



DaSCI

Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

UJa Universidad
de Jaén

Autoencoders

¿Qué son, para qué sirven y cómo funcionan?

Francisco Charte

fcharte@ujaen.es

27 enero 2021

Estructura de la charla

¿Qué es un autoencoder?

- Características básicas
- Partes del autoencoder
- Mecanismo de aprendizaje

¿Para qué sirven los autoencoders?

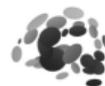
- Reducción de dimensionalidad
- Identificación de anomalías
- Eliminación de ruido
- Imputación de datos perdidos
- Anonimización de datos
- Criptografía y seguridad
- Agrupamiento semántico
- Generación de nuevos datos

¿Cómo funciona un autoencoder?

- Mecanismo de autoaprendizaje
- Funciones objetivo
- Referencias útiles

Construcción de autoencoders

- En Python - Usando Keras
- En R - El paquete ruta



Contenidos

¿Qué es un autoencoder?

- Características básicas

- Partes del autoencoder

- Mecanismo de aprendizaje

¿Para qué sirven los autoencoders?

- Reducción de dimensionalidad

- Identificación de anomalías

- Eliminación de ruido

- Imputación de datos perdidos

- Anonimización de datos

- Criptografía y seguridad

- Agrupamiento semántico

- Generación de nuevos datos

¿Cómo funciona un autoencoder?

- Mecanismo de autoaprendizaje

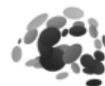
- Funciones objetivo

- Referencias útiles

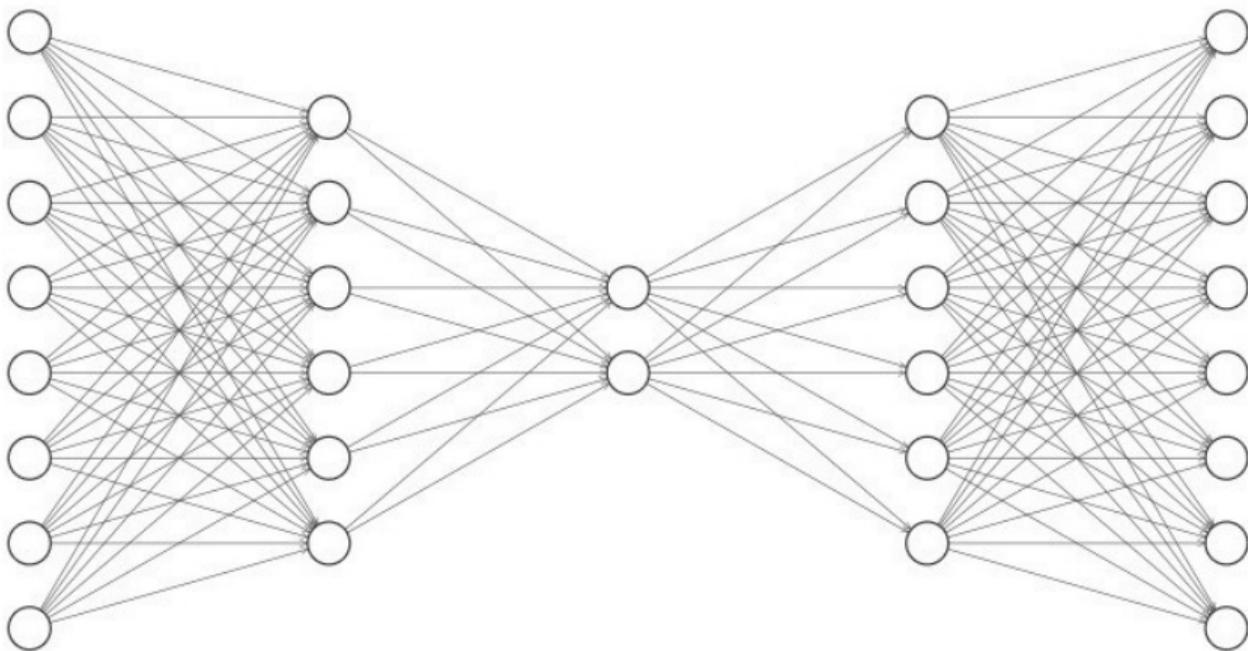
Construcción de autoencoders

- En Python - Usando Keras

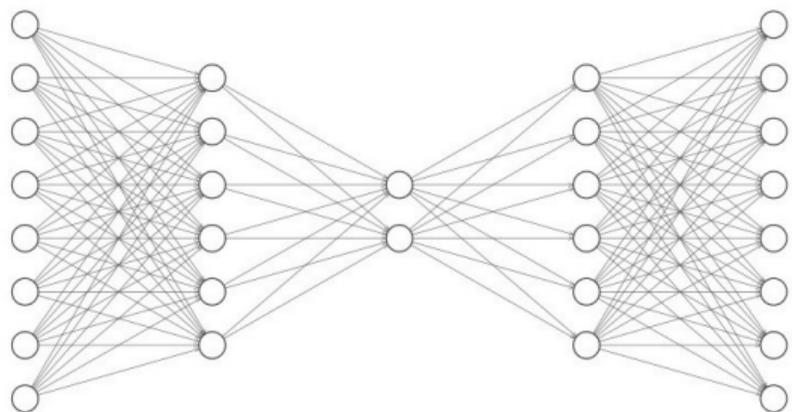
- En R - El paquete ruta



Red neuronal simétrica y no supervisada

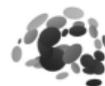


Red neuronal completamente conectada

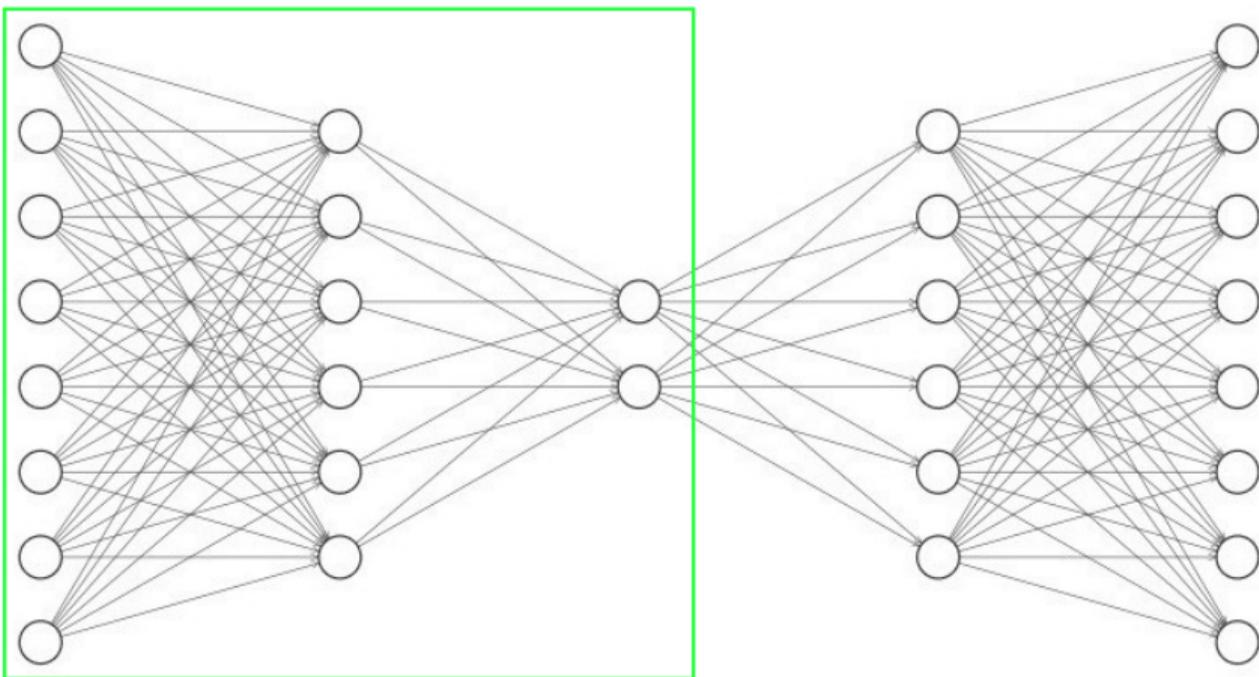


Funciona como un MLP cualquiera

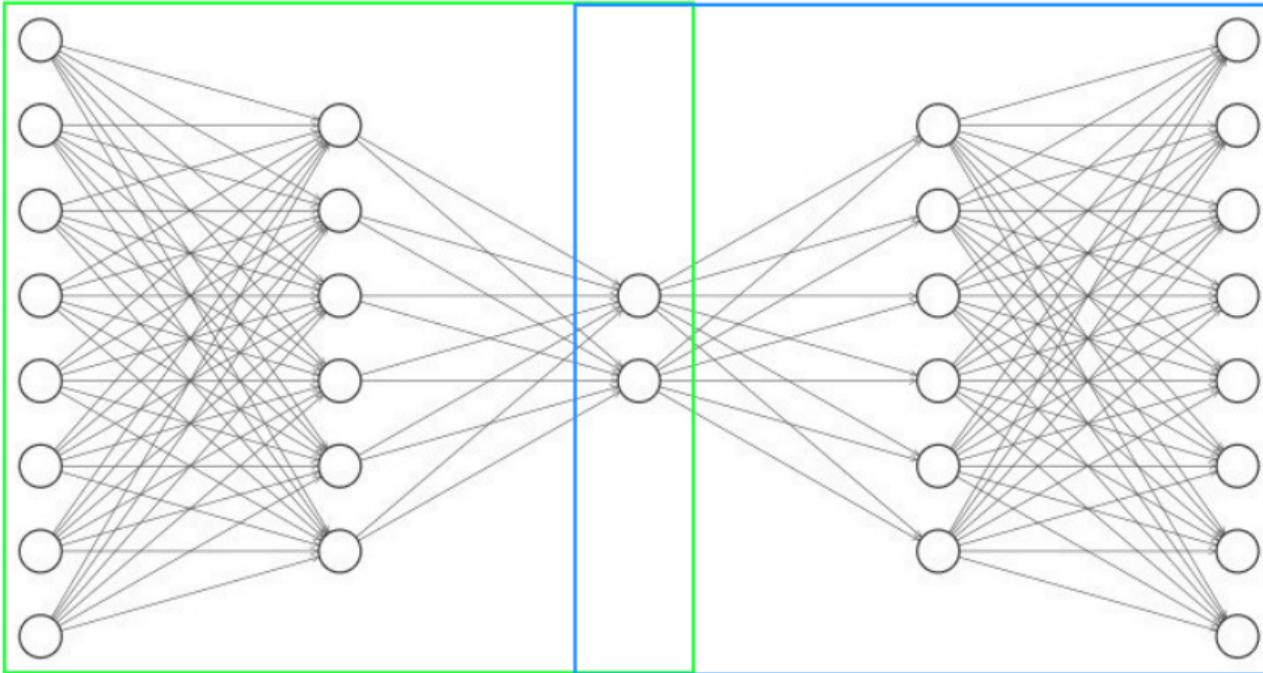
- ◆ cada neurona se conecta con todas las de la capa siguiente
- ◆ cada conexión tiene un peso asociado
- ◆ cada neurona aplica a la entrada una función de activación
- ◆ ajuste de los pesos con algoritmo de aprendizaje tipo *backpropagation*



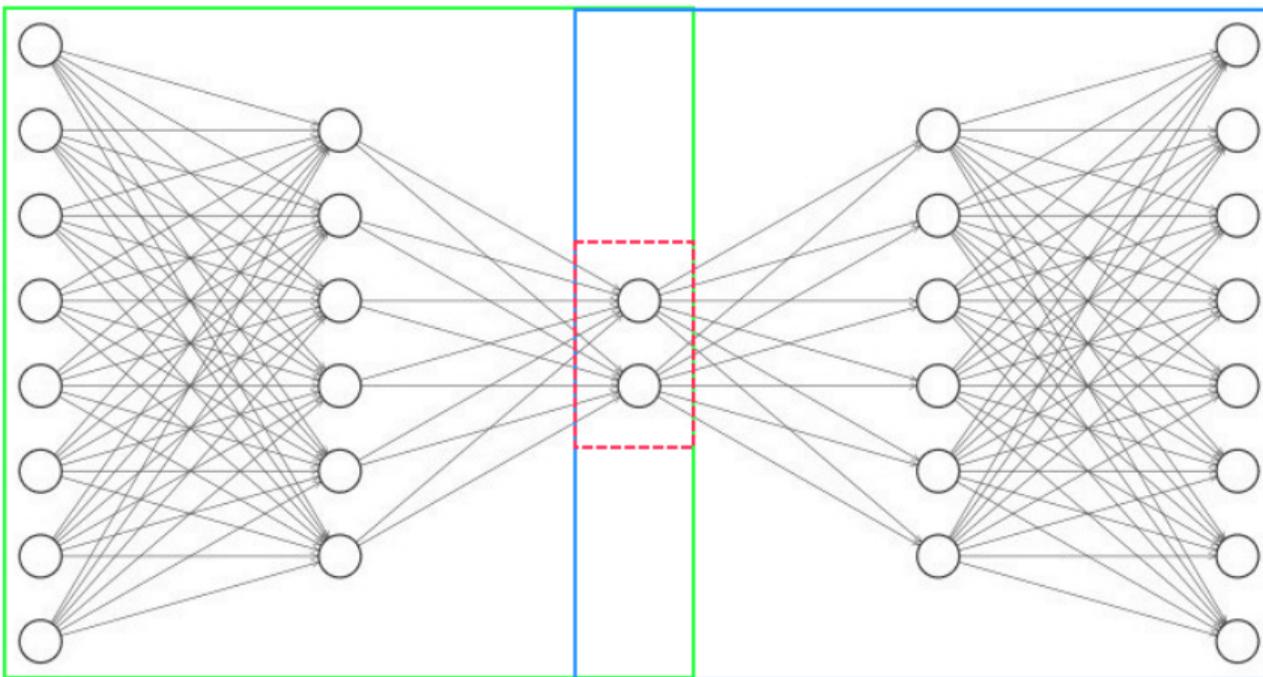
Simétrica - Codificador



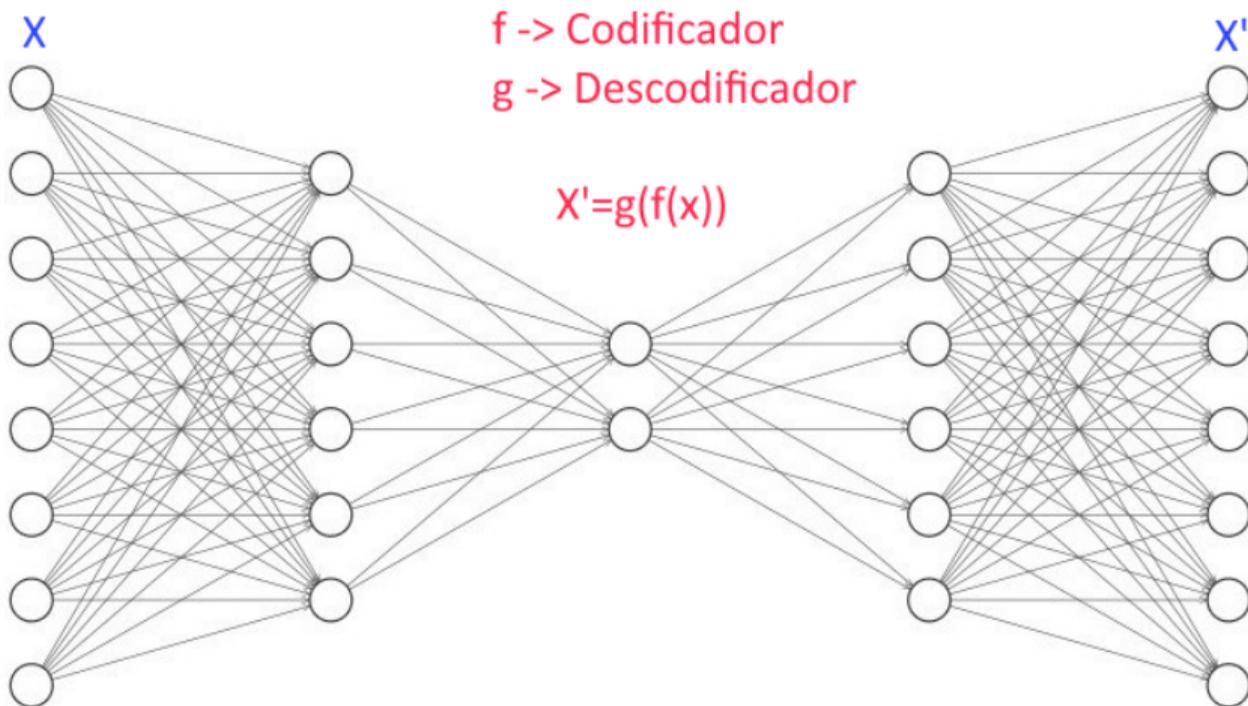
Simétrica - Decodificador



Parte común - Codificación



Autoaprendizaje



Contenidos

¿Qué es un autoencoder?

- Características básicas

- Partes del autoencoder

- Mecanismo de aprendizaje

¿Para qué sirven los autoencoder?

- Reducción de dimensionalidad

- Identificación de anomalías

- Eliminación de ruido

- Imputación de datos perdidos

- Anonimización de datos

- Criptografía y seguridad

- Agrupamiento semántico

- Generación de nuevos datos

¿Cómo funciona un autoencoder?

- Mecanismo de autoaprendizaje

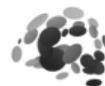
- Funciones objetivo

- Referencias útiles

Construcción de autoencoders

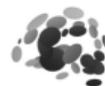
- En Python - Usando Keras

- En R - El paquete ruta



Tipos de aplicaciones

- ◆ Reducción de dimensionalidad
- ◆ Identificación de anomalías
- ◆ Eliminación de ruido
- ◆ Imputación de datos perdidos
- ◆ Anonimización de datos
- ◆ Agrupamiento semántico
- ◆ Generación de nuevos datos



La maldición de la alta dimensionalidad

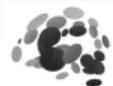
Verleysen, M., & François, D. (2005, June). The curse of dimensionality in data mining and time series prediction. In International work-conference on artificial neural networks (pp. 758-770). Springer, Berlin, Heidelberg.

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001, January). On the surprising behavior of distance metrics in high dimensional space. In International conference on database theory (pp. 420-434). Springer, Berlin, Heidelberg.

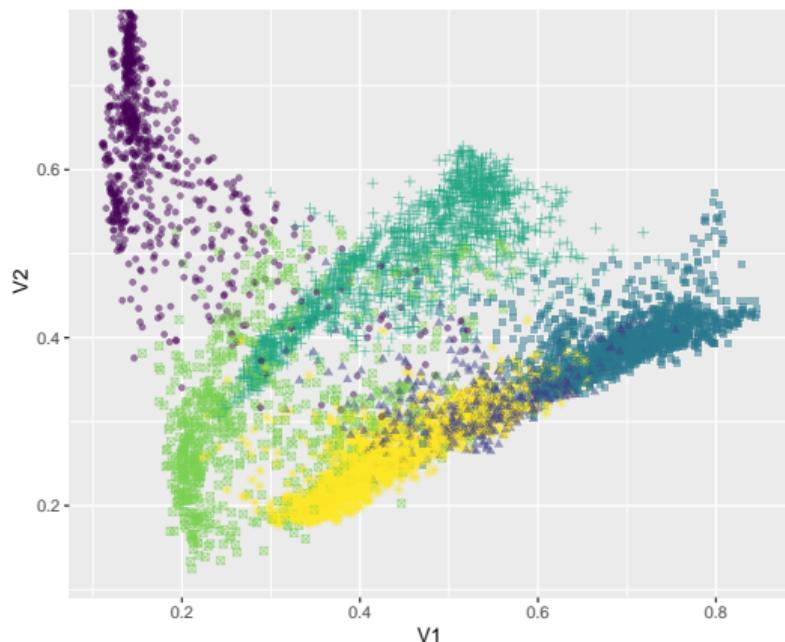
Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999, January). When is “nearest neighbor” meaningful?. In International conference on database theory (pp. 217-235). Springer, Berlin, Heidelberg.

Proyección en un espacio de menor dimensionalidad

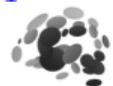
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.



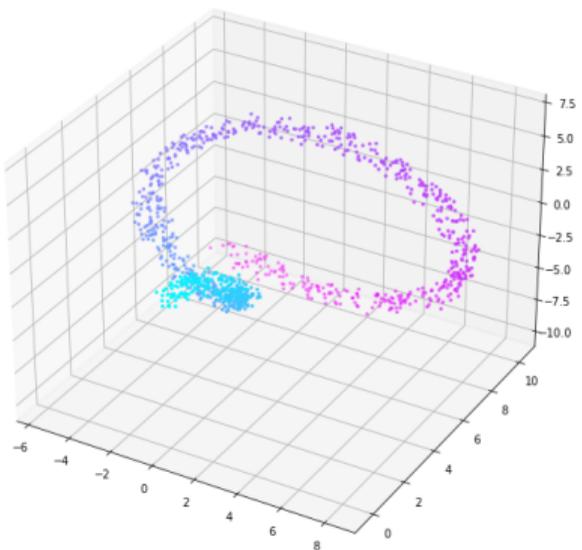
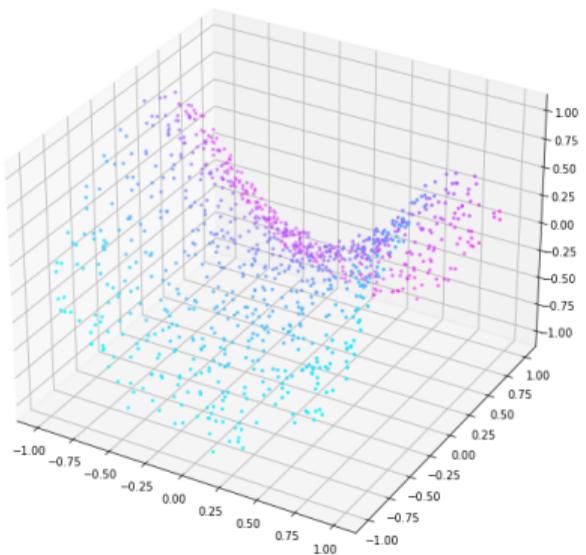
Visualización - Proyección 2D



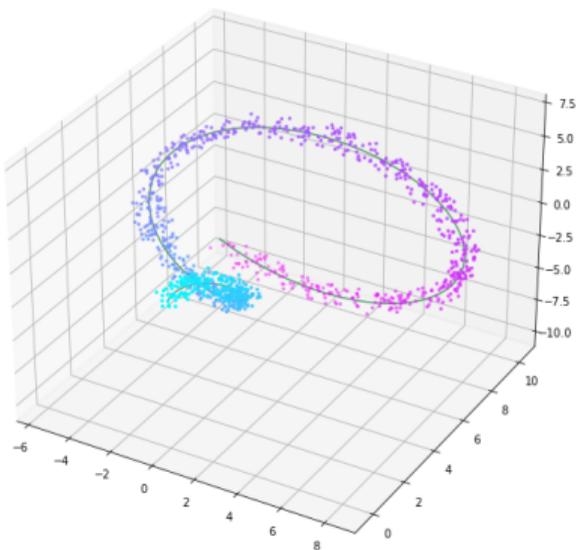
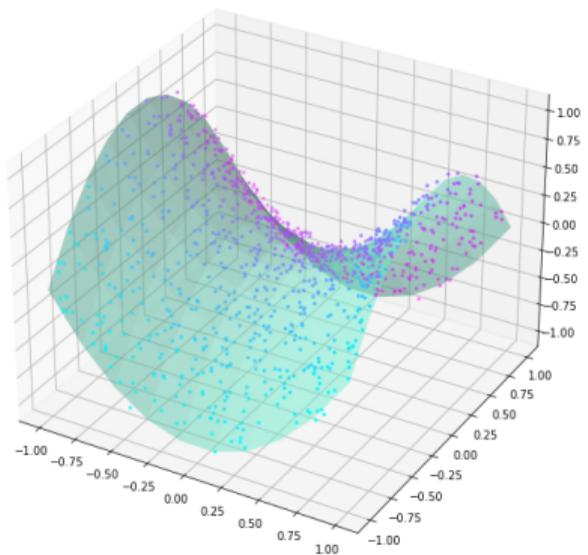
satellite_image - Datos espectrales que identifican el tipo de suelo. Espacio original con 36 dimensiones - <https://www.openml.org/d/294>



Manifold Learning



Manifold Learning



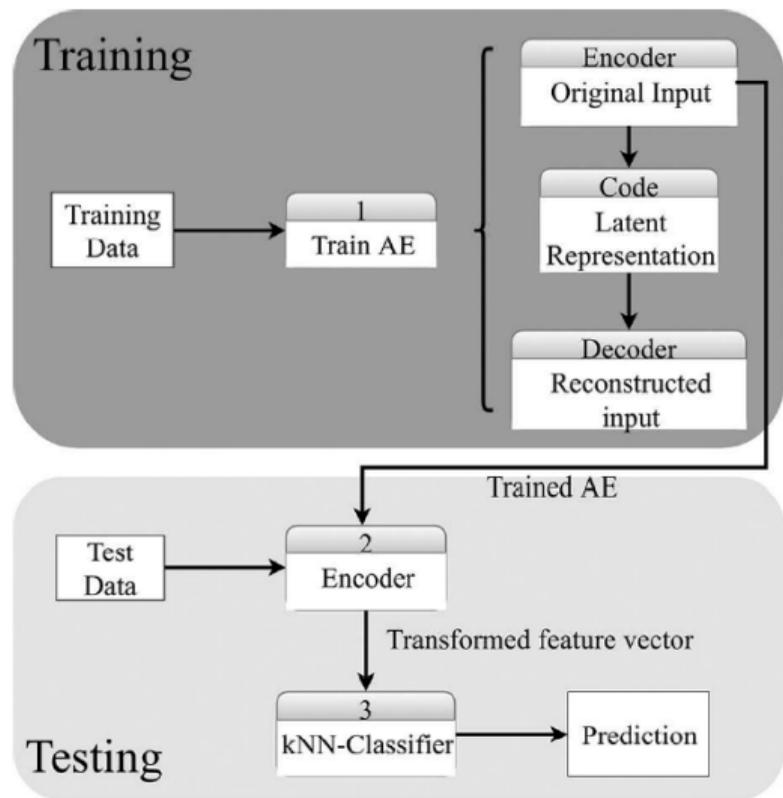
Los datos pueden estar distribuidos en un espacio de menor dimensionalidad (manifold) que es posible aprender con un AE



DaSCI

Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

AEkNN - Mejora de kNN con un autoencoder



Pulgar, F. J., Charte, F., Rivera, A. J., & del Jesus, M. J. (2019). AEkNN: An AutoEncoder kNN-Based Classifier With Built-in Dimensionality Reduction. *International Journal of Computational Intelligence Systems*, 12(1), 436-452.

Datos anómalos \neq ruido o datos erróneos

Anomalías

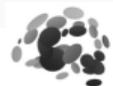
Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *IEEE Communications surveys & tutorials*, 16(1), 303-336.

