Unsupervised Learning

Unsupervised Learning

- Supervised learning: The **transfer of existing knowledge**, for example from human experts to a neural network.
- Unsupervised learning: The **discovery of new knowledge** from raw data.
  - New knowledge/information: Existence of categories, relations, rules, etc.
  - There is no “correct answer” or “ground truth”. The objective is to produce results that are reasonable and informative.
- We will start with some discussion on “**clustering**”, the main research and application field in unsupervised learning.

What is Clustering?

In cluster analysis, a group of objects is split up into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity, such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups.

- Backer & Jain, 1981

Clustering as Partition of Data

- Let X be the set of all the data to be clustered.
- A cluster is a non-empty subset of X. it can be considered a set containing all the items (data points, feature vectors, samples, patterns, etc.) in it.
- A ***clustering*** or a ***partition*** of X is a set containing all the clusters extracted from X by some clustering process.
- A few new issues to consider:
  - Desired shapes or distributions of the clusters
  - Whether the clusters contain all the data points
  - Whether overlaps between clusters are allowed
  - Whether the number of clusters is known beforehand
  - …

Proximity Measure

- Since our goal is to group "similar" patterns into clusters, we need a measure of how similar (or conversely, dissimilar) two items are.
- The choice of proximity measure directly affects the "shape" of clusters in the feature space.
- Proximity between two clusters can be defined using the proximity between their members.
- The most widely used proximity measure is some form of **distance measure**, such as Euclidean.

Cluster Representatives

It is often useful to represent a cluster by a few parameters (instead of a list of all its members).

The most common form is a point representative. Examples:

mean point $\boldsymbol{m}_p$:

$$\boldsymbol{m}_p = \frac{1}{n_C} \sum_{\boldsymbol{y} \in C} \boldsymbol{y}$$

medoid $\boldsymbol{m}_c$ (one of the vectors in the cluster):

$$\boldsymbol{m}_c = \arg\max_{\boldsymbol{z} \in C} \sum_{\boldsymbol{y} \in C} s(\boldsymbol{y}, \boldsymbol{z})$$

similarity measure

k-Means

- The most popular and widely used clustering algorithm.
- An example of **competitive learning**, a main type of unsupervised learning methods.
- The number of clusters is predetermined (the "*k*").
- Cluster representatives (prototypes) are required. The standard form uses “mean points” (the “*means*”).
- There are many variations:
  - Non-point prototypes
  - Partially overlapping clusters (soft partitions)
  - etc.

k-Means

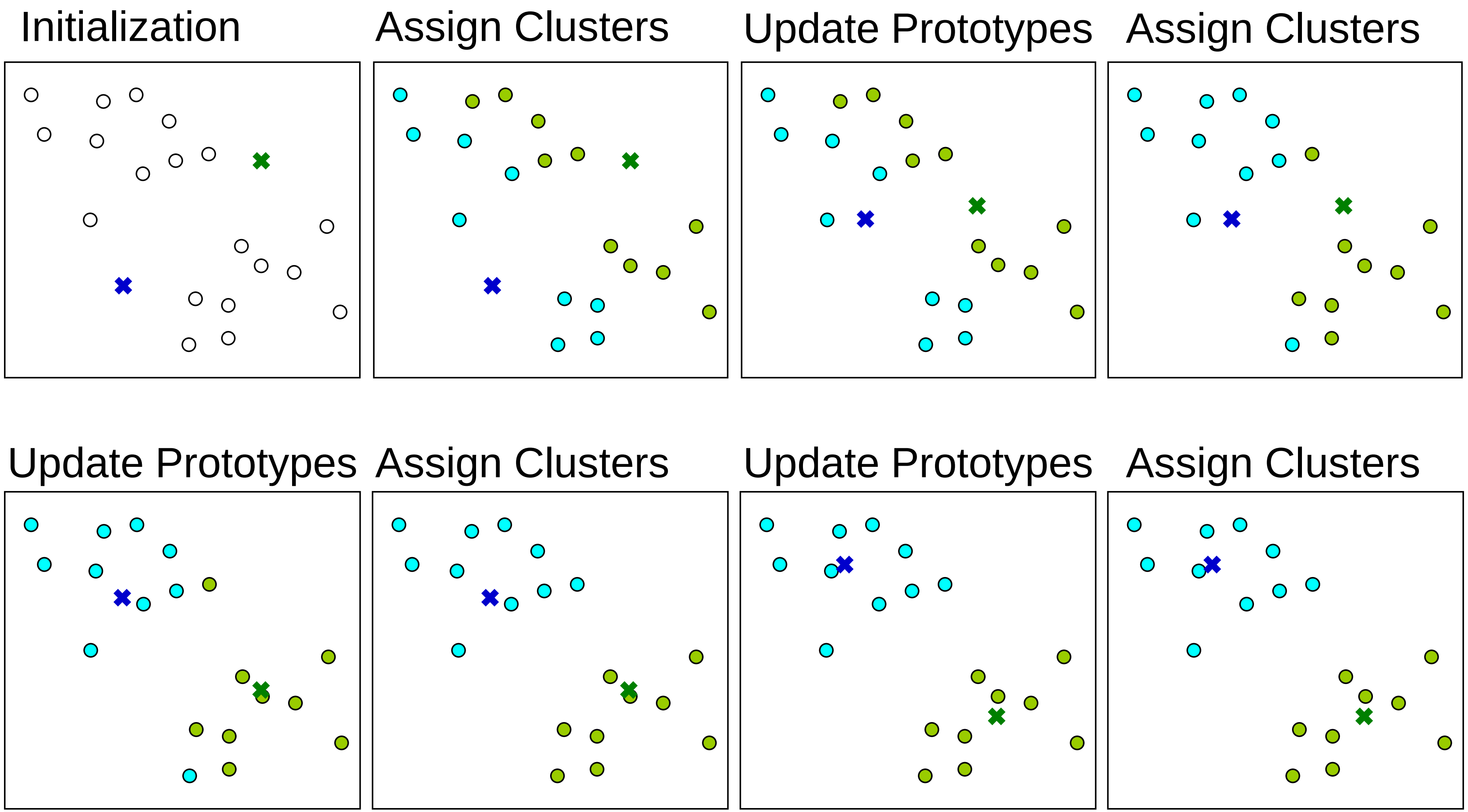
Each sample belongs to exactly one cluster

Algorithm:

- (Randomly) initialize the prototypes
- Repeat until "stopping criteria"
  - Assign each sample to the cluster of the closest prototype (the *competitive* part)
  - Recalculate each prototype as the mean of all $\boldsymbol{x}$ belonging to that cluster

Common stopping criterion: no more change of prototypes (or cluster assignments) between iterations.

