

# Redes Neuronales de Base Radial aplicadas a la mejora de la calidad

Recepción: Abril de 2008 / Aceptación: Junio de 2008

<sup>(1)</sup>Juan Cevallos Ampuero

## ABSTRACT

Esta investigación ha permitido construir una Red Neuronal Artificial RNA con Función de Base Radial, y que utiliza la distancia de Mahalanobis RND, para la mejora de la calidad de diseño de procesos, obteniendo mejores resultados que los obtenidos con los análisis estadísticos tradicionales para los diseños experimentales y las RNA ya existentes, para los casos que se trabaje con varias variables dependientes e independientes y en los que sus relaciones no sean lineales. Asimismo, al RND permite obtener parámetros de entrada para lograr un nivel de calidad deseado; para ello se aplica una metodología que usa las RNA Inversa y Directa a la vez.

**Palabras clave:** Redes neuronales de base radial, Funciones de base radial, Redes neuronales de diseño exacto, Perceptrón multicapa con aprendizaje backpropagation.

## NEURAL NETWORKS WITH RADIAL BASIS APPLIED TO THE IMPROVE OF QUALITY

## RESUMEN

This research has led to construct an artificial neural network ARN with Radial Basis Function, and using Mahalanobis distance RND, for improving the quality of process design, which have performed better than those obtained with the for traditional statistical analysis of design of experiments and other RNA that already exist, for cases that are working with several independent and dependent variables in which its relations are not linear. It also allows with the RND obtain input parameters to achieve a desired level of quality, for it applies a methodology that uses RNA Reverse and Direct at once.

**Keyword:** Neural networks with radial basis, Radial basis functions, Neural networks of exact design, Multilayer perceptron with backpropagation learning.

## INTRODUCCIÓN

Las herramientas que mayoritariamente se utilizan para la mejora de la calidad mediante el diseño de procesos y productos solo permiten trabajar con relaciones lineales y consideran pocas variables de entrada y salida; las redes neuronales artificiales son una alternativa para poder superar esta limitación. El análisis estadístico tradicional para el diseño de experimentos, los Métodos Taguchi, entre otros, no permiten hacer este tipo de cálculos. Los modelos de redes neuronales más utilizados son: perceptrón, adaline, perceptrón multicapa, redes de base radial, redes recurrentes (Red de Hopfield), redes de aprendizaje supervisado (Mapas autoorganizados de Kohonen), y de resonancia adaptativa ART, redes de series temporales, redes de control de procesos dinámicos, redes de clasificación (redes de cuantización vectorial)<sup>6</sup>. Con base al marco teórico de las RNA más utilizadas se aprecian que algunas podrían servir de base para desarrollar una nueva RNA a ser utilizada en temas relacionados con la mejora de la calidad. Las Redes Neuronales Artificiales (RNA) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. En la Figura 1, se aprecia el funcionamiento de una neurona, así se tiene un conjunto de entradas  $X_1, X_2, \dots, X_3$ ; que pueden ser definidas por un vector  $\bar{X}$ . Cada entrada se multiplica por un peso asociado  $W_1, W_2, \dots, W_n$  para calcular la respectiva sumatoria  $\Sigma$  que es procesada por la función de activación respectiva y se obtienen las salidas correspondientes.

El conjunto de pesos se representa por un vector  $\bar{W}$ . La sumatoria que corresponde al cuerpo de la neurona suma todas las entradas ponderadas por sus respectivos pesos, produciendo una salida E:

$$E = x_1W_1 + x_2W_2 + \dots + x_nW_n$$

Lo cual se puede definir en forma vectorial como sigue:  $E = \bar{X}^T \bar{W}$

La señal E es procesada por la función de activación, que produce la salida de la neurona S. Hay diversas funciones de activación, como por ejemplo:

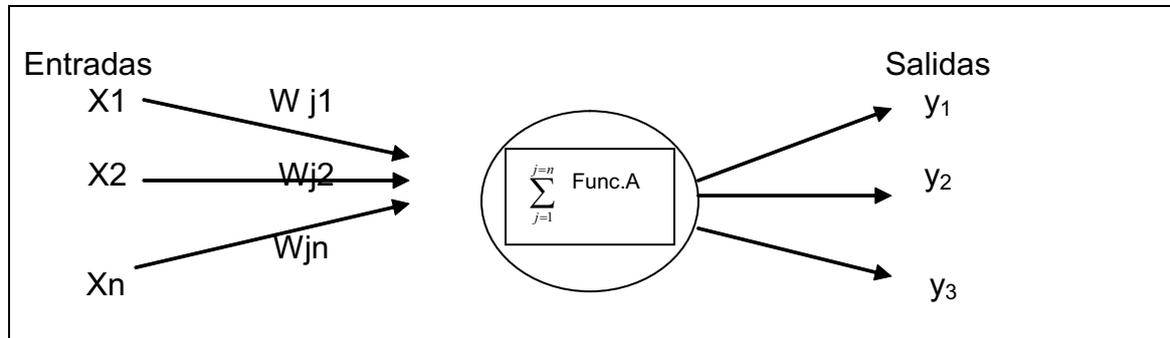
Lineal:  $S = K E$  con K constante.

Umbral:  $S = 1$  si  $E > \theta$ ,  $S = 0$  si  $E < \theta$ ; siendo  $\theta$  el umbral constante.  
 Cualquier función:  $S = F(I)$ ; siendo F una función cualquiera.

1 Doctor en Ingeniería. Profesor en la Facultad de Ingeniería Industrial. Departamento Académico de Producción y Gestión Industrial de la UNMSM. E mail. jceval@terra.com.pe/ jcevallosa@unmsm.edu.pe  
 2 ISASI, Pedro y GALVAN, Inés. (2004). Redes de Neuronas Artificiales. Editorial Pearson- Prentice Hall. España. p. 45.

Combinando varias neuronas de este tipo, se pueden construir diversos tipos de redes que calculen cualquier función lógica finita. La forma cómo se organizan las neuronas que se combinan se conoce como su arquitectura.

**Figura N.º 1. Esquema de una Neurona Artificial (McCulloch-Pitts)**

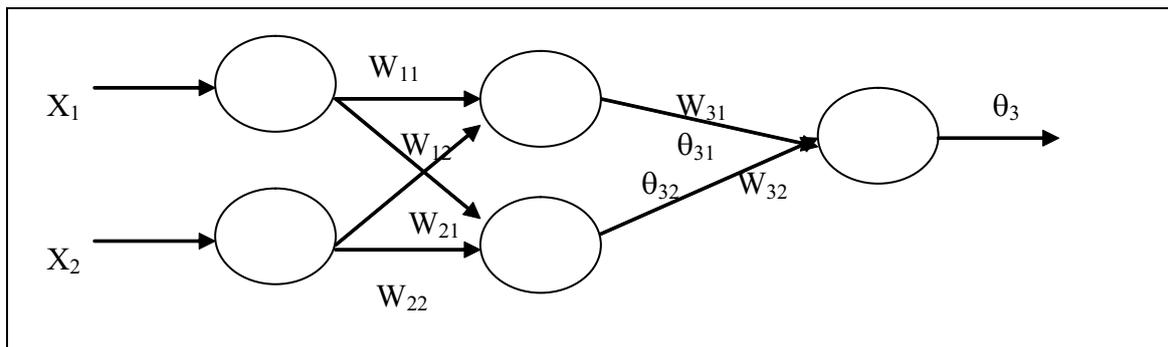


FUENTE: Elaboración propia.

Se considera que las Arquitecturas de las neuronas tienen capas, y en cada capa puede haber una o varias neuronas. Así se considera que existe una capa de entrada, una de salida y entre ambas pueden presentarse capas ocultas o intermedias.<sup>4,5,11</sup>

En la Figura N.º 2, se muestra un ejemplo de arquitectura típica de red de neuronas artificiales:

**Figura N.º 2. Red Neuronal Artificial con una capa oculta**



FUENTE: Elaboración propia.

## REDES NEURONALES DE BASE RADIAL

### Arquitectura

Se encontró que las redes de neuronas de base radial son las que mejor se adecuan a nuestro objetivo, debido a que son redes multicapa con conexiones hacia delante, que se caracterizan porque están formadas por una única capa oculta y cada neurona de esta capa posee un carácter local, en el sentido de que cada neurona oculta de la red se activa en una región diferente del espacio de patrones de entrada. Este carácter local viene dado por el uso de las llamadas funciones de base radial,

generalmente la función gaussiana, como funciones de activación. Las neuronas de la capa de salida de las redes de base radial simplemente realizan una combinación lineal de las activaciones de las neuronas ocultas. La capa de entrada la componen un conjunto de neuronas que reciben las señales del exterior, transmitiéndolas a la siguiente capa sin realizar ningún procesado sobre dichas señales. Las neuronas de la capa oculta reciben las señales de la capa de entrada y realizan una transformación local y no lineal sobre dichas señales. Activación de las neuronas de la red de base radial. Si la red tiene "p" neuronas en la capa de entrada, "m" neuronas en la

4 HAGAN M., DEMUTH H., BEALE, M. (1996). Neural Network Design. Editorial Thomson. EEUU. p. 2.1.

5 HAYKIN, S. (2005). Neural Networks. Editorial Macmillan College Publishing Company. EEUU. P.278

11 WASSERMAN, P. (1993). "Advanced Methods in Neural Computing". Ed. Van Nostrand Reinhold. EEUU. P.147

capa oculta y "r" neuronas en la capa de salida, las activaciones de las neuronas de salida para el patrón de entrada "n":

$$X(n) = (x_1(n), x_2(n), \dots, x_p(n),)$$

son denotadas como  $y_k(n)$ , vienen dadas por la siguiente ecuación:

$$y_k(n) = \sum w_{ik} \theta_i(n) \pm u_k ; \text{para } k = 1, 2, \dots, r$$

donde:  $w_{ik}$  es el peso de la conexión de la neurona oculta  $i$  a la neurona de salida  $k$ ;  $u_k$  es el umbral de la neurona de salida  $k$ ; y  $\theta_i(n)$  son las activaciones de las neuronas ocultas para el patrón de entrada  $X(n)$ .