Pumsirirat, A., & Yan, L. (2018). Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine. *International Journal of advanced computer science and applications*, 9(1), 18-25.

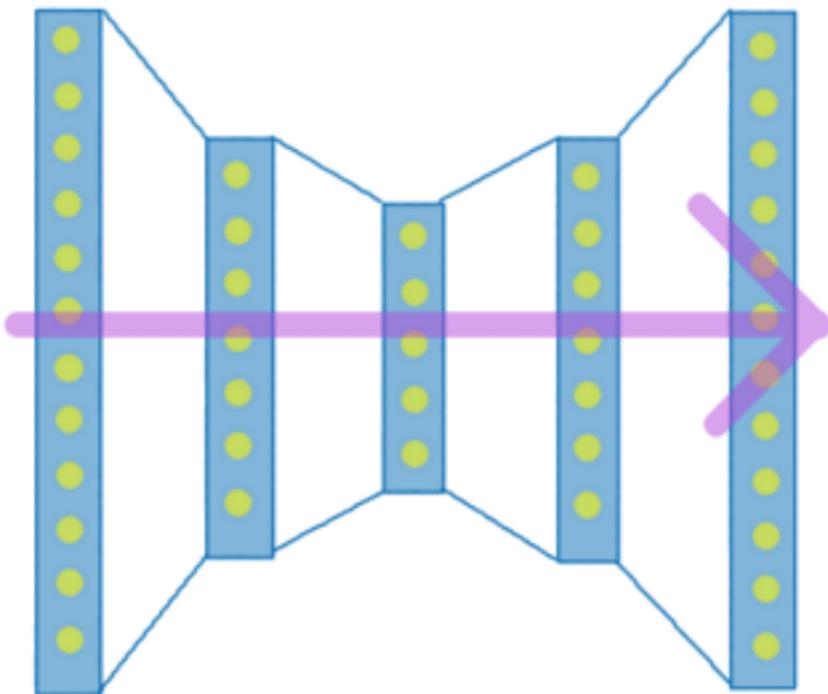
Detección de anomalías con autoencoders

M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: *Proceedings of the MLSDA(2014) Second Work- shop on Machine Learning for Sensory Data Analysis*, ACM, 2014, pp. 4–11, doi: 10.1145/26 89746.26 89747.

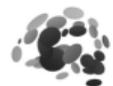
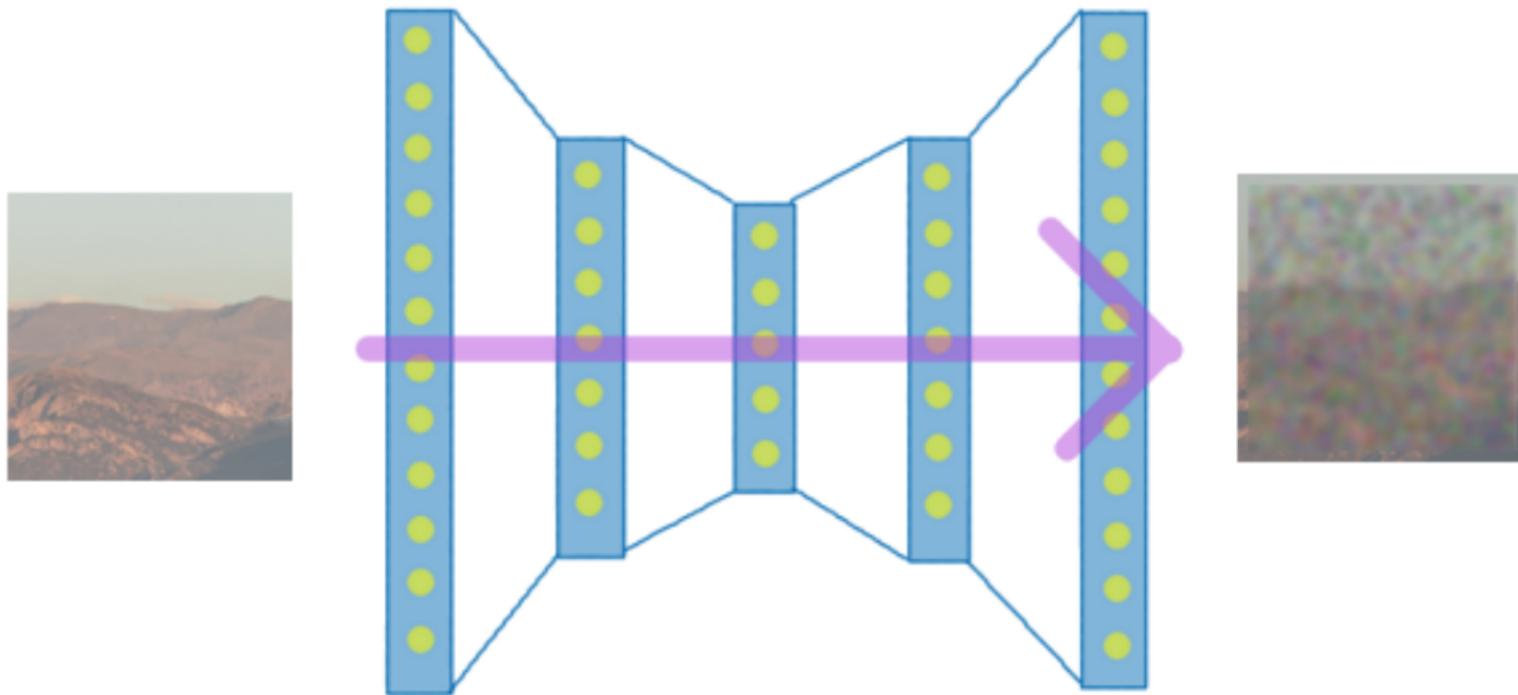
S. Park, M. Kim, S. Lee, Anomaly detection for HTTP using convolutional autoencoders, *IEEE Access* 6 (2018) 70884–70901, doi: 10.1109/ACCESS.2018. 2881003.



Cómo detectar anomalías con un autoencoder



Cómo detectar anomalías con un autoencoder

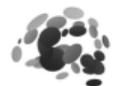


Detección de animales en imágenes de fototrampeo



Autoencoder entrenado con imágenes de una misma localización pero con distintas condiciones

- ◆ Las cámaras se disparan espontáneamente muchas veces
- ◆ El descarte de imágenes vacías supone muchas horas de trabajo
- ◆ El autoencoder identifica como una anomalía la presencia de cualquier animal o cambio importante en la escena



Ruido en señales y otros datos

Tradicional: filtros a medida para eliminación dependiendo del tipo de señal: audio, imágenes, etc.

Autoencoders para eliminación de ruido

X. Lu , Y. Tsao , S. Matsuda , C. Hori , Speech enhancement based on deep de- noising autoencoder, Interspeech (2013) 436–440.

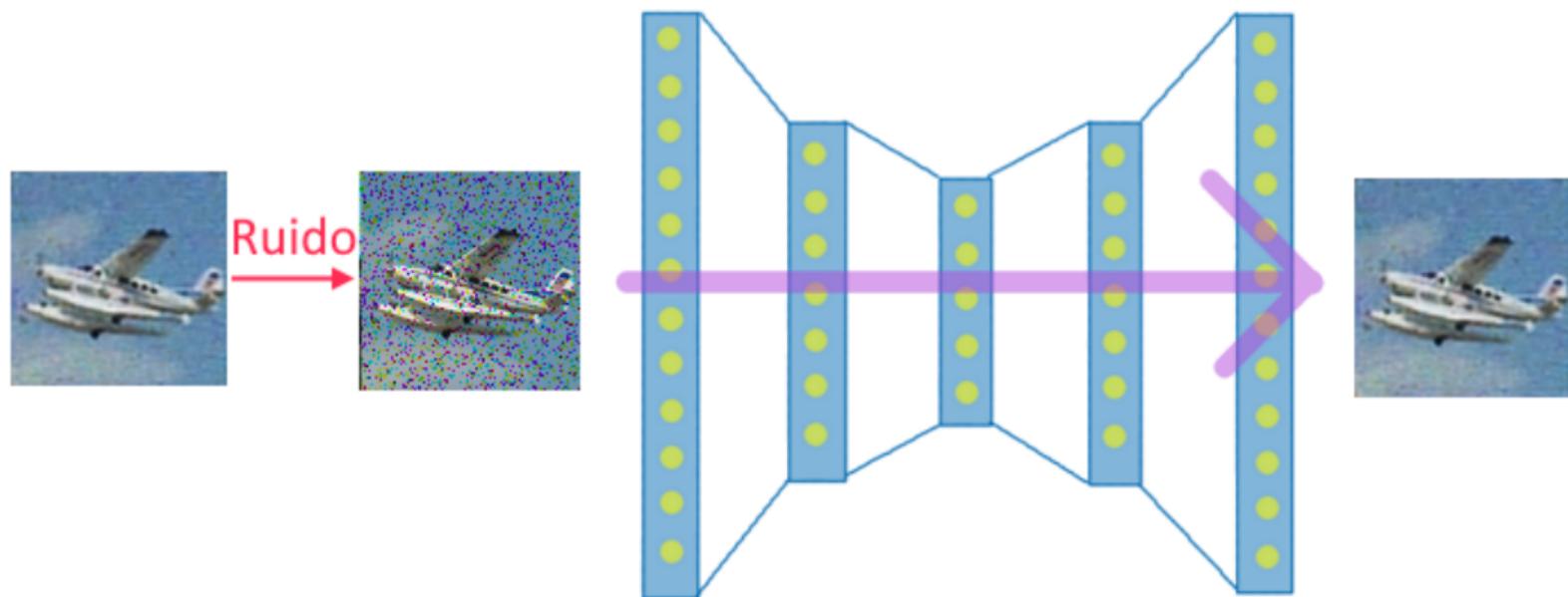
P. Xiong, H. Wang, M. Liu, S. Zhou, Z. Hou, X. Liu, Ecg signal enhancement based on improved denoising auto-encoder, Eng. Appl. Artif. Intell. 52 (2016) 194–202.

J. Xie , L. Xu , E. Chen , Image denoising and inpainting with deep neural networks, Adv. Neural Inf. Process. Syst. (2012) 341–349.

P. Vincent , H. Larochelle , I. Lajoie , Y. Bengio , P.A. Manzagol , Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.



Cómo eliminar ruido con un autoencoder



Eliminar ruido de imágenes



Eliminar ruido de imágenes



Estimar datos que no han podido obtenerse

Tradicional: métodos estadísticos, vecinos más cercanos

Zhang, Z. (2016). Missing data imputation: focusing on single imputation. *Annals of translational medicine*, 4(1).

Malarvizhi, M. R., & Thanamani, A. S. (2012). K-nearest neighbor in missing data imputation. *International Journal of Engineering Research and Development*, 5(1), 5-7.

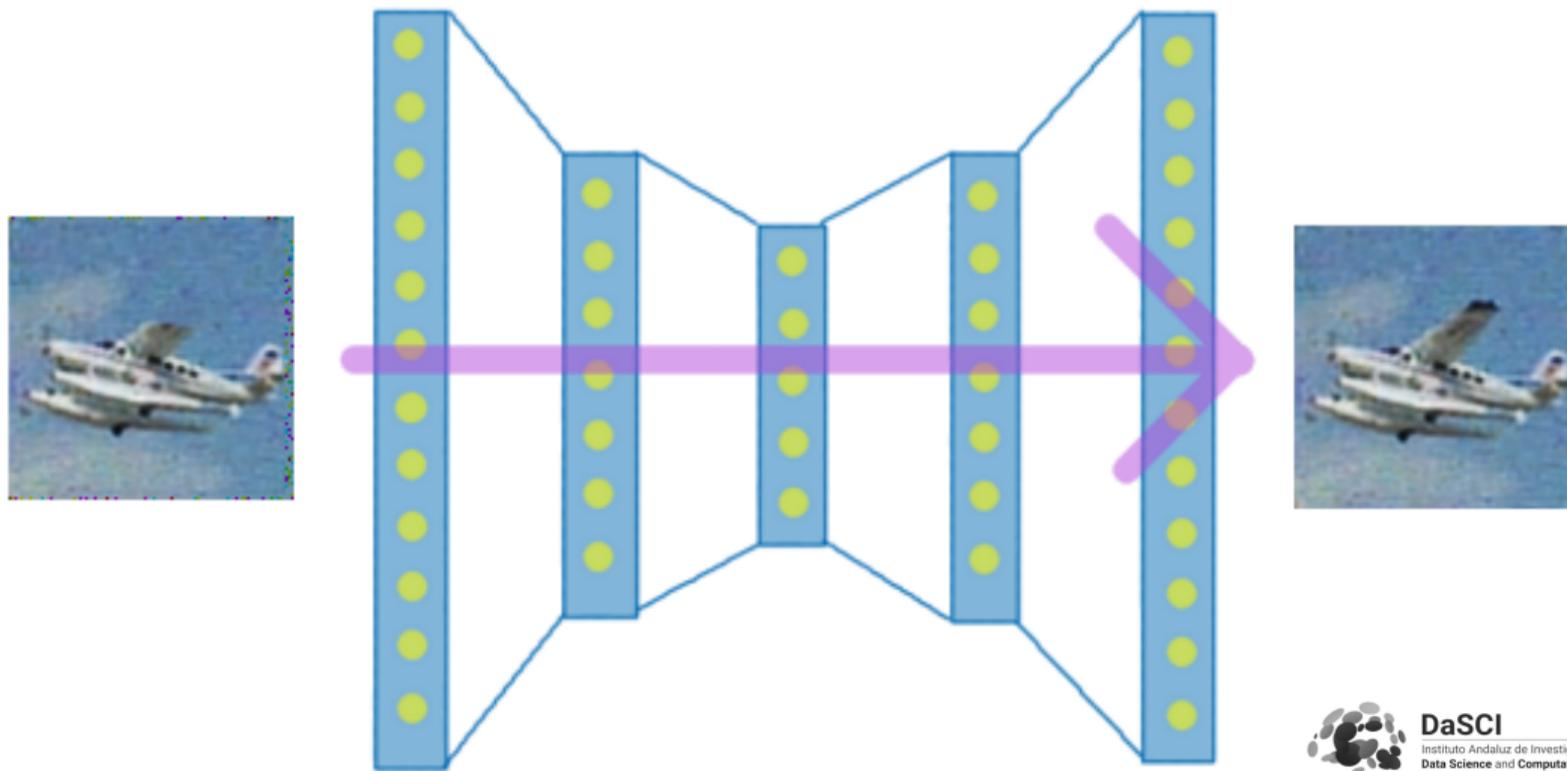
Imputación de datos con autoencoders

Beaulieu-Jones, B. K., Moore, J. H. (2017). Missing data imputation in the electronic health record using deeply learned autoencoders. In *Pacific symposium on biocomputing 2017* (pp. 207-218).

Tran, L., Liu, X., Zhou, J., & Jin, R. (2017). Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1405-1414).



Cómo imputar datos con un autoencoder



Normativas y guías gubernamentales

Gil, E. (2016). Big data, privacidad y protección de datos. Madrid: Agencia Estatal Boletín Oficial del Estado.

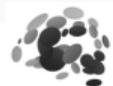
Cuadrada, E. B. (2007). La protección de datos en España y en la Unión Europea. Especial referencia a los mecanismos jurídicos de reacción frente a la vulneración del derecho a la intimidad. IDP. Revista de Internet, Derecho y Política, (5), 78-92.

Técnicas de anonimización de datos

Zhou, B., Pei, J., & Luk, W. (2008). A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM Sigkdd Explorations Newsletter, 10(2), 12-22.

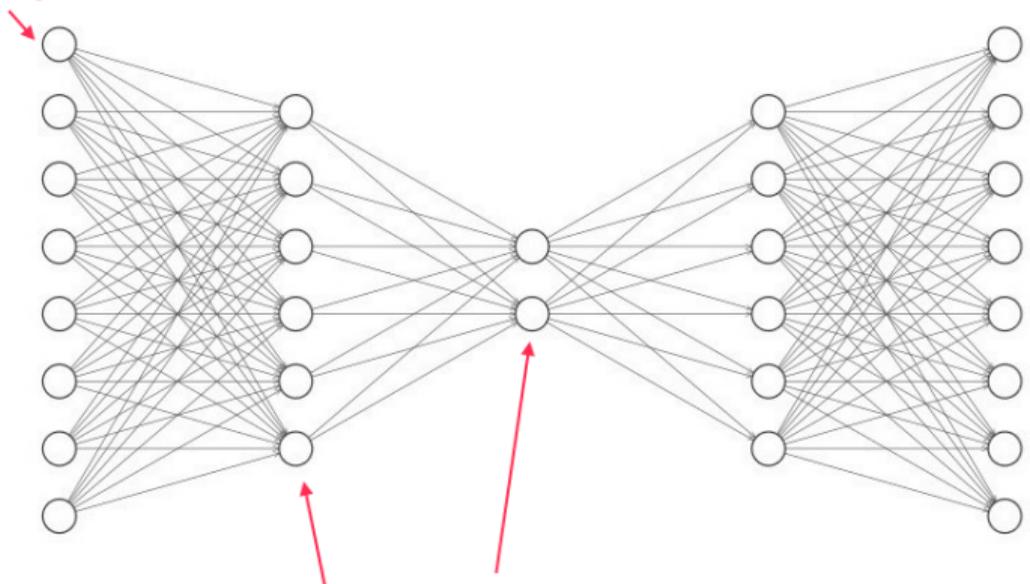
O Hajihassani, O Ardakanian, & H Khazaei. Latent representation learning and manipulation for privacy-preserving sensor data analytics. In The second Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML). IEEE, 2020.

Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, & Hamed Haddadi. Mobile sensor data anonymization. In Proceedings of the International Conference on Internet of Things Design and Implementation (IoTDI), pages 49–58. ACM, 2019



Privacidad mediante el uso de representaciones alternativas

Datos
originales



Mismos datos con representaciones alternativas

La reconstrucción de los datos originales no es posible si no se cuenta con el decodificador del autoencoder



DaSCI

Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

El cifrado y la ocultación de datos

Castells, M. (2003). Internet, libertad y sociedad: una perspectiva analítica. Polis. Revista Latinoamericana, (4).

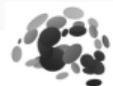
Radlo, E. J. (1997, February). Legal issues in cryptography. In International Conference on Financial Cryptography (pp. 259-286). Springer, Berlin, Heidelberg.

Criptografía con autoencoders

Sreelakshmi, K., & Ravi, R. V. (2020, July). An Encryption-then-Compression Scheme Using Autoencoder Based Image Compression for Color Images. In 2020 7th International Conference on Smart Structures and Systems (ICSSS) (pp. 1-5). IEEE.

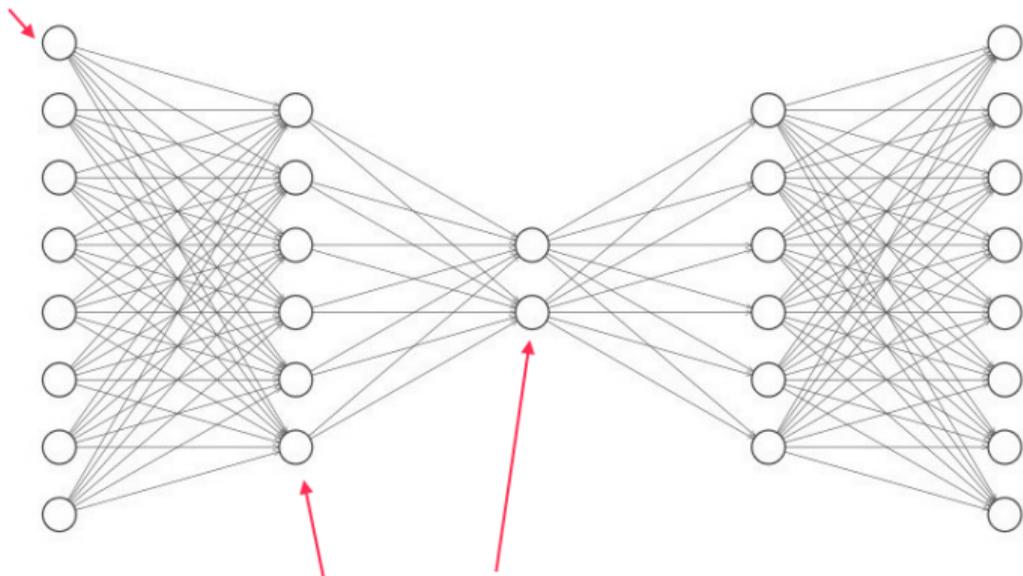
Suhail, K. A., & Sankar, S. (2020). Image Compression and Encryption Combining Autoencoder and Chaotic Logistic Map. Iranian Journal of Science and Technology, Transactions A: Science, 44(4), 1091-1100.

Quinga-Socasi, F., Velastegui, R., Zhinin-Vera, L., Valencia-Ramos, R., Ortega-Zamorano, F., & Chang, O. (2020). Digital Cryptography Implementation using Neurocomputational Model with Autoencoder Architecture. In ICAART (2) (pp. 865-872).



Mismo principio usado en la anonimización de datos

Datos
originales



Mismos datos con representaciones alternativas

La codificación puede ser de más longitud que el dato original (*overcomplete*) para reducir la pérdida en la reconstrucción.

La función de pérdida puede forzar que la codificación tenga una representación específica para reforzar el cifrado.



DaSCI

Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

Agrupamiento/recuperación de datos semánticamente similares

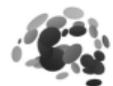
Análisis por métodos estadísticos

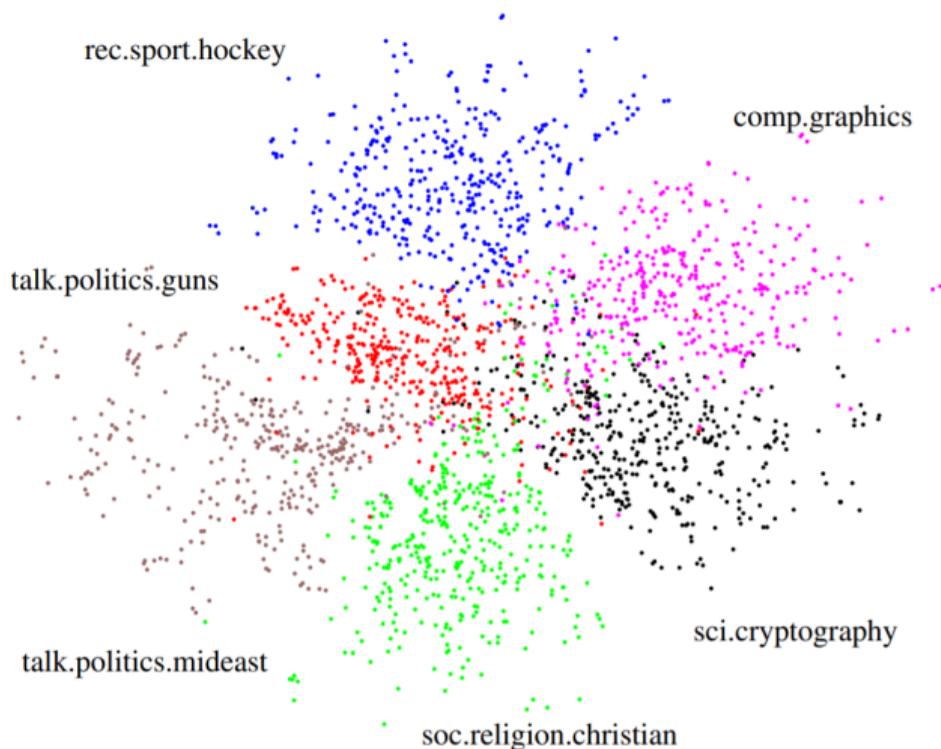
Albitar, S., Fournier, S., & Espinasse, B. (2014, October). An effective TF/IDF-based text-to-text semantic similarity measure for text classification. In International Conference on Web Information Systems Engineering (pp. 105-114). Springer, Cham.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988, May). Using latent semantic analysis to improve access to textual information. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 281-285).

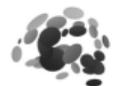
Hashing semántico con autoencoders

Krizhevsky, A., & Hinton, G. E. (2011, April). Using very deep autoencoders for content-based image retrieval. In ESANN (Vol. 1, p. 2).

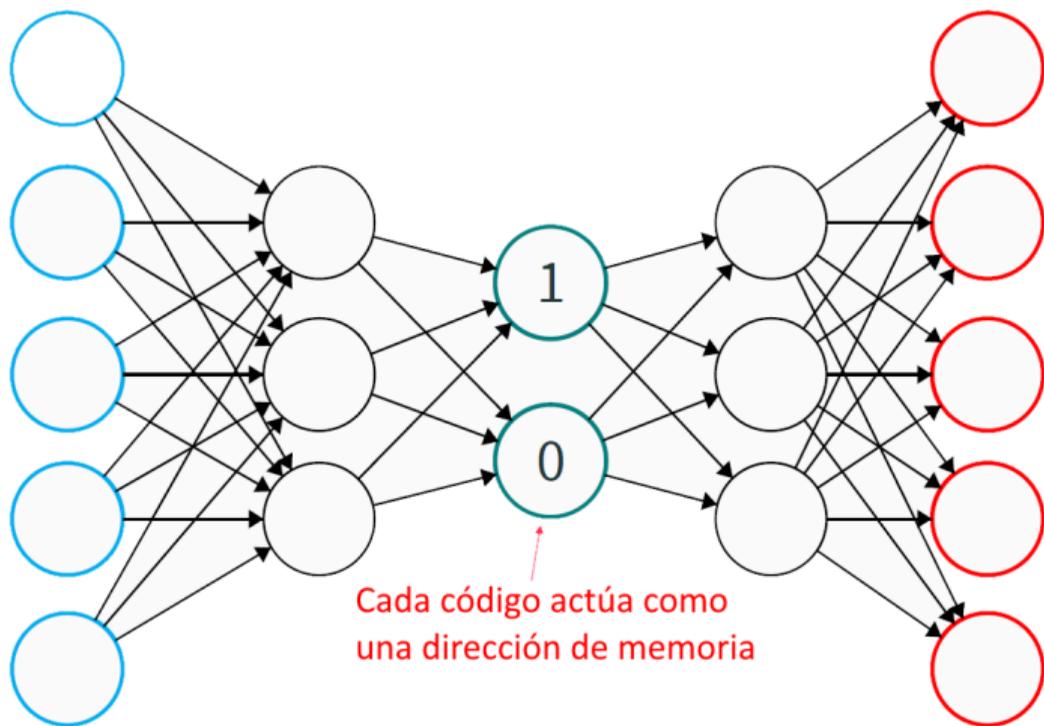




Salakhutdinov, R., & Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7), 969-978.



Cómo agrupar semánticamente con un autoencoder



Se fuerza una codificación binaria

- ◆ patrones similares generan el mismo código
- ◆ se asocian a un mismo grupo con semántica similar

Los datos sintéticos tienen multitud de aplicaciones

Datos no reales para entrenar modelos IA

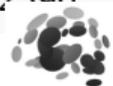
Dahmen, J., & Cook, D. (2019). SynSys: A synthetic data generation system for healthcare applications. *Sensors*, 19(5), 1181.

Lundin, E., Kvarnström, H., & Jonsson, E. (2002, December). A synthetic fraud data generation methodology. In *International Conference on Information and Communications Security* (pp. 265-277). Springer, Berlin, Heidelberg.

Generación de datos sintéticos con autoencoders

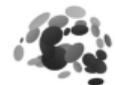
Lee, D., Yu, H., Jiang, X., Rogith, D., Gudala, M., Tejani, M., ... & Xiong, L. (2020). Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association*, 27(9), 1411-1419.

Xu, W., Keshmiri, S., & Wang, G. (2019). Adversarially approximated autoencoder for image generation and manipulation. *IEEE Transactions on Multimedia*, 21(9), 2387-2396



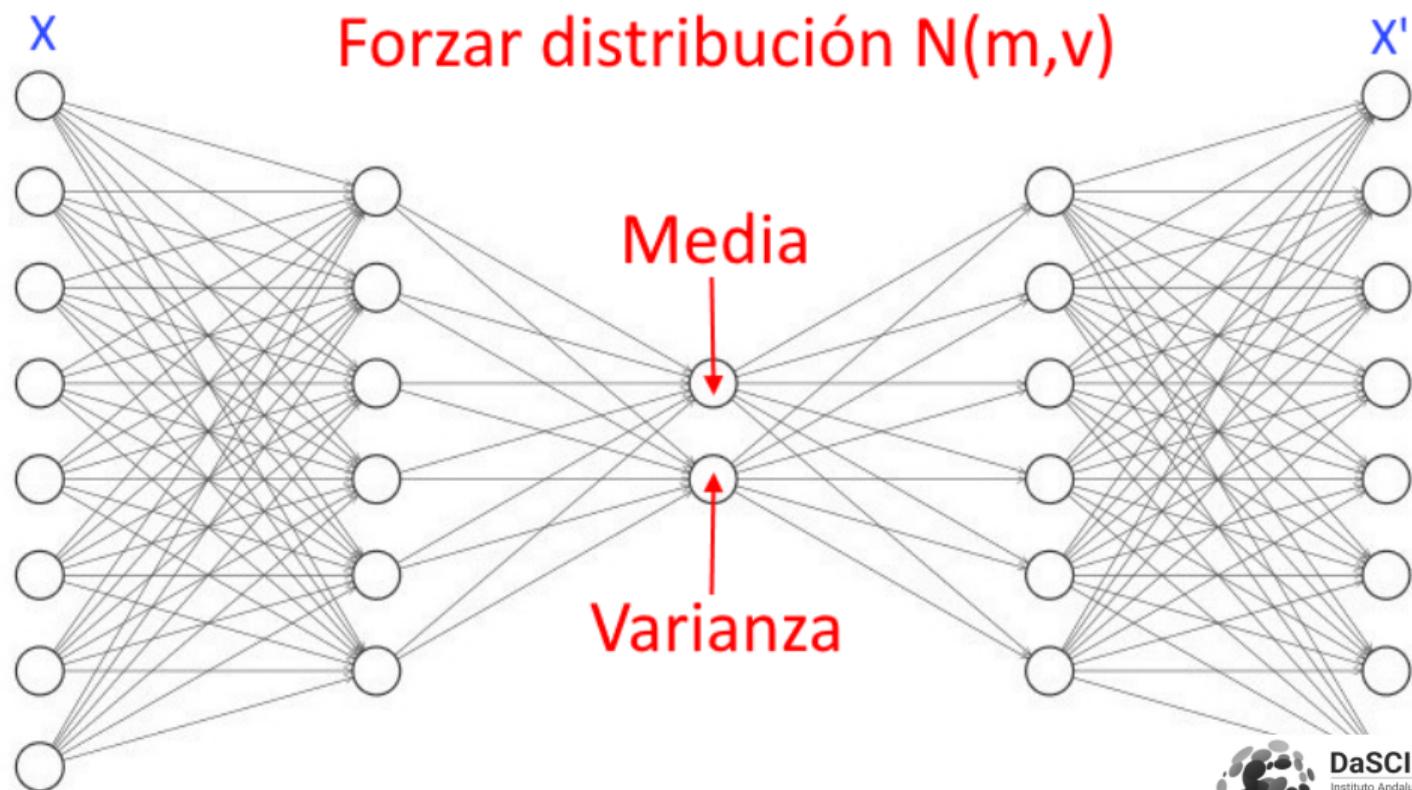


thispersondoesnotexist.com



DaSCI
Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

Cómo generar datos sintéticos con un autoencoder



Contenidos

¿Qué es un autoencoder?

Características básicas

Partes del autoencoder

Mecanismo de aprendizaje

¿Para qué sirven los autoencoders?

Reducción de dimensionalidad

Identificación de anomalías

Eliminación de ruido

Imputación de datos perdidos

Anonimización de datos

Criptografía y seguridad

Agrupamiento semántico

Generación de nuevos datos

¿Cómo funciona un autoencoder?

Mecanismo de autoaprendizaje

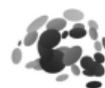
Funciones objetivo

Referencias útiles

Construcción de autoencoders

En Python - Usando Keras

En R - El paquete ruta



Codificador y decodificador son funciones

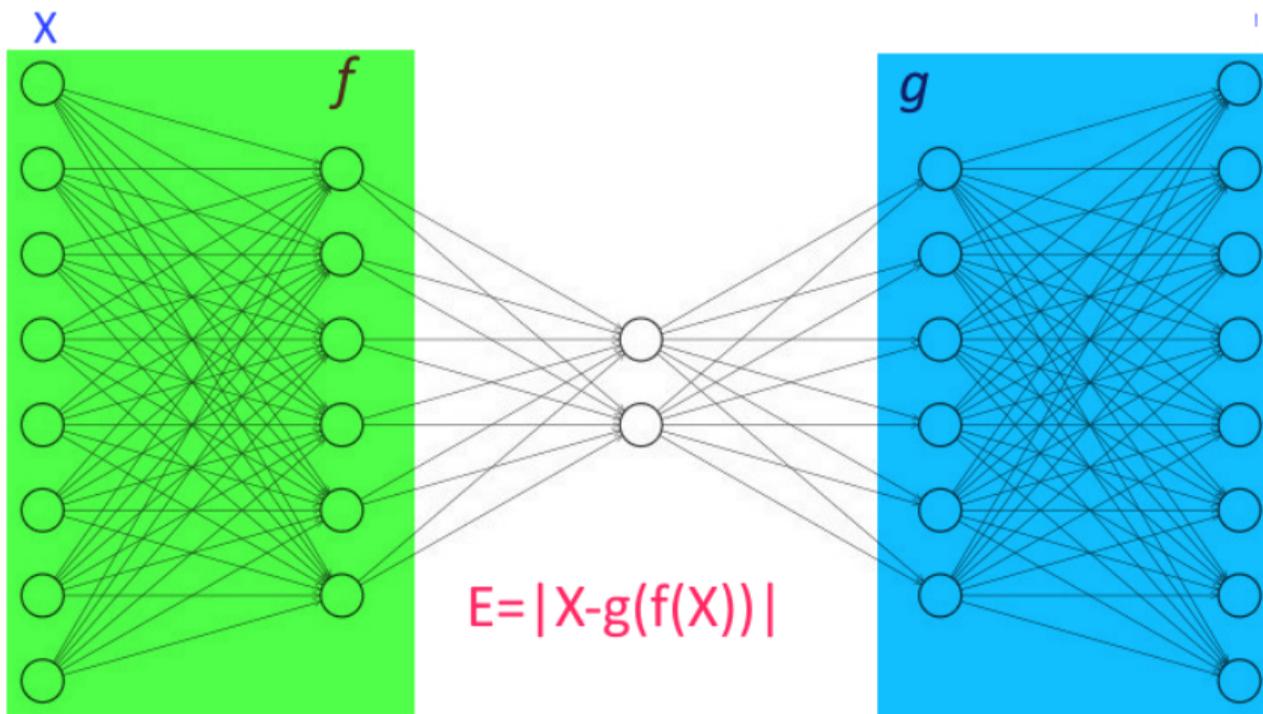
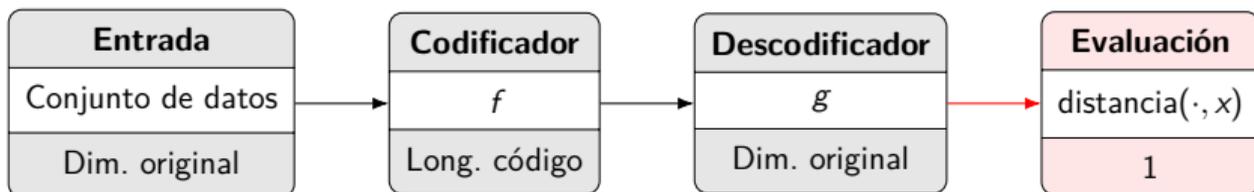
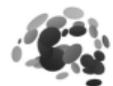


Diagrama del proceso



- ◆ Los pesos de la red almacenan la información general sobre la distribución de los datos de los que se ha aprendido: **estructura del subespacio** de menor dimensionalidad o *manifold*
- ◆ Cada código contiene la información más importante a nivel individual: **posición del dato** en el subespacio
- ◆ La función de error, en principio calculada como la distancia entre el dato original y el resultado de la reconstrucción, es la que rige el **ajuste de pesos** durante el proceso de *backpropagation*
- ◆ El objetivo es conseguir una **configuración de pesos que minimice la distancia** entre las muestras del conjunto de datos y sus reconstrucciones



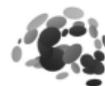
Función objetivo del autoencoder básico

Formalmente tenemos una función objetivo, dependiendo de cuya salida se ajustarán los pesos entre los nodos

$$\text{Objetivo} = \min_{\theta} \sum_{x \in \mathcal{X}} d(x, g_{\theta}(f_{\theta}(x)))$$

Siendo d una métrica de distancia adecuada, por ejemplo el RMSE, el proceso iterativo de entrenamiento permite encontrar una **configuración de pesos θ que genera la mejor reconstrucción** posible de los datos de entrada

La función objetivo puede forzar la obtención de representaciones con unas características concretas, dando lugar a **diferentes categorías de autoencoders**: dispersos, contractivos, variacionales, etc.



Variaciones sobre la función básica

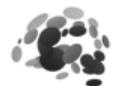
- ◆ **Denoising autoencoder.** La función $r(x) = x + \xi$ introduce en los datos ruido atendiendo a un cierto modelo

$$\text{Objetivo} = \min_{\theta} \sum_{x \in \mathcal{X}} d(x, g_{\theta}(f_{\theta}(r(x))))$$

- ◆ **Sparse autoencoder.** La función $r(\mathcal{Z}) = \sum_{j=1}^k (\rho - \rho_j)^2$ ajusta el número de nodos activos en el código según el parámetro ρ

$$\text{Objetivo} = \min_{\theta} \sum_{x \in \mathcal{X}} d(x, g_{\theta}(f_{\theta}(x))) + r(f_{\theta}(x))$$

- ◆ **Variational autoencoder.** La codificación modela una distribución normal de la que pueden muestrearse nuevas reconstrucciones



Taxonomía, tutorial y revisión

Charte, D., Charte, F., García, S., del Jesus, M. J., & Herrera, F. (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. Information Fusion.

Herramientas para crear autoencoders

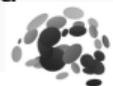
Charte, D., Herrera, F., & Charte, F. (2019). Ruta: implementations of neural autoencoders in R. Knowledge-Based Systems.

Usos de autoencoders para aprendizaje de representaciones

Charte, D., Charte, F., del Jesus, M. J., & Herrera, F. (2020). An analysis on the use of autoencoders for representation learning: Fundamentals, learning task case studies, explainability and challenges. Neurocomputing.

Cómo determinar la mejor arquitectura de un autoencoder

Charte, F., Rivera, A. J., Martínez, F., & del Jesus, M. J. (2020). EvoAAA: An evolutionary methodology for automated neural autoencoder architecture search. Integrated Computer-Aided Engineering.



Contenidos

¿Qué es un autoencoder?

- Características básicas

- Partes del autoencoder

- Mecanismo de aprendizaje

¿Para qué sirven los autoencoder?

- Reducción de dimensionalidad

- Identificación de anomalías

- Eliminación de ruido

- Imputación de datos perdidos

- Anonimización de datos

- Criptografía y seguridad

- Agrupamiento semántico

- Generación de nuevos datos

¿Cómo funciona un autoencoder?

- Mecanismo de autoaprendizaje

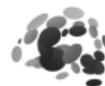
- Funciones objetivo

- Referencias útiles

Construcción de autoencoders

- En Python - Usando Keras

- En R - El paquete ruta



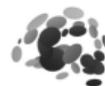
Construcción como cualquier otra red en Keras

<https://colab.research.google.com/drive/13ZUy6A6ouJoDxen2UBAbR8lrzu4MMGeY?usp=sharing>

```
>>> import keras
>>> from keras import layers

>>> entrada = keras.Input(shape=(784,))
>>> codificación = layers.Dense(32, activation='relu')(entrada)
>>> salida = layers.Dense(784, activation='sigmoid')(codificación)

>>> autoencoder = keras.Model(entrada, salida)
```



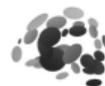
Construcción como cualquier otra red en Keras

```
>>> keras.utils.plot_model(autoencoder, show_shapes=True, show_layer_names=False)
```

InputLayer	input:	[(None, 784)]
	output:	[(None, 784)]

Dense	input:	(None, 784)
	output:	(None, 32)

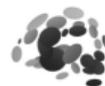
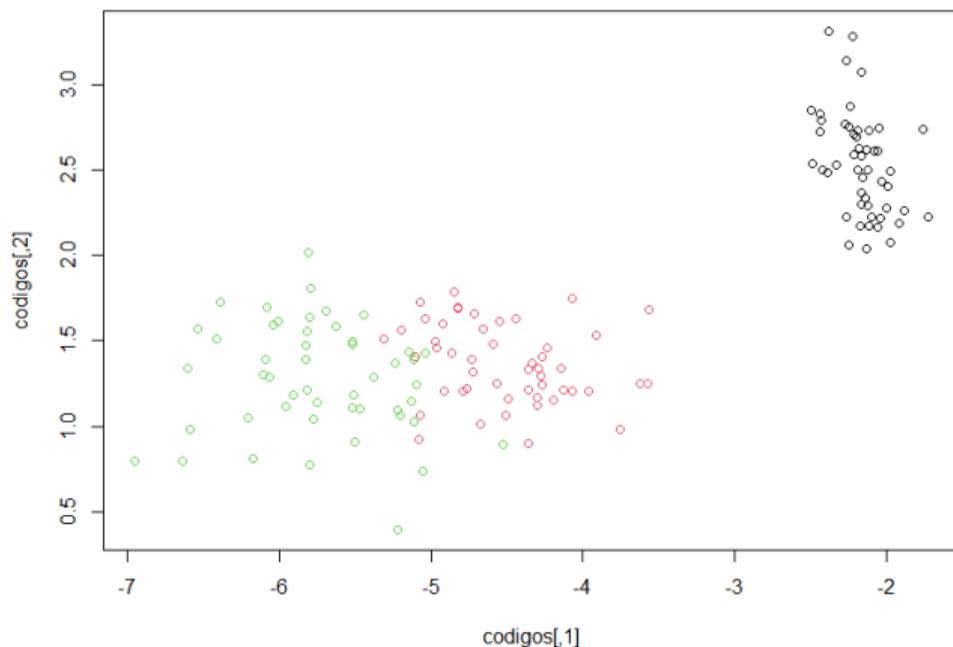
Dense	input:	(None, 32)
	output:	(None, 784)



Paquete R que usa Keras y automatiza el proceso

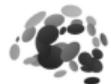
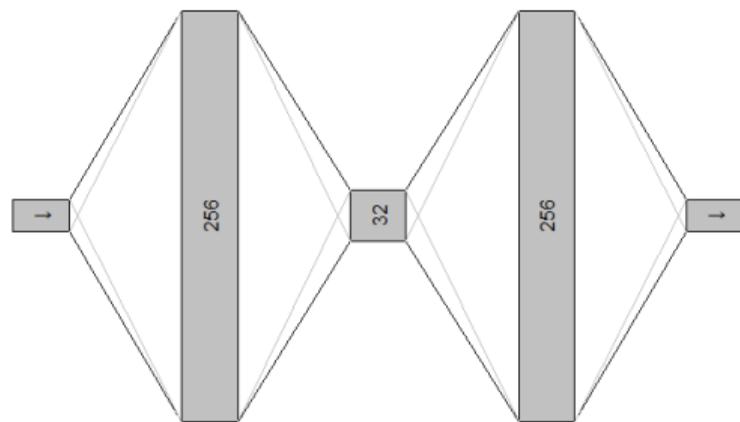
<https://ruta.software/>

```
> library(ruta)
>
> codigos <- iris[, 1:4] %>%
+   as.matrix %>%
+   autoencode(2, type = "denoising")
>
> plot(codigos, col = iris$Species)
```



Implementaciones de los tipos más habituales de autoencoder

```
> ae <- autoencoder_robust(
+   input() +
+   dense(256, activation = "relu") +
+   dense(32, activation = "relu") +
+   dense(256, activation = "relu") +
+   output("linear"))
>
> plot(ae$network)
>
> train(ae,
+   x_train,
+   epochs = 5,
+   batch_size = 512,
+   optimizer = "adam",
+   validation_data = x_test,
+   metrics = list("binary_accuracy"))
```



Ejemplos adicionales de autoencoders, tanto en R usando el paquete ruta como en Python empleando Keras, en los siguientes repositorios:

◆ **Autoencoder case studies**

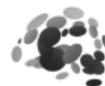
<https://github.com/ari-dasci/S-autoencoder-case-studies>

◆ **Autoencoder showcase**

<https://github.com/ari-dasci/S-autoencoder-showcase>

◆ **Paquete ruta**

<https://github.com/fdavidcl/ruta>





DaSCI

Instituto Andaluz de Investigación en
Data Science and Computational Intelligence

UJa Universidad
de Jaén

Autoencoders

¿Qué son, para qué sirven y cómo funcionan?

Francisco Charte

fcharte@ujaen.es

27 enero 2021