

RESUMEN

El artículo describe la implementación del algoritmo **Forward Select** para la construcción de una Red Neuronal RBF en el intérprete Scilab. El procedimiento eurístico para la elección de los nodos se describe y se realiza un experimento computacional que permite comprobar la bondad del algoritmo para aproximar funciones no lineales.

Palabras claves

Modos deslizante, **Forward Select**, .

1. INTRODUCCION

El tema de redes neuronales es muy extenso, un tipo de redes (de funciones radiales) permite un análisis formal que facilita su aplicación práctica. El problema de aproximación no lineal, cuando se dispone de una gran cantidad de datos afectados de ruido, es interesante para las aplicaciones. Si en un determinado problema se desea aproximar una función no lineal, partiendo de una serie de datos afectados de ruido, el procedimiento más común es realizar una Regresión Funcional, para ello es necesario suponer que una función determinada, expresa de la mejor forma la relación entre las variables dependientes e independientes. Para confirmar el tipo de función que mejor expresa esta relación existen técnicas estadísticas tales como la determinación del coeficiente de correlación. Sin embargo, el hecho de introducir la suposición, sobre el tipo de función, es un factor que limita la obtención de la mejor aproximación a un determinado conjunto de datos efectuados por ruido. Adicionalmente el problema de regresión funcional es inherentemente mal condicionado y por lo tanto una solución única no existe. Las redes con funciones de base radial presentan un entrenamiento mucho más rápido que el que se logra con técnicas como la de retro-propagación y si se le agrega además la técnica de planos deslizantes se pueden tener mejoras adicionales.

2. DESCRIPCION

El diagrama de la Figura 1 muestra el esquema del sistema completo utilizando la red de base radial. A partir del error y su derivada, se calcula un plano deslizante llamado pS , luego

aplicando una función $\psi(\)$ se obtiene el error de la señal de control e_c , y con base en ésta se actualizan los pesos de la red de base radial, la cual hace las veces de controlador.

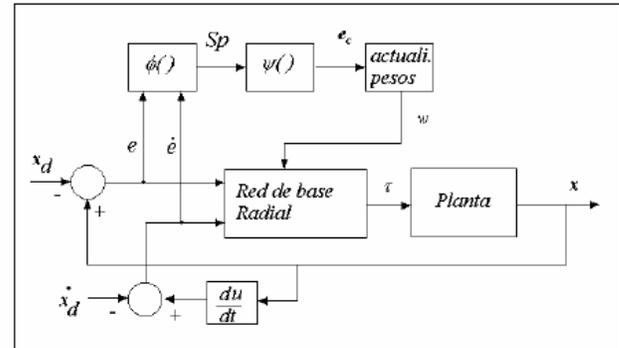


FIGURA 1

3. ESTRUCTURA DE UN RED

Para una Red Neuronal de Funciones de Base Radial (RBF), la idea es centrar funciones de base radial alrededor de los datos a aproximar, una de las funciones más utilizadas es la de tipo Gaussiana:

$$(f_i(u) = e^{-\frac{\|u-c_i\|^2}{\rho_i^2}})$$

En esta expresión c_i , ρ_i son el centro y ancho de la función i . Utilizando la propiedad de estas Redes Neuronales de aproximar cualquier función continua se puede escribir para la salida de una red RBF con n funciones Gaussianas:

$$y = \sum_{i=1}^n w_i \phi(u)$$

En la expresión anterior w_i son los pesos que necesitan ser adaptados a través de un algoritmo de aprendizaje. El problema que se plantea cuando se utiliza este tipo de aproximación, está en la cantidad de funciones radiales, centros c_i y anchos ρ_i a utilizar.

Características:

- Capa de entrada: En la primera capa se tiene como entradas el error, la derivada del error y una constante, reciben las señales del exterior, no realizan ningún preprocesado.
- Capa Oculta: la capa oculta está formada por neuronas con funciones de base radial, reciben las señales de la capa de entrada y realizan una **transformación local** y no lineal sobre dichas señales (Diferencia con el MLP).
- Capa de Salida: Se realiza una combinación lineal de las activaciones de las neuronas de la capa oculta y actúa como salida de la red.

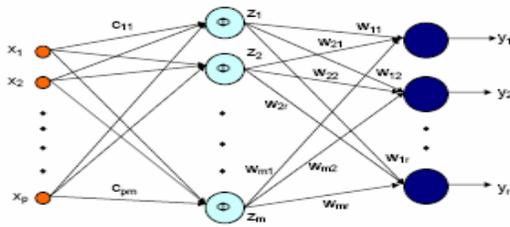


FIGURA 2 Red neuronal de función radial

Los pesos entre la capa de entrada y la oculta no reciben entrenamiento y su valor es de uno. El entrenamiento de los pesos de la capa oculta a la de salida se realiza utilizando un plano deslizante, como se muestra en la sección del algoritmo de aprendizaje. El error y la derivada del error se obtienen como:

$$e = x - x_d \quad (1)$$

$$\dot{e} = \dot{x} - \dot{x}_d \quad (2)$$

donde x es la salida de la planta y x_d es la salida deseada.

La salida de cada neurona de la capa oculta, la cual esta formada por funciones de base radial es:

$$y_i = \sum_{j=0}^m \exp\left(\frac{-(u_j - c_{ij})^2}{\sigma_{ij}^2}\right) \quad (3)$$

La salida total de la red se define por:

$$\tau = \sum_{i=1}^h y_i w_i \quad (4)$$

donde w_i denota el peso i de entrenamiento, y τ representa la salida total de la red.

4. TECNICA FORWARD SELECT.

Esta técnica consiste en añadir nodos a la red, uno a la vez en base a un criterio. La base para realizar la selección está formada de aquellos nodos que se obtienen considerando que cada dato es suplementado con un nodo. Por supuesto esta no es la solución, pues si usamos todos los datos, el resultado sería una función que aproxima el ruido. Los nodos se añaden a la red y se verifica el criterio de selección. Este criterio eurístico, está basado en la estimación del error de predicción en un futuro conjunto de prueba. Se dispone de los siguientes criterios.

4.1 Criterios de Selección de Modelos.-

Estos criterios no son sino estimaciones del error de predicción, es decir como se comportará nuestra red neuronal ante valores futuros desconocidos. Para su utilización es necesario definir la Matriz de Proyección y el Número Efectivo de parámetros en la red. La Matriz de Proyección está definida mediante:

$$P = I_p - HP^{-1}H^T$$

La matriz $A^{-1} = (H^T H - L)^{-1}$ está relacionada con la matriz H , que a su vez está formada por las evaluaciones de las funciones radiales para cada nodo. La matriz L , es una matriz diagonal que contiene los parámetros de regulación l_i utilizados para encontrar los pesos w_i de la red neuronal. El número efectivo de parámetros $g = p - \text{trace}(P)$. Adicionalmente se define S , como la suma al cuadrado de los errores entre los valores precedidos y los de prueba.

LOO Leave One Out Cross Validation.-

Este criterio se basa en la utilización de la siguiente expresión para la estimación del error futuro

$$\sigma_{LOO}^2 = \frac{y^T P (\text{diag}(P))^{-2} P y}{p}$$

Una variación de la expresión anterior que permite manejar el término que contiene a $\text{diag}(P)$ es Validación Generalizada Cruzada (GCV).

4.1.1 GCV Generalized Cross Validation.-

El error de predicción se calcula mediante:

$$\sigma_{GCV}^2 = \frac{py^T P^2 y}{(\text{trace}(P))^2}$$

Otros criterios que tienen la misma forma de GCV son:

4.1.2 UEV Unbiased Estimate of Variance.-

$$\sigma_{UEV}^2 = \frac{y^T P^2 y}{p - \gamma}$$

4.1.3 FPE Final Prediction Error.-

$$\sigma_{FPE}^2 = \frac{1}{p} (y^T P^2 y + 2 \gamma \sigma_{UEV}^2)$$

4.1.4 BIC Schwartz Bayesian Information Criterion.-

$$\sigma_{BIC}^2 = \frac{1}{p} (y^T P^2 y + \ln(p) \gamma \sigma_{UEV}^2)$$

5. ALGORITMO DE APRENDIZAJE

El aprendizaje se logra calculando el plano deslizante p S y el error de la señal de control e como en (5) y (6).

$$S_p \dot{=} e + \lambda \dot{e} \quad (5)$$

$$e_c = \tau - \tau_d = \frac{1}{\lambda} \left\{ \dot{S}_p + \zeta \text{signo}(S_p) \right\} \quad (6)$$

λ y ζ constantes.

Donde d τ es la señal de control deseada y como ésta no está disponible, es necesaria la expresión del lado derecho.

Con e se calcula el cambio en los pesos de la red neuronal.

$$\Delta w_i = \frac{-ky_i}{\rho \sum_{j=1}^h y_j + \mu} \text{signo}(e_c) \quad (7)$$

donde k, ρ, μ son constantes, y k es la tasa de aprendizaje.

5.1 Entrenamiento de un RBF

El entrenamiento de este tipo de redes, determina todos los parámetros de la red.

- Parámetros de la capa de salida: Pesos, W
- Parámetros de la capa Oculta: Centros, C y desviaciones asociadas d.

La determinación de los parámetros de la capa oculta, se realiza mediante la optimización en el espacio de entradas, ya que cada neurona va a representar una zona diferente en dicho espacio. La determinación de los parámetros de la capa de salida, la optimización se realiza en base a las salidas que se desea obtener (salidas deseadas), ya que en su globalidad, las redes de base radial se utilizan para aproximar relaciones entre el conjunