Neural gas algorithm

Algorithm:

1. Initialize:

- inputs, $\{X\}_1^N = \{x_1, x_2, ...x_N\}$,
- neurons, $\{U\}_1^K = \{u_1, u_2, ...u_K\}$, $K < N$ // "U" stands for "unit"
- topology (connection matrix) $A = [\mathbf{0}]_{K,K}$

where $x_i, u_i \in \mathbb{R}^n$

2. Set maximum age for connections, τ

3. Loop until maximum number of iterations M: // loop through inputs

i. Randomly select an input x_i

ii. Loop over neurons $\{U\}_1^K$:

- Let current neuron be u_c
- Find $|B_c|$ where $B_c = \{u_b \mid norm(u_b, x_i) < norm(u_c, x_i)\}$

iii. Sort neurons by distance to x_i, $\{U_{sorted}\} = \{u_{i_0}, u_{i_1}...u_{i_{K-1}}\}$

iv. Update/replace all neurons with,

$$u_c = u_c + \epsilon.e^{-|B_c|/\lambda}(x_i - u_c)$$

for c={1,2,...K}.

v. Make connections between closest neurons (u_{i_0}, u_{i_1}) if connection doesn't exist already else increase connection age, $C_{u_{i_0},u_{i_1}} = C_{u_{i_0},u_{i_1}} + 1$

vi. if $C_{u_{i_0},u_{i_1}} > \tau$, set $C_{u_{i_0},u_{i_1}} = 0$

In the original paper, the parameters λ, ϵ, τ were defined by a function of time (iteration),

$g(t) = g_i \frac{g_f}{g_i}^{\frac{t}{t_{max}}}$ where,

$$\lambda_i = 30, \lambda_f = 0.01, \epsilon_i = 0.3, \epsilon_f = 0.05, \tau_i = 20, \tau_f = 200, t_{max} = 40000$$

> Written with StackEdit.