k-Means Example



Initialization

Assign Clusters

Update Prototypes

Assign Clusters

Update Prototypes

Assign Clusters

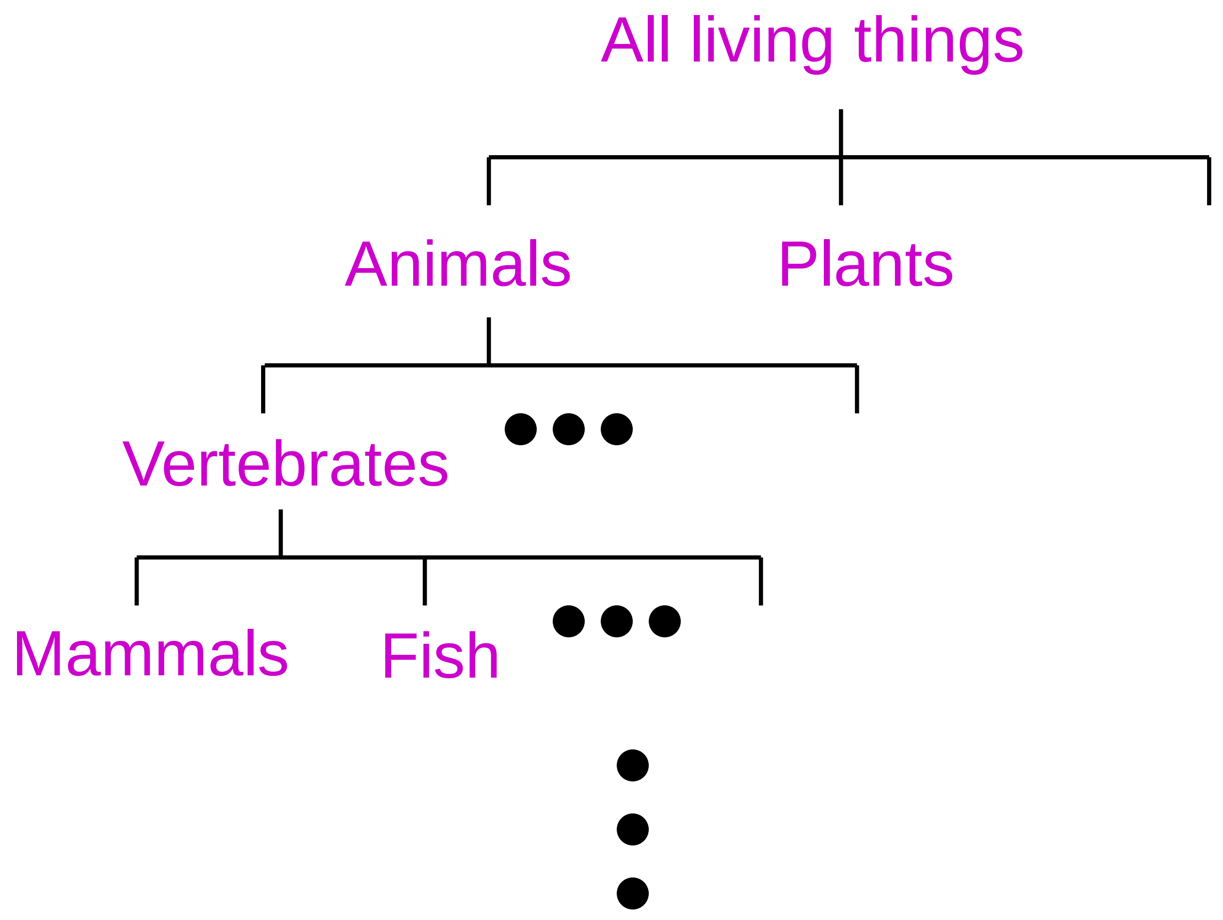
Update Prototypes

Assign Clusters

No change ➔ Stop!

Hierarchy Clustering

Hierarchical clustering algorithms create a hierarchical organization of all the samples. Each cut (horizontally) gives a different partition of the data.


All living things

Animals

Plants

Vertebrates

Mammals

Fish

Relational Data Clustering

Hierarchical clustering is the most common approach for clustering **relational data**:

- The only information used is the "relations" (similarities or distances) among the samples.
- No need to represent the samples with vectors of attributes.
- Relations represented as a matrix or a graph.
- Particularly useful in applications where it is difficult or meaningless to use factored representation. Examples:
  - Biology (see the previous slide)
  - Document, music, etc.
  - Network structure (e.g., social network)

Hierarchical Algorithms

Two main types of hierarchical clustering algorithms:

- **Agglomerative Clustering** Algorithms:
  - Bottom-up
  - Initial state: One cluster for each data point
  - Each iteration: A pair of clusters are merged
- **Divisive Clustering** Algorithms:
  - Top-down
  - Initial state: One single cluster containing all data points
  - Each iteration: One of the clusters is divided into two

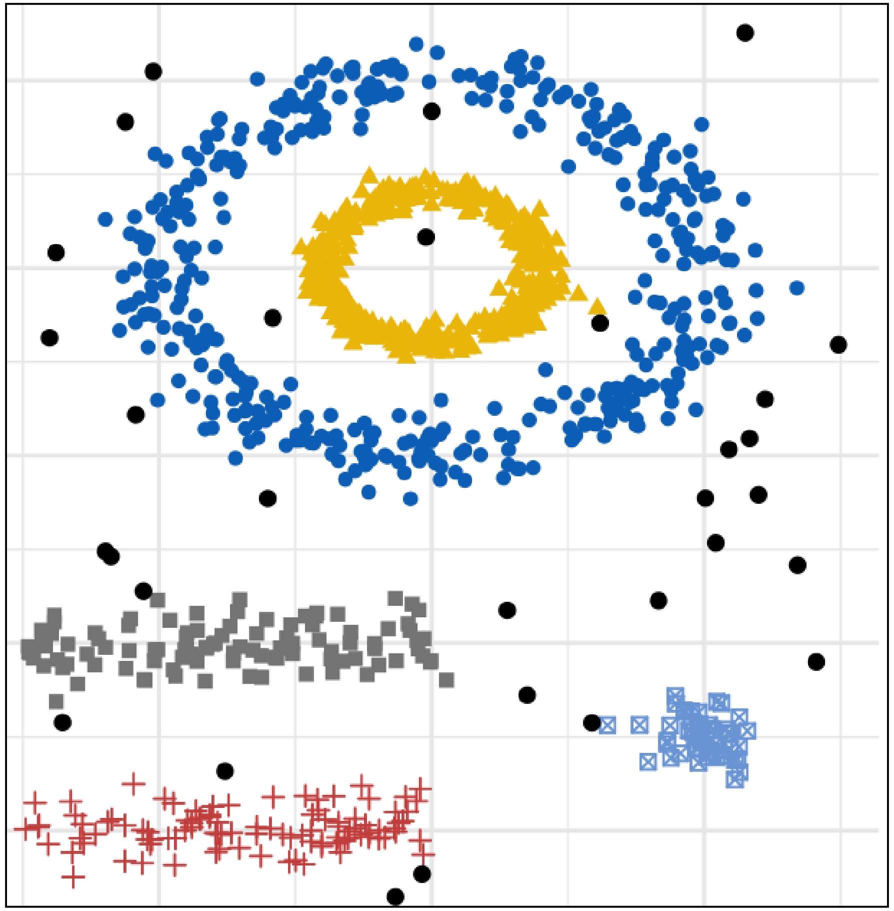
More Clustering Approaches

There are a large number of different approaches used in clustering. Several major categories are:

- Mode-seeking (e.g., mean-shift): Iterative update of cluster representatives so that they move toward local maximums of data density.
- Mixture decomposition: Approximate the data distribution with the weighted combination of several parametric distributions.
- Density-based (e.g., DBSCAN): Use local data density for the decision of whether to put samples in the same cluster.
- Graph-cut (e.g., spectral clustering): Represent the similarities among the samples as a weighted graph, then find a “cut” of the graph.
- Many more …

More Clustering Approaches

- **DBSCAN** has become quite popular given its robustness to noise and flexibility of cluster shapes.
- Basic ideas:
  - Identify data points with sufficiently high local density as core points.
  - Connect the core points to form clusters.
  - Non-core points are assigned to their nearest clusters.

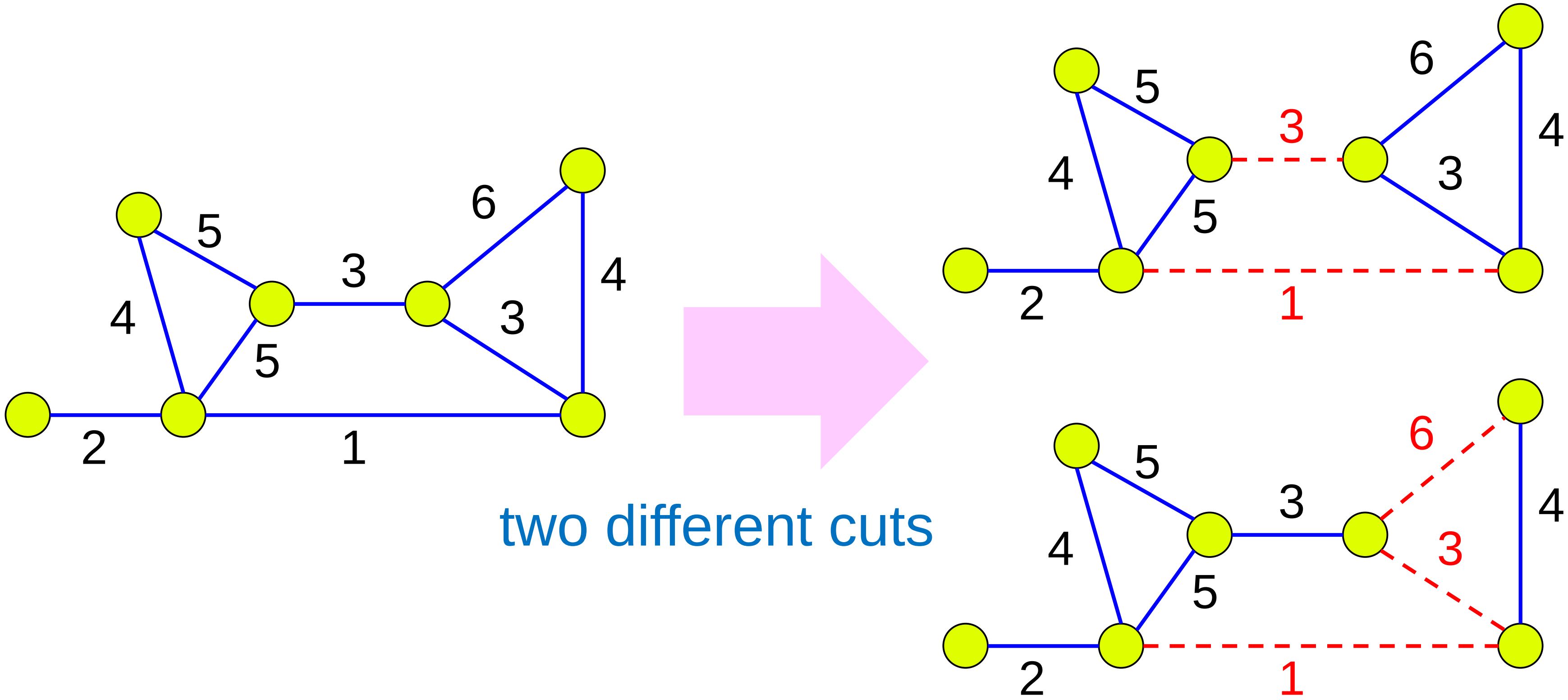


www.datanovia.com

More Clustering Approaches

- **Graph-cut** based clustering: Treat the samples to be clustered as a weighted graph. Use a similarity measure between samples as the weights.

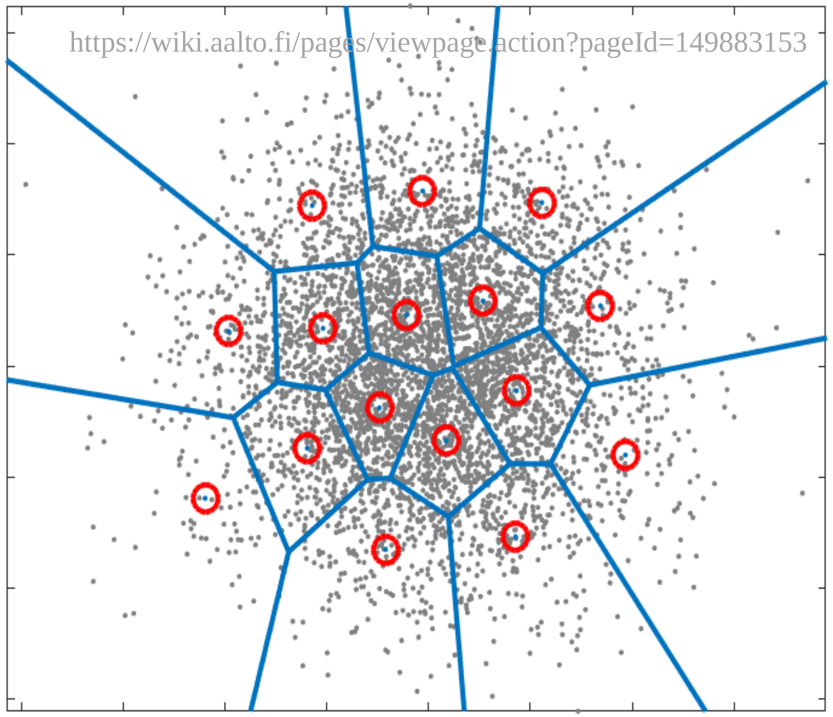
- Idea: To cut the graph into two sub-graphs while minimizing the total weights of the removed edges.



