Redes de Neuronas de Base Radial

Introducción

- Redes multicapa con conexiones hacia delante
- Única capa oculta
- Las neuronas ocultas poseen carácter local
  - Cada neurona oculta se activa en una región distinta del espacio de entrada
  - El carácter local se debe al uso de las funciones de base radial como funciones de activación. Generalmente la función gaussiana.
- Las neuronas de salida realizan una combinación lineal de las activaciones de las neuronas ocultas
- Se deben fundamentalmente a
  - Moody y Darken (1989)
  - Renals (1989)
  - Poggio y Girossi (1990)

Introducción

- Son aproximadores universales (como el MLP)
  - Demostrado formalmente por Park y Sandberg (1991)
- Las funciones de base radial (RBF) definen hiperesferas o hiperelipses que dividen el espacio de entrada
- Cada RBF (cada neurona) construye una aproximación local no lineal en una determinada región del espacio de entrada
- Las RBNR construyen aproximaciones que son combinaciones lineales de múltiples funciones locales no lineales

- Se han aplicado a gran variedad de problemas
  - Análisis de series temporales
  - Procesamiento de imágenes
  - Reconocimiento automático del habla
  - Diagnósticos médicos, etc

Arquitectura

- **Tres capas de neuronas**
  - Capa de entrada

    Transmiten las señales de entrada a las neuronas ocultas sin realizar procesamiento

    Las conexiones de la capa de entrada a la capa oculta no llevan pesos asociados

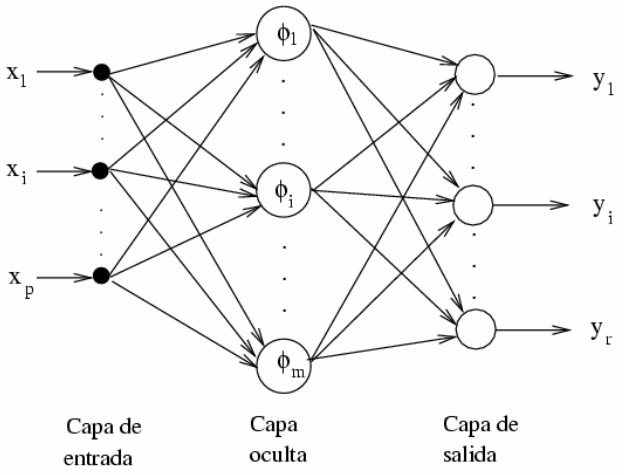
  - Capa oculta

    Realizan una transformación local y no lineal de dichas señales

  - Capa de salida

    Realiza una combinación lineal de las activaciones de las neuronas ocultas

Arquitectura



Arquitectura. Activaciones de las neuronas

- Espacio de entrada de **p** dimensiones
- Espacio de salida de **r** dimensiones
- **m** neuronas ocultas

Patrón de entrada: $X(n)=(x_1(n), x_2(n),.... x_p(n))$

Las activaciones de las neuronas de salida $y_k(n)$ para el patrón de entrada n serán

$$y_k(n) = \sum_{i=1}^{m} w_{ik}\phi_i(n) + u_k \text{ para } k = 1, 2, ..., r$$

- w_{ik}: peso de la conexión de la neurona oculta i a la de salida k
- $\phi_i(n)$: activación de la neurona oculta i
- u_k : umbral de la neurona de salida k

Arquitectura. Activaciones de las neuronas

Funciones de base radial: determinan las activaciones de las neuronas ocultas en función del vector de entrada

$$\phi_i(n) = \phi\left(\frac{\|X(n) - C_i\|}{d_i}\right) \text{ para } i = 1, 2, ..., m$$

ϕ es una función de base radial

$C_i=(c_{i1}, c_{12}, ...c_{ip})$ son vectores: centros de las funciones de base radial

d_i son números reales: desviaciones de las funciones

$\|\cdot\|$ es la distancia euclídea desde el vector de entrada al centro de la función

$$\|X(n) - C_i\| = \sqrt{\sum_{j=1}^{p} (x_j(n) - c_{ij})^2}$$

Arquitectura. Activaciones de las neuronas

Funciones de base radial más habituales:

- Función gausiana:

$$\phi(r) = exp(\frac{-r^2}{2})$$

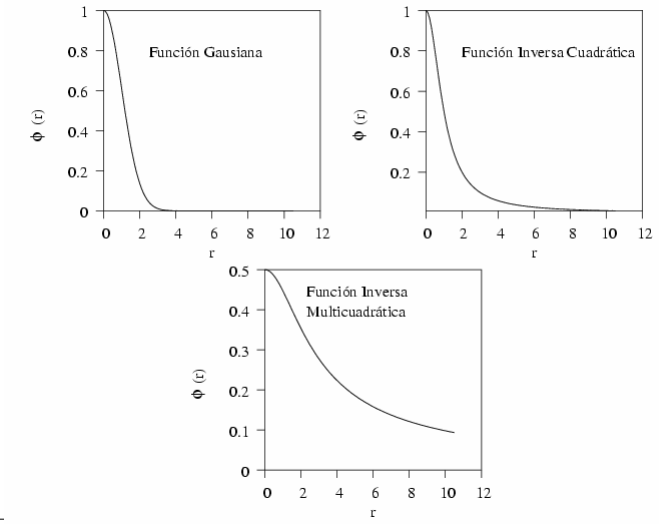
- Función inversa cuadrática:

$$\phi(r) = \frac{1}{1 + r^2}$$

- Función inversa multicuadrática:

$$\phi(r) = \frac{1}{\sqrt{1 + r^2}}$$

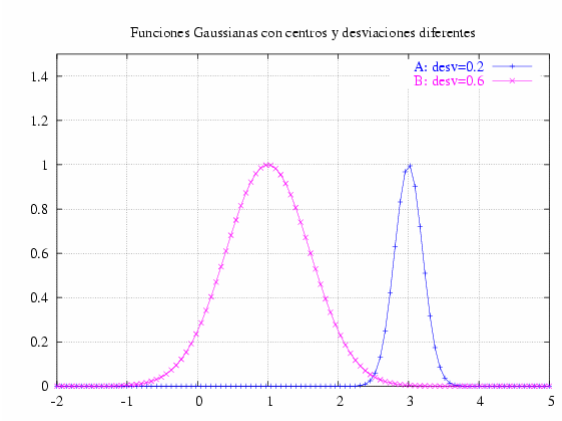
Arquitectura. Activaciones de las neuronas



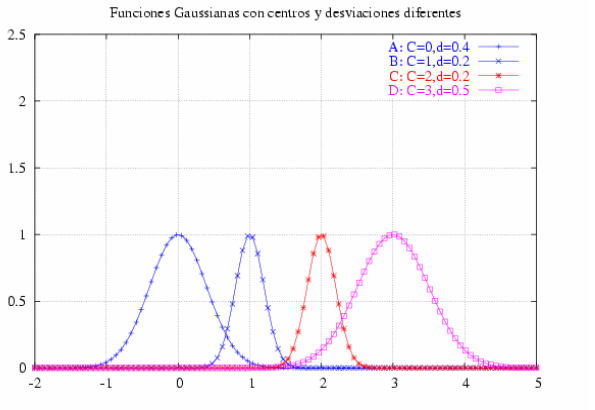
Arquitectura. Activaciones de las neuronas

La más utilizada es la función gausiana. Por tanto:

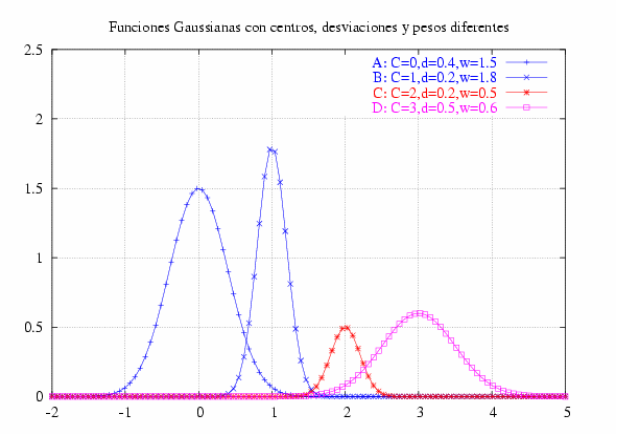
$$\phi_i(n) = e^{-\frac{\|X(n)-C_i\|^2}{2d_i^2}}$$



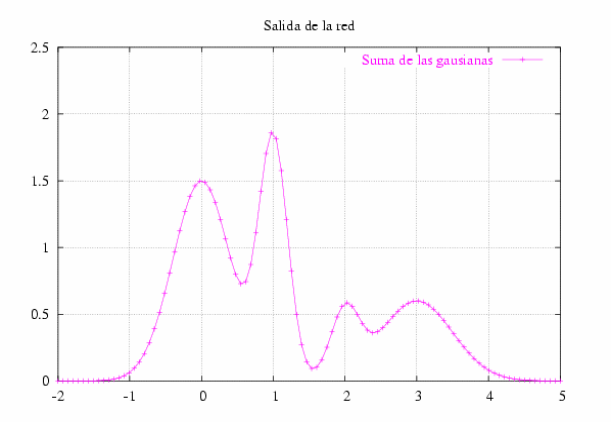
Arquitectura. Activaciones de las neuronas



Arquitectura. Activaciones de las neuronas



Arquitectura. Salida de la red



Aprendizaje

- Consiste en la determinación de todos los parámetros
  - **Centros**
  - **Desviaciones**
  - **Pesos** de la capa oculta a la capa de salida

- Como las capas de la red realizan tareas diferentes, parece razonable separar el proceso de optimización de los parámetros de la capa oculta, y los de la capa de salida utilizando diferentes técnicas
  - Centros y desviaciones: proceso guiado por una optimización en el espacio de entrada
  - Pesos: optimización en base a las salidas que se desea obtener

Aprendizaje

- **Métodos de aprendizaje más importantes**
  - Método híbrido
    - Realiza el aprendizaje en dos fases:
      - **Fase no supervisada**: determinación de los centros y amplitudes de las neuronas de la capa oculta
      - **Fase supervisada**: determinación de pesos y umbrales de la capa de salida
    - Es el más utilizado
    - Se basa en lo dicho en el apartado anterior sobre la separación de técnicas de optimización
    - Conserva las características locales de la red
  - Método totalmente supervisado
    - realiza una adaptación supervisada de todos los parámetros de la red

Aprendizaje. Método híbrido

- **Fase no supervisada**
  - Se determinan de forma no supervisada los siguientes parámetros de las neuronas ocultas:
    - Centros
    - Desviaciones
  - Los centros y las desviaciones de las funciones de base radial deben ser determinados con el objetivo de agrupar el espacio de entrada en diferentes clases
  - El representante de cada clase será el centro de la función de base radial y la desviación vendrá dada por la amplitud de cada clase
- **Fase supervisada**
  - Se determinan de forma supervisada los pesos y umbrales de la capa de salida

Aprendizaje. Método híbrido

Determinación de los centros

Se utilizará un algoritmo de clasificación no supervisado que permita dividir el espacio de entrada en clases o clusters

- El número de clusters es el número de neuronas ocultas en la red de base radial
- El método más utilizado es el algoritmo de K-medias, aunque cualquier algoritmo de clasificación no supervisado podría ser utilizado

- Algoritmo de K-medias
  - J. MacQueen, 1967
  - algoritmo de clasificación no supervisado mediante el cual el espacio de patrones de entrada se divide en K clases o regiones
  - El representante de cada una de estas clases, C_i, será el centro de la neurona oculta i.

Aprendizaje. Método híbrido

Algoritmo de K-medias

- Dichos centros se determinan con el objetivo de minimizar las distancias euclídeas entre los patrones de entrada y el centro más cercano

$$J = \sum_{i=1}^{K} \sum_{n=1}^{N} M_{in} \|X(n) - C_i\|$$

donde *N* es el número de patrones, || . || es la distancia euclídea, *X(n)* es el patrón de entrada *n* y M_{in} es la función de pertenencia, que vale 1 si el centro *Ci* es el más cercano al patrón *X(n)*, y 0 en otro caso, es decir:

$$M_{in} = \begin{cases} 1 & \text{si } \|X(n) - C_i\| < \|X(n) - C_s\| \ \forall s \neq i,\ s = 1, 2, ..., K \\ 0 & \text{en otro caso} \end{cases}$$

Aprendizaje. Método híbrido

Algoritmo de K-medias

Dado el número de clases K, el conjunto de patrones de entrada y los centros de las clases, los pasos para la aplicación del algoritmo son los siguientes:

1. Se inicializan aleatoriamente los centros de los K clusters (centroides)

2. Se asignan N_i patrones de entrada a cada cluster i del siguiente modo
   - El patrón X(n) pertenece al cluster i si

$$\|X(n) - C_i\| < \|X(n) - C_s\|$$

$$\forall\ s \neq i \text{ con } s = 1, 2, ..., K.$$

   - Por tanto, cada cluster tendrá asociado un determinado número de patrones de entrada, aquellos más cercanos a su centroide

Aprendizaje. Método híbrido

Algoritmo de K-medias

3. Se calcula la nueva posición de los centroides como la media de todos los patrones que pertenecen al cluster, es decir:

$$c_{ij} = \frac{1}{N_i} \sum_{n=1}^{N} M_{in} x_j(n) \text{ para } j = 1, 2, ...p\ ,\ i = 1, 2, ..., K$$

4. Se repiten los pasos 2 y 3 hasta que las nuevas posiciones de los centroides no se modifiquen respecto a su posición anterior, es decir hasta que:

$$\|C_i^{nuevo} - C_i^{anterior}\| < \varepsilon\ \forall\ i = 1, 2, ..., K$$

Aprendizaje. Método híbrido

Algoritmo de K-medias

Características

- El algoritmo de K-medias es un método fácil de implementar y usar
- suele ser un algoritmo bastante eficiente en problemas de "clusterización", pues converge en pocas iteraciones hacía un mínimo de la función J, aunque podría tratarse de un mínimo local.
- Principal inconveniente: su dependencia de los valores iniciales asignados a cada centroide (mínimos locales)

Aprendizaje. Método híbrido

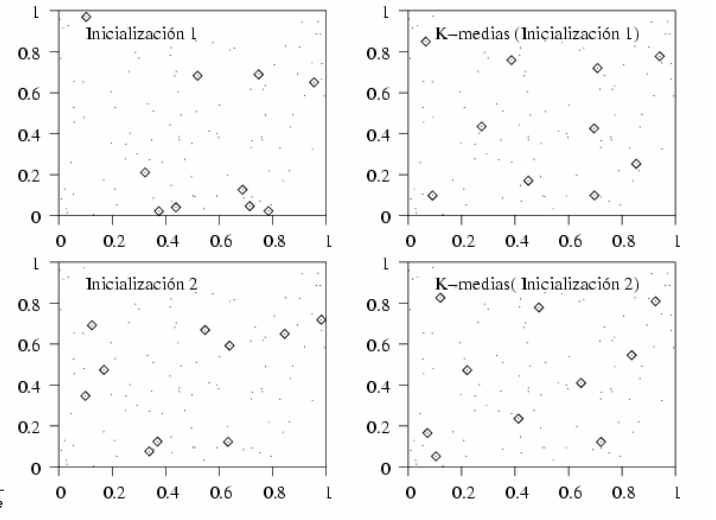
Algoritmo de K-medias

Dependencia de la inicialización

- Ejemplo:
  - Centros obtenidos después de aplicar el algoritmo de K-medias (con K=10) a una distribución aleatoria de puntos en el intervalo [0,1] para dos inicializaciones diferentes
  - Se observa que los centros convergen a puntos diferentes del plano
  - Primera inicialización, J=1.5532
  - Segunda inicialización, J=1.6705
- Algunos autores han propuesto mejoras del algoritmo de K-medias, con el objetivo de obtener soluciones óptimas o globales que no dependan de la inicialización de los centros:
  - C. Chinrungrueng and C. H. Sequin, 1995
    Optimal Adaptove K-means Algorithm with Dynamic Adjustment of Learning Rate
  - S. Chen, 1995
    Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning

Aprendizaje. Método híbrido

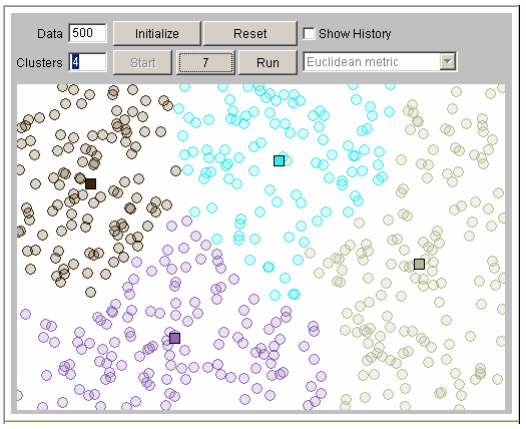
Algoritmo de K-medias. Ejemplo de inicializaciones diferentes



Aprendizaje. Método híbrido

Algoritmo de K-medias. Applet

- http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html



Aprendizaje. Método híbrido

Determinación de las Desviaciones

- Las amplitudes deben calcularse de manera que cada neurona oculta se active en una región del espacio de entrada y de manera que el solapamiento de las zonas de activación de una neurona a otra sea lo más ligera posible, para suavizar así la interpolación
- Una opción bastante efectiva es determinar la amplitud de la función de base radial como la media geométrica de la distancia del centro a sus dos vecinos más cercanos:

$$d_i = \sqrt{\|C_i - C_t\| \, \|C_i - C_s\|}$$

siendo C_t y C_s los dos centros más cercanos al centro C_i.

Aprendizaje. Método híbrido

Fase Supervisada

- En esta fase se calculan los pesos y umbrales de las neuronas de salida de la red
- El objetivo es minimizar las diferencias entre las salidas de la red y las salidas deseadas
- el proceso de aprendizaje está guiado por la minimización de una función error computada en la salida de la red:

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n) \qquad e(n) = \frac{1}{2}\sum_{k=1}^{r}(s_k(n) - y_k(n))^2$$

  - Como la salida de la red (y_k) depende linealmente de los pesos, puede utilizarse un método directo (Método de la seudoinversa), o bien el método de mínimos cuadrados

Aprendizaje. Método híbrido

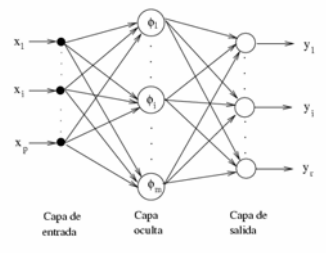
Fase Supervisada. Método de Mínimos cuadrados

- La salida de la red depende linealmente de los pesos y umbrales
- los pesos y umbrales de la red se determinan mediante un proceso iterativo gobernado por la siguiente ley:

$$w_{ik}(n) = w_{ik}(n-1) - \alpha_1 \frac{\partial e(n)}{\partial w_{ik}}$$

$$u_k(n) = u_k(n-1) - \alpha_1 \frac{\partial e(n)}{\partial u_k}$$

$$\text{para } k = 1, 2, ..., r \text{ y para } i = 1, ..., m$$



Aprendizaje. Método híbrido

Fase Supervisada. Método de Mínimos cuadrados

- Teniendo en cuenta la expresión del error y que el peso w_{ik} y el umbral u_k únicamente afecta a la neurona de salida k, se obtiene que:

$$\frac{\partial e(n)}{\partial w_{ik}} = -(s_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial w_{ik}}$$

$$\frac{\partial e(n)}{\partial u_k} = -(s_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial u_k}$$

$$y_k(n) = \sum_{i=1}^{m} w_{ik}\phi_i(n) + u_k \text{ para } k = 1, 2, ..., r \quad [1]$$

- Derivando la salida $y_k(n)$ de la red respecto a los pesos y umbrales, teniendo en cuenta la expresión [1], se obtiene que:

$$\frac{\partial y_k(n)}{\partial w_{ik}} = \phi_i(n) \qquad \frac{\partial y_k(n)}{\partial u_k} = 1$$

Aprendizaje. Método híbrido

Fase Supervisada. Método de Mínimos cuadrados

– Por tanto las leyes de modificación de los pesos y umbrales quedan de la siguiente forma:

$$w_{ik}(n) = w_{ik}(n-1) + \alpha_1(s_k(n) - y_k(n))\phi_i(n)$$
$$u_k(n) = u_k(n-1) + \alpha_1(s_k(n) - y_k(n))$$
$$\text{para } k = 1, 2, ..., r \text{ y para } i = 1, ..., m$$

[2]

Aprendizaje. Método híbrido

Fase Supervisada. Método de la seudoinversa

– Método que proporciona una solución directa al problema de optimización
– Dicha solución viene dada por la siguiente expresión matricial:

$$W = G^+ \cdot S = (G^t \cdot G)^{-1} \cdot G^t \cdot S$$

donde W es la matriz de pesos y umbrales de la red de base radial, de orden $(m+1) \times r$:

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1r} \\ w_{21} & w_{22} & \dots & w_{2r} \\ \dots & \dots & \dots & \dots \\ w_{m1} & w_{m2} & \dots & w_{mr} \\ u_1 & u_2 & \dots & u_r \end{pmatrix}$$

$G^+ = (G^t \cdot G)^{-1} \cdot G^t$ es la matriz seudo-inversa de G, siendo G^t la matriz traspuesta de G. G es una matriz de orden $N \times (m+1)$ que contiene las activaciones de las neuronas ocultas de la red para los patrones de entrada:

Aprendizaje. Método híbrido

Fase Supervisada. Método de la seudoinversa

$$\mathbf{G} = \begin{pmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_m(1) & 1 \\ \phi_1(2) & \phi_2(2) & \dots & \phi_m(2) & 1 \\ \dots & \dots & \dots & \dots & \\ \phi_1(N) & \phi_2(N) & \dots & \phi_m(N) & 1 \end{pmatrix}$$

donde $\phi_i(n)$ es la activación de la neurona oculta i para el patrón de entrada $X(n)$.

Y S es la matriz de salidas deseadas para la red, de orden $N \times r$:

$$\mathbf{S} = \begin{pmatrix} s_1(1) & s_2(1) & \dots & s_r(1) \\ s_1(2) & s_2(2) & \dots & s_r(2) \\ \dots & \dots & \dots & \dots \\ s_1(N) & s_2(N) & \dots & s_r(N) \end{pmatrix}$$

donde $s_k(n)$ es la coordenada k de la salida deseada para el patrón $X(n)$.

Aprendizaje. Método híbrido

- **Resumen del método híbrido**

  Dado el conjunto de patrones de entrada y sus salidas deseadas

  - Se aplica el algoritmo K-medias sobre el conjunto de patrones de entrada, para calcular los centros de las funciones de base radial, siendo K el número de neuronas ocultas.
  - Se calculan las amplitudes o desviaciones de las funciones de base radial.
  - Se determinan los pesos y umbrales de la capa de salida siguiendo el siguiente proceso iterativo:
    1. Se inicializan aleatoriamente los pesos y umbrales
    2. Se toma un patrón X(n) y se calcula la salida de la red Y(n)
    3. Se evalúa el error e(n) cometido por la red para dicho patrón
    4. Se modifican los pesos y umbrales utilizando las leyes de aprendizaje dadas por las ecuaciones [2]
    5. Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento
    6. Se repiten los pasos 2,3,4, y 5 hasta conseguir la convergencia, es decir hasta que la suma de los errores para todos los patrones se estabilice.

Aprendizaje. Método totalmente supervisado

- Todos los parámetros de la RNBR –centros, desviaciones, pesos y umbrales- se determinan de forma supervisada con el objetivo de minimizar el error cuadrático medio.
- El método totalmente supervisado no conserva, en principio, las propiedades o características locales de las RNBR
  - El proceso no se guía para que las amplitudes sean tales que el solapamiento de las gaussianas sea lo más suave posible, sino para minimizar el error cuadrático. Por tanto, pueden perderse las características locales
- Las salidas de la red dependen linealmente de los pesos, pero no de los centros y desviaciones
- Se aplicará el método de descenso del gradiente

Aprendizaje. Método totalmente supervisado

Los pesos, umbrales, centros y amplitudes se modifican de acuerdo con las siguientes leyes:

Pesos: $$\omega_{ik}(n) = \omega_{ik}(n-1) - \alpha_1 \frac{\partial e(n)}{\partial \omega_{ik}}$$

Umbrales: $$u_k(n) = u_k(n-1) - \alpha_1 \frac{\partial e(n)}{\partial u_k}$$

Centros: $$c_{ij}(n) = c_{ij}(n-1) - \alpha_2 \frac{\partial e(n)}{\partial c_{ij}}$$

Desviaciones: $$d_i(n) = d_i(n-1) - \alpha_3 \frac{\partial e(n)}{\partial d_i}$$

La aplicación del método de descenso del gradiente implica el cálculo de las derivadas del error con respecto a cada uno de los parámetros.

Aprendizaje. Método totalmente supervisado

Pesos y umbrales

Las derivadas se han calculado en el entrenamiento híbrido, obteniéndose las siguientes leyes de aprendizaje:

$$\omega_{ik}(n) = \omega_{ik}(n-1) + \alpha_1 (s_k(n) - y_k(n)) \phi_i(n)$$

$$u_k(n) = u_k(n-1) + \alpha_1 (s_k(n) - y_k(n))$$

Para $k=1,2,\ldots, r$ y para $i=1,2, \ldots, m$

Aprendizaje. Método totalmente supervisado

Centros

La expresión final es la siguiente:

$$c_{ij}(n) = c_{ij}(n-1) + \alpha_2 \left(\sum_{k=1}^{r} (s_k(n) - y_k(n)) \omega_{ik}\right) \phi_i(n) \frac{(x_j - c_{ij})}{d_i^2}$$

Para j$=1,2,\ldots, p$ y para $i=1,2, \ldots, m$

Aprendizaje. Método totalmente supervisado

Amplitudes o desviaciones

La expresión final es la siguiente:

$$d_i(n) = d_i(n-1) + \alpha_3 \left(\sum_{k=1}^{r} (s_k(n) - y_k(n))\omega_{ik}\right)\phi_i(n)\frac{\|X(n) - C_i\|^2}{d_i^3}$$

Para $i=1,2, \ldots, m$

Aprendizaje. Método totalmente supervisado

- Al ser un proceso iterativo se deben inicializar todos los parámetros
- Podría hacerse una inicialización aleatoria con valores próximos a 0
- Es más aconsejable inicializar los centros de manera que representen zonas del espacio de entrada, limitando así la búsqueda a ciertas regiones del espacio.
- Podrían combinarse el método híbrido y el totalmente supervisado:
  - Se fijan los parámetros con el método híbrido
  - Se adaptan posteriormente de forma supervisada

- Regiones de Voronoi
  http://www.pi6.fernuni-hagen.de/GeomLab/VoroGlide/index.html.en