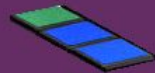


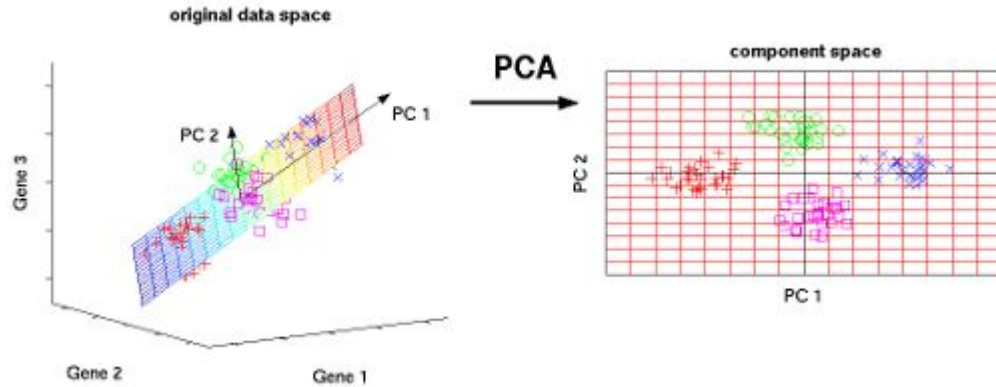
Reducing the Dimensionality of Data with Neural Networks



Por qué?

- Clasificación
- Visualización
- Comunicación
- Almacenamiento

Principal Components Analysis (PCA)

