Unsupervised Learning Algorithms

Francesco Pugliese, PhD

neural1977@gmail.com

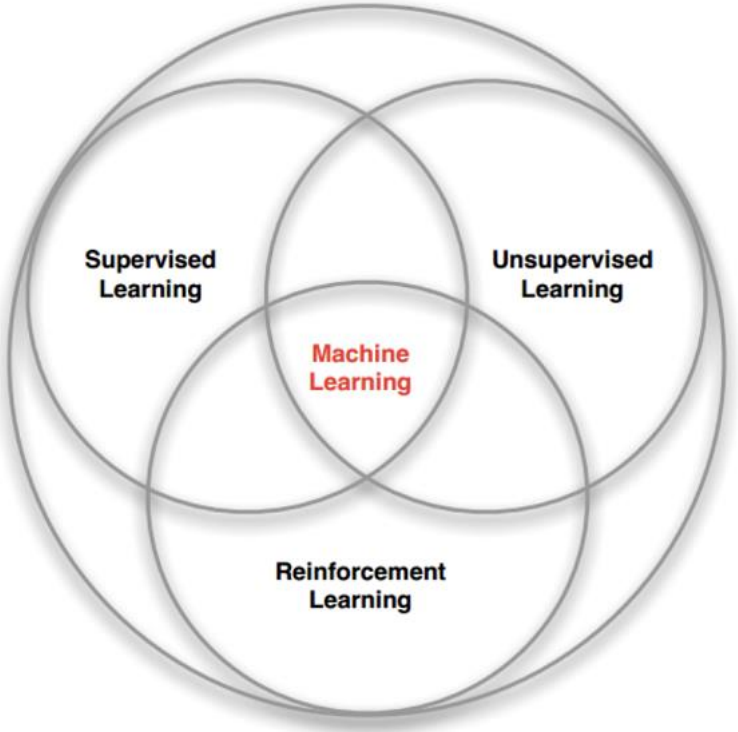
We will discuss about…

- ✓ Unsupervised Learning
- ✓ K-Means
- ✓ Principle component analysis (PCA)
- ✓ Autoencoders

Unsupervised Learning

- ✓ **Unsupervised learning** is one of the 3 fundamental ways that Machine Learning "learns" data: Supervised Learning, Reinforcement Learning, Unsupervised Learning. Unsupervised learning is characterized by "Unlabelled" datasets, and its algorithm must try to make sense of on it.

- ✓ Most of Data on the **Web** is Unsupervised, this is because Unsupervise Learning is so important.