two different cuts

Vector Quantization (VQ)

- The algorithm looks like an online version of k-means; the prototypes become the code vectors.
- The purpose is not to find cluster structures, but to code a large number of samples more compactly.
- The most common application is lossy compression. (It's a part of many earlier image codecs.)



https://wiki.aalto.fi/pages/viewpage.action?pageId=149883153

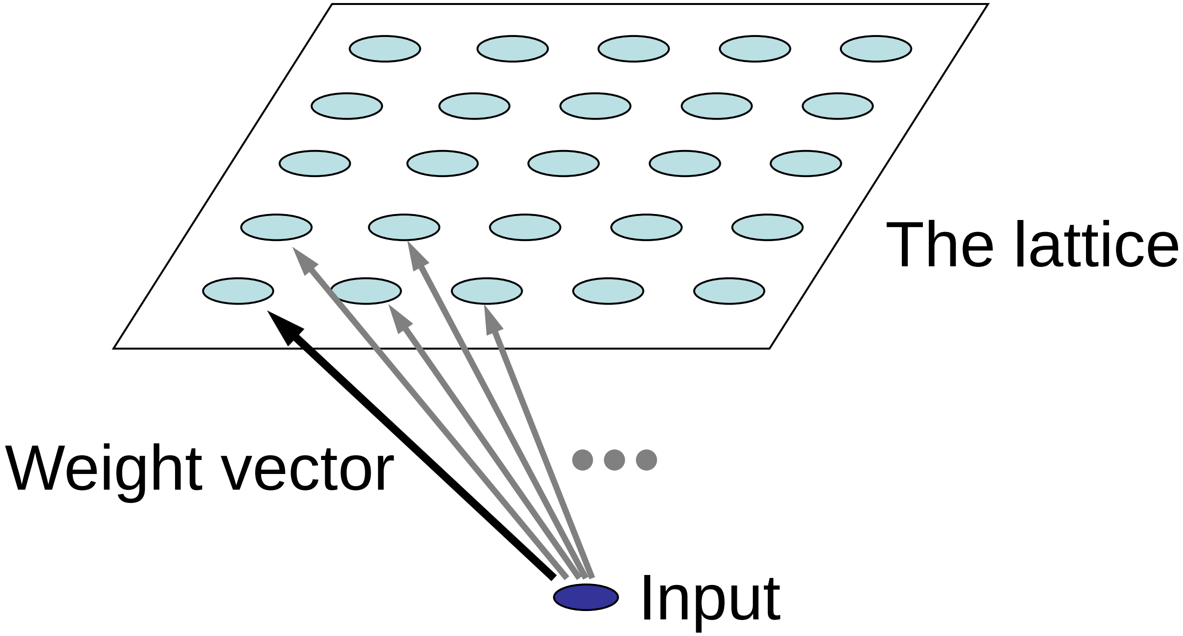
- A fixed number of randomly initialized vector representatives.
- For each step:
  - Select an input sample.
  - Move the nearest representative towards it (with a learning rate).
- Repeat until convergence.

Dimensionality Reduction

- Another important subfield of unsupervised learning is dimensionality reduction, which is to generate more compact (lower-dimensional) representations of samples than in their original space.
- Principle component analysis (PCA) discussed along with supervised learning is an important technique of dimensionality reduction.
- Dimensionality reduction leads to a lossy representation, so it’s important to preserve as much information useful for the task as possible.

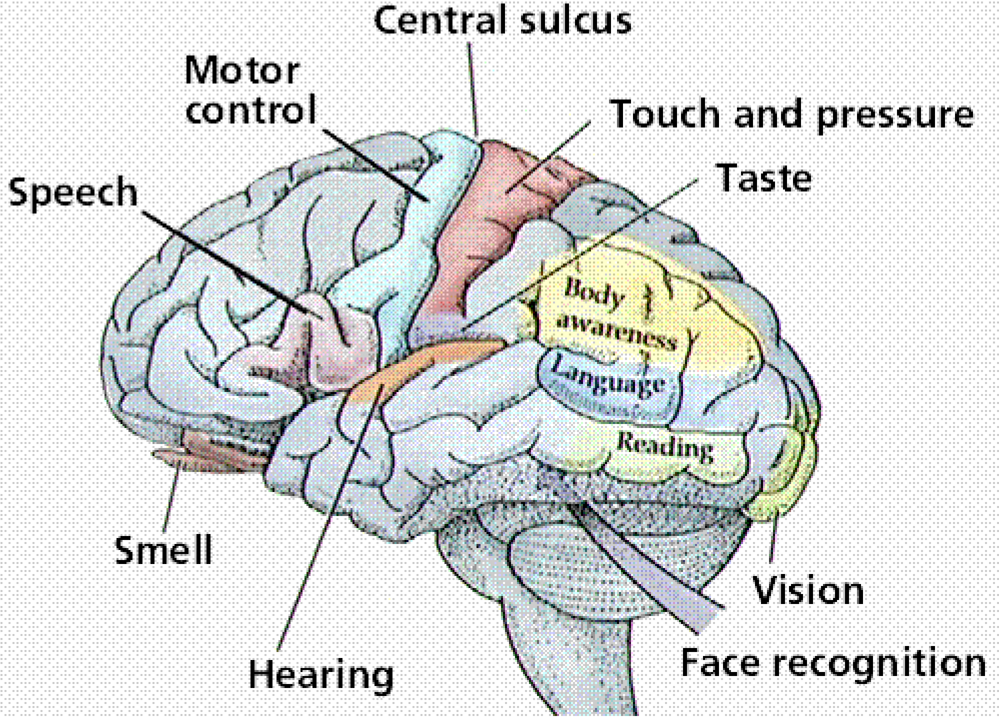
Self-Organizing Maps (SOM)

