Algoritmo de la GROWING NEURAL GAS

La red la consideramos formada por:

- Un conjunto $\mathcal{A}$ de unidades y/o nodos. Cada unidad $c \in \mathcal{A}$ tiene asociado un vector de referencia $w_c \in \mathbb{R}^n$. Los vectores de referencia pueden verse en correspondencia a posiciones del espacio de entrada de las correspondientes unidades de $\mathcal{A}$.
- Un conjunto $\mathcal{N}$ de conexiones (o aristas) entre pares de unidades de $\mathcal{A}$. Estas conexiones no se encuentran ponderadas, simplemente se encuentran definidas para propósitos de definición de la estructura topológica que la red está representando.

El algoritmo [1] presenta los siguientes pasos:

1. La red comienza con dos unidades a y b, las cuales se posicionan de forma aleatoria en w_a y w_b en $\mathbb{R}^n$.
2. Generamos/Procesamos una señal de entrada ξ en concordancia con una función densidad de probabilidad $P(\xi)$.
3. Encontrar *entre las unidades que conforman la red en esta iteración la unidad más cercana* s_1*, y la unidad segunda más c*ercana s_2.
4. Incrementar la edad de todos los arcos que comunican s_1 con el resto de unidades de $\mathcal{A}$.
5. Añadir a la variable contadora local $\Delta\text{error}(s_1)$ el cuadrado de la distancia entre la señal de entrada ξ y la unidad más cercana s_1:

$$\Delta\text{error}(s_1) = \left\| w_{s_1} - \xi \right\|^2$$

6. Mover s_1 y sus vecinos topológicos (unidos mediante $\mathcal{N}$) directos hacia ξ mediante las fracciones ϵ_b y ϵ_n, respectivamente, de la distancia total que los separa:

$$\Delta w_{s_1} = \epsilon_b(\xi - w_{s_1})$$

$$\Delta w_{s_n} = \epsilon_n(\xi - w_n) \text{ para todos los vecinos directos } n \text{ de } s_1.$$

7. Si s_1 y s_2 están conectados por una arista de $\mathcal{N}$, establecer la edad de la arista a 0. Si tal arista no existe en $\mathcal{N}$, entonces se debe crear.
8. Eliminar todas las aristas de $\mathcal{N}$ que tengan una edad mayor que $a_{\max}$. Si este proceso provoca que aparezcan puntos/unidades donde no tienen aristas en $\mathcal{N}$, entonces debemos borrar también dichos puntos/unidades.
9. Si se han generado/procesado tantas señales de entrada ξ como un número entero que es múltiplo de λ, insertar una nueva unidad siguiendo el siguiente procedimiento:
    a. Determinar la unidad q con el máximo error acumulado.
    b. Inserta una nueva unidad r en medio de q y de su vecino f con el mayor error:

$$w_r = {}^1\!/_2\,(w_q + w_f)$$

c. Insertar las nuevas aristas en $\mathcal{N}$ que conectan la nueva unidad r con las unidades q y f, y elimina la arista original entre q y f.
d. Decrementa las variables de error de q y f mediante la multiplicación de estas por una constante α. Inicializa la variable de error de r con un nuevo valor de la variable de error de q.

10. Decrementa todas las variables de error mediante la multiplicación de estas con una constante d.
11. Si un criterio de parada (por ejemplo, tamaño de la red o alguna medida de rendimiento) no ha sido alcanzado, ir al paso 1.

Referencias

1. Fritzke, B., A Growing Neural Gas Network Learns Topologies, ¿…pendiente de mirar…?

No parece que tenga mucho que ver con el movimiento de un gas (…), parece que tiene mucha más implicación el proceso de Triangulación de Delaunary. Tengo que mirar los artículos de Martinetz, T. M. (*1993, Competitive Hebbian learning rule forms perfectly topology preserving maps*) y Martinetz, T. M. y Schulten, K. J. (*1991, A "neural-gas" network learns topologies; 1994, Topology representing networks*).

1.- Parámetros que son necesarios establecer para poder lanzar el proceso de aprendizaje de la GNG:

epsilon_b

epsilon_n

lambda >>> eta

Alpha

D >>> delta