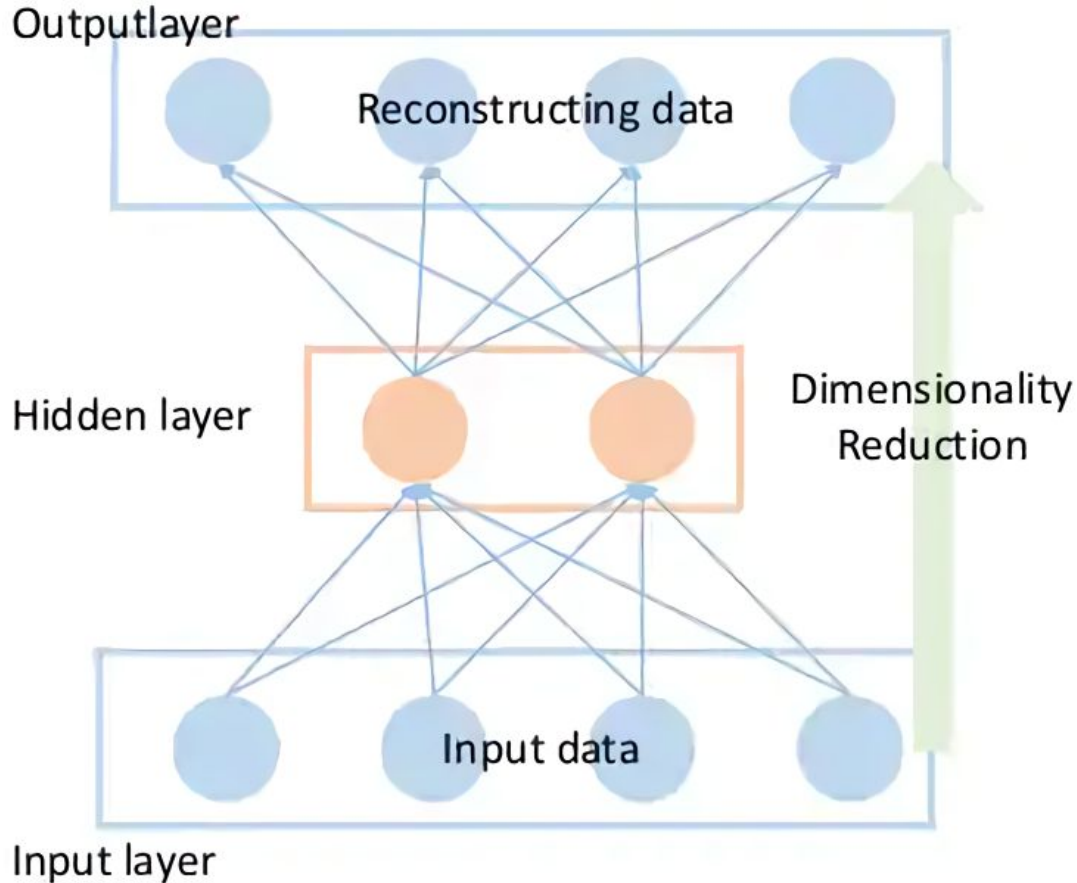


- Método simple y ampliamente usado.
- Encuentra las direcciones de la mayor variancia del data set.
- Representa cada punto de los datos por sus coordenadas a lo largo de estas direcciones.

Idea a presentar

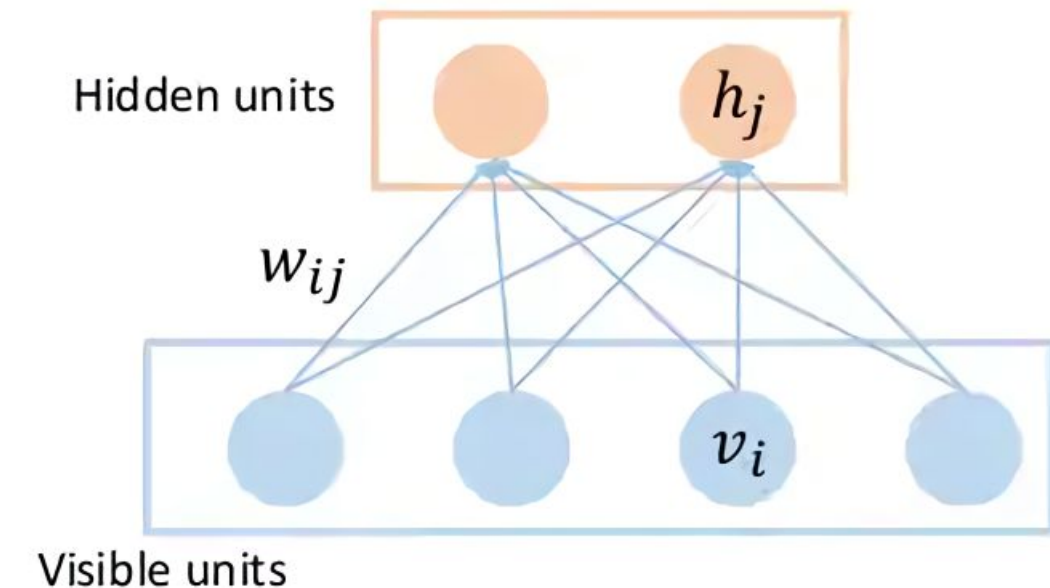
- Generalización no lineal de PCA a partir de una red neuronal
- Método de inicialización de pesos

Autoencoder



- Son entrenados reduciendo la discrepancia entre los datos originales y su reconstrucción.
- Los gradientes son obtenidos por la regla de la cadena para propagar el error desde la red decoder hasta la red encoder.

Restricted Boltzman Machine (RBM)



$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i \in \text{pixels}} b_i v_i - \sum_{j \in \text{features}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

- Los datos de input corresponden a las “visible units” de la RBM y los feature detectors corresponden a las “hidden units”.
- La configuración conjunta (\mathbf{v}, \mathbf{h}) de las “visible units” y “hidden units” tiene una energía dada por la fórmula en (1).
- La red le asigna una probabilidad a cada dato posible con esta fórmula.