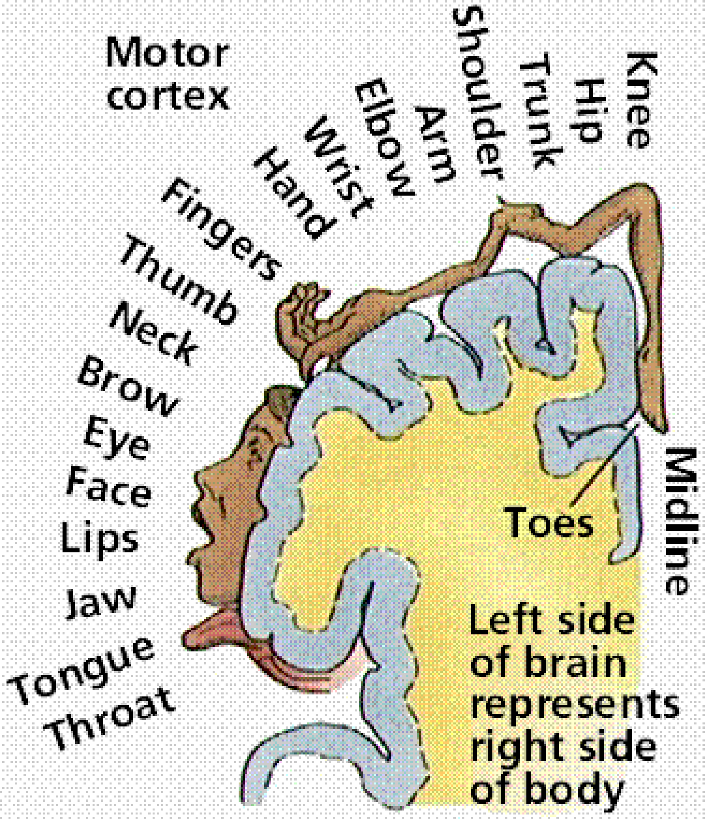
- A SOM has a single layer of neurons arranged on a lattice (2-D is most common).
- The weight vector of each neuron represents a point in the feature space.
- Nearby neurons on the lattice should behave similarly (have similar responses to inputs). Overall, the resulting lattice gives an organized representation of the inputs.



Biological Origin of SOM

SOM is inspired by how the information processing in a human brain is distributed and organized at the cerebral cortex:



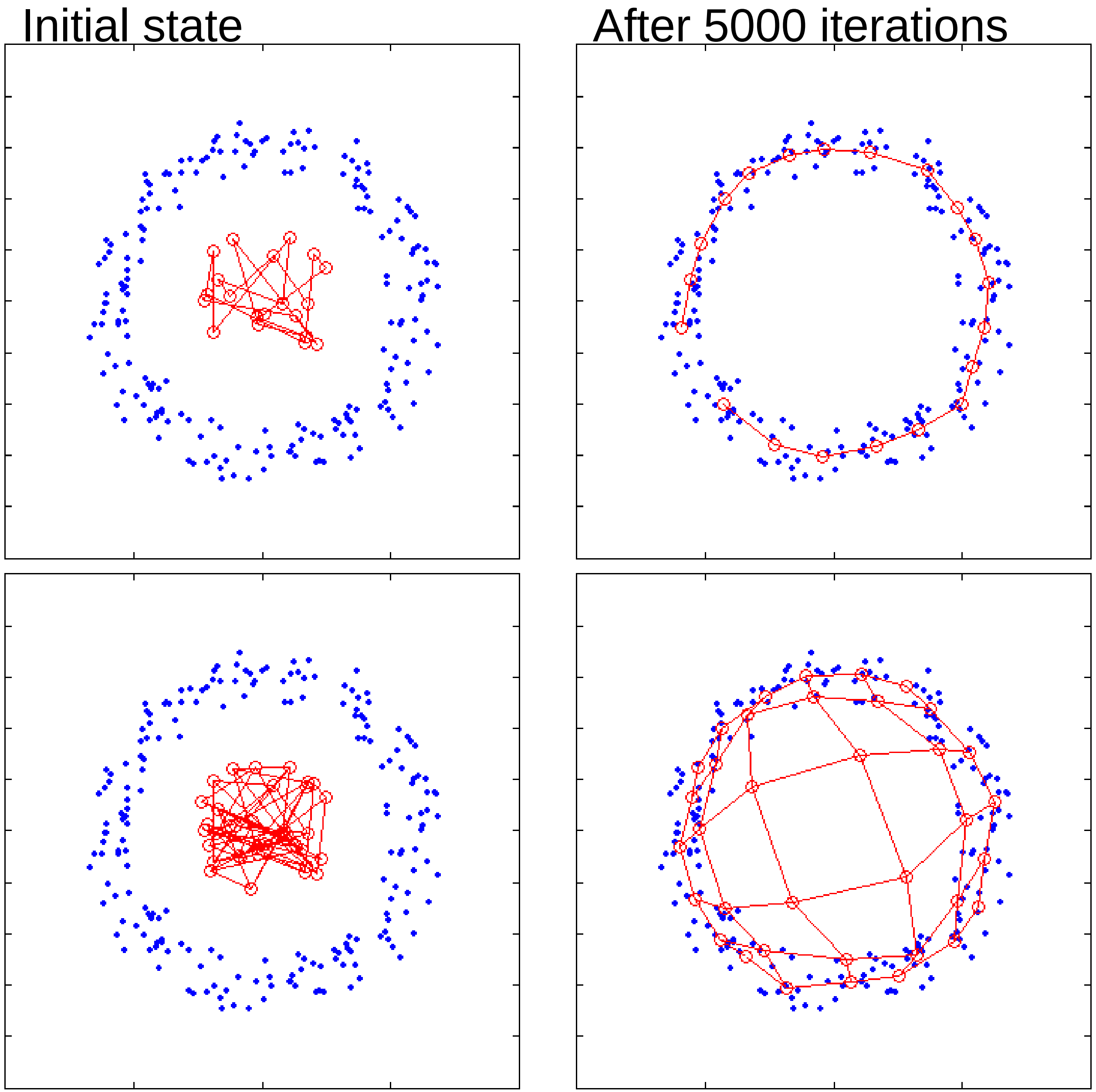


http://www.cartage.org.lb/en/themes/sciences/lifescience/generalbiology/Physiology/NervousSystem/Brain/Brain.htm

Training a SOM

- Random initialization of the weight vectors.
- During training, for each input sample:
  - Competition: Select the winning neuron as the one with its weight vector closest to the input.
  - Adaptation: The weight vector of the winning neuron is moved toward the input (the amount is affected by the learning rate) so that it will respond stronger to the same input next time.
  - Cooperation (the part different from VQ): The weight vectors of other neurons are moved as well, with the amount depending on theirs distances to the winning neuron.
- Repeat the above until convergence.

