A Growing Neural Gas
Network Learns Topologies

Presentation by Jonathan Beall
on the paper by Bernd Fritzke
Growing Neural Gas

- Adjust units and connections to approximate input data topology.
- Unsupervised learning
- Dimensionality reduction
- Topology learning
Competitive Hebbian Learning (CHL)

- For each input signal, connect the closest two centers.
- Can add in aging to eliminate now-irrelevant connections.
- Creates a subgraph of the Delaunay triangulation.
CHL and Neural Gas

- Place all centers
- For each input
  - Perform CHL
  - Move the k closest centers toward the input
- k decays over time—annealing.
GNG in Action
GNG in Action
GNG in Action
GNG in Action
GNG in Action
GNG in Action
Growing Neural Gas Algorithm

- Start with 2 random units
- Generate a random signal $\xi$ and find two nearest units, $s_1$ and $s_2$.
- Increment the age of all edges coming from $s_1$.
- Add the squared distance $||s_1 - \xi||^2$ to $s_1$'s error counter.
Move $s_1$ and its direct neighbors $s_n$ towards $\xi$ by fractions $\epsilon_b$ and $\epsilon_n$, respectively, of the total distance.

If $s_1$ and $s_2$ are connected, set the age of the edge connecting them to 0, otherwise create an edge between them.

Remove any edges with ages larger than $a_{\text{max}}$. If this leaves any units with no edges, delete them.
Every \( \lambda \) input signals, insert a new unit:

- Find the unit \( q \) with the maximum error.
- Find \( q \)'s neighbor \( f \) with the largest error.
- Insert a new unit \( r \) between \( q \) and \( f \)
- Decrease the errors of \( q \) and \( f \) by multiplying them by a constant \( \alpha \), and then set the error of \( r \) to \( q \)'s new error.
Decrease all error variables by multiplying them by a constant $d$.

If the network is not yet complete (size, performance...), begin again by generating a new signal, $\xi$. 
GNG Applet

Benefits of GNG

- Variable dimensionality
- Constant parameters
- No set termination time
- Soft competition helps skip over local minima
- Insertion based on batch accumulated errors helps to avoid overfitting
- Efficient—time increases linearly with the number of links
Drawbacks of GNG

- No cooling schedule, so no long term stability
- 7 parameters must be set
- CHL is not batch-driven, is sensitive to noise
- Can leave stranded units when exposed to rapidly changing data (GNG-U)
- Cannot visualize higher dimension GNGs easily.
Applications of GNG

- Incremental radial basis function networks
- 2-stage hybrid classifier (Baraldi and Blonda)
- Stability assessment of electric power systems (Rehtans and Leder)
Sources