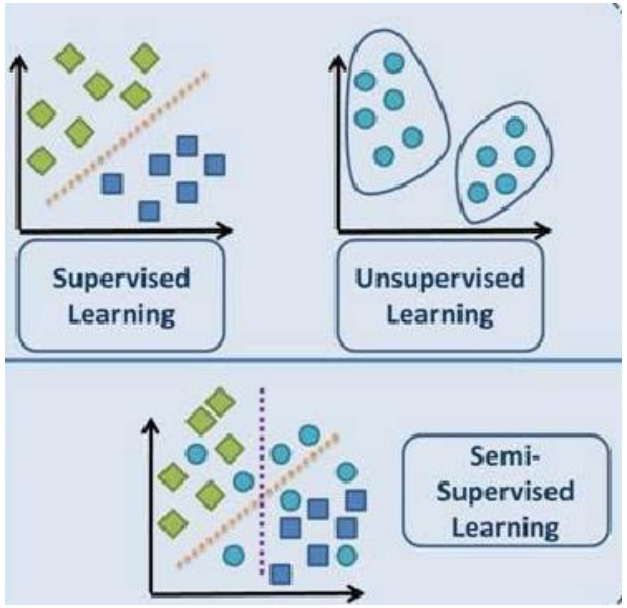


Unsupervised Learning

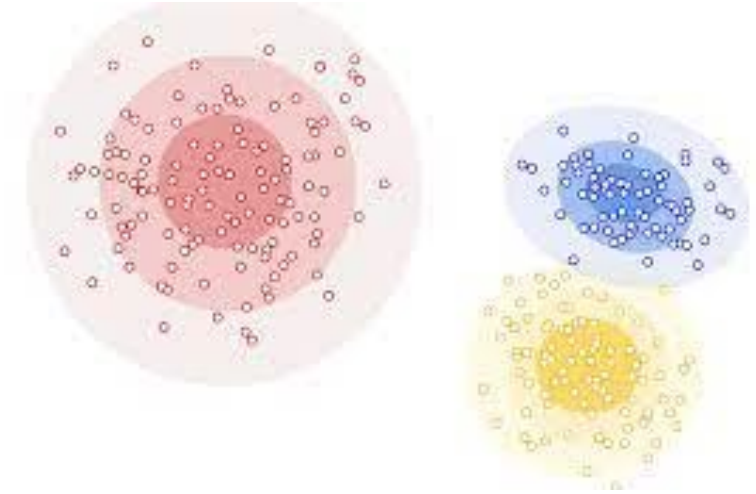
- ✓ The purpose is to exploit mimicry, which is an important mode of learning in people, to make machines able top to build a compact internal representation of its world and then generate imaginative content from it.

- ✓ Generally, Unsupervised methods exhibit self-organizing behaviours capturing patterns as probability densities or a combination of neural feature preferences.

- ✓ A variation of Unsupervised Learning is the **Semi-Supervised Learning** where a smaller portion of the data is labelled.

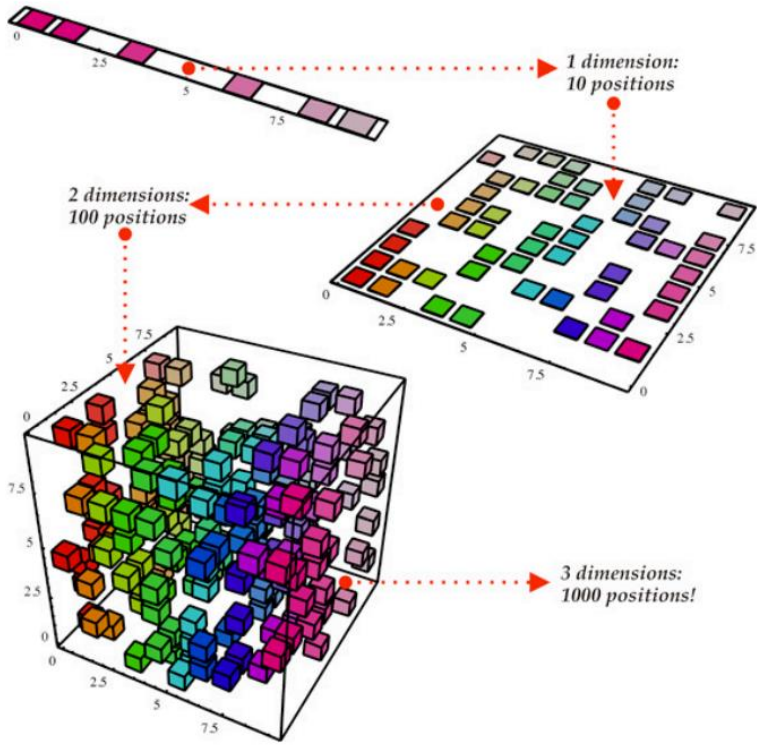


Unsupervised Learning



✓ The principal Tasks of Unsupervised Learning are:

1) **Clustering**: Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects belonging to the same group (**cluster**) are more similar (in some sense) to each other than to those in other groups (clusters). Clustering is one of the best method for the **Exploratory Data Analysis (EDA)**.



2) **Dimensionality Reduction**: Dimensionality Reduction minimizes the data inputs into smaller dimensional data without loosing the integrity of data, it is useful when dataset has a high value density. This technique is pretty much used to prevent overfitting. One of the most important method for Dimensionality Reduction is the **Principal Component Analysis (PCA)**.

