Mapas Auto-organizáveis com Topologia Variante no Tempo

Aluizio Fausto Ribeiro Araújo
aluizioa@cin.ufpe.br
Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
Av. Professor Luís Freire, s/n; Cidade Universitária
50670-901, Recife, PE, Brasil

Conteúdo

- Mapas Auto-organizáveis com Topologia Variante no Tempo:
  - TRN
  - GCS
  - GNG

Self-organizing Maps (SOM)

- Limitações
  - A estrutura pré-determinada limita o mapa resultante por causa do:
    - Número fixo de nodos.
    - Conexões pré-definidas entre nodos.

Topology Representing Networks (TRN)

- Martinetz & Schulten (1993)
- Proposals:
  - Distribute a number of nodes according to some probability distribution.
  - Topology Learning: Generate a topology in which the dimensionality is equal to the *local* dimensionality of the input data.

Centro de Informática
UFPE

Topology Representing Networks (TRN)

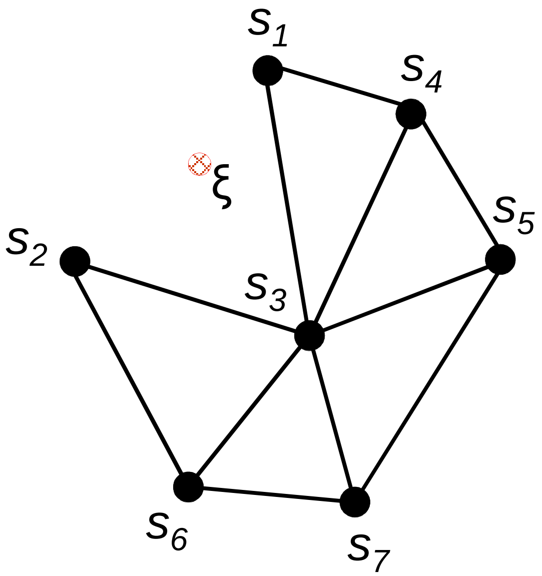
- Martinetz & Schulten (1993)
- Proposals:
  - Distribute a number of nodes according to some probability distribution. *Neural Gas*
  - Topology Learning: Generate a topology in which the dimensionality is equal to the *local* dimensionality of the input data. *Competitive Hebbian Learning*

UFPE

Centro de Informática UFPE

Topology Representing Networks (TRN)

- Neural Gas:



For each sample (ε) from set *A*:

– Order the nodes according to their distance do *ε*.

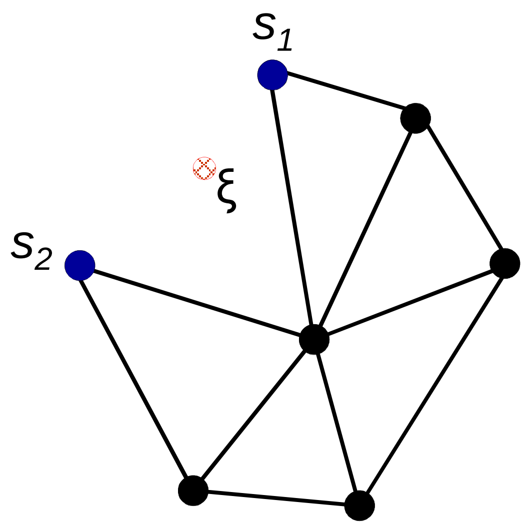
– Adapt the nodes according to their rank order with respect to *ε*.

– Decrease the number of significantly moved centers over time until only the winner is moved.

Centro de Informática UFPE

Topology Representing Networks (TRN)

- Competitve Hebbian Learning:



- Generate at random an input signal according to P(ξ).
- Determine units s_1, s_2, such that:

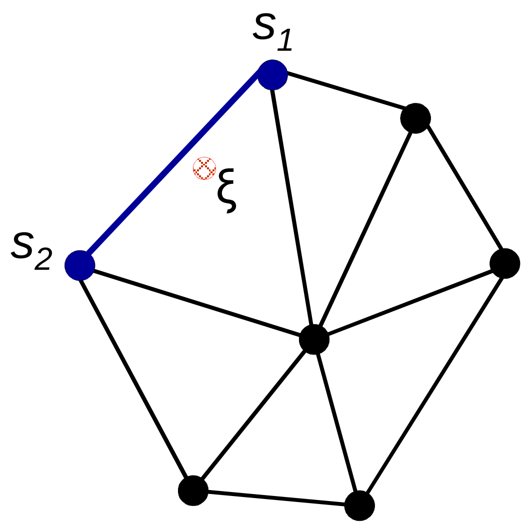
$$\|\mathbf{w}_{s_1} - \xi\| \le \|\mathbf{w}_s - \xi\| \quad (\forall\, s \in A)$$

$$\|\mathbf{w}_{s_2} - \xi\| \le \|\mathbf{w}_\mathbf{s} - \xi\| \quad (\forall\, s \in A - \{s_1\})$$

- If it does not exist already, create a connection between s_1 and s_2.

Topology Representing Networks (TRN)

- Competitve Hebbian Learning:



- Generate at random an input signal according to P(ξ).
- Determine units s_1, s_2, such that:

$$\| \mathbf{w}_{s_1} - \xi \| \le \| \mathbf{w}_s - \xi \| \quad (\forall\, s \in A)$$

$$\| \mathbf{w}_{s_2} - \xi \| \le \| \mathbf{w}_\mathbf{s} - \xi \| \quad (\forall\, s \in A - \{s_1\})$$

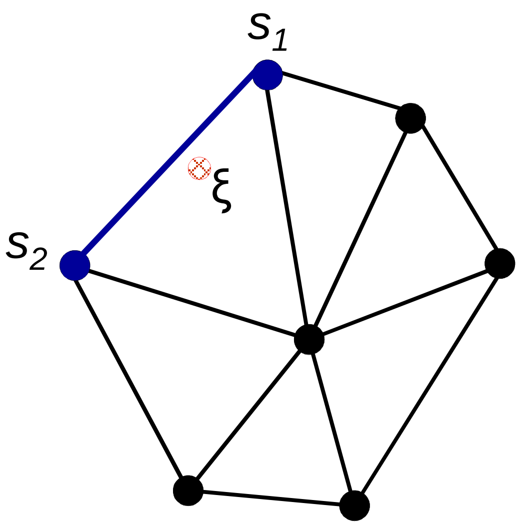
- If it does not exist already, create a connection between s_1 and s_2.

Topology Representing Networks (TRN)

- Adaptation may make edges previously created invalid.
  - An edge aging scheme is used to remove such edges.
  - Each edge has an associated age that is set to zero when the edge is created.

Topology Representing Networks (TRN)

- Edge aging scheme:

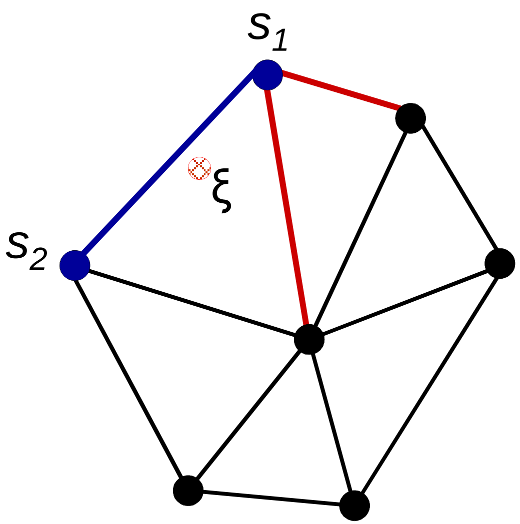


- If the connection between s_1 and s_2 already exists then set its age to zero.
- Increase by one the age of all edges emanating from the winner (s_1).

Centro de Informática
UFPE

Topology Representing Networks (TRN)

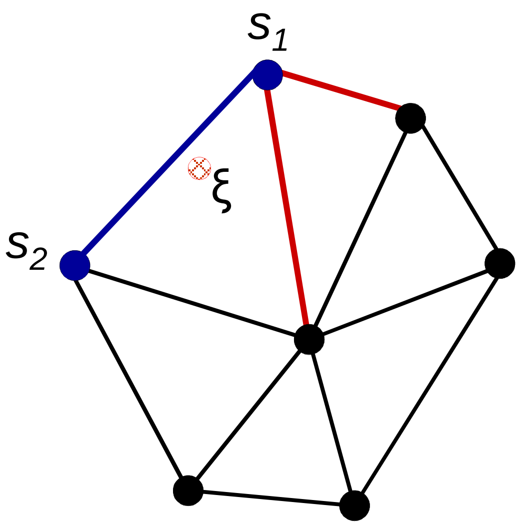
- Edge aging scheme:



- If the connection between s_1 and s_2 already exists then set its age to zero.
- Increase by one the age of all edges emanating from the winner (s_1).

Topology Representing Networks (TRN)

- Edge aging scheme:



- If the connection between s_1 and s_2 already exists then set its age to zero.
- Increase by one the age of all edges emanating from the winner (s_1).
- Remove the connections with an age larger then a threshold (a_{max}).

U F P E

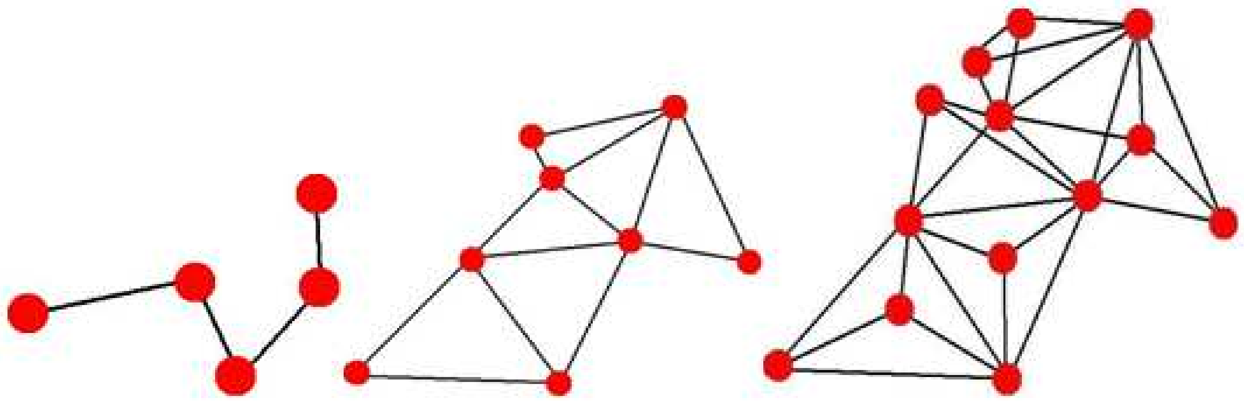
Centro de Informática
UFPE

Growing Cell Structures (GCS)

- Limitation of TRN
  - Fixed number of nodes.
- Growing Cell Structures - GCS (Fritzke, 1993)
  - Same purpose of SOM but does not rely on a predetermined topology.
  - Nodes can be inserted and removed from the map.
  - Edges are learned under some restrictions in order to preserve the dimensionality of the map.

Growing Cell Structures (GCS)

- GCS topology is strictly *k*-dimensional
  - The basic building block is a *k*-dimensional simplex.
  - *k* is chosen in advance.



k=1 for lines   k=2 for triangles   k=3 for tetrahedrons

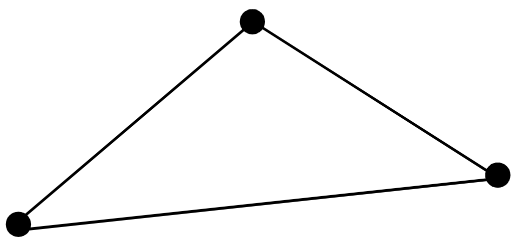
U F P E

Centro de Informática UFPE

Growing Cell Structures (GCS)

- Learning Rule

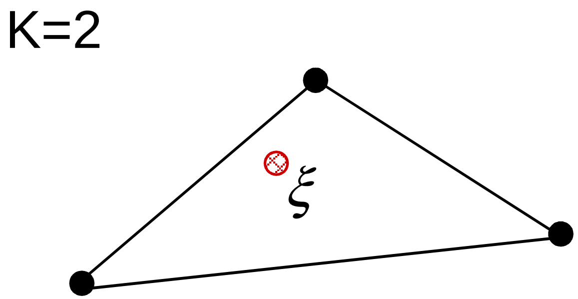
K=2



1. Start with a *k*-dimensional simplex
2. Choose an input signal ξ according to the input distribution $P(\xi)$

Growing Cell Structures (GCS)

- Learning Rule

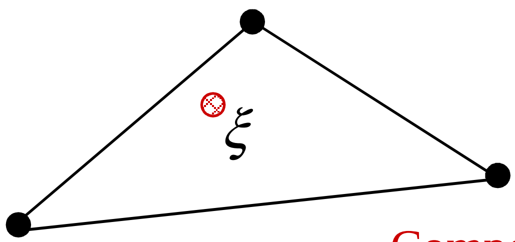


1. Start with a *k*-dimensional simplex
2. Choose an input signal ξ according to the input distribution P(ξ)

Growing Cell Structures (GCS)

- Learning Rule

K=2

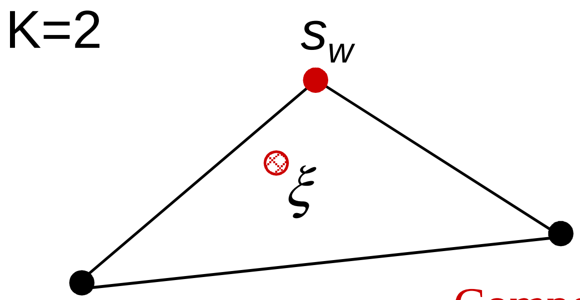


1. Start with a *k*-dimensional simplex
2. Choose an input signal ξ according to the input distribution P(ξ)
3. Determine the winner node (s_w)

Competition $$\| \xi - \mathbf{w}_{s_w} \| < \| \xi - \mathbf{w}_{s_i} \| \quad (\forall s_i \in A)$$

Growing Cell Structures (GCS)

- Learning Rule

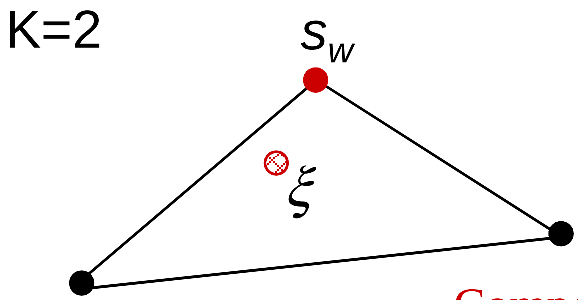


1. Start with a *k*-dimensional simplex
2. Choose an input signal ξ according to the input distribution $P(\xi)$
3. Determine the winner node (s_w):

Competition $\quad \| \xi - \mathbf{w}_{s_w} \| < \| \xi - \mathbf{w}_{s_i} \| \quad (\forall s_i \in A)$

Growing Cell Structures (GCS)

- Learning Rule



1. Start with a *k*-dimensional simplex
2. Choose an input signal ξ according to the input distribution $P(\xi)$
3. Determine the winner node (s_w):

Competition

$$\| \xi - \mathbf{w}_{s_w} \| < \| \xi - \mathbf{w}_{s_i} \| \quad (\forall s_i \in A)$$

4. Move s_w and its direct topological neighbors towards ξ.

Adaptation and Colaboration

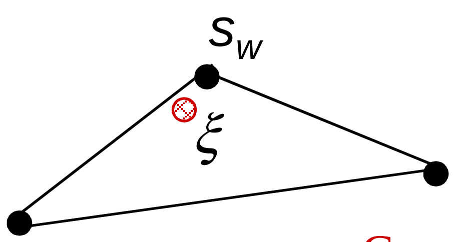
$$\Delta \mathbf{w}_{s_w} = \varepsilon_b (\xi - \mathbf{w}_{s_w})$$

$$\Delta \mathbf{w}_{s_i} = \varepsilon_n (\xi - \mathbf{w}_{s_i}), \quad \forall\, s_i \in N_{s_w}$$

Growing Cell Structures (GCS)

- Learning Rule

K=2



1. Start with a k-dimensional simplex
2. Choose an input signal ξ according to the input distribution P(ξ)
3. Determine the winner node (s_w):

Competition

$$\| \xi - \mathbf{w}_{s_w} \| < \| \xi - \mathbf{w}_{s_i} \| \quad (\forall s_i \in A)$$

4. Move s_w and its direct topological neighbors towards ξ.

Adaptation and Colaboration

$$\Delta \mathbf{w}_{s_w} = \varepsilon_b (\xi - \mathbf{w}_{s_w})$$

$$\Delta \mathbf{w}_{s_i} = \varepsilon_n (\xi - \mathbf{w}_{s_i}), \quad \forall\, s_i \in N_{s_w}$$

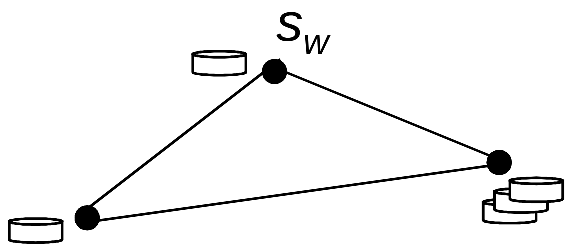
UFPE

Centro de Informática UFPE

Growing Cell Structures (GCS)

- Learning Rule

K=2



Growing Step

Each node have an error variable that is used to determine where to insert a new node in the growing step of the learning algorithm.

UFPE

Centro de Informática UFPE

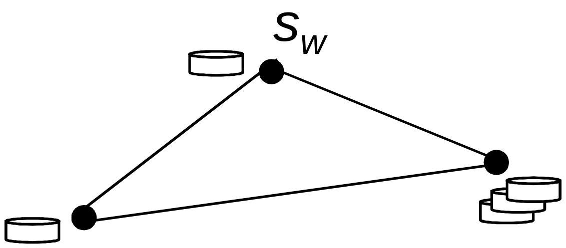
Growing Cell Structures (GCS)

- Learning Rule

5. Add ΔE_{s_w} to the s_w local error variable

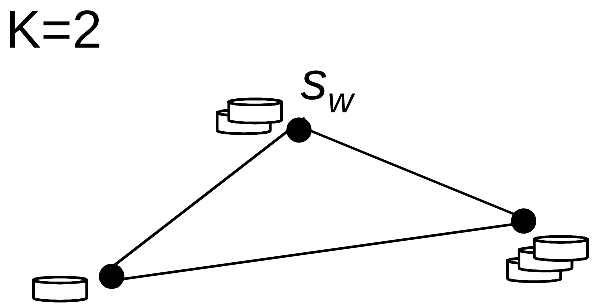
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

K=2



Growing Cell Structures (GCS)

- Learning Rule

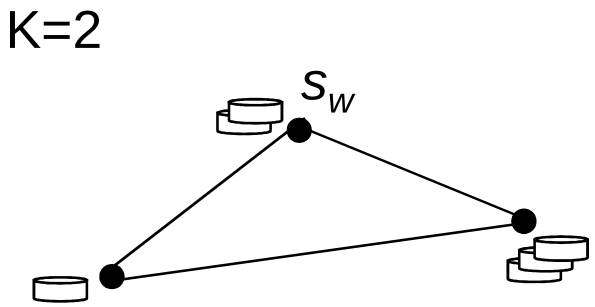


5. Add ΔE_{s_w} to the s_w local error variable

$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

Growing Cell Structures (GCS)

- Learning Rule



5. Add ΔE_{s_w} to the s_w local error variable

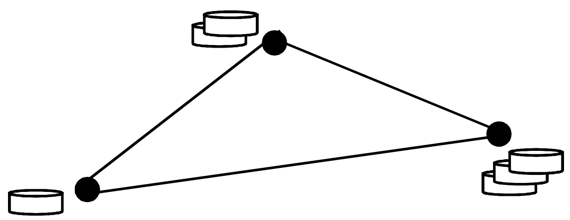
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.

Growing Cell Structures (GCS)

- Learning Rule

K=2



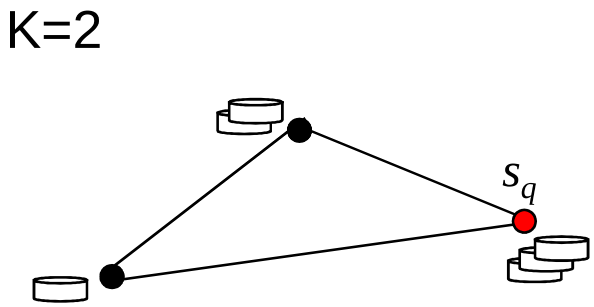
5. Add ΔE_{s_w} to the s_w local error variable

$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node q with the maximum accumulated error:

Growing Cell Structures (GCS)

- Learning Rule



5. Add ΔE_{s_w} to the s_w local error variable

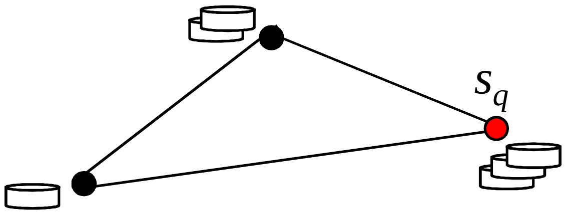
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node q with the maximum accumulated error.

Growing Cell Structures (GCS)

- Learning Rule

K=2



5. Add ΔE_{s_w} to the s_w local error variable

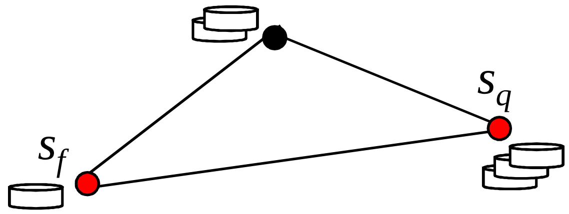
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node q with the maximum accumulated error.
   - Determine the topological neighbor (f) with the larger distance for q.

Growing Cell Structures (GCS)

- Learning Rule

K=2



5. Add ΔE_{s_w} to the s_w local error variable

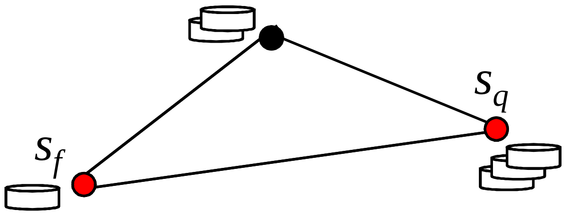
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node q with the maximum accumulated error.
   - Determine the topological neighbor (f) with the larger distance for q.

Growing Cell Structures (GCS)

- Learning Rule

K=2



5. Add ΔE_{s_w} to the s_w local error variable

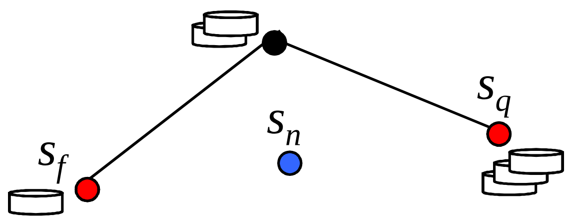
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node q with the maximum accumulated error.
   - Determine the topological neighbor (f) with the larger distance for q.
   - Insert a new node by splitting the edge between q and f.

Growing Cell Structures (GCS)

- Learning Rule

K=2



Edge Split Procedure

• Connect between s_n to s_q and to s_f, and undo the connection between s_q and s_f.

5. Add ΔE_{s_w} to the s_w local error variable

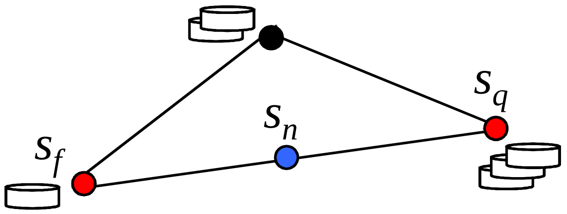
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the topological neighbor (s_f) with the larger distance for s_q.
   - Insert a new node (s_n) by splitting the edge between s_q and s_f..

Growing Cell Structures (GCS)

- Learning Rule

K=2



Edge Split Procedure

• Connect between s_n to s_q and to s_f, and undo the connection between s_q and s_f.

5. Add ΔE_{s_w} to the s_w local error variable

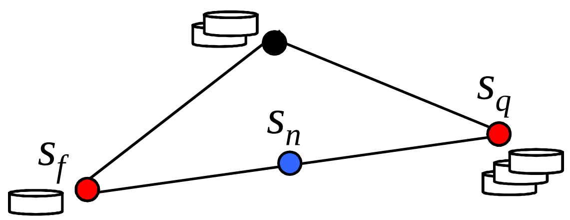
$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the topological neighbor (s_f) with the larger distance for s_q.
   - Insert a new node (s_n) by splitting the edge between s_q and s_f..

Growing Cell Structures (GCS)

- Learning Rule

K=2



Edge Split Procedure

• Connect between s_n to s_q and to s_f, and undo the connection between s_q and s_f.

• Connect s_n to all common neighbors of s_q and s_f

5. Add ΔE_{s_w} to the s_w local error variable

$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the topological neighbor (s_f) with the larger distance for s_q.
   - Insert a new node (s_n) by splitting the edge between s_q and s_f..

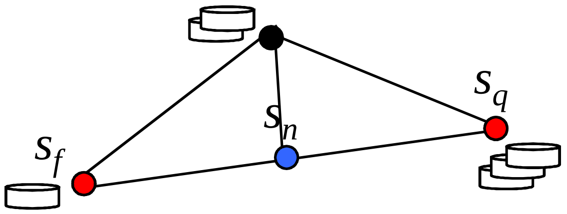
UFPE

Centro de Informática
UFPE

Growing Cell Structures (GCS)

- Learning Rule

K=2



Edge Split Procedure

• Connect between s_n to s_q and to s_f, and undo the connection between s_q and s_f

• Connect s_n to all common neighbors of s_q and s_f

5. Add ΔE_{s_w} to the s_w local error variable

$$\Delta E_{s_w} = \| \mathbf{w}_{s_{wi}} - \xi \|^2$$

6. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the topological neighbor (s_f) with the larger distance for s_q.
   - Insert a new node (s_n) by splitting the edge between s_q and s_f..

Growing Neural Gas (GNG)

- Fritzke (1995)
- Growing Neural Gas (GNG) can be seen as:
  - A variant of the GCS without its strict topological constraints, or
  - An incremental variant of the TRN
- Purpose:
  - To generate a graph structure which reflects the topology of the input data manifold (topology learning).
    - This graph has a dimensionality which varies with the dimensionality of the input data.

Growing Neural Gas (GNG)

- Learning Rule:
  - An edge aging scheme is used to remove edges that become invalid during the learning process (as TRN).
  - An error variable is attached to each node and used to determine where to insert a new node (as GCS).

Centro de Informática UFPE

U F P E

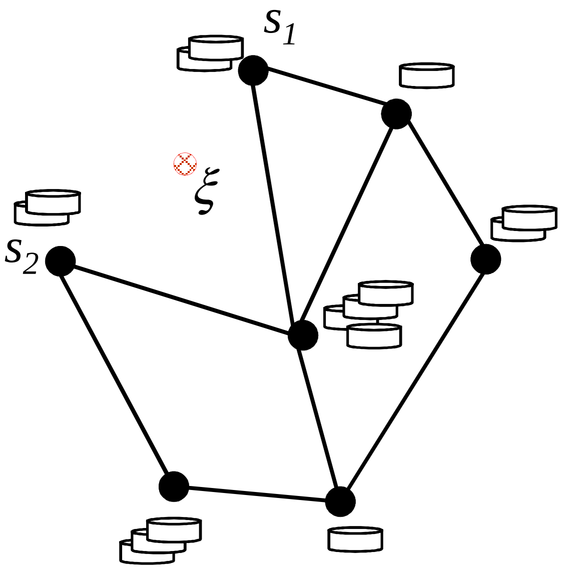
Growing Neural Gas (GNG)

- Learning Rule:

1. Start the map with two units s_a and s_b at random positions in $\boldsymbol{R}^n$

Growing Neural Gas (GNG)

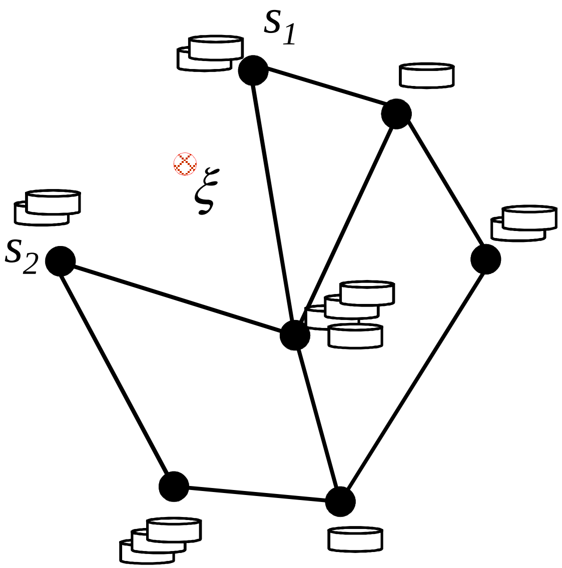
- Learning Rule:



2. Generate an input signal ξ according to P(ξ)

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

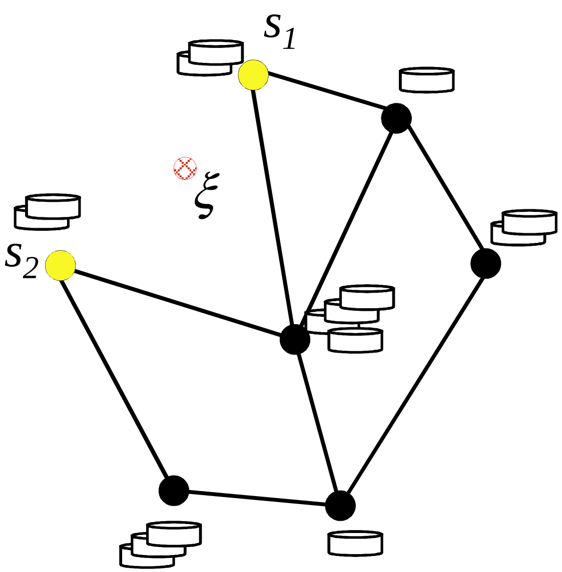
$$\| \mathbf{w}_{s_1} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \quad \text{and}$$

$$\| \mathbf{w}_{s_2} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

Centro de Informática
UFPE

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

$$\| \mathbf{w}_{s_1} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \quad \text{and}$$

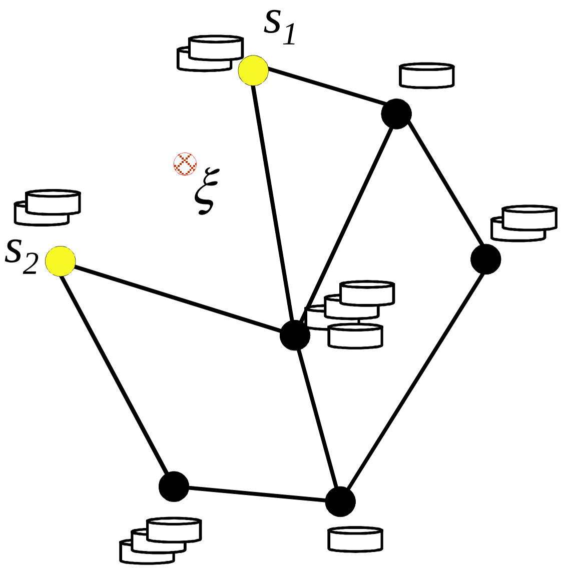
$$\| \mathbf{w}_{s_2} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

Competition

Centro de Informática
UFPE

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

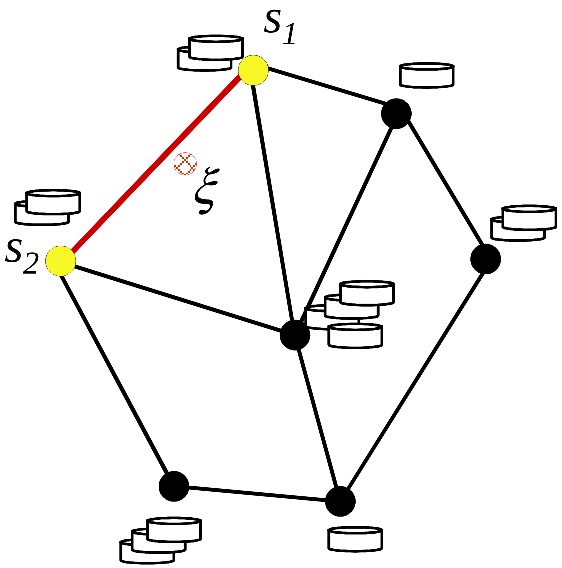
$$\| \mathbf{w}_{s_1} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \qquad \text{and}$$

$$\| \mathbf{w}_{s_2} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to $P(\xi)$
3. Determine the units s_1 e s_2 nearest to ξ

$$\| \mathbf{w}_{s_1} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \qquad \text{and}$$

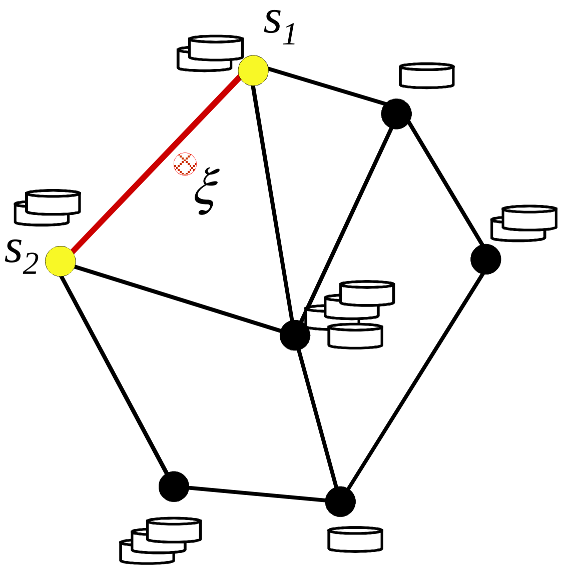
$$\| \mathbf{w}_{s_2} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero

Topology Learning

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

$$\| \mathbf{w}_{s_1} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \qquad \text{and}$$

$$\| \mathbf{w}_{s_2} - \xi \| \le \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero
5. Move s_1 and its direct topological neighbors towards ξ.

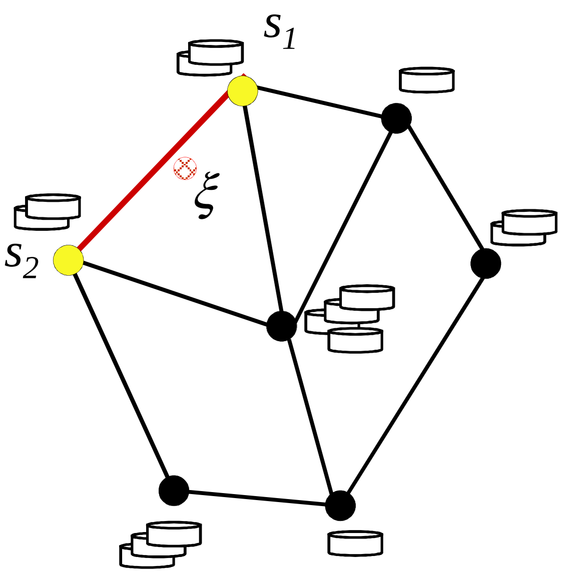
$$\Delta \mathbf{w}_{s_1} = \varepsilon_b (\xi - \mathbf{w}_{s1})$$

$$\Delta \mathbf{w}_{s_i} = \varepsilon_n (\xi - \mathbf{w}_{s_i}) \quad \forall s_i \in N_{s_1},$$

Centro de Informática
UFPE

Growing Neural Gas (GNG)

- Learning Rule:



2. Generate an input signal ξ according to $P(\xi)$
3. Determine the units s_1 e s_2 nearest to ξ

$$\|\mathbf{w}_{s_1} - \xi\| \leq \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A \qquad \text{and}$$

$$\|\mathbf{w}_{s_2} - \xi\| \leq \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero
5. Move s_1 and its direct topological neighbors towards ξ.

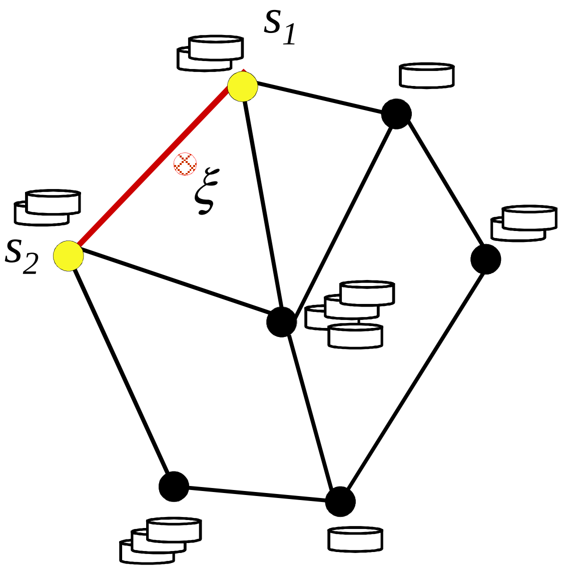
$$\Delta\mathbf{w}_{s_1} = \varepsilon_b(\xi - \mathbf{w}_{s1})$$

$$\Delta\mathbf{w}_{s_i} = \varepsilon_n(\xi - \mathbf{w}_{s_i}) \quad \forall s_i \in N_{s_1},$$

Competition and Collaboration

Growing Neural Gas (GNG)

- Learning Rule:



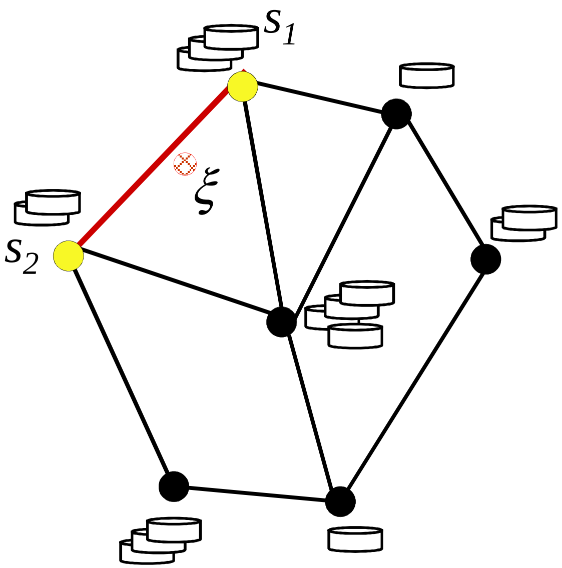
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

Centro de Informática UFPE

Growing Neural Gas (GNG)

- Learning Rule:

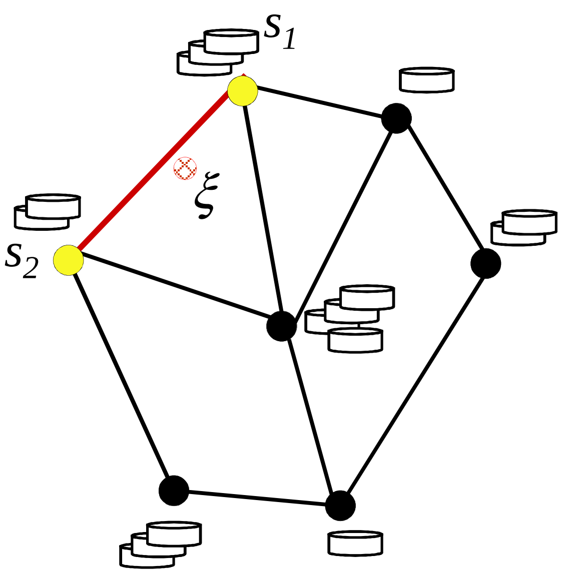


6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

Growing Neural Gas (GNG)

- Learning Rule:



6. Add ΔE_{s_1} to the s_1 local error variable

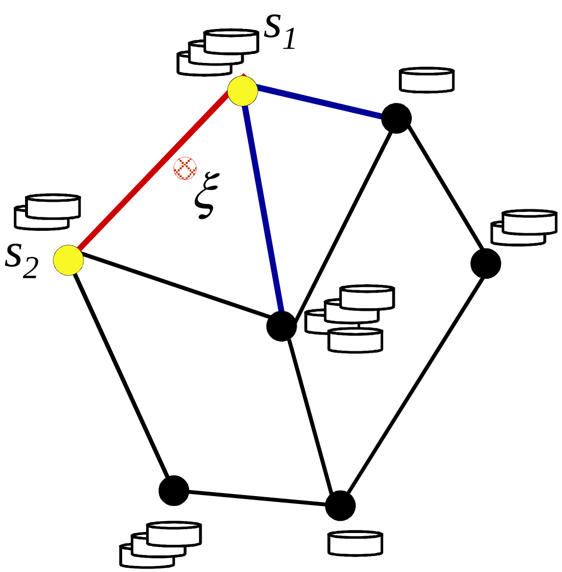
$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.

Centro de Informática UFPE

Growing Neural Gas (GNG)

- Learning Rule:



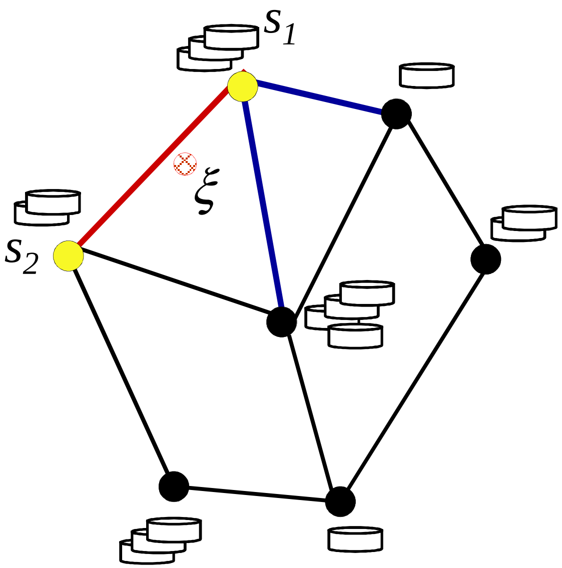
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.

Growing Neural Gas (GNG)

- Learning Rule:



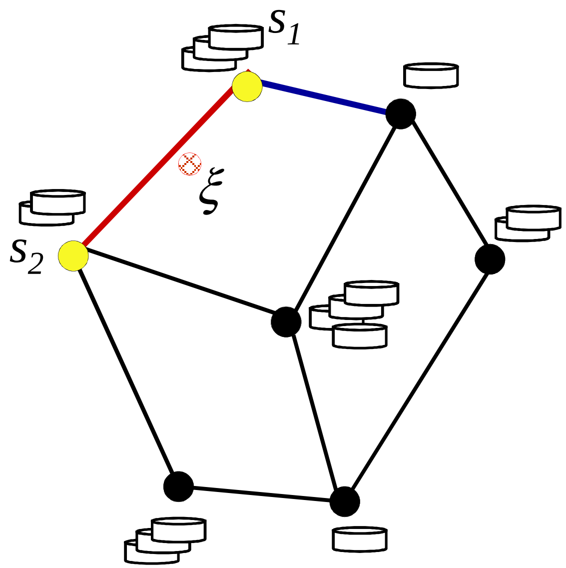
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.
8. Remove the connections with an age larger then a threshold (a_{max})

Growing Neural Gas (GNG)

- Learning Rule:



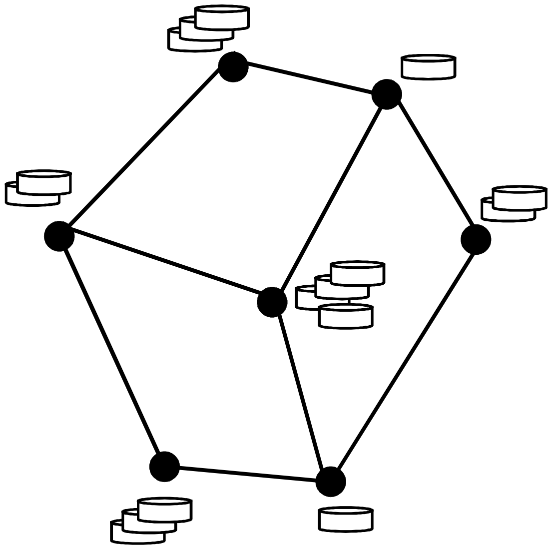
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.
8. Remove the connections with an age larger then a threshold (a_{max})
   - If this results in units having no emanating edges, remove them as well.

Growing Neural Gas (GNG)

- Learning Rule:

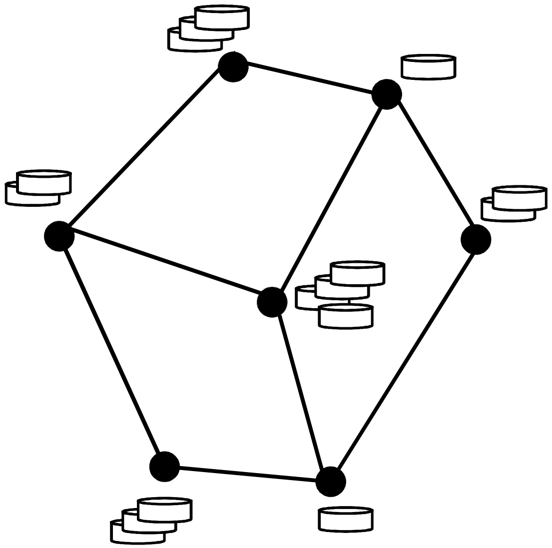


9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.

Growing Step

Growing Neural Gas (GNG)

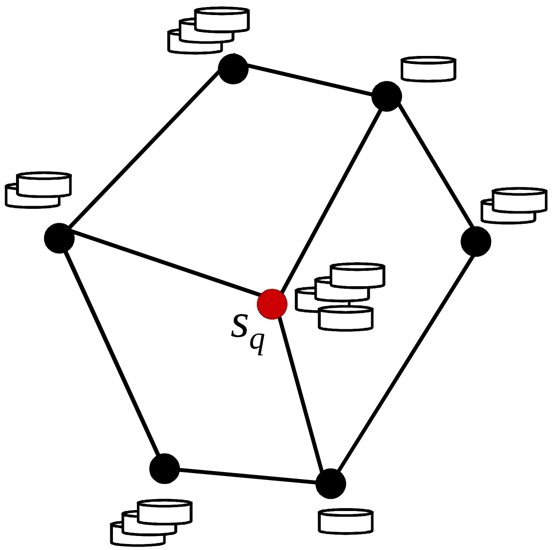
- Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.

Growing Neural Gas (GNG)

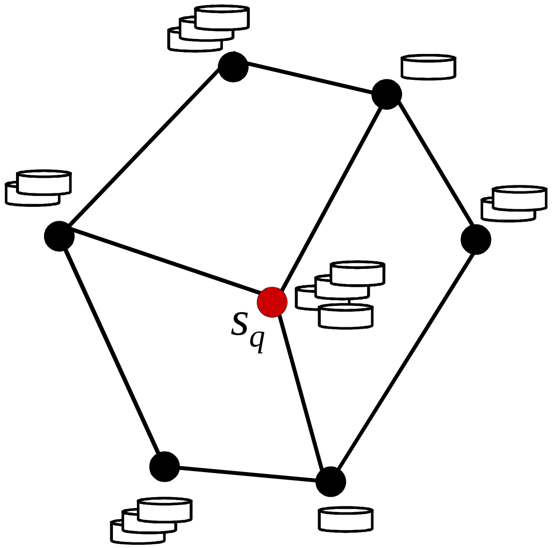
- Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.

Growing Neural Gas (GNG)

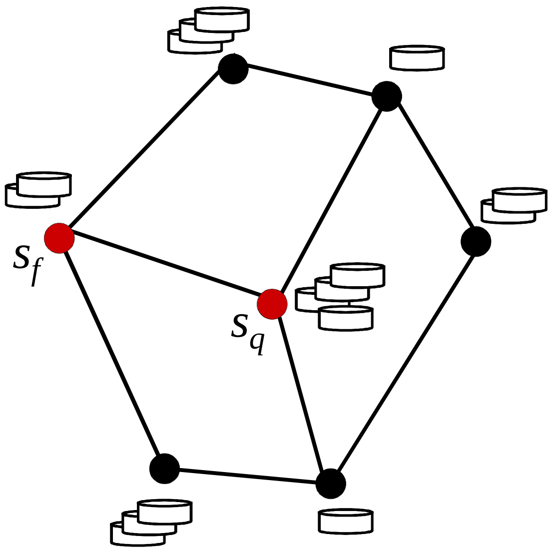
- Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of $s_{q,}$ with the largest error variable.

Growing Neural Gas (GNG)

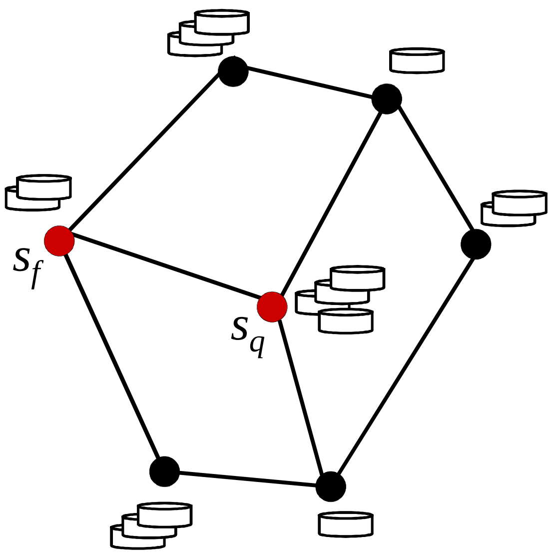
- Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of s_q, with the largest error variable.

Growing Neural Gas (GNG)

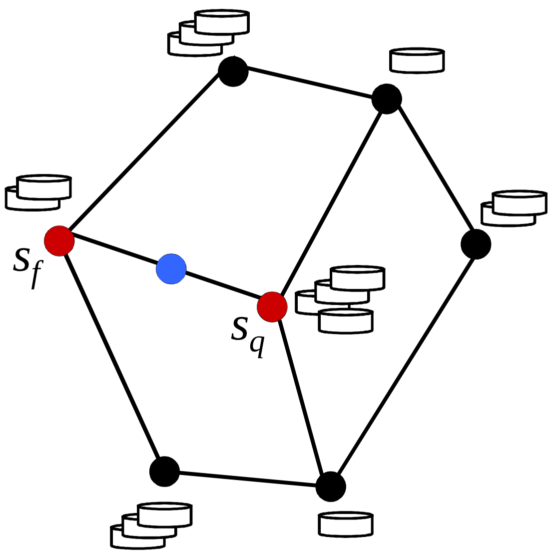
- Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of s_q, with the largest error variable.
   - Insert a new unit (s_r) between s_q and s_f.

Growing Neural Gas (GNG)

- Learning Rule:



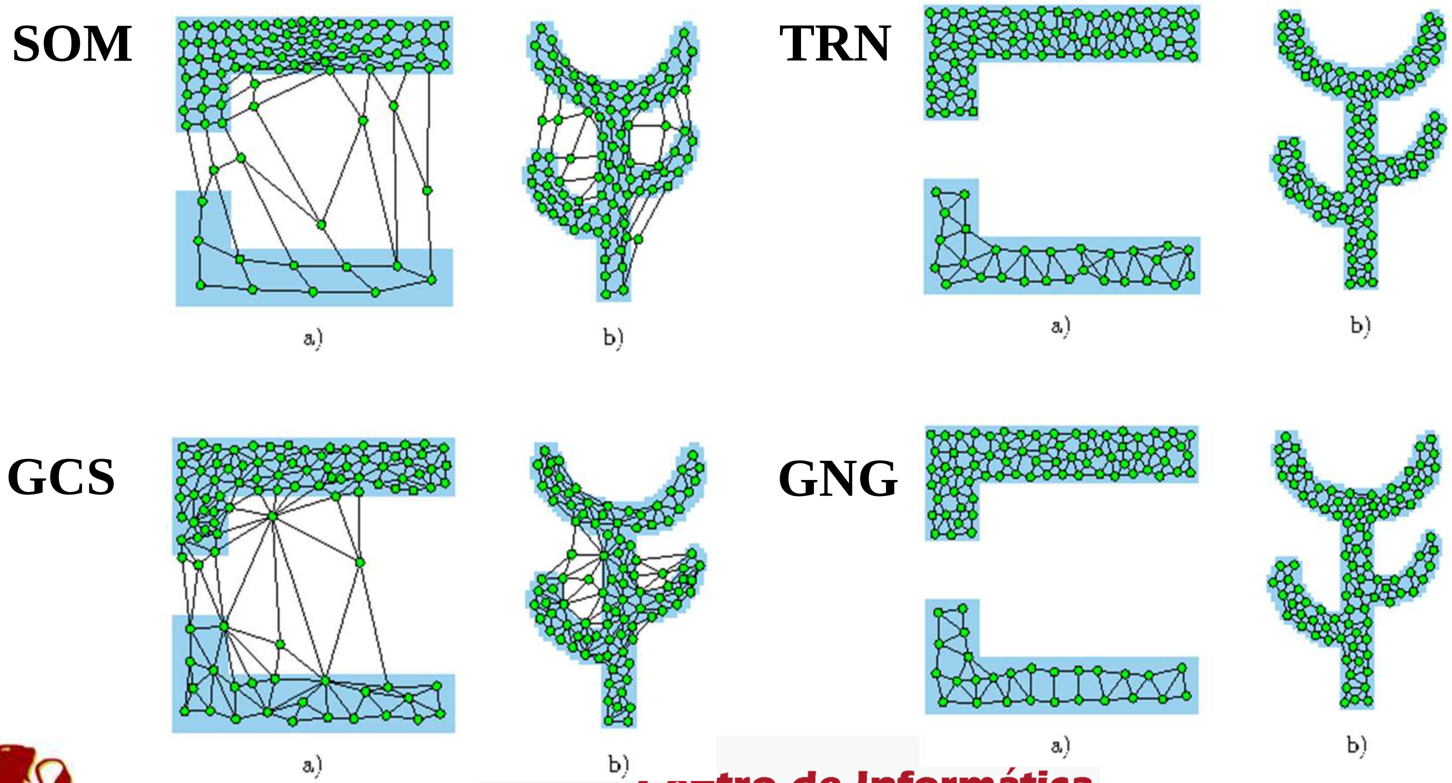
9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f, topological neighbor of s_q, with the largest error variable.
   - Insert a new node (s_r) between s_q and s_f
   - Insert edges connecting the new node s_r with nodes s_q and s_f , and remove the original edge between *sq* and s_f .

Centro de Informática UFPE

U F P E

Comparison of Results

- Mappings created by the SOM and its dynamic variants.



References

- Fritzke, B. (1994a). Growing cell structures - a self-organizing network for unsupervised and supervised learning. Neural Networks, 7(9): 1441 – 1460.
- Fritzke, B. (1995a). A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, 7: 625-632.
- Fritzke, B. (1996). Unsupervised ontogenetic networks. *Handbook of Neural Computation*, IOP Publishing and Oxford University Press.
- Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1): 59-69.
- Martinetz, T. M. & Schulten, K. J. (1994). Topology representing networks. *Neural Networks*, 7 (3): 507-522.