SOM Example



Initial state

After 5000 iterations

1-D SOM
(20 nodes)

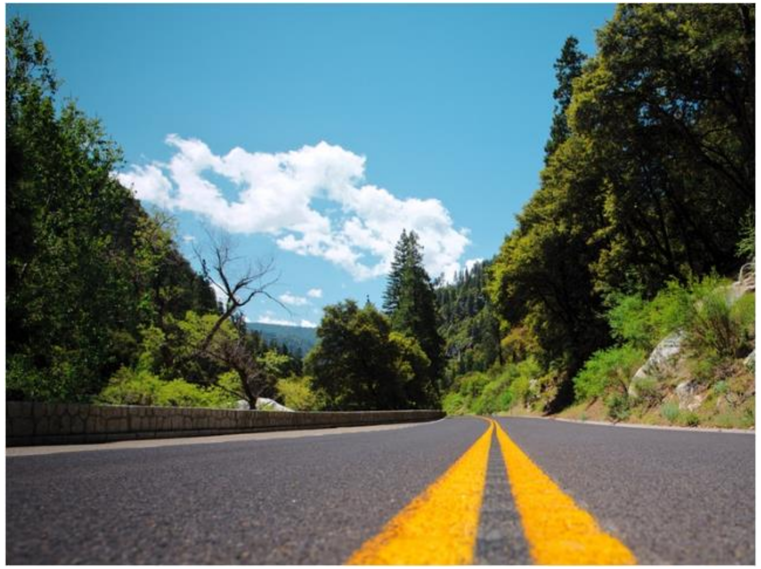
2-D SOM
(6x6)

Applications of SOM

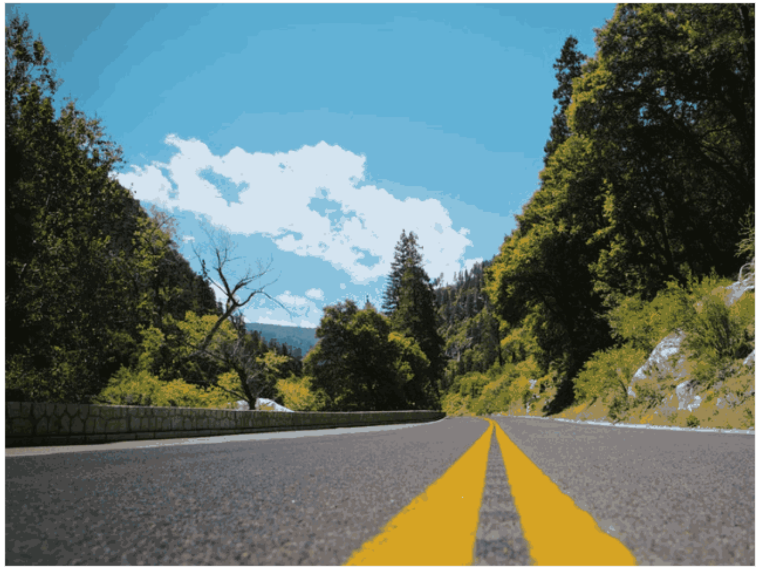
- A reduced-dimension representation of the original data by projecting input samples to the space of the weight vectors.
- The distribution of neuron weight vectors approximates the "density" of the training data.
- The approximate topology (proximity) is preserved. (This is how SOM differs from methods like VQ.)
- Many tasks (such as training a classifier) can be done on this lower dimensional representation. As a result, SOM is often called **SOFM** (self-organizing feature map) as well.
- Visualization of high-dimensional data.

Applications of SOM: Example

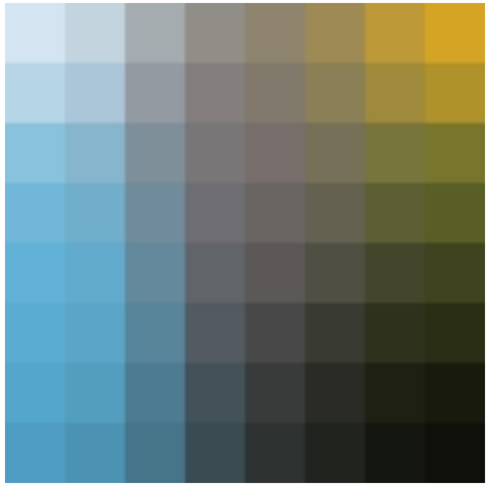
- SOM applied to the RGB values of image pixels.



