Chapter 7: Competitive learning, clustering, and self-organizing maps

António R. C. Paiva

EEL 6814 – Spring 2008

Outline

Competitive learning

Clustering

Self-Organizing Maps

What is competition in neural networks?

- Competition means that, given the input, the PEs in a neural network will compete for the “resources,” such as the output.
- For every input the PEs will produce an output. Only the “most suitable” output is utilized. Only the winner PE is updated.
- As an analogy, consider bidding in the stock market. The stock are the input, and each broker competes by bidding with a value. The most suitable output is the highest value!

Why is competition necessary?

- Competition creates specialization in the network.
  - Specialization means that, through competition, the PEs are tuned for different areas of the input space.
- In many situations, resources are limited, so competition recreates these natural constraints in the environment.
- Competition is the base concept for clustering and self-organizing maps (SOMs).

Characteristics of competitive learning

- Competitive learning is typically applied to a single-layer topology.
    - Formulations using multi-layer topologies exist but typically employ independent competition on each layer.
- Competitive learning is unsupervised learning.
- Competition is by itself a non-linear process and thus difficult to treat mathematically.

Criteria for competitive learning I

- Error minimization
  - Select the PE such that the output yields the minimum 'error',

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \|\mathbf{x} - \mathbf{y}\|$$

    (Notice that error can be defined with different metrics and depends on the application.)
  - Utilized in the formulation of clustering methods and SOM.

Criteria for competitive learning II

- Entropy maximization
  - Selects the PE such that, on average, all PEs are equally likely to be a winner. Put differently, an histogram of how many times each PE was a winner is approximately uniform.
  - Important for density estimation.
  - Depends on the formulation, but entropy maximization can be achieved by error minimization.

Outline

Competitive learning

Clustering

Self-Organizing Maps

Clustering

- Clustering is a particular example of competitive learning, and therefore unsupervised learning.
- Clustering aims at representing the input space of the data with a small number of reference points.
  - The reference points are called centroids and each centroid defines a cluster.
  - The difference with PCA is that a cluster is a hard neighborhood.
  - That is, any point in the neighborhood of the reference point is represented by that point.

K-means

- K-means is perhaps the simplest and most widely used clustering method.
- K-means minimizes the reconstruction MSE.
- Cost function:

$$J = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{y}_i - \mathbf{x}_j\|^2 ,$$

where K is the number of clusters (or centroids), $\mathbf{y}_i$ is the ith centroid and C_i are the input data points within the corresponding cluster.

Optimization of K-means

- Take the gradient of the cost function

$$\frac{\partial J}{\partial \mathbf{y}_k} = \sum_{i=1}^{K} \sum_{\mathbf{x}_j \in C_i} 2(\mathbf{y}_i - \mathbf{x}_j) \frac{\partial(\mathbf{y}_i - \mathbf{x}_j)}{\partial \mathbf{y}_k}$$

$$= \sum_{\mathbf{x}_j \in C_k} 2(\mathbf{y}_k - \mathbf{x}_j)$$

- Setting the gradient to zero gives the *fixed-point update rule*,

$$\Leftrightarrow \sum_{\mathbf{x}_j \in C_k} \mathbf{x}_j = \sum_{\mathbf{x}_j \in C_k} \mathbf{y}_k = N_k \mathbf{y}_k$$

$$\Leftrightarrow \mathbf{y}_k = \frac{1}{N_k} \sum_{\mathbf{x}_j \in C_k}$$

Algorithm

1. Initialization: Select K random data points as centroids.
2. While “change in J is large”
   2.1 Assign each data point to the “nearest” centroid (i.e., smaller error).
   2.2 Compute new location of centroids.

Note:

- For finite data, this algorithm is known to converge to a local minima in a finite number of steps.

Demo

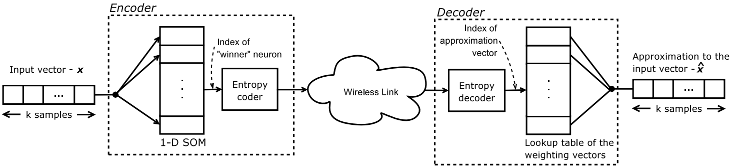
From: `http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/LBG_2.html`

Clustering and vector quantization I

- Vector quantization is an important application of clustering in engineering.
- The K-means is equivalent to the “Linde-Buze-Gray” (LBG) algorithm commonly known in vector quantization.