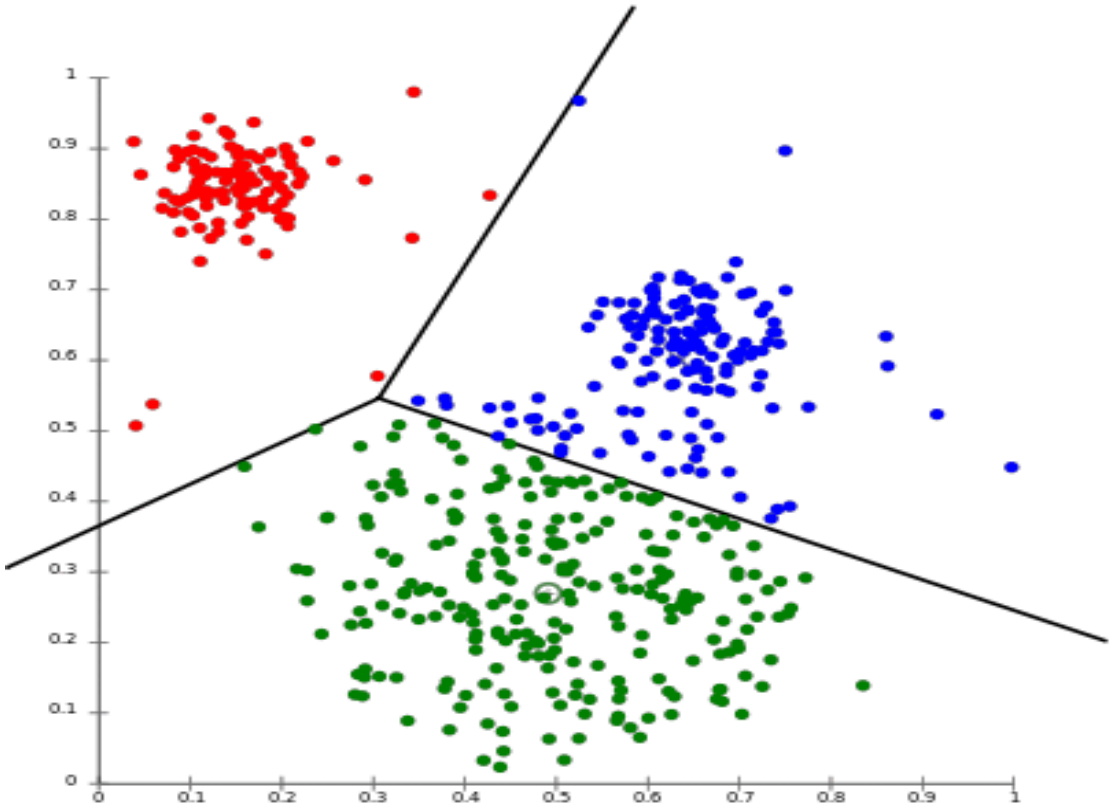
K-Means

- ✓ K-means algorithm identifies **k number of centroids**, and then allocates every data point to the **nearest cluster**, while keeping the centroids as small as possible.
- ✓ The 'means' in the K-means refers to **averaging** of the data; that is, **finding the centroid**.
- ✓ K-means algorithm starts with a first group of **randomly** selected centroids, which are used as the **beginning** points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids.
- ✓ It halts creating and optimizing clusters when either the centroids have **stabilized** or a defined number of **iterations** have been achieved.

K-Means Algorithm

- ✓ Specify the number of clusters K.
- ✓ Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement
- ✓ Keep iterating until the centroids are stabilized
- ✓ Compute the sum of the squared distance between data points and all centroids
- ✓ Assign each data point to the closest cluster (centroid)
- ✓ Compute the centroids for the clusters by taking the average of the data points that belong to each cluster.

K-Means



kNN vs K-Means

	k-NN	k-Means
Type	Supervised	Unsupervised
Meaning of k	Number of closest neighbors to look at	Number of centroids
Calculation of prediction error	Yes	No
Optimization done using	Cross validation, and confusion matrix	Elbow method, silhoutte method
Convergence	When all observations classified at the desired accuracy	When cluster memberships don't change anymore
Complexity	Train: O(d) Test: O(nd) Where: d: Dimensions/features n: Number of observations	O(nkId) Where: n: Number of points k: Number of clusters I: Number of iterations d: Number of attributes

Applications of K-Means

- ✓ Identifying fake news
- ✓ Spam detection and filtering
- ✓ Classify books or movies by genre
- ✓ Popular transport routes while town planning

Principle component analysis

- ✓ PCA is an unsupervised, statistical technique primarily used for **dimensionality reduction** by feature extraction in machine learning.
- ✓ When we talk about high-dimensionality, it means that the dataset has a large number of features. and that requires a large amount of memory and computational power.
- ✓ PCA uses **orthogonal** transformation which converts a set of correlated variables to a set of uncorrelated variables.
- ✓ It is used to explain the variance-co variance structure of a set of variables through linear combinations.
- ✓ It is a also the most widely used tool in exploratory data analysis and predictive modeling.
- ✓ The idea behind PCA is simply to find a **low-dimension** set of axes that **summarize** data.

Principle component analysis

- ✓ Say for example, we have a dataset composed by a set of car properties; size, color, number of seats, number of doors, size of trunk, circularity, compactness, radius… However, many of these features will indicate the same result and therefore can be redundant.