Las funciones de base radial  $\theta_i$  determinan las activaciones de las neuronas de las neuronas ocultas de la red en función de un vector de entrada a la red  $X(n)$  y vienen dadas por expresiones que dependen de los centros de la función de base radial, la desviación o amplitud de la función de base radial y la distancia del vector de entrada  $X(n)$  al centro  $C_i$ .

Las entradas  $x_1, x_2, \dots, x_m$  conforman un vector de entrada  $x$ , y son aplicadas a todas las neuronas en una capa oculta. Según la topología de la red que se muestra en la siguiente figura:  
Cada neurona de la capa oculta calcula la siguiente

función exponencial (radial):

$$h_i = \text{expo}[-D_i^2 / \sigma^2]$$

Donde:

$X$  = un vector de entrada

$w_{1,i}$  y  $w_{2,j}$  = vector peso de la neurona  $i$  de la capa oculta y  $j$  de la capa de salida

$b_{1,i}$  y  $b_{2,j}$  = umbrales de de la neurona  $i$  de la capa oculta y  $j$  de la capa de salida

$u_i$  = centros de las neuronas ocultas

$$D_i^2 = (x - u_i)^T (x - u_i) = \text{Distancia Euclídea}$$

$x$  y  $u$  = vectores columna

$T$  = indica la transpuesta del vector

Los pesos de cada neurona de capa oculta son asignados desde los valores de un vector de entrenamiento de entrada.

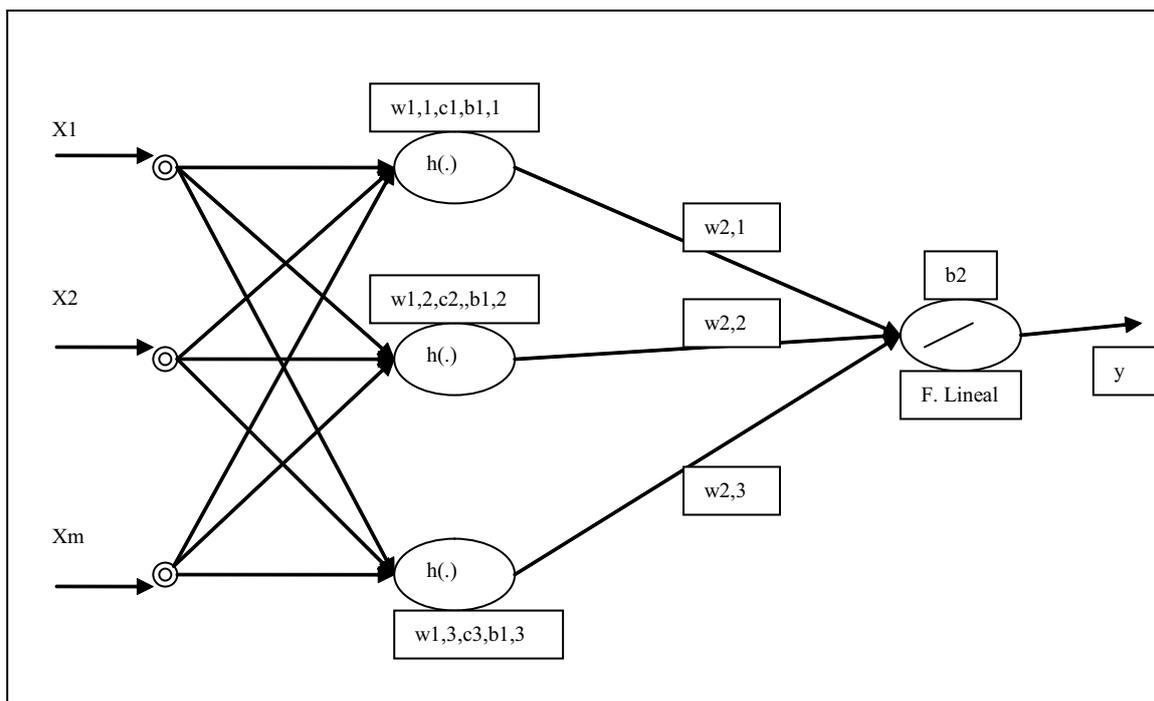
La neurona de salida produce la suma de pesos lineal de estos:

$$y = \sum h_i w_i + b_{ij}$$

donde  $w_{i,j}$  = peso en la capa de salida.

Las Redes de Base Radial son aproximadores universales de carácter local.

Figura N.º 3. Red Neuronal de Base Radial. Arquitectura



FUENTE: Elaboración propia

## APRENDIZAJE DE LAS REDES NEURONALES DE BASE RADIAL

Se pueden dar dos casos: Híbrido y totalmente supervisado.

