Identifying Clusters Using Growing Neural Gas: First Results

Riccardo Rizzo and Alfonso Urso

ICAR-CNR
Italian National Research Council,
Viale delle Scienze,
90128 Palermo, Italy
{ricrizzo,urso}@pa.icar.cnr.it
http://www.pa.icar.cnr.it

Abstract. Growing Neural Gas is a self organizing network capable to build a lattice of neural unit that grows in the input pattern manifold. The structure of the obtained network often is not a planar graph and can be not suitable for visualization; cluster identification is possible only if a set of not connected subgraphs are produced. In this work we propose a method to select the neural units in order to extract the information on the pattern clusters, even if the obtained network graph is connected. The proposed method creates a new structure called Labeling Network (LNet) that repeats the topology of the GNG network and a set of weights to the links of the neuron graph. These weights are trained using an anti-Hebbian algorithm obtaining a new structure capable to label input patterns according to their cluster.

Keywords: Growing Neural Gas, Self organization, Cluster Identification.

1 Introduction

What we call self organization is a mechanism that changes a complex interconnected structure of simple units in something completely different through the emergence of a global order out of local simple interactions. Often self organization is related to a different scale of observation: at the lower level the only observable thing is the connection of simple units, but if the point of view changes a completely new behavior can be observed.

An example is the Self Organizing Map (SOM) [1] where, at the lower level the units try to approximate the input patterns, and, on the upper level, we see the patterns organized in a topographic map.

Many other neural networks with similar mechanisms were developed after the SOM: Neural Gas (NG) [2] and Growing Neural Gas (GNG) [3] are among them. Growing Neural Gas is an interesting algorithm that is able to "produces" new neurons when it is necessary to represent new input patterns. Another characteristic of this algorithm is that neurons are connected using links that

C. Alippi et al. (Eds.): ICANN 2009, Part I, LNCS 5768, pp. 536–545, 2009.

are created during training, when the neurons are excited together. The training stage of the GNG network produce a network of units that completely fills the input pattern manifold. This can be clearly observed in the original paper [3] and will be clear in the following section. A complex and not planar neural graph, as the one produced by GNG, is difficult to use in clustering applications because there is not visualization (the network of the neural units has a complex structure even for simple datasets) nor cluster identification (see Fig. 2 for example).

In the past few approaches to cluster analysis using GNG were proposed. One method obtains a not connected graph using an edge removal procedure based on an utility factor and a suitable threshold [8]. Another method was focused on the GNG algorithm robustness and outliers filtering. The Minimum Description Length principle was then used to obtain the number of clusters on several attempts with different number of neural units [9].

In this work we propose a neural structure, called Labeling Network (LNet), that has the same topology of a trained GNG but uses an algorithm inspired by the EXIN mechanism (Excitatory-INhibitory) [6] in order to have one or few "active" neurons for each pattern cluster.

The paper is organized as follows: in the next section the details of the problem are explained with reference to a particular dataset and the GNG algorithm is introduced, then the proposed method is explained and in section 4 the algorithm is presented. The last two sections present some results and conclusions.

2 Self Organizing Networks and Cluster Identification

The goal of algorithms like SOM, NG and GNG is to approximate the input patterns at the best, in order to minimize the approximation error (this error is in [2] for NG and can be found as energy function in [7] for a modified SOM network). In all these algorithms communication among neurons is a fundamental part of learning algorithm: in SOM this exchange is supported by the neighborhood function defined over the lattice, in GNG networks the neural units exchange information about weight update using links, in NG the weight modification is propagated (with attenuation) from the best matching unit (bmu) to the neighborhood neurons. This communication among neurons is the basic mechanism of self organization.

From now on we focus on GNG network. The GNG network starts from a set of two neural units $U = \{u_a, u_b\}$ each of them with a set of weights $\mathbf{w}_i \in \Re^n$ and a cumulative error variable $e_i \in \Re$. These neurons are connected using a link $c_{a,b}$ that has an age $a_{a,b} \in \Re$ but not a weight. The training algorithm starts picking a random initial patten $\mathbf{x} \in X \subset \Re^n$ and finding the best matching unit (bmu) u_{bmu} and the second bmu, u_{bmu2}, from the neuron set:

$$bmu = \arg\min_{u_i \in U} d(\mathbf{x}, \mathbf{w}_i), \quad bmu2 = \arg \min_{u_i \in U} \min_{, \, u_i \neq u_{bmu}} d(\mathbf{x}, \mathbf{w}_i). \tag{1}$$

The edge $c_{bmu,bmu2}$, that connects u_{bmu} and u_{bmu2}, is set to zero or is created, then the error of bmu is updated as $\Delta e_{bmu} = \|\mathbf{w}_{bmu} - \mathbf{x}\|^2$ together with its

weight $\Delta \mathbf{w}_{bmu} = \varepsilon_b(\mathbf{x} - \mathbf{w}_{bmu})$, where ε_b is the learning parameter. The weight correction is propagated to the units connected as $\Delta \mathbf{w}_n = \varepsilon_n(\mathbf{x} - \mathbf{w}_n)$. At the end of the procedure all the connections emanating from *bmu* are updated adding a constant value, and all the error variables are decreased by multiplying for a constant $d < 1$. The edges that has an age larger than age_{MAX} are removed together with the units that have no emanating links.

Each λ learning steps a new unit u_r is added between the unit with the maximum accumulated error u_q and its neighborhood with the larger error variable u_f; the weights $\mathbf{w}_r$ of the new unit are mediated between $\mathbf{w}_q$ and $\mathbf{w}_f$ and the error of units u_q and u_f are decreased multiplying for a constant $\delta < 1$; finally $e_r = e_q$.

When the learning stage ends each neural unit approximate a subset of the input pattern; the number of patterns associated to a single neuron depends on the number of available neurons and it is roughly the same for each neuron. The neural units in some way represent the input patterns that are inside the corresponding Voronoi regions. Now we want to reduce the number of neurons that activate when the input patterns are presented, in order to identify the clusters inside the patterns set and o produce few clear signals that allows to recognize the input patterns. This problem can be easily visualized using a real dataset.

The "Iris" dataset [10] is maybe the simplest and most common real dataset available. It is made of three clusters of objects in four dimensions: an idea of the spread of the pattern in their space is given in Fig. 1. This dataset was chosen because the two clusters "Iris Versicolor" and "Iris Virginica" are somewhat difficult to separate.

Training GNG on this dataset (for example with the following parameters $\epsilon_b = 0.5$, $\epsilon_n = 0.2$, $age_{MAX} = 50$, $\lambda = 50$, $d = 0.8$,$\delta = 0.5$ and 30000 learning steps; the meaning of the parameters can be found in [3]) produce the typical result in Fig. 2: a set of connected neurons (50 in the present trial). The two clusters "Iris Virginica" and "Iris Versicolor" are connected even if a separation among them can be easily spotted. This situation is no more than a simplified snapshot of the dataset (see Fig. 2).

3 Overview of the LNet Structure

In [4] the hidden units of a backpropagation network are considered as candidate hypothesis and trained using backpropagation algorithm, then the training is refined in order to better represent the target concept. The same consideration can be made for a trained GNG network: all the neurons are candidates to be the cluster centers for the clusters in input pattern set, what we need is a mechanism to select these neurons. To have emerging cluster centers from a set of neurons it is necessary that the activation of one neuron rise, and inhibits the activation of the other neurons.

The proposed Labeling Network has the same topology of the GNG, as depicted in right side of Fig.3; neurons of the LNet receive an input E_i (given by

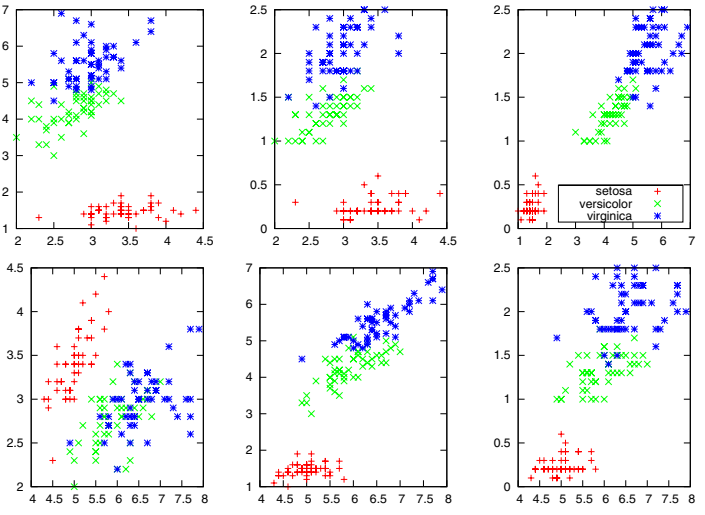


Fig. 1. The iris dataset plotted using all the combination of the four dimensions. It is possible to see that it is difficult to separate "Iris Versicolor" from "Iris Virginica", while "Iris Setosa" forms a well separated cluster.

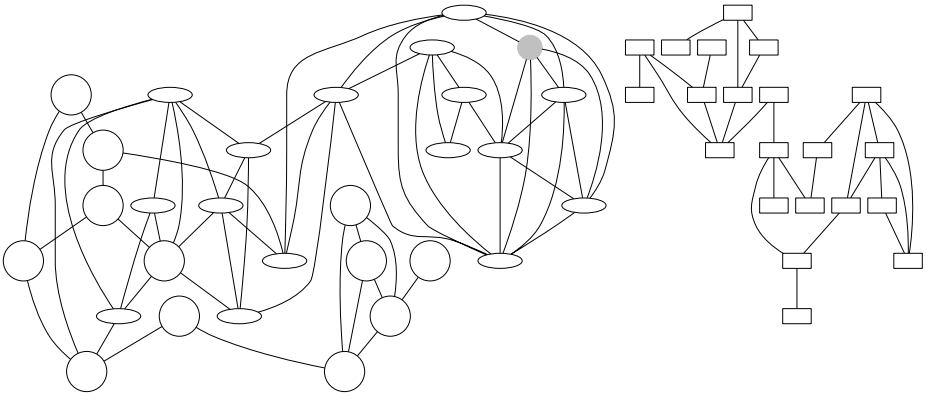


Fig. 2. The result of the learning of the GNG using the iris dataset. The units represented as boxes classify the "Iris Setosa", the "circle" units classify the "Iris Versicolor" and the "ellipse" units the "Iris Virginica"; the gray unit is the "dead unit", the one that does not classify any pattern.

eq. 2), from the trained GNG in the lower layer; these neurons have an activation a_i and excites and inhibits each other using the available links that are modulated by a new set of weights p_{ij}, see left side of Fig.3. These weights are trained using an anti-Hebbian rule, and the activation is calculated using an iterative procedure that is explained in the next section.

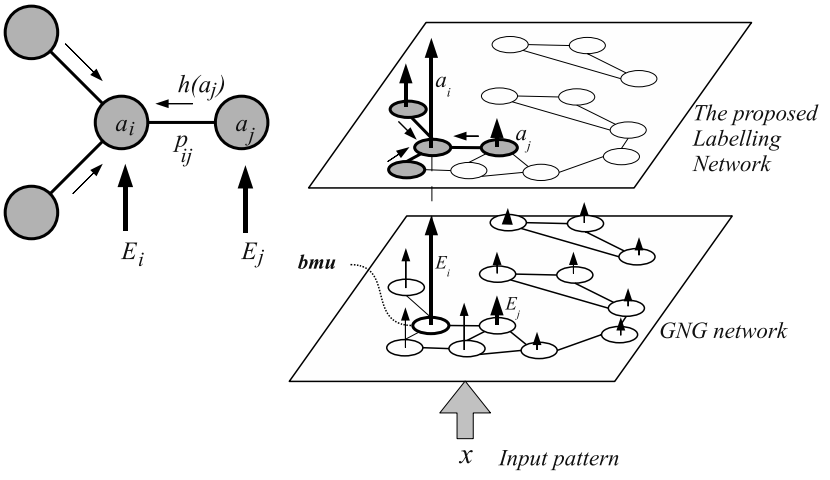


Fig. 3. A representation of the GNG and the Labeling Network on the right and a magnification of the interaction between two neurons, i and j, in the LNet network on the left

In order to build a cluster identification system it is necessary to use a two stage procedure that will produce the two layers on the right of Fig.3:

1. train a GNG network using the available pattern set;
2. add the new LNet structure with the same topology of the GNG.

In the following we will refer only to this second structure, assuming that we have the trained GNG.

At the end of the training procedure we should submit an input to the network and obtain the activation of a single unit that will label the input pattern. In iris dataset, that is made of 3 clusters, 50 pattern each (150 pattern), we should ideally have 3 neural units that have the maximum activation, and "label" the input patterns. This is the kind of result we show in section 5.

4 The Proposed LNet Algorithm

In our network the activation a_i of the single neuron i is not related only to the similarity between the input pattern and the weight vector of the neuron, but it is calculated using a shunting equation that takes into account the excitation due to the input stimulus E_i and the excitation that came from the activation of the other connected neurons I_i using the weights p_{ij} as indicated in the left side of Fig.3 .

The excitation from the external input E_i in LNet neurons is calculated using the distance of the input pattern x from the weight vector of the neuron in GNG network w_i, according to:

$$E_i = \alpha \frac{d_{MAX} - d(\mathbf{w}_i, \mathbf{x})}{d_{MAX} - d_{min}}. \tag{2}$$

where $\alpha < 1$ is a coefficient necessary in order to do not saturate the neuron input, $d(\mathbf{w}_i, \mathbf{x})$ is the Euclidean distance between the pattern $\mathbf{x}$ and the weight vector $\mathbf{w}_i$ of the neuron i, and d_{MAX} and d_{min} are the maximum and minimum distances $d(\mathbf{w}_j, \mathbf{x})$ among the neurons j of the GNG network. This is necessary because we want the maximum excitation for the nearest neuron, usually called *bmu* (best matching unit).

The excitation due to other neurons is obtained from:

$$I_i = \sum_{j=1}^{n} h(a_j(t)) * p_{ij}. \tag{3}$$

where $h(a_i) = \max(a_i, 0)$ and p_{ij} is the weight of the connections.

As said before the input components E_i and I_i are used to calculate the activation of the single neuron using a shunt equation:

$$\frac{da_i}{dt} = -Aa_i(t) + \beta(1 + a_i(t))E_i - \gamma(C + a_i(t))I_i. \tag{4}$$

where A is the decay factor, β is the parameter that regulates the activation due to the external inputs, i.e. the term $(1 + a_i(t))E_i$ that also fixes the maximum value of $a_i(t)$ to 1, and γ is the parameter that regulate the inhibition from other neurons to the neuron activation, $-C$ is the minimum value of $a_i(t)$. The weights p_j are modified using an anti-Hebbian rule:

$$\frac{dp_{ij}}{dt} = \eta h(a_j(t)) * [h(a_i(t)) - p_{ij}]. \tag{5}$$

where η is the learning rate.

The equations 4 and 5 are discretized in order to carry on the simulations [5], and the time quantum Δt is added as a parameter of the simulation:

$$\Delta a_i(t+1) = -A * a_i(t) * \Delta t + \beta(1 + a_i(t))E_i * \Delta t - \gamma(C + a_i(t))I_i * \Delta t. \tag{6}$$

$$\Delta p_{ij}(t+1) = \eta h(a_j(t)) * [h(a_i(t)) - p_{ij}]. \tag{7}$$

the new values $a_i(t+1)$ and $p_{ij}(t+1)$ are calculated as: $a_i(t+1) = a_i(t) + \Delta a_i(t+1)$ and $p_{ij}(t+1) = p_{ij}(t) + \Delta p_{ij}(t+1)$. The learning algorithm of LNet is below. Note that the activation of the units is calculated using an interactive procedure that stops when the update of the solution is below the threshold ϵ.

The learning algorithm

1. $t = 0$
2. `Select the input pattern` x
3. `Calculate the excitation` E_i `of the neurons`
4. `do`
   - `Calculate the excitation` I_i `of the neurons`
   - `Calculate` $\Delta a_i(t)$ `using eq.6 and the new` $a_i(t+1)$ `value`

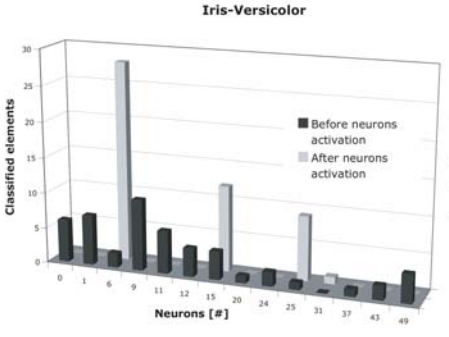
   `while`$(\sum_i \Delta a_i(t+1) > \epsilon)$

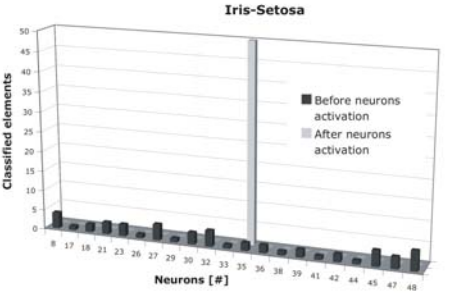
5. `update the weights` p_{ij} `using eq.7`
6. $t++$
7. `if` $(t < t_{max})$ `goto 2 else exit`

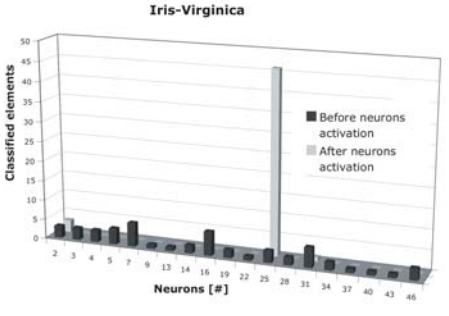
After the training the network is ready to classify the input patterns, for each pattern in input the activation of the neurons is calculated using the steps 3 and 4 of the learning algorithm.

5 Results

We show a typical result obtained with "Iris dataset". As is possible to see from the foreground black bars in the graphs in Fig.4 all the units of the GNG network are *bmu* for a numbers of patterns that varies from 0 to 10 (there is only one dead unit). In LNet the activation of the neurons is calculated using eq. 4 (step 3 and 4 of the algorithm) and this completely changes the landscape: as it is possible to see from the gray bars in the graphs, for all the patterns very few neurons are always activated, these neurons can clearly identify the input clusters. The results are repeated in the table down right of Fig.4.







Neuron	Iris Setosa	Iris Versicolor	Iris Virginica
35	50	0	0
6	0	28	0
15	0	12	0
31	0	1	1
25	0	9	46
2	0	0	3

Fig. 4. Units of the GNG network representing the candidates to be cluster centers (before units activation) and emerging exemplars for the Iris pattern set

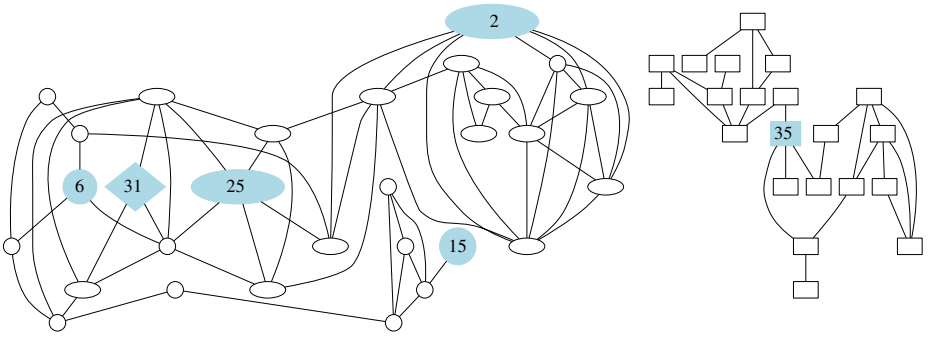


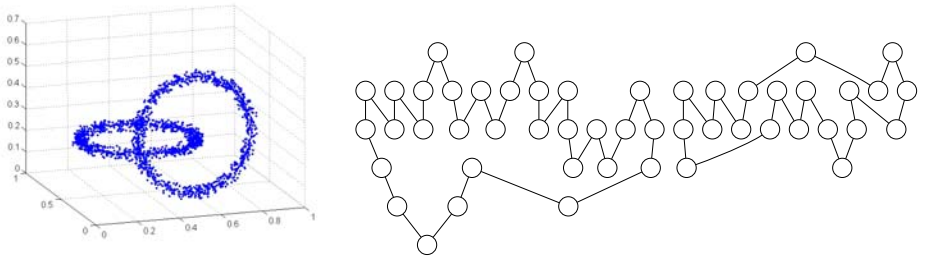
Fig. 5. The neural units that are activated with the input patterns. The gray units are the active one reported in the table down right of Fig.4. The circle units (6 and 15) are the ones that label the "Iris Versicolor" cluster; the diamond unit 31 is the one that label one pattern "Iris Versicolor" and one pattern "Iris Virginica"; the oval units (25 and 2) are the one that label "Iris Virginica" patterns; the square unit 35 is the one that label the "Iris Setosa" cluster.

It is possible to notice that for the cluster Iris-Setosa there is only one exemplar that emerges, the unit number 35. For "Iris Versicolor" and Iris Virginica there is a small number of emerging exemplars, some of them are mixed (unit 25 and 31), although for "Iris Versicolor" most of the elements are classified by unit 6 and for Iris Virginica most of elements are classified by unit 25. If we assume that the unit 25 label "Iris Virginica" then we will have 9 patterns of "Iris Versicolor" that will be misclassified (the one in the bottom graph in Fig. 4).

In our experiments, to avoid the initial saturation of the neuron input, we carried out various trials varying values of the α coefficient defined in eq. 2; the best results were obtained for $\alpha = 0.3$. The other parameters were setup as follows: $A = 10.0$, $C = 10$, $\gamma = 1.0$, $\beta = 1.0$, the threshold $\epsilon = 0.00005$, $\eta = 0.6$ and $\Delta t = 0.00014$.

Some final remarks can be made looking at the topological position of the activated units that is reported in fig. 5: for cluster of "Iris-Setosa" the activated unit is in a "center" position: if we remove the units and the connected edges the subgraph is split in two parts; "Iris Virginica" are classified by the ellipse shaped units and the number 25 (that makes the highest bar in the histogram) seems to be at the crossing with the "Iris Versicolor" cluster; the round units are the ones that cluster "Iris Versicolor". One of them (number 6) is again at the crossing with the cluster of "Iris Versicolor" while the other (number 15) is at the border of the cluster. The proposed algorithm seems to privilege the units that have a small number of active neighborhoods, This needs further investigations.

Another observation come from trials on other datasets that are not compact as the "Iris" dataset. For example in the "2 Rings" artificial dataset in Fig. 6(a) GNG is capable to grow a graph with two connected parts as the one in Fig.6(b) but the LNet is unable to isolate two neurons that can label the patterns in the two clusters, the neurons that are activated are the same of the GNG network.



(a) (b)

Fig. 6. 6(a) The "2 Rings" dataset and the GNG structure obtained 6(b)

6 Conclusions and Future Works

In this paper a new neural network capable to identify clusters from trained GNG is proposed. The new network have the same topology of the GNG but works in a very different way because has weight on the connections between neurons and take the input from excitation of the neurons in GNG. The network is capable to identify the clusters by selecting and activating few neurons that "label" the input patterns and allows to clearly identify clusters. The network was tested on the "Iris dataset" with good results, but more investigations are needed in order to understand if there is some meaning in the position of the active neural units. Moreover while the algorithm is able to isolate an active neuron for a globular cluster it is still difficult to identify clusters in some artificial datasets.

References

1. Kohonen, T.: Self Organizing Maps. Springer, Heidelberg (1997)
2. Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: Neural Gas Network for Vector Quantization and its Application to Time-Series Prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)
3. Fritzke, B.: A growing neural gas network learns topologies. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.) Advances in Neural Information Processing Systems, pp. 625–632. MIT Press, Cambridge (1995)
4. Fu, L.: A Concept Learning Network Based on Correlation and Backpropagation. IEEE Trans. on System, Man and Cybernetics part. B 29(6), 912–916 (1999)
5. Barreto, G.A., Arajo, A.F.R.: The role of excitatory and inhibitory learning in EXIN networks. In: Proceedings of the IEEE World Congress on Computational Intelligence, pp. 2378–2383. IEEE Press, Los Alamitos (1998)
6. Marshall, J.A.: Adaptive perceptual pattern recognition by self-organizing neural networks: Context, uncertainty, multiplicity and scale. Neural Networks 8, 335–362 (1995)
7. Heskes, T.: Energy functions for self-organizing maps. In: Oja, E., Kaski, S. (eds.) Kohonen Maps, pp. 303–316. Elsevier, Amsterdam (1999)

8. Costa, J.A.F., Oliveira, R.S.: Cluster Analysis using Growing Neural Gas and Graph Partitioning. In: Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA (2007)
9. Qin, A.K., Suganthan, P.N.: Robust growing neural gas algorithm with application in cluster analysis. Neural networks 17(8-9), 1135–1148 (2004)
10. Iris Dataset in Orange website, http://www.ailab.si/orange/doc/datasets/iris.htm