- ✓ We as smart technologists should try to remove these redundancies and describe each car with **fewer** properties, making the **computation** simple. This is exactly what PCA aims to do.

- ✓ PCA does not take information of attributes into account. It concerns itself with the **variance** of each attribute because the **presence of high variance would indicate a good split between classes**, and that's how we reduce the dimensionality.

- ✓ PCA never just considers some while discards others. It takes the attributes into account **statistically**.

Applications of PCA

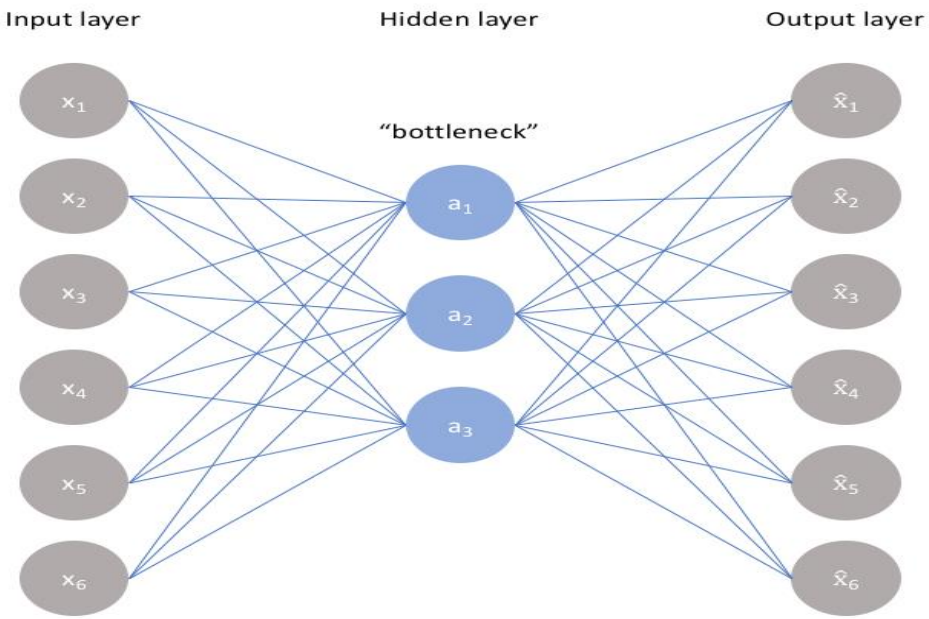
- ✓ Optimize power allocation in multiple communication channels
- ✓ Image Processing
- ✓ Movie recommendation system

Autoencoders

- ✓ **Autoencoders** are an unsupervised learning technique belonging to the field of representation learning and then they became to belong to Deep Learning with the rise of **Stacked Autoencoders**.

- ✓ Nowadays, Autoencoders are the unique solution of Deep Learning for the discriminative Unsupervised Learning (there is also a generative field), where from 2006 we had also **Deep Belief Networks (DBN)** from one of Deep Learning Godfathers, Hinton. DBN fell into disuse during the last decade.

- ✓ With Autoencoders we leverage the power of Artificial Neural Networks for the task of **compressing knowledge** representation of the original input. Namely, we impose a **bottleneck** layer where extract the compressed information.



Autoencoders

- ✓ If the input features are independent of one another, the compression and subsequent reconstruction would be a very difficult task.

- ✓ However, if a sort of structure exists within the data (ie. correlations between input features), this structure can be learned and consequently leveraged when forcing the input through the network's bottleneck.

- ✓ Therefore, **Autoencoding** is a data compression algorithm where the compression and decompression functions are:
    1) Data-specific: they only compress data similar to what they have trained on;
    2) Lossy: decompressed outputs will be degraded with regard to the inputs
    3) Learned automatically from examples rather than engineered by a human.

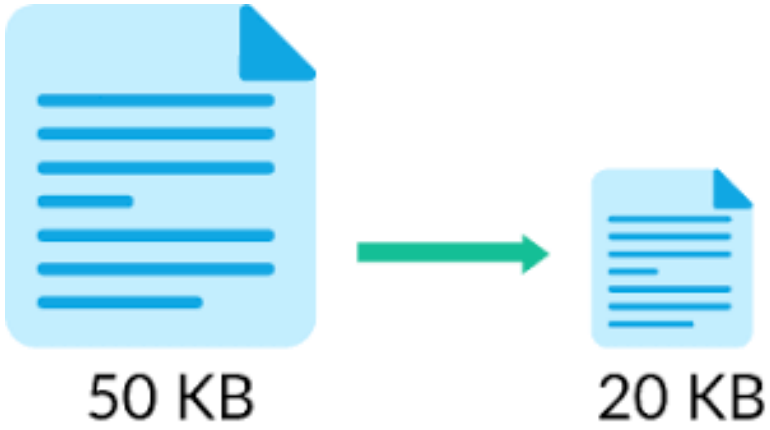
- ✓ Compression and decompression are implemented by Neural Networks.