### Aprendizaje Híbrido

En el caso híbrido: se tiene la primera fase no supervisada y la segunda supervisada.

#### Fase no supervisada

Los centros de las funciones de base radial se determinan mediante un algoritmo de clasificación no supervisado que permita dividir el espacio de patrones de entrada en clases. El número de clases es el número de neuronas ocultas en la red de base radial. Se puede utilizar el algoritmo de K medias, o cualquier otro, e incluso los mapas autoorganizados de Kohonen. Las amplitudes o desviaciones se calculan de manera que cada neurona oculta se active de una región del espacio de entrada y 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 ligero posible, para supervisar así la interpolación. Entre otras, se pueden usar las heurísticas de los vecinos más cercanos, la media uniforme de las distancias euclídeas del centro "Ci" a los centros más cercanos, o la media geométrica de las distancias centro Ci a los p centros más cercanos o incluso valores determinados que permitan un valor de salida predeterminado luego se aplica la función de base radial.

#### Fase supervisada

Se busca minimizar el error entre el valor de salida de la red y el de la salida deseada que corresponde a su respectivo par "p" de entrada. Para ello se puede seguir el método de mínimos cuadrados o el de la pseudoinversa.

#### Mínimos Cuadrados

Las ecuaciones que se obtienen para ser aplicadas son:

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

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

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

#### Pseudoinversa

Se aplica la ecuación:

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

Donde W es la matriz de pesos y umbrales de la red;  $G^+$  es la matriz pseudoinversa de G, G es la matriz

que contiene los valores de las funciones de base radial (salida de la capa oculta) y S la matriz que tiene todas las salidas deseadas.

### Aprendizaje totalmente supervisado

Dado  $\{(X(n), S(n))\}_{n=1..N}$  el conjunto de patrones de entrada y sus salidas deseadas, el método de aprendizaje totalmente supervisado para las redes de neuronas de base radial se resume en los siguientes pasos:

- 1) Se inicializan todos los parámetros de la red. En el caso de las amplitudes, pesos y umbrales, esta inicialización suele hacerse de manera aleatoria con valores cercanos a cero. Para los centros es, generalmente, preferible inicializarlos aleatoriamente a patrones de entrada o a la salida de un algoritmo de clasificación aplicado en el espacio de entrada.
- 2) Se toma un patrón del conjunto de patrones disponibles  $(X(n), S(n))$  y se calcula la salida de la red,  $Y(n)$ , para el patrón de entrada  $X(n)$ .
- 3) Se evalúa el error  $e(n)$  cometido por la red para dicho patrón, con la ecuación:

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

donde  $Y(n) = (y_1(n), \dots, y_r(n))$  y  $S(n) = (s_1(n), \dots, s_r(n))$  los vectores de salida de la red y salida deseada para el patrón de entrada  $X(n)$ , respectivamente.

- 4) Se modifican los pesos, umbrales, centros y amplitudes de la red utilizando las ecuaciones:

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

umbrales:

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

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

centros:

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

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

amplitudes:

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

Para  $i=1,\dots,m$

- 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, según la ecuación:

$$E = 1/N \sum e(n)$$

en este momento se alcanza un mínimo de dicha función.

La ubicación de los centros del campo de recepción es un tema crítico y hay muchas alternativas para su determinación. Por ejemplo, un centro y las correspondientes neuronas de capa oculta pueden ser consideradas como cada vector input del conjunto de entrenamiento. Los vectores de entrenamiento tienden a poder agruparse, para no tener más neuronas de la capa oculta que las que son necesarias. Ya que de lo contrario, el resultado sería un tiempo de entrenamiento más largo y la operación más lenta durante la etapa de entrenamiento, por la mayor cantidad de cálculos. Aunque se pierda en exactitud.

La Distancia de Mahalanobis es una medida de distancia introducida por Mahalanobis en 1936. Su utilidad radica en que es una forma de determinar la similitud entre dos variables aleatorias multidimensionales. Se diferencia de la Distancia Euclídea en que tiene en cuenta la correlación entre las variables aleatorias. Formalmente, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad  $X$  y con  $Y$  matriz de covarianza, cuyo símbolo es  $\Sigma$ , se define como:

$$d_m(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}$$

Se puede construir una Red de Base Radial de Diseño Exacto, que produzca error cero sobre los vectores de entrenamiento. Es decir con datos de entrada en pares  $p$ ,  $t$  (input y output) y con una determinada amplitud se puede obtener una red con pesos y umbrales cuya salida sea exactamente  $t$  cuando la entrada es  $p$ . Para ello se tienen que crear tantas neuronas ocultas con función radial como vectores input se tiene en  $p$ . En este caso se establece como umbral de la capa oculta  $0.8326/\text{amplitud}$ , y los pesos de la capa oculta a la de salida y los umbrales de la capa de salida se determinan mediante la expresión:

$$[W \{2,1\} b \{2\}] * [A \{1\}; ones] = T$$

Teniendo en cuenta que podemos conocer  $A\{1\}$  y  $T$  se puede obtener  $W$  y  $b$  aplicando:

$$[W \ b] = T / [P; ones(1, Q)]$$

Donde  $Q$  es el número de vectores de entrada (que en este caso equivale al número de neuronas de la capa de entrada).

Del estudio de las redes se encontró el Perceptrón Multicapa con aprendizaje de backpropagation, y las Redes Neuronales de Base Radial las más adecuadas para desarrollar una red neuronal artificial específica para la mejora de la calidad de diseño de procesos.

## METODOLOGÍA

Se analizaron las RNA existentes para aplicarlas en la mejora de calidad de procesos y se desarrolló una RNA para aplicarla a la mejora de la calidad de procesos RND. Se evaluó la RND con relación a otras RNA y las técnicas estadísticas tradicionales de mejora de la calidad de procesos (análisis estadísticos) a fin de evaluar su desempeño a través del error de sus resultados.

Con base a siete casos de mejora de calidad se comprobó las ventajas de la RND.

Los casos fueron:

- 1) "Modified Desirability Functions for Multiple Response Optimization", Del castillo *et al.*<sup>3</sup>
- 2) "Design of Experiments. Improve Molded Part Quality", Vasconcellos *et al.*<sup>10</sup>
- 3) "Diseño y Análisis de Experimentos", Montgomery.<sup>7</sup>
- 4) "Understanding Industrial Designed Experiments", Schmidt y Launsby.<sup>9</sup>
- 5) "Taguchi Techniques for Quality Engineering", Ross.<sup>8</sup>
- 6) "How to Formulate the Ultimate Margarita: A Tutorial on Experiments with Mixtures", Bowles *et al.*<sup>2</sup>
- 7) "Improving Parafin Wax Yield Through Process Optimization Using Taguchi's Method of Experimentation", Annand.<sup>1</sup>

1 Anand, K. (1993). "Improving Parafin Wax Yield through process optimization using Taguchi's Method of Experimentation". Quality Engineering, 6(1):39-56.  
 2 Bowles, M. y Montgomery, D. (1997). "How to formulate the ultimate Margarita: A tutorial on experiments with mixtures". Quality Engineering 10(2):239-253.  
 3 Del Castillo, E., Montgomery, D. y Mccarville, D. (1996). "Modified Desirability Functions for Multiple Response Optimization". Journal of Quality Technology 28(3):337-345  
 7 Montgomery, Douglas. (2004). "Diseño y análisis de experimentos". Editorial Limusa Wiley. México. p. 260.  
 8 Ross, Philip. (1988). "Taguchi Techniques for Quality Engineering". Editorial McGrawHill. EEUU. p. 269.  
 9 Schmidt, S. y Launsby, R. (1994). "Understanding Industrial Designed Experiments". Editorial Air Academy Press. EEUU. pp. 8-23.  
 10 Vasconcellos, A., Sodre, S. Y Rekab, K. (2003). "Design of Experiments. Improve Part Quality". Quality Progress. 2003(7):72-76.

**RESULTADOS**  
**LA RED NEURONAL DESARROLLADA RND**

Se desarrolló una Red Neuronal Artificial con Función de Base Radial, que tiene las siguientes características:

- a) Arquitectura con tres capas: capa de entrada, capa oculta y capa de salida.
- b) La capa de entrada tiene neuronas que reciben las señales del exterior, que la transmiten a la siguiente capa: capa oculta, sin realizar ningún proceso.
- c) Las conexiones de la capa de entrada a la oculta no están asociadas a algún peso. Las conexiones de la capa oculta a la de salida si van asociadas a un peso.
- d) La capa oculta tiene igual número de neuronas que vectores de entrada y la capa de salida tiene igual número de vectores que salidas requeridas.
- e) La capa oculta tiene como función de transferencia una Función de Base Radial, la Función de Gauss, cuya distancia es calculada con la Distancia de Mahalanobis. La expresión matemática de la función de base radial modificada es:

$$\phi(r) = e^{-(x-c)^T K(x-c)}$$

donde K es:  
 $K = [E(x-m)(x-m)^T]^{-1}$   
 m= el vector medio calculado considerando todos los vectores de entrada.  
 T= símbolo de la transpuesta de la matriz.  
 X= los vectores de entrada  
 c= los centros de la función de base radial.  
 K=Distancia de Mahalanobis.  
 siendo K la inversa de la matriz de la covariancia de los vectores de entrada.