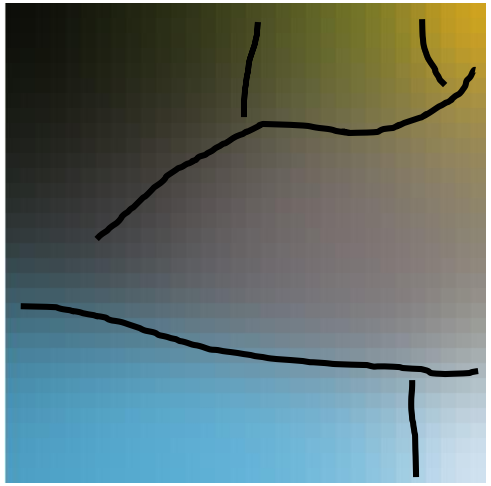
64
colors



8x8
SOFM

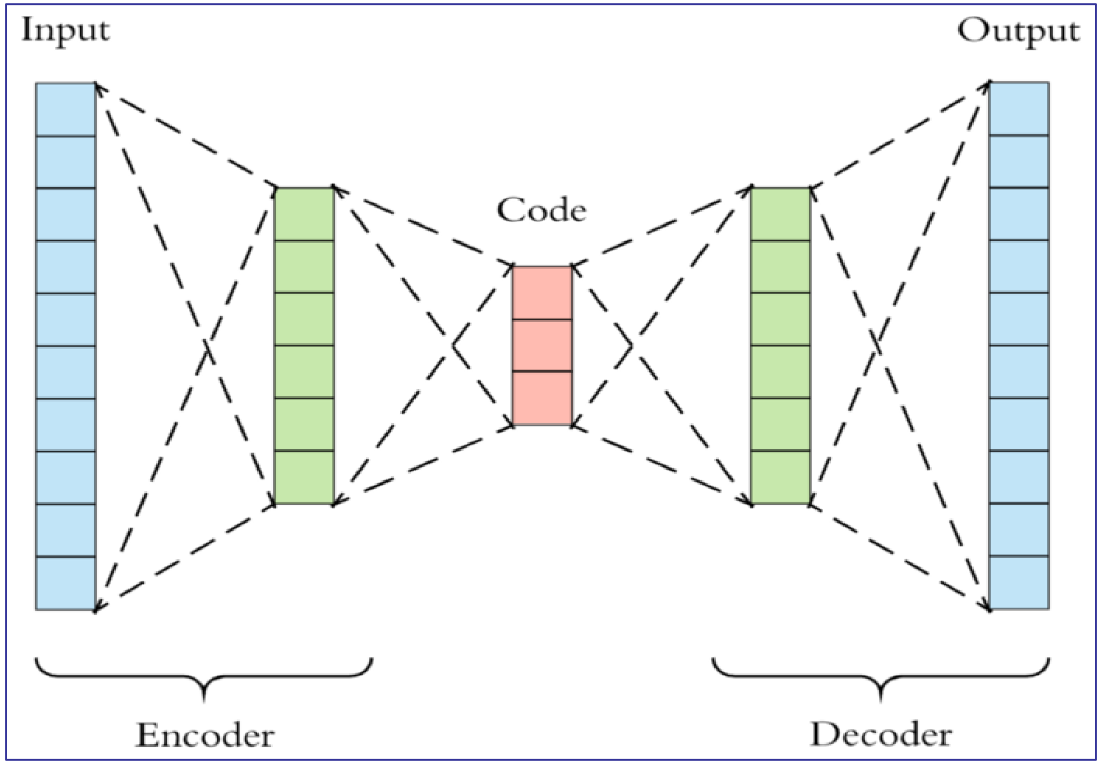


32x32
SOFM



Autoencoder

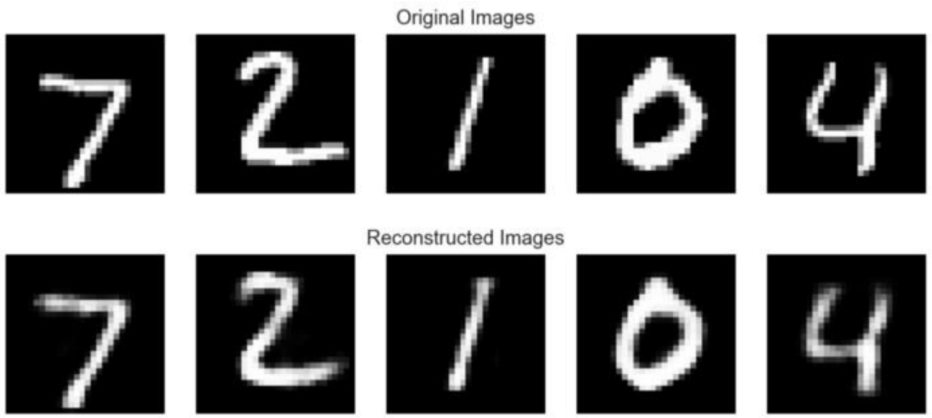
- **Autoencoder** is a learning based technique for dimensionality reduction.
- The objective is for the output to "reproduce" the input.
- The "information bottleneck" leads to dimensionality reduction, forcing the network to learn only the "important" information from the training samples.



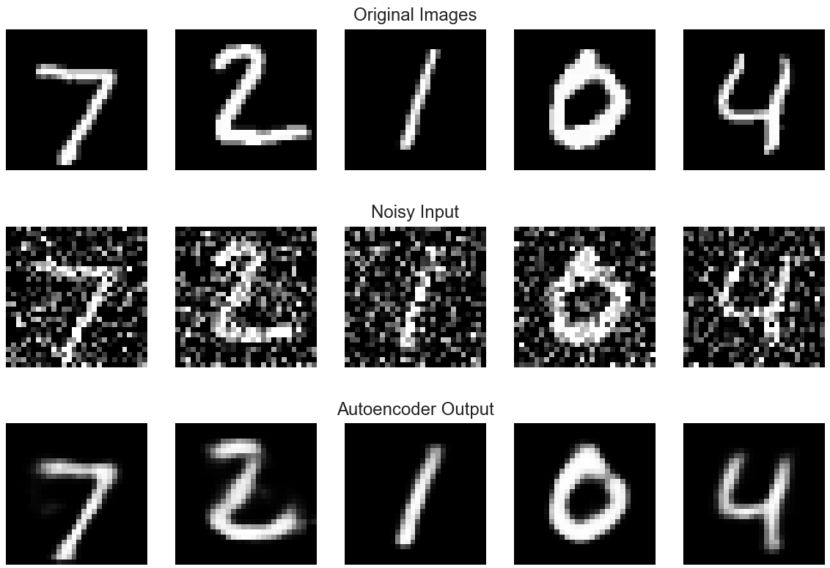
https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Autoencoder Examples

- Reconstruction from a 32-element code. (Original input space has 784 dimensions.



- Image denoising. (Trained with noisy-clean image pairs.)



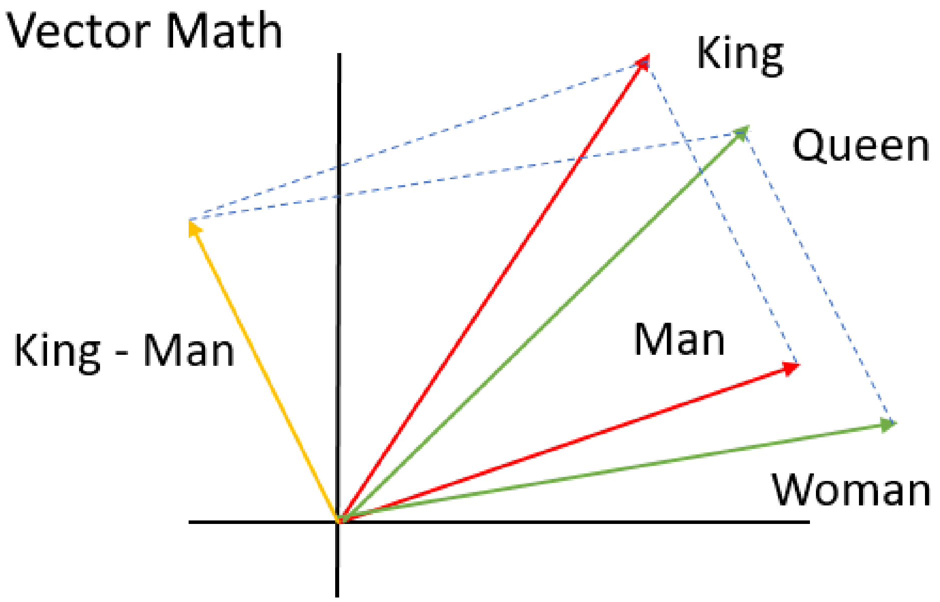
- Many other applications …

https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Word2Vec: Word Embedding

- In natural language processing, the space of words has very high dimensionality with one-hot coding.
  - Inefficient processing
  - Relations between words missed (no context in the codes)
- A lower dimensional embedding is desired where the dimensions can capture meanings of words.