Clustering and vector quantization II

- Vector quantization is a form of clustering aimed at lossy data compression.
  - Idea: The centroids are known both by the encoder and decoder. Instead of transmit the data, send the index of the centroid that is nearest. At the decoder end, use the centroid itself as an approximation to the original data data point.



Outline

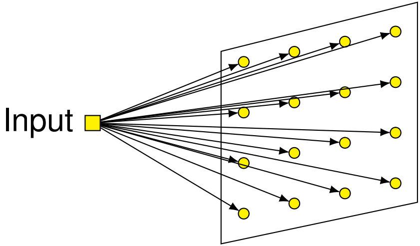
Competitive learning

Clustering

Self-Organizing Maps

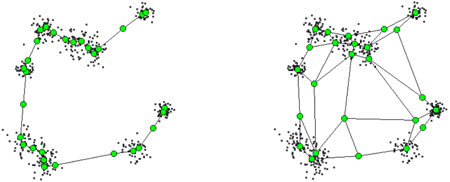
Self-Organizing Maps

- Self-organizing maps (SOMs; also known as Kohonen SOM maps) are another example of competitive learning.
- *"The goal of SOM is to transform the input space into a 1-D or 2-D discrete map in a topologically ordered fashion."*



Distinct feature

- SOM builds a topologically preserving map.
  - Topologically preserving means that data points close in the input space are represented by nearby points in the SOM.



  - Self-organizing means that the competitive learning process finds this topology directly from data.

Implementation

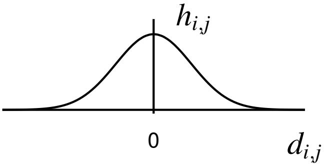
- Each step of training involves three processes:
  1. *Competition*: The network computes the winner PE based on some criterion.
  2. *Cooperation*: The winner PE defines a neighborhood PEs that are updated.
  3. *Adaptation*: All PEs in the neighborhood of the winner PE are adapted to optimize the criterion, *weighted* by the topological distance to the winner PE.
- Training of the SOM can be divided in two phases: ordering and convergence.

Cooperation process I

- Cooperation between neighboring PEs implements lateral interaction between neurons in biological systems.
- Cooperation is obtained by *soft-competition*.
  - Although there is still only one winner, a neighborhood of the winner is updated.

Cooperation process II

- The neighborhood function $h_{i,j}$ must be:
  - Symmetric around the origin. This implies that the function is *shift-invariant*.
  - The amplitude must decrease monotonically to zero.
  - The "width" must be adjustable.
- A typically choice for the neighborhood function is the Gaussian:



$$h_{i,j} = \exp\left(-\frac{d_{i,j}^2}{2\sigma^2(n)}\right)$$

($d_{i,j}$ is the *topological* distance; i.e., the distance in the map.)

Cooperation process III

- Cooperation forces neighboring PEs to tune for neighboring areas of the input space.
- Cooperation is the process responsible for self-organization (and topology preservation) in SOMs.

Adaptation process I

- At each step the PEs are adapted according to

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n)h_{i,j}(n)(\mathbf{x} - \mathbf{w}_i(n)),$$

where i is the index of the winner PE.

Adaptation process II

- At each epoch, the stepsize η and "width" of the neighborhood function σ are reduced according to some rule. For example,

$$\eta(n) = \frac{\eta_0}{1 + \alpha \frac{n}{n_{\max}}},$$

$$\sigma(n) = \frac{\sigma_0}{1 + \beta \frac{n}{n_{\max}}}.$$

  - Decreasing η creates simulated annealing.
  - Decreasing σ means that cooperation exist at first, but towards the end, the PEs will fine tune to their specific areas.

Demo

From: `http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/SOM.html`

Applications

- Visualization of higher dimensional data or process.
- Density estimation.