- f) En la capa de salida hay un aprendizaje supervisado para obtener los valores de salida de la red, buscando que el error de la salida de la red con relación al vector inicialmente dado como salida sea lo más pequeño. Sus ecuaciones son:

pesos:  

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

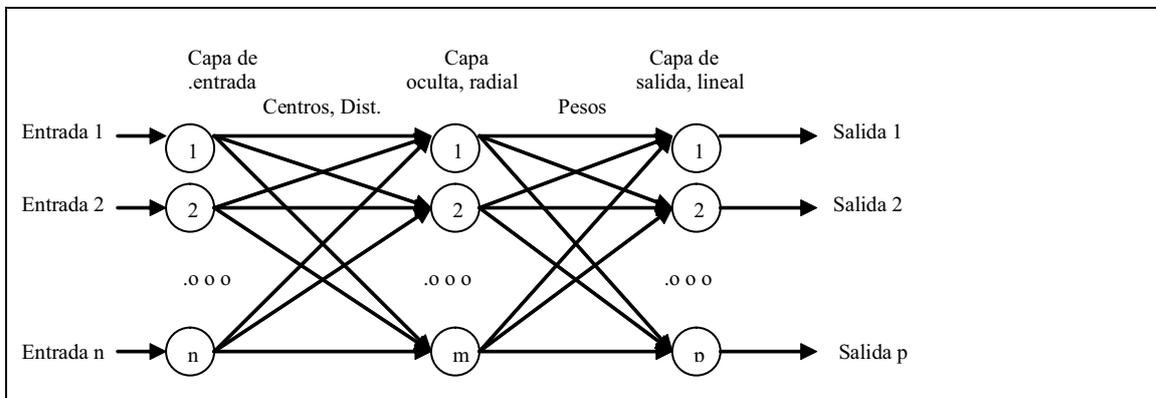
umbrales:  

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

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

siendo:  
 wik= los pesos de la capa oculta a la capa de salida.  
 uk= umbrales de la capa de salida.  
 sk(n)=salida deseada, de los datos  
 Yk(n)= salidas de la capa de salida.  
 $\alpha$  = constante de aprendizaje  
 $\Phi$ = la función de base radial, función de Gauss.  
 La activación de las neuronas de salida para el patrón de entrada viene dada por la siguiente expresión:  
 $y_k(n) = \sum w_{ik}A_i(n) + u_k$ ; para  $k = 1, 2, \dots, r$   
 siendo:  
 wik= los pesos de la capa oculta a la capa de salida.  
 $A_i(n)$  = es la salida de la capa oculta  
 uk= umbrales de la capa de salida.  
 yk(n)= salidas de la capa de salida.  
 Luego de obtener la salida de la capa oculta, se determinan los pesos de la capa oculta a la capa de salida y los umbrales de la capa de salida. Se determinan siguiendo el método de la matriz pseudoinversa.

**Figura 4. Modelo de la Red Neuronal Desarrollada**



FUENTE: Elaboración propia.

La ecuación que se aplica es:

$$Wb = A(1)+T;$$

donde:

A(1) es la salida de la capa oculta

A(1)+ es la pseudoinversa

T es la salida dada como dato inicial

En la Figura N.º 4 se aprecia el modelo de la red desarrollada:

### MODELO DE RED NEURONAL DESARROLLADA

Para efectuar los cálculos se elaboró el siguiente programa para MATLAB:

Se tienen las Fases de Entrenamiento y Simulación, que se detallan a continuación.

#### FASE I. Entrenamiento de la red

Datos iniciales: Matriz de Entrada: p, Matriz de Salida: t, Amplitud d (sc, en Matlab), Salidas: Peso de Entrada: w1, Umbral de entrada b1, Peso de Salida: w2, Umbral de salida: b2, Inversa de la matriz de covariancia: K.

Los pasos que se siguen son: 1) Ingresar datos, 2) Determinar dimensiones de P y T, 3) Determinar pesos w1, 4) Calcular umbrales B1, 5) Calcular las distancias Z, 6) Calcular las salidas de la capa oculta A, 7) Calcular la matriz pseudoinversa, 8) Calcular los pesos w2, 9) Calcular los umbrales B2, 10) Mostrar resultados.

Código de Matlab:

```
function [w1,b1,w2,b2,K] = newrbfcv15r(p,t,sc)
[r,q] = size(p);
[s2,q] = size(t);
w1 = p';
b1 = ones(q,1)*sqrt(-log(0.5))/sc;
m=sum(p)/r;
copies=zeros(1,r);
M = m(1+copies,:),1;
H=(p-M)'*(p-M);
K=pinv(H);
```

```
D=dist(w1,p);
Z=K*D;
a1=radbas(Z.*(b1*ones(1,q)));
x=t/[a1; ones(1,q)];
w2=x(:,1:q);
b2=x(:,q+1);
```

#### FASE II. Simulación

Dados iniciales: Valores de Entrada: pe, Peso de Entrada: w1, Umbral de entrada b1, Peso de Salida: w2, Umbral de salida: b2, Amplitud d (sc, en Matlab). Salidas: Salida de la red: ts

Los pasos que se siguen son: 1) Ingresar datos, 2) Determinar dimensiones de Pe y W1, 3) Calcular las distancias Z, 4) Calcular las salidas de la capa oculta A, 5) Calcular la salida por el peso, 6) Calcular la salida ts, 7) Mostrar resultados.