An example in English:



https://medium.com/analytics-vidhya/word-embeddings-in-nlp-word2vec-glove-fasttext-24d4d4286a73

Learning Word Embedding

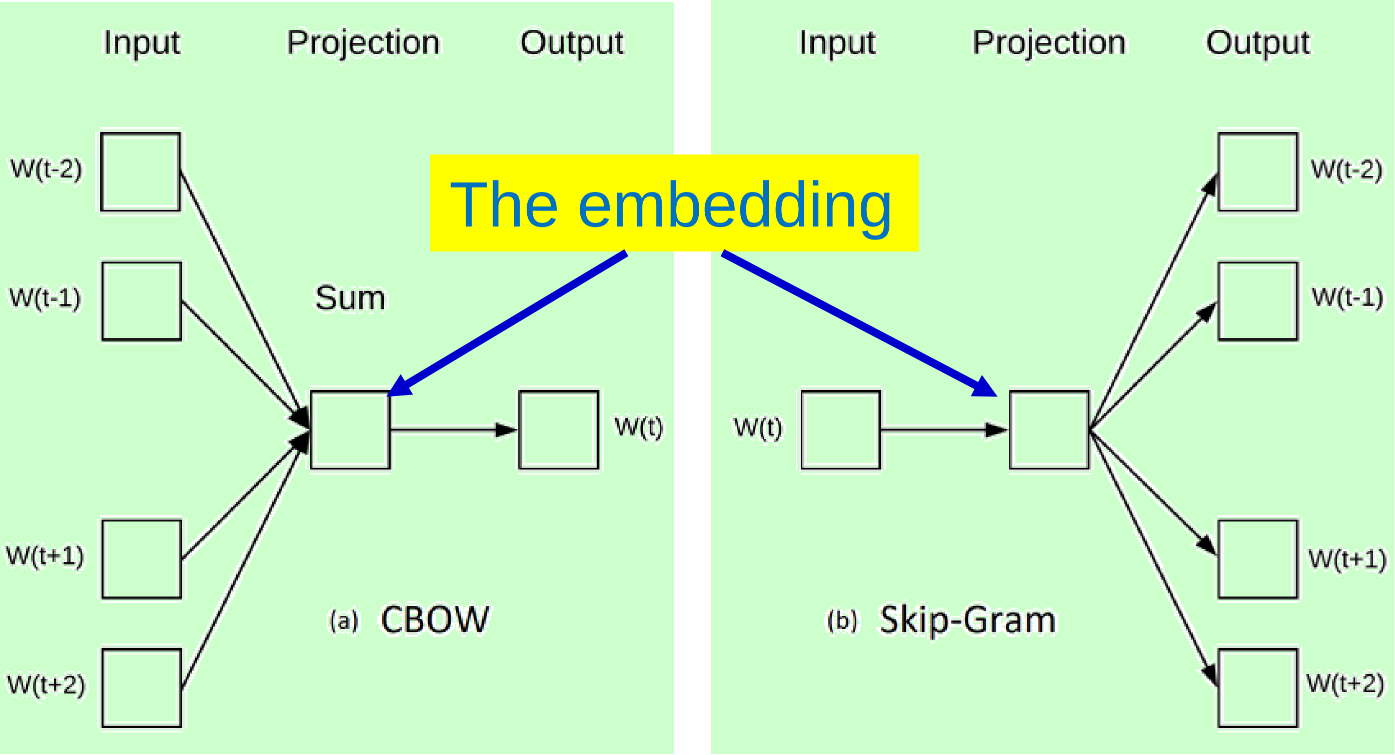
- But how do we get the word embedding (the vector space of the embedding, and the vectors presenting individual words)?
- We have to learn from the context (in practice, this means the surrounding words):

  "*A word is characterized by the company it keeps.*" (Firth, 1950s)

- Train with (central-word, context-words) pairs from a large corpus.

Learning Word Embedding

(individual words are on-hot coded)



The embedding

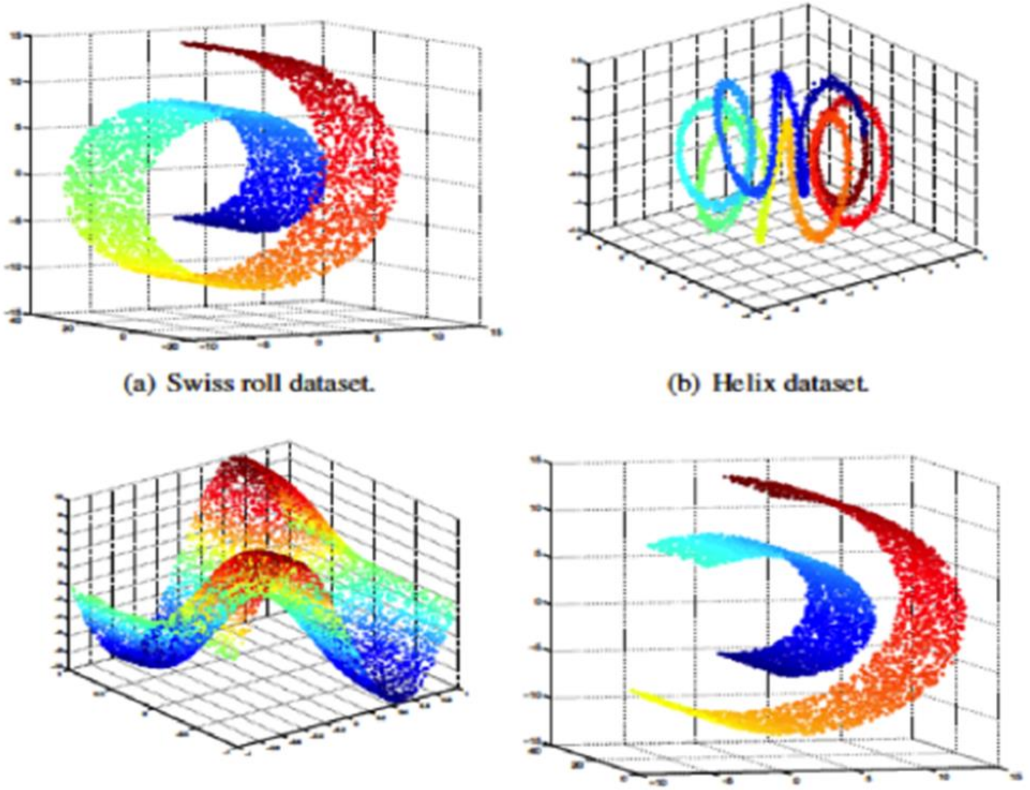
Two versions:

- **CBOW** (continuous bag-of-words): Learn to predict the central word from context words.
- **Skip-gram**: Learn to predict the context words from the central word.

https://www.researchgate.net/figure/CBOW-and-Skip-Gram-neural-architectures_fig14_328160770

Nonlinear Embedding and Manifolds

- **Manifolds**: Low dimensional structures in a high dimensional space.
- A few examples are shown here.
- A 1-D manifold is also called a **principal curve**. 1-D SOM is a technique that can be used to find principal curves.
- For data points distributed in a manifold, the distribution is locally linear. This condition is used in many algorithms that try to do dimensionality reduction for such data.



(a) Swiss roll dataset.

(b) Helix dataset.

(c) Twinpeaks dataset.

(d) Broken Swiss roll dataset.

Locally Linear Embedding (LLE)

- A class of algorithms for dimensionality reduction that utilize conditions of local linearity.
- A basic two-step method:
  - Approximate each data point as a linear combination of its neighbors. (For example, using only the K nearest neighbor.)
  - Fix the coefficients of the linear combinations, and optimize the low-dimensional coordinates of the data points (their embeddings) to minimize the reconstruction errors.