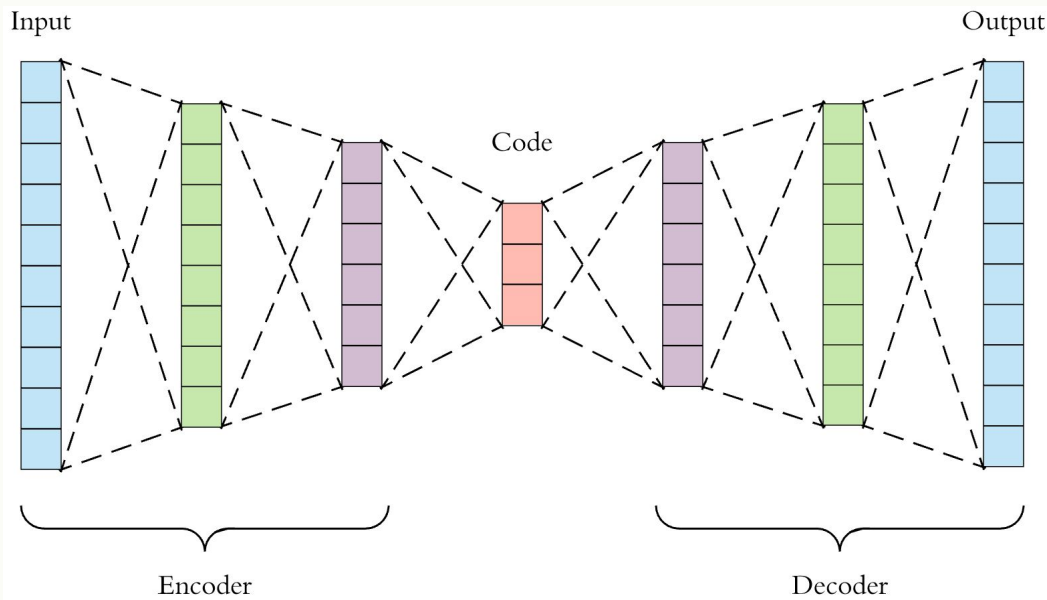


3

Para qué
sirven los AE

¿Esto para qué?

La clave está en el “code” o la codificación. Se fuerza que aprenda las características más representativas de los datos.

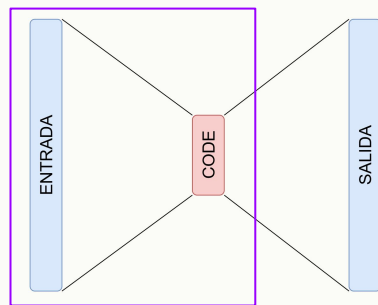


Aplicaciones

Reducción de dimensionalidad

Los datos en bruto tienen alta dimensionalidad, añadiendo complejidad a la tarea a realizar.

Un AE puede reducir la dimensionalidad de los datos usando el “code” e incluso si es necesario, volver a reconstruirlos usando el *decoder*.

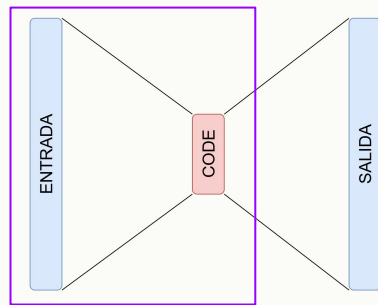


Aplicaciones

Favorecer *downstream tasks*

Gracias a un estado codificado de los datos en bruto, de menor dimensión, la tarea (clasificación, regresión,...) se vuelve más sencilla.

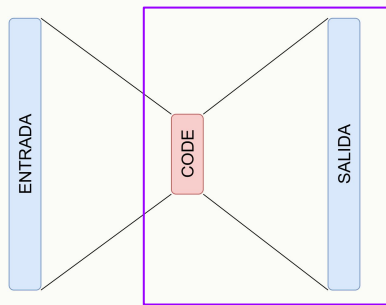
Además, en el entrenamiento se fuerza a que “code” capture las propiedades más relevantes, sin pérdida de información.



Aplicaciones

Generación de instancias

El espacio latente captura las propiedades de las instancias originales. Cogiendo un punto cualquiera del espacio latente, idealmente obtenemos una reconstrucción útil y lógica.

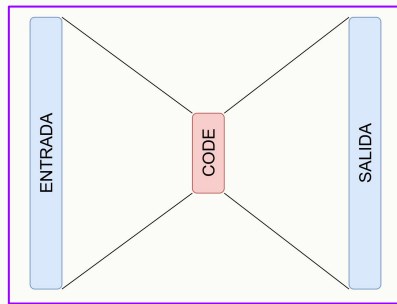


Aplicaciones

Denoining

Eliminar el ruido de las instancias, como imágenes.

Entrenamos al autoencoder para que, en el proceso de codificación y decodificación, sea capaz de reconstruir eliminando el ruido inicial.

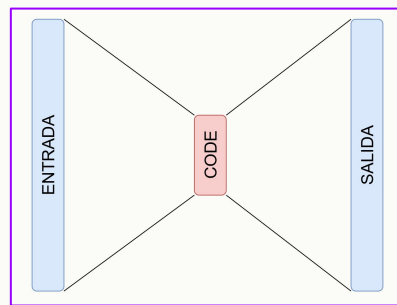


Aplicaciones

Detección de anomalías

El autoencoder aprende a reconstruir lo que le pasemos como entrada, es decir, sabe reconstruir cosas “similares” a las que se ha entrenado.

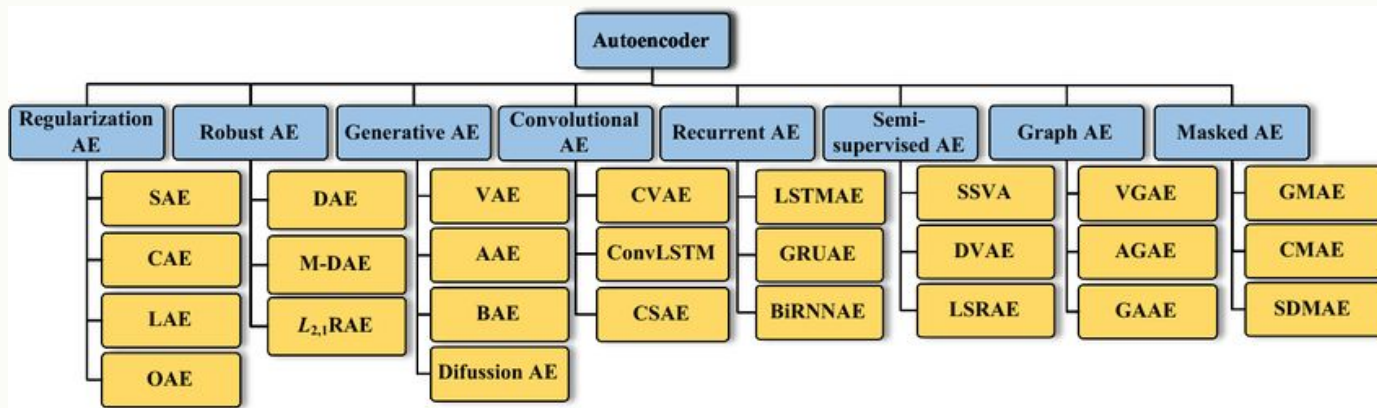
Cuando le pedimos que reconstruya algo diferente, una *anomalía*, no puede reconstruirlo de forma exacta.



4

Tipos de AE

Tipos de AE



Vamos a programar un Autoencoder!