

## **FUNCION DE BASE RADIAL**

### **Radial Basis Function (RBF)**

Características generales y diferencias resaltantes con los modelos neuronales multicapas (Backpropagation):

Broomhead y Lowe, 1988, introducen un método alternativo al perceptrón multicapa (MLP) (ej.: backpropagation) para hacer ajuste a funciones no lineales. Esto es clasificación no lineal: las redes RBF.

A diferencia de la disposición que se tiene en las funciones de activación que permite construir modelos de entrenamiento mediante backpropagation, estas nuevas redes basadas en RBF construyen sus modelos con funciones de activación que son diferentes tanto en la capa oculta como la de salida. Esto es, una red RBF está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros gravitacionales propios y en la capa de salida mediante funciones lineales.

A diferencia de las MLP, el modelo clásico de las redes RBF está construido con una arquitectura rígida de tres capas: la de entrada, la oculta y la de salida. (Broomhead y Lowe, 1988)

En general, una red RBF tiene un mejor desempeño con un mayor volumen de datos de entrenamiento.

La construcción de una red RBF requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes que usan backpropagation.

Aunque las redes RBF no son comúnmente utilizadas en aplicaciones que impliquen un alto volumen de patrones de entrenamiento, se le reconoce como una red con una alta eficiencia en la fase de entrenamiento.

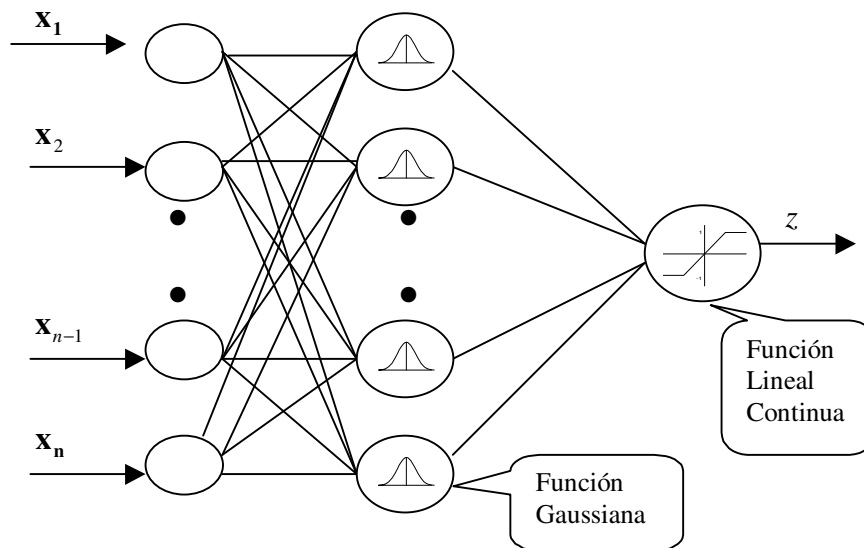
Método alternativo para aproximar funciones y clasificar patrones.

## COMO FUNCIONA UNA RBF

Tal como ya se dijo anteriormente, una red RBF, a diferencia de una MLP, está conformada de tres capas.

1. La capa de entrada que sirve para los ejemplos o patrones de entrenamiento y prueba,
2. la capa oculta completamente interconectada entre todos sus nodos con la capa de entrada y activada a través de la función radial (gaussiana) y,
3. la capa de salida, también completamente interconectada a la capa oculta y activada a través de una función lineal continua.

El entrenamiento, a diferencia de la red usando backpropagation, es solamente hacia delante. De este modo, la salida  $z$  de una red RBF, en general, está influenciada por una transformación no lineal originada en la capa oculta a través de la función radial y una lineal en la capa de salida a través de la función lineal continua.



Topología particular de la RBF:

- Los nodos ocultos contienen una función base radial, la cual tiene como parámetros a **centro** y **ancho**.

- Existe un **centro** para cada función radial involucrada en la capa oculta. Regularmente, definen un vector de la misma dimensión del vector de entrada y hay normalmente un centro diferente por cada nodo de la capa oculta.
- Por otro lado, el **ancho** es el término empleado para identificar a la amplitud de la campana de gauss originada por la función radial. Es decir, la desviación estándar de la función radial. Algunos autores (Lowe, 1989) consideran a este ancho como un valor constante para cada una de las funciones radiales consideradas en la capa oculta y de este modo, así contribuiría a simplificar los pasos de construcción del modelo de entrenamiento de la red.