Autoencoders

- ✓ To build an autoencoder, you need three things:
  1) an **encoding function**;
  2) a **decoding function**;
  3) a **loss function**, namely a distance function between the amount of information loss between the compressed representation of your data and the decompressed representation.

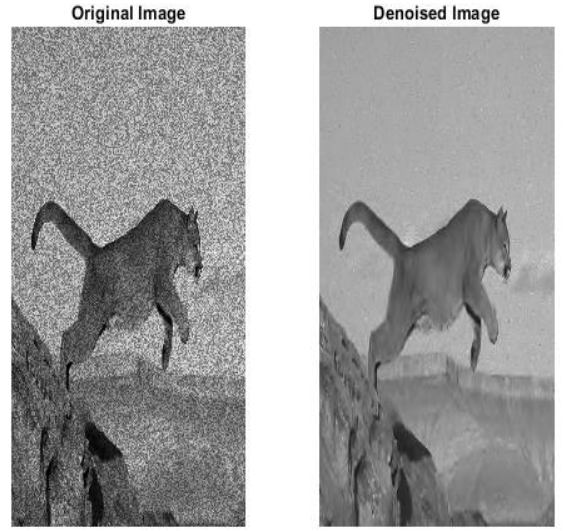
- ✓ However, in **Picture Compression**, it is difficult to train an autoencoder outperforming the JPEG compression.

- ✓ Since **Autoencoders** are data-specific makes them impractical for real-world data compression problems: you can only use them on data that is similar to what they were trained on, and making them more general thus requires lots of training data.



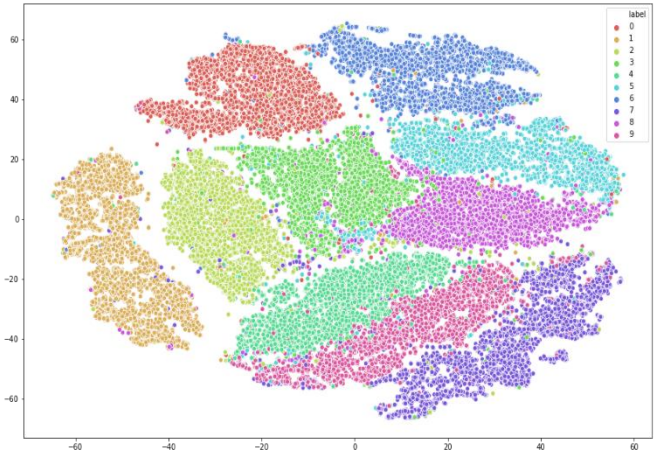
Applications of Autoencoders

- ✓ **Greedy layer-wise pretraining** for deep convolutional neural networks: but this quickly fell out of fashion as we started realizing that better random weight initialization schemes were sufficient for training deep networks from scratch. In **2014**, batch normalization started allowing for even deeper networks, and from late **2015** we could train arbitrarily deep networks from scratch using residual learning.



- ✓ **Data denoising**: We can an autoencoder to map noisy images to clean digit images. Here Autoencoder can substitute classical **Wavelets**.

- ✓ **Dimensionality Reduction for Data Visualization**: With appropriate dimensionality and sparsity constraints, autoencoders can learn data projections that are more interesting than PCA or other basic techniques. For 2D visualization specifically, t-SNE (pronounced "tee-snee") is probably the best algorithm around, but it typically requires relatively low-dimensional data. So a good strategy for visualizing similarity relationships in high-dimensional data is to start by using an autoencoder to compress your data into a low-dimensional space (e.g. 32-dimensional), then use t-SNE for mapping the compressed data to a 2D plane. Scikit-Learn has a simple and practical implementation of t-SNE.



References

- ✓ https://towardsdatascience.com/top-10-algorithms-for-machine-learning-beginners-149374935f3c

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, *9*(8), 1295.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433-459.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433-459.

Francesco **Pugliese**