Código de Matlab:

```
function [ts] = simnewrbfcv15r(w1,b1,w2,b2,p)
[s,r] = size(w1);
[r2,q] = size(p);
if(r~r2),error('matriz interna de dimensiones que no concuerdan. '), end
m=sum(p)/r;
copies=zeros(1,r);
M = m(1+copies,:),1;
H=(p-M)'*(p-M);
K=pinv(H);
D=dist(w1,p);
Z=K*D;
a1=radbas(Z.*(b1*ones(1,q)));
ts=(w2*a1)+b2;
```

#### EJEMPLO DE CORRIDA DEL PROGRAMA DESARROLLADO PARA APLICAR LA RND, CASO 2

Diseño de experimentos del artículo de Vasconcelos et al, "Design of Experiments. Improve Molded Part Quality", publicado en Quality Progress, N° 7 del 2003, donde se busca establecer los parámetros de proceso de un envase plástico elaborado por moldeo de inyección, cuyos parámetros de proceso del diseño son los de la Tabla 1.

**Tabla N.º 1. Parámetros de proceso del caso de Vasconcelos et al.**

Factores del proceso	Nivel bajo	Nivel alto
A. Velocidad de inyección (%)	40	75
B. Temperatura de moldeo °C	25	45
C. Temperatura de mezcla °C	205	235
D. Presión de trabajo, bar	25	45
E. Tiempo de trabajo	2	3
F. Tiempo de enfriamiento	10	25
G. Velocidad de eyección %	5	25

Solución con la RND:

Datos de Entrada: Se ingresa p, t y sc.

**CON LA RED NEWRBFCEV15r DESARROLLADA PARA APLICAR CON MATLAB**

**COMANDOS:**

Fase I: Entrenamiento

```
>> [w1,b1,w2,b2,K]=newrbfcev15r(p,t,sc);
```

Fase II. Simulación

```
>> [ts]=simnewrbfcev15rr(w1,b1,w2,b2,p)
```

CON LA ENTRADA DE p, EN LA SIMULACION, SE OBTIENEN LOS VALORES DE t:

ts =	69.5700	69.5700	69.5700	41.1367	41.1367	41.1367	74.6733
	74.6733	74.6733	62.3467	62.3467	62.3467	63.2700	63.2700
	63.2700	40.9700	40.9700	40.9700	70.3367	70.3367	70.3367
	42.3167	42.3167	42.3167				

**Resultados Variando la Entrada**

Con 1 vector Pe de entrada se obtiene un valor de t:

Sea Pe la entrada: 75;45;235;45;3;25;25

```
[ts]=simnewrbfcev15rr(w1,b1,w2,b2,pe)
```

ts = 69.5700

Si se modifica Pe1= 75;45;245;45;3;25;25

ts obtenido con la RND es 74.6733; mientras que con la ecuación de regresión se obtiene 72.04 y con la Red Newrbe de Matlab 42.3167. Ello demuestra la ventaja de la RND sobre la Red Newrbe.

Con la Calidad Deseada:

Suponiendo que se desea una calidad de salida de 85, con los datos iniciales:

- a) Se construye la Red Inversa, con entradas como salidas y viceversa; y se obtiene la red de aprendizaje.
- b) Se construye la Red Directa, y se ingresa la salida deseada (85) a la red antes obtenida y ob-

tener valores de entrada optimizados.

En el caso de usar los 24 datos se obtiene (con sc=0.2): 75; 24.84; 235; 24.84; 3; 9.88; 4.84.

**ANÁLISIS DE RESULTADOS**

Sobre los procedimientos:

Las RNA de FBR Función de Base Radial, Directa, se construyen como sigue:

- a. Dados pares de datos de entrada y de salida reales.
- b. Se calculan los centros y amplitudes con base a los datos de entrada. Inicialmente se prueba con los centros iguales o muy cercanos a los valores de entrada. De ser compleja la determinación se puede utilizar el algoritmo de K-medias, para agrupar datos.
- c. Las amplitudes se determinan con base a las distancias de Mahalanobis de los centros a los vectores de entrada.

**Tabla 2. Comparativo de errores obtenidos en los casos analizados**

Caso		1	2	3	4	5	6	7
Error al simular entrada 1	Red Desarrollada	0	0	0	0.0011 0.0004	0	0 0	0 0
	Análisis de Exp. Tradic.	No hace este cálculo	0.300	1.22	Sólo* compara promedios	No calcula	0.6 0.1	0.6 0.2
Error de simulación global	Red Desarrollada	24.31	9.16	0	0	0	0 0	0 0
	Análisis de Exp. Tradic.	96.,86	12.27	22.44	Sólo compara promedios	77.47	0.6 0.1	0.6 0.2

\* En este caso no calcula el error, sólo compara promedios y permite obtener niveles de factores recomendados.

- d. Se determinan los pesos y umbrales de la capa de salida con base a los valores objetivos de calidad que se persiguen, aplicando el método de mínimos cuadrados o el de la matriz inversa.
- e. Con esta red, se puede predecir el desempeño futuro del proceso, es decir, dados unas entradas determinar los valores de salida. Por otro lado también se puede obtener valores de entrada del proceso que han sido optimizados en función de una salida deseada.

Las RNA de FBR Función de Base Radial, Inversa, se construyen como sigue:

- a. De manera similar a la Red de FBR Directa, pero considerando las entradas como salidas y viceversa.

Si se requiere calcular los valores para una calidad deseada, se deben calcular las RNA Inversa y Directa, según el siguiente procedimiento:

- a. Dados pares de datos de entrada y de salida reales.
- b. Construir una RNA Inversa que tiene los datos de salida reales como entradas y los datos de entrada reales como salida.
- c. Obtenida la red inversa se aplican los datos de salida (deseados ó calidad deseada) para obtener los valores de entrada optimizados..
- d. Construir una RNA Directa que tiene los datos de entrada reales como entradas y los datos de salida reales como salida; es decir, se obtiene la Red Directa con los datos iniciales.
- e. Obtenida la Red Directa, allí se pueden aplicar los datos obtenidos en c. (valores de entrada optimizados), para obtener la salida deseada (calidad deseada).

A continuación se presenta el cuadro comparativo de los errores obtenidos para los 7 casos:

En la Tabla 2 se aprecia como los errores de la RND son menores que los de las otras metodologías, tanto para el caso de simular una entrada o todas a la vez.

Asimismo:

- a. Entre los diversos tipos de RNA perceptrón multicapa con aprendizaje backpropagation, la que usó el algoritmo Levenberg-Marquardt tuvo mejor desempeño.
- b. El desempeño de las RNA de Función de base radial fue mejor que las demás RNA, dentro de ellas las de Diseño Exacto, tanto la red NEWRBE como la RNA desarrollada. Dicho desempeño fue mejor que las RNA de Regresión Generalizada GRNN; por tener menor error.

La Red desarrollada tuvo mejor desempeño que la Red NEWRBE.

- c. En los 7 casos trabajados la RND tuvo mejor desempeño que las otras RNA y que el Método estadístico tradicional.

## CONCLUSIONES

1. Las Redes Neuronales Artificiales con Función de Base Radial, que utilizan la distancia de Mahalanobis, aplicadas a los diseños de experimentos para mejora de la calidad dan mejores resultados que los obtenidos mediante los métodos de análisis estadísticos tradicionales y del Perceptrón Multicapa con aprendizaje Backpropagation. Es así que se construyó la "Red desarrollada".
2. Las características principales de la Red neuronal artificial de función de base radial "Red desarrollada", que fue construida son:
  - Arquitectura con tres capas
  - En la capa oculta el aprendizaje es no supervisado.
  - En la capa de salida se da un aprendizaje supervisado
3. La Red desarrollada tiene como Función de Base Radial la Función de Gauss, se caracteriza por: centros, que son iguales a vectores de entrada; y los umbrales de la capa oculta de 0.8326 y por defecto con una amplitud de 1. Los pesos de la capa oculta a la capa de salida y los umbrales de la capa de salida se determinan siguiendo el método de de la matriz seudo inversa

## REFERENCIAS BIBLIOGRÁFICAS

1. Anand, K. (1993). "Improving Parafin Wax Yield through process optimization using Taguchi's Method of Experimentation". *Quality Engineering*, 6(1):39-56.
2. Bowles, M. y Montgomery, D. (1997). "How to formulate the ultimate Margarita: A tutorial on experiments with mixtures". *Quality Engineering* 10(2):239-253.
- 3.- Del Castillo, E., Montgomery, D. Y Mccarville, D. (1996). "Modified Desirability Functions for Multiple Response Optimization". *Journal of Quality Technology* 28(3):337-345.
- 4.- Hagan, Martín T., Demuth, Howard B., Beale, Mark. (1996). *Neural Network Design*. Editorial Thomson. China. p. 2.1.

5. Haykin, S. (2005). *Neural Networks*. Editorial Macmillan College Publishing Company. EEUU. p. 278.
6. Isasi, Pedro y Galván, Inés. (2004). *Redes de Neuronas Artificiales*. Editorial Pearson– Prentice Hall. España. p. 45.
7. Montgomery, Douglas. (2004 ). “Diseño y análisis de experimentos”. Editorial Limusa Wiley. México. p. 260.
8. Ross, Philip. (1988). “Taguchi Techniques for Quality Engineering”. Editorial McGrawHill. EEUU. p. 269.
9. Schmidt, S. y Launsby, R. (1994). “Understanding Industrial Designed Experiments”. Editorial Air Academy Press. EEUU. pp. 8-23.
10. Vasconcellos, A., Sodre, S. Y Rekab, K. (2003). “ Design of Experiments. Improve Part Quality”. *Quality Progress*. 2003(7):72-76.
11. Wasserman, P. (1993). “Advanced Methods in Neural Computing”. Editorial Van Nostrand Reinhold. EEUU. p. 147.