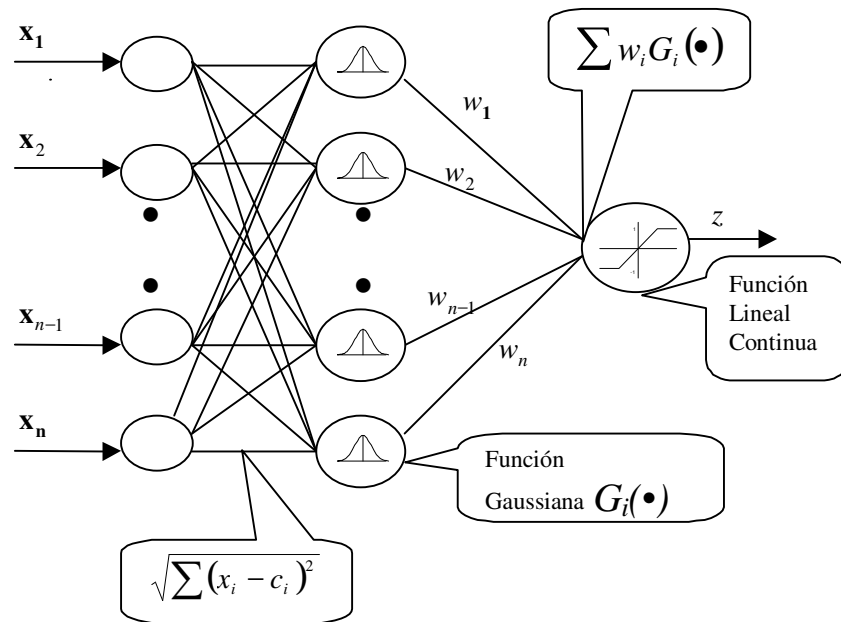
El primer cálculo efectuado en la capa oculta es hallar en un nodo de la capa oculta la distancia radial (distancia euclidiana)  $d$  entre el vector de entrada  $\mathbf{x}$ , con  $n$  observaciones, a ese nodo en particular y el centro de gravedad  $\mathbf{c}$  de ese mismo nodo. Es decir:

$$d = \|\mathbf{x} - \mathbf{c}\| = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \dots + (x_n - c_n)^2}$$

Este valor  $d$  es un componente de la entrada para activar la función radial  $G(\bullet)$ . Este valor establece la principal diferencia con las redes MLP, entre ellas la backpropagation, quienes incluyen el producto interno en sus capas ocultas de las entradas por sus respectivos pesos.

En cuanto a la función radial  $G(\bullet)$ , siendo una de las más comunes  $\exp(-r^2)$ , siendo  $r$  el contenido evaluado en cada nodo de la capa oculta. En este caso particular, el contenido evaluado en cada nodo es la distancia euclidiana  $d$ . De ahí la expresión, entonces sería  $\exp(-d^2)$ .

Una de las derivaciones del modelo RBF es emplear el **ancho** (desviación estándar) para activar la función  $G(\bullet)$ . En este caso se estaría trabajando con algo como  $\exp(d^2/a)$ , donde  $a$  es el ancho para ese nodo oculto.



Entre la capa oculta y la capa de salida se derivan un conjunto de pesos  $w$  que se verían afectados de acuerdo al algoritmo de aprendizaje. En este caso particular, sería la combinación lineal entre los pesos y la resultante de cada función radial para determinar la salida  $z$ .

Tal como hemos visto con anterioridad, sería,

$$z = \sum w_i G(\bullet),$$

donde  $G(\bullet)$  es la salida de la capa oculta y se corresponde con la función radial aplicada a la distancia euclidiana en cada una de las unidades ocultas.

Del resultado de este tipo de entrenamiento podemos observar que:

1. Los valores de entrada se recomiendan que previamente sean de algún modo transformados a una escala.
2. En la capa oculta, en la medida que los valores de entrada se parezcan más a un centro su distancia tenderá a **cero** y de este modo la función gaussiana se dispararía a las vecindades de **uno**. Por otro lado, en la medida que los valores de entrada no se

parezcan a su centro la distancia será mayor y la función radial parecería tender a **cero**. Este proceso es una clasificación no lineal de las entradas.

3. En la capa de salida del modelo RBF, los valores obtenidos en las salidas de la capa oculta serían transformados por la función lineal que permite aproximar los valores  $z$  a los valores deseados, mediante la combinación lineal que se sucede en esta capa entre sus pesos y el resultado de aplicar la función radial. Es decir,  $z = \sum w_i G(\bullet)$ .
4. El tiempo de entrenamiento es substancialmente inferior al requerido por otros algoritmos. Es una pasada hacia adelante en la mayoría de los casos. La diferencia la establece si se incorpora en la salida del modelo de entrenamiento, una supervisión a través del control del error que se produce entre los valores calculados y los observados, conduciendo a una retropropagación del error.
5. Alrededor del algoritmo clásico iniciado por Broomhead y Lowe, se han implementado algoritmos que contribuyan a la mejor selección de los centros y anchos de las funciones radiales.
6. Nuevos cambios incorporados a las funciones de activación de salida originan nuevos modelos de entrenamiento. Tal es el caso de las redes neuronales GRNN (Generalized Regression Neural Network) y PNN (Probabilistic Neural Network), PCANN (Principal Component Analysis Neural Network). Para una mayor referencia acerca de estas redes les sugiero revisar los textos electrónicos de Matlab (<http://www.mathworks.com>) y Statistica (<http://www.statsoft.com>).

## ENTRENAMIENTO DE UNA RBF

Diferente a las redes supervisadas vistas anteriormente, en este caso, suponiendo un hiperplano definiendo un espacio  $N$ -dimensional, lo que pretende una red RBF es ejecutar una **correspondencia no lineal** entre los patrones de entrenamiento que definen el espacio de entrada al espacio oculto definido por la capa oculta y una **correspondencia lineal** desde este espacio al espacio de salida. Es decir definir a la salida una superficie que describa las entradas.

En vista de que esta superficie es desconocida, se acude un proceso de entrenamiento usando ejemplos representativos tanto para la entrada como para la salida.

De acuerdo a ello, han surgido variantes como producto fundamentalmente de las siguientes desventajas:

- de no conocer los centros (a veces el ancho) para cada función radial,
- de situaciones de singularidad presentes en la implementación del algoritmo con problema de dimensionalidad,
- de un gran volumen de entradas haciendo inmanejable la aplicación del algoritmo. Se presentan problemas de regularización (Simon Haykin, 1995)

De acuerdo a Broomhead y Lowe el proceso de aprendizaje de la red RBF puede ser visto en dos fases:

**Fase de Entrenamiento:** constituye la optimización de un procedimiento de ajuste de una superficie que se define como producto de los ejemplos de entrada-salida presentados a la red.

**Fase de Generalización:** una interpolación entre los datos o interpolación realizada a lo largo de la superficie generada por un procedimiento de ajuste de la aproximación óptima de la superficie real.

De este modo en el sentido estricto de interpolación, podemos decir que existe una función que satisfaga la condición de interpolación  $F(x_i)=d_i$ , donde  $x_i$  son los puntos que

describen la superficie de un espacio  $N$  dimensional y  $d_i$  representa su respuesta. Tal como lo describe esta función, la interpolación estricta se refiere a que la función está restringida a pasar por todos los puntos. Es decir, es la aproximación óptima de la superficie real.

### Clásico.

Bajo esta premisa, tenemos que la función que puede describir dicha interpolación, de acuerdo a Powel sigue la siguiente forma.

$$F(\mathbf{x}) = \sum_1^N w_i G(\|\mathbf{x} - \mathbf{x}_i\|)$$

donde la función  $F(\mathbf{x})$  está involucrada con la función lineal  $G(\bullet)$  y la combinación lineal con los pesos.

En forma matricial, sería

$$\mathbf{G}\mathbf{w} = \mathbf{z}$$

Cada elemento  $g_{j,i} = g(\|\mathbf{x}_j - \mathbf{x}_i\|)$ ,  $j, i = 1 \dots N$

$$\mathbf{z} = [z_1, z_2, z_3, \dots, z_N]^T$$

$$\mathbf{w} = [w_1, w_2, w_3, \dots, w_N]^T$$

Provistos que todas las observaciones son distintas, entonces  $G$  se podría decir que es positiva definida y por lo tanto los pesos podrían ser calculados mediante la inversa de  $\mathbf{G}$ . Es decir

$$\mathbf{w} = \mathbf{G}^{-1}\mathbf{z}$$

Sin embargo se puede correr el riesgo de que la inversa de la matriz de interpolación  $G$  está próxima a ser singular. En este caso se procedería mediante la teoría de la regularización para perturbar la matriz mediante  $\mathbf{G} = \mathbf{G} + \lambda \mathbf{I}$ . (Simon Haykin, pp. 245)

De esta manera sería un aprendizaje directo, provocando cambio a los pesos que están ubicados entre la capa oculta y la capa de salida.

### EJEMPLO DE UNA RBF

Veamos el ejemplo que está ilustrado en la página 260 del texto de Simon Haykin.

El se refiere al problema XOR, que de acuerdo a lo que podemos recordar, no pudo ser clasificado por una TLU.

Supongamos que tenemos los siguientes patrones de entrenamiento

$x_1$	$x_2$	$z$
1	1	1
0	1	0
0	0	1
1	0	0

La función gaussiana es  $G(\|\mathbf{x} - \mathbf{c}_i\|) = \exp(-\|\mathbf{x} - \mathbf{c}_i\|^2)$ ,  $i=1,2$ .

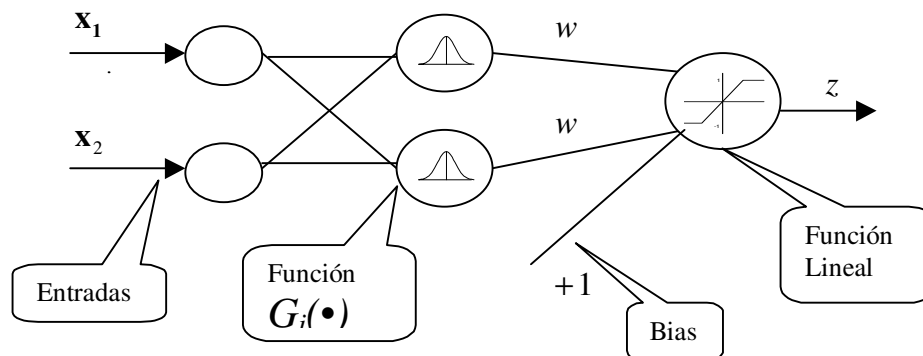
Deben existir dos centros y ellos ya son conocidos. Vienen dados por:  $\mathbf{c}_1 = [1 \ 1]^T$ ,  $\mathbf{c}_2 = [0 \ 0]^T$ .

Por las características del problema, se asume lo siguiente:

Los pesos son compartidos por la simetría del problema.

La capa de salida incluye un bias.

De este modo la arquitectura de la red RBF es:





La relación entrada-salida está expresada por:

$$F(\mathbf{x}) = \sum_1^2 w_i G(\|\mathbf{x} - \mathbf{t}_i\|) + b$$

donde  $b$  es el bias y  $F(\mathbf{x})$  es  $\mathbf{z}$ .

Definamos el problema en forma matricial como  $\mathbf{z}=\mathbf{G}\mathbf{w}$ .

Resolviendo mediante **Matlab** tenemos:

```
%  
% Resolviendo el problema XOR usando la expresión  $\mathbf{z}=\mathbf{G}\mathbf{w}$ .  
%  
x=[1 1; 0 1; 0 0; 1 0]  
x =  
    1    1  
    0    1  
    0    0  
    1    0  
  
z=[1; 0; 1; 0]  
z =  
    1  
    0  
    1  
    0  
  
%  
% los centros están dados por  
%  
c1=[1 1];  
c2=[0 0];  
%  
% La función gaussiana es  $\exp(-\text{norm}(\mathbf{x}-\mathbf{c})^2)$   
%  
% Redefinamos los centros en una matriz  
%
```

```
c=[c1;c2]
```

```
c =
```

```
    1    1
    0    0
```

```
% De este modo el cálculo de las funciones radiales y las
% distancias la vamos a realizar por un procedimiento continuo
% con dos lazos donde todos los patrones de entrada van a ser
% revisados.
```

```
%
```

```
for j=1:4
```

```
for i=1:2
```

```
    G(j,i) = exp(-norm(x(j,:)-c(i,:))'^2);
```

```
end
```

```
end
```

```
G
```

```
G =
```

```
    1.0000    0.1353
    0.3679    0.3679
    0.1353    1.0000
    0.3679    0.3679
```

```
% La matriz ampliada incluyendo el bias es
```

```
G=[G ones(4,1)]
```

```
G =
```

```
    1.0000    0.1353    1.0000
    0.3679    0.3679    1.0000
    0.1353    1.0000    1.0000
    0.3679    0.3679    1.0000
```

```
% Ahora debemos resolver el cálculo de los pesos entre la capa
% oculta y la capa de salida mediante la expresión  $Gw=z$ .
```

```
% Desde donde w es  $w=inv(G)z$ 
```

```
% Aquí podremos tener ciertos problemas con la singularidad de la matriz G.
```

```
% Para cuidarnos, usaremos la seudoinversa de la siguiente
```

```
% manera  $w=inv(G'G)G'z$ . Donde G' es la transpuesta de G.
```

```
%
w=inv(G'*G)*G'*z
w =
    2.5027
    2.5027
   -1.8413

% Procedamos a realizar una prueba para conocer el nivel de error
% con respecto a los valores deseados.
Z_estimado=G*w;
%
% Luego los valores de la salida calculados son:
%
Z_estimado

Z_estimado =
    1.0000
    0.0000
    1.0000
    0.0000
%
% Los valores deseados de la salida son:
%
z

z =
    1
    0
    1
    0

% los resultados son exactamente iguales.
% Por tanto la red entrenada para el XOR para los centros dados
% incluiría los siguientes pesos
w
w =
    2.5027
    2.5027
   -1.8413
```

## **GENERALIZACION DE UNA RBF**

Luego de entrenada una red RBF se mide su capacidad de generalizar antes nuevos ejemplos de entrada. Este proceso se le conoce como Interpolación.