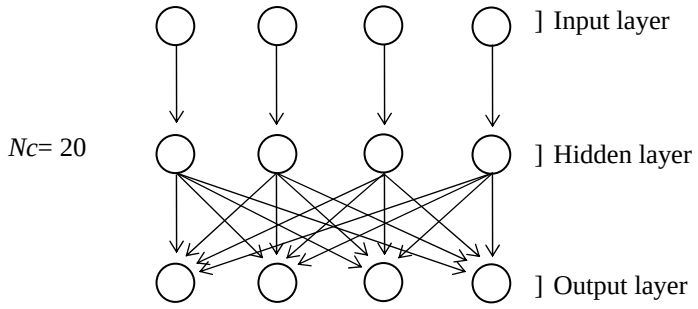
Supplementary Material: AutoCM Neural Network Basic Equations and Convercy Properties, MST, and MRG

Learning Equations

The Auto Contractive Map (CM) presents a three layer architecture: an Input layer, where the signal is captured from the environment; a Hidden layer, where the signal is modulated inside the CM; and an Output layer by which the CM influences the environment according to the stimuli previously received (Figure 1). Each layer is composed by *N* units. Then the whole CM is composed by *3N* units. The connections between the Input layer and the Hidden layer are Mono-dedicated, whereas the ones between the Hidden layer and the Output layer are at maximum gradient. Therefore, in relation to the units the number of the connections *Nc*, is given by:

$Nc = N\,(N + 1)$.

Figure 1. The figure gives an example of a Auto CM with $N = 4$.



All the connections of CM may be initialized both by equal values and by values at random. The best practice is to initialize all the connections with the same positive value, close to zero.

The learning algorithm of CM may be summarized in four orderly steps:

1. Signal Transfer from the Input into the Hidden layer;
2. Adaptation of the connections value between the Input layer and the Hidden layer; *

3. Signal Transfer from the Hidden layer into the Output layer; *
4. Adaptation of the connections value between the Hidden layer and the Output layer.

(*): step 2 and 3 may take place in parallel.

We define as $m^{[s]}$ the units of the Input layer (sensors), scaled between 0 and 1; as $m^{[h]}$, the ones of the Hidden layer; and as $m^{[t]}$, the ones of the Output layer (system target). We define **v** the vector of monodedicated connections; **w** the matrix of the connections between Hidden layer and Output layer; and *n*, the discrete time of the weights evolution or rather, *n* is the number of cycles of elaboration that, starting from zero increases itself of a unit at each successive cycle: $n \in N$.

The signal forward transfer equations and the learning ones are as follows:

a. Signal transfer from the Input to the Hidden:

(1) $$m_{i_{(n)}}^{[h]} = m_i^{[s]} \left(1 - \frac{v_{i_{(n)}}}{C} \right)$$

Where *C* = Positive real number not lower than 1, named Contractive Factor; and where *(n)* subscript has been omitted from input layer units, these being constant at every elaboration cycle.

b. Adaptation of the connections $v_{i_{(n)}}$ through the $\Delta v_{i_{(n)}}$ trapping the energy difference generated by the equation (1):

(2) $$\Delta v_{i_{(n)}} = \left(m_i^{[s]} - m_{i_{(n)}}^{[h]} \right) \cdot \left(1 - \frac{v_{i_{(n)}}}{C} \right) ;$$

(3) $$v_{i_{(n+1)}} = v_{i_{(n)}} + \Delta v_{i_{(n)}} ;$$

c. Signal transfer from the Hidden to the Output:

(4) $$Net_{i_{(n)}} = \sum_{j}^{N} m_{j_{(n)}}^{[h]} \cdot \left(1 - \frac{w_{i,j_{(n)}}}{C} \right);$$

(5) $$m_{i_{(n)}}^{[t]} = m_{i_{(n)}}^{[h]} \left(1 - \frac{Net_{i_{(n)}}}{C} \right).$$

d. Adaptation of the connections $w_{i,j_{(n)}}$ through the $\Delta w_{i,j_{(n)}}$ trapping the energy differences generated by the equation (5):

(6) $$\Delta w_{i,j_{(n)}} = \left(m_{i_{(n)}}^{[h]} - m_{i_{(n)}}^{[t]} \right) \cdot \left(1 - \frac{w_{i,j_{(n)}}}{C} \right) \cdot m_{j_{(n)}}^{[h]};$$

(7) $$w_{i,j_{(n+1)}} = w_{i,j_{(n)}} + \Delta w_{i,j_{(n)}}$$

The value $m_{j_{(n)}}^{[h]}$ of (6) is used for proportioning the change of the connection $w_{i,j_{(n)}}$ to the quantity of energy liberated by the node $m_{j_{(n)}}^{[h]}$ in favor of node $m_{i_{(n)}}^{[t]}$.

In CM the learning process, conceived as adjustment of the connections in relation to the minimization of Energy, corresponds to the continuous acceleration and deceleration of velocities of the learning signals (corrections $\Delta w_{i,j_{(n)}}$ and $\Delta v_{i_{(n)}}$) inside the ANN connection matrix.

AutoCM convercy properties

Auto Contractive Maps do not behave as a regular ANN. Rather, they learn starting from all connections set up with the same values so they do not suffer the problem of the symmetric connections. During training, they develop for each connection only positive values. Therefore, Auto CM do not present inhibitory relations among nodes, but only different strengths of excitatory connections. Auto CM can learn also in hard conditions, that is, when the connections

of the main diagonal of the second connections matrix are removed. When the learning process is organized in this way, Auto CM seems to find a specific relationship between each variable and any other. Consequently, from an experimental point of view, it seems that the ranking of its connections matrix is equal to the ranking of the joint probability between each variable and the others.

After learning process, any input vector, belonging to the training set, will generate a null output vector. So, the energy minimization of the training vectors is represented by a function trough which the trained connections absorb completely the input training vectors. Auto CM seems to learn to transform itself in a dark body.

At the end of the training phase ( ), all the components of the weights vector v reach up the same value:

(8) $\lim_{n\to\infty} v_{i_{(n)}} = C$.

The matrix w, then, represents the CM knowledge about all the dataset. It is possible to transform the **w** matrix also in probabilistic joint association among the variables *m*:

(9) $$p_{ij} = \frac{w_{ij}}{\sum_{j=1}^{n} w_{ij}};$$

(10) $$P(m_j^{[n]}) = \sum_{j}^{n} p_{ij} = 1$$

The new matrix **p** can be read as the probability of transition from any state-variable to anyone else:

(11) $P(w_i^{[1]} | w_j^{[1]}) = p_{i,j}$.

At the same time the matrix **w** may be transformed into a non Euclidean distance metric (semimetric), when we train the CM with the main diagonal of the **w** matrix fixed at value N. Now, if we consider N as a limit value for all the weights of the **w** matrix, we can write:

(12) $d_{i,j} = N - w_{i,j}$

The new matrix **d** is also a squared symmetric matrix where the main diagonal represents the zero distance between each variable from itself.

AutoCM and minimum spanning tree (MST)

Kruskal in the 1956 found an algorithm able to determinate the MST of any undirected graph in a quadratic number of steps, in the worst case. Obviously, the Kruskal algorithm generates one of the possible MST. In fact in a weighted graph more than one MST are possible if there are equal distances among variables

From the conceptual point of view the MST represents the **energy minimization** state of a structure. In fact, if we consider the atomic elements of a structure as vertices of a graph and the strength among them as the weight of each edge, linking a pair of vertex; the MST represents the minimum of energy needed because all the elements of the structure continue to stay together.
In a closed system, all the components tend to minimize the overall energy. So the MST, in specific situations, can represent the most probable state where a system tends to.

To define the MST of an undirected graph, each edge of the graph has to be weighted. We employ the weights assigned by AutoCM to compute a non linear matrix of weights among variables usable for MST development.

Maximal regular graph.

The MST represents the nervous system of any dataset. In fact, the summation of the strength of the connection among all the variables represents the total energy of that system. The MST selects only the connections **that minimize this energy**. Consequently, all the links shown by MST are fundamental, but not every fundamental link of the dataset is shown by MST. Such limit is intrinsic to the nature of MST itself: every link able to generate a cycle into the graph is eliminated, however its strength. To avoid this limit and to explain better the intrinsic complexity of a dataset, it is necessary to add more links to the graph according to two criteria:

1. The new links have to be **relevant** from a quantitative point of view;
2. The new links have to be able to generate new **cyclic regular microstructures**, from a qualitative point of view.

Consequently, the MST Tree-graph is transformed into an undirected graph with cycles. Because of the cycles, the new graph is a dynamic system, involving in its structure the **time** dimension. This is the reason why this new graph should provide information not only about the structure but also about the **functions** of the variables of the dataset.

Very briefly this is the procedure:

1. The MST structure is assumed as a starting point of the new graph;
2. The sorted list of the connections skipped during the MST generation is taken into consideration;

3. The complexity of the new graph each time is estimated with a special function (H function) while each new connection is added to the MST structure, to monitor the variation of the complexity of the new graph at every step
4. We have named **Maximally Regular Graph** (M.R.G.) the graph whose **H Function** is the highest, among all the graphs generated adding to the original MST the new connections skipped before to complete the MST itself.

Consequently, starting from the equation (32), the M.R.G. is given by the following equations:

(40)

$$H_i = f(G(A_p, N)); \quad \text{/* Generic Function on a graph with } A_p \text{ arcs e } N \text{ Nodes */}$$

$$H_i = \frac{\mu_p \cdot \varphi_p - 1}{A_p}; \quad \text{/* Calculattion of H Function, where } H_0 \text{ represents MST complexity */}$$

$$MRG = Max\{H_i\}. \quad \text{/* Graph with highest H */}$$

$$i \in [0,1,2,...,R]; \quad \text{/* Index of H Function*/}$$

$$p \in [N-1, N, N+1, ..., N-1+R]. \text{ /* index for the number of graph arcs */}$$

$$R \in \left[0,1,..,\frac{(N-1)\cdot(N-2)}{2}\right]; \text{ /* Number of the skipped arcs during the M.S.T. generation */}$$

The “R” variable is a key variable during the M.R.G. generation. “R” variable, in fact, could also be the null, when the generation of M.S.T. implies no connections to be skipped. In this case, there is no M.R.G. for that dataset. The “R” variable, further, makes sure that the last and consequently the weakest connection added to generate M.R.G. is always more relevant that the weakest connection of M.S.T. The M.R.G., finally, generates, starting from the M.S.T., the graph presenting the **highest number of regular microstructures using the most important connections** of the dataset. In addition, the H Function selected to generate the M.R.G. is high, more meaningful the microstructures of the M.R.G..