Pretraining

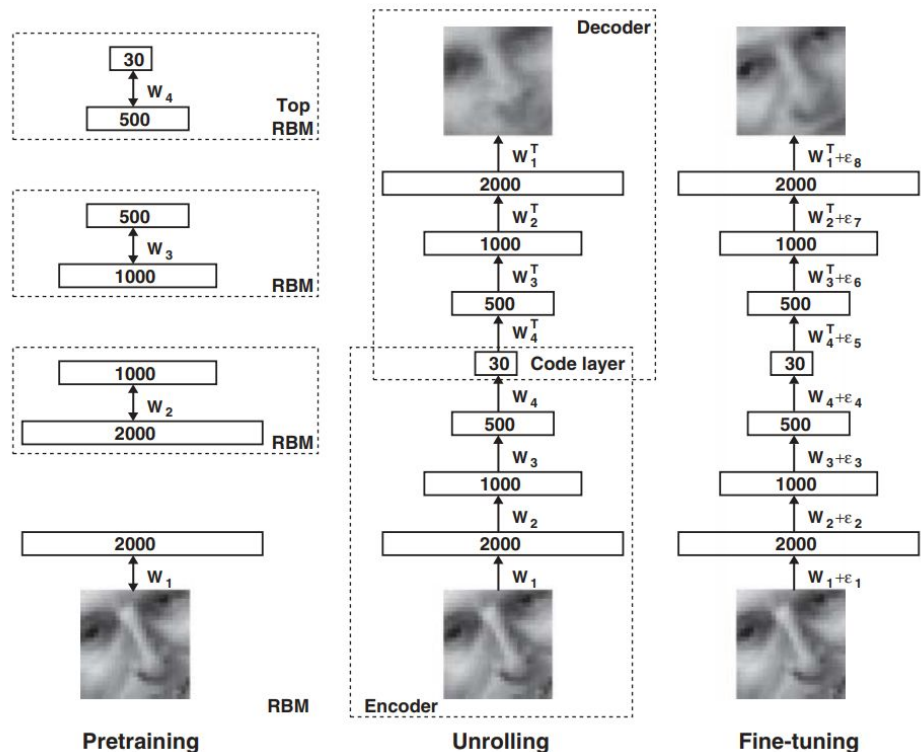


Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the “data” for training the next RBM in the stack. After the pretraining, the RBMs are “unrolled” to create a deep autoencoder, which is then fine-tuned using backpropagation of error derivatives.

“Greedy layer-wise pretraining”

La primer capa de feature detectors se vuelve las unidades visibles para aprender la próxima RBM.

Este aprendizaje capa a capa puede ser repetido cuantas veces querramos.

Ejemplos: Curvas

Fig. 2. (A) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by “logistic PCA” (8) using six components; reconstructions by logistic PCA and standard PCA using 18 components. The average squared error per image for the last four rows is 1.44, 7.64, 2.45, 5.90.



Ejemplos: Números escritos a mano y caras

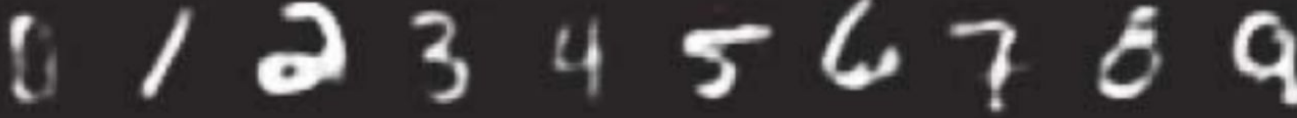
B.1



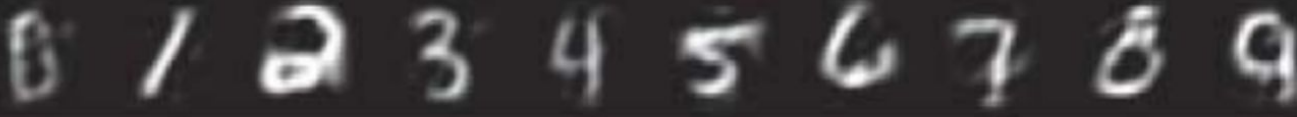
B.2



B.3



B.4



(B) Top to bottom: A random test image from each class; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional logistic PCA and standard PCA. The average squared errors for the last three rows are 3.00, 8.01, and 13.87.

(C) Top to bottom: Random samples from the test data set; reconstructions by the 30-dimensional autoencoder; reconstructions by 30-dimensional PCA. The average squared errors are 126 and 135.

C.1



C.2

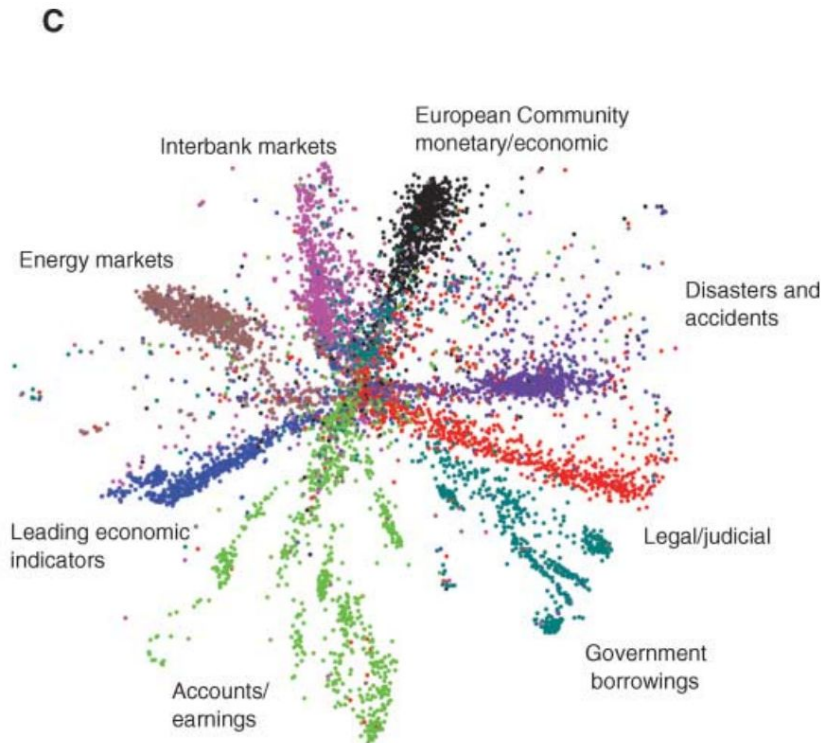
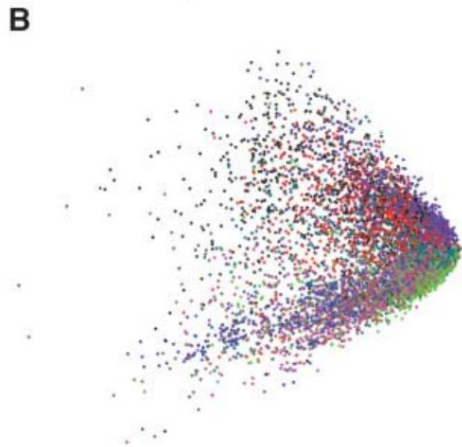
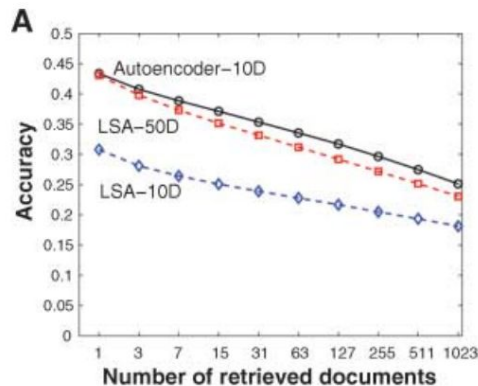


C.3



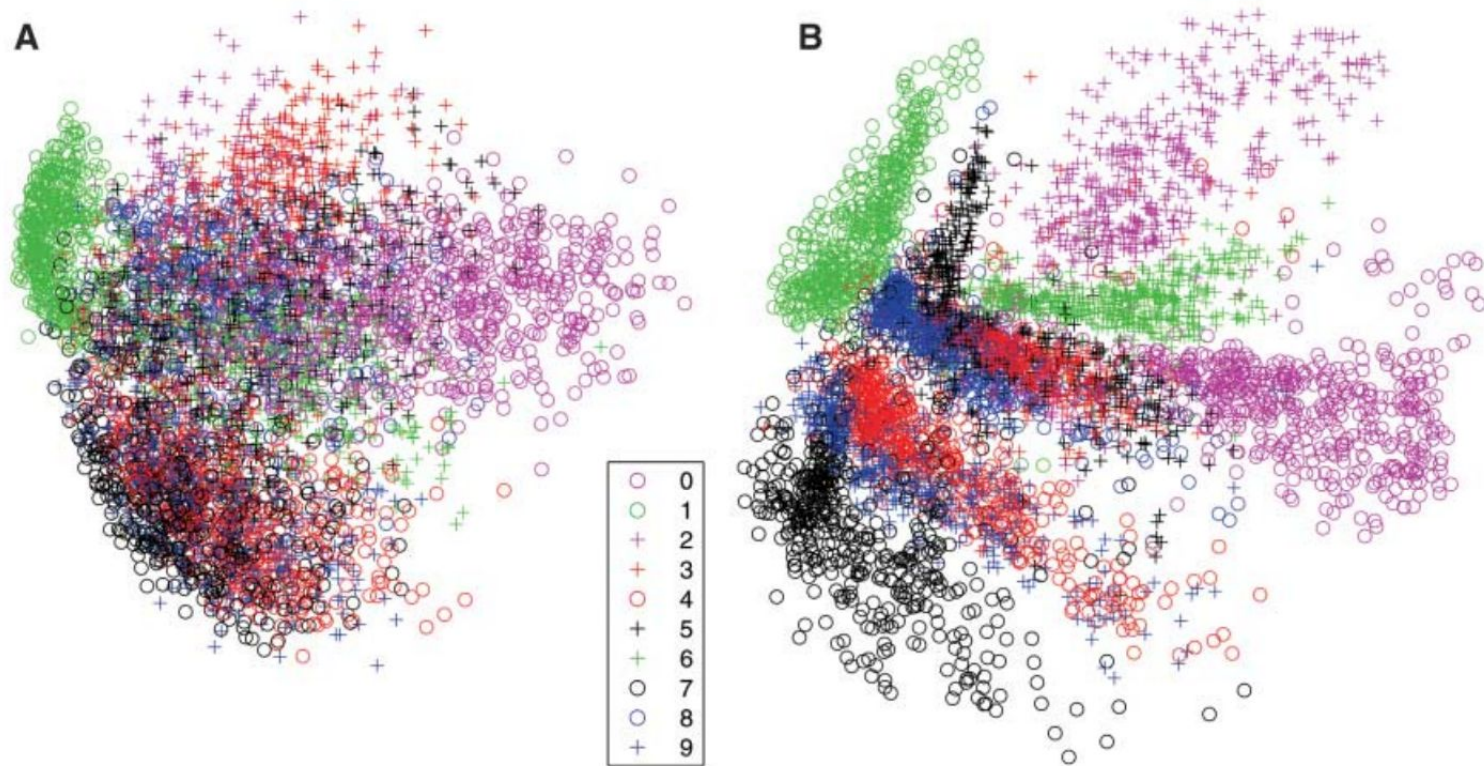
Ejemplos: Documentos

Fig. 4. (A) The fraction of retrieved documents in the same class as the query when a query document from the test set is used to retrieve other test set documents, averaged over all 402,207 possible queries. (B) The codes produced by two-dimensional LSA. (C) The codes produced by a 2000-500-250-125-2 autoencoder.



Ejemplo: Clasificación de dígitos

Fig. 3. (A) The two-dimensional codes for 500 digits of each class produced by taking the first two principal components of all 60,000 training images. (B) The two-dimensional codes found by a 784-1000-500-250-2 autoencoder. For an alternative visualization, see (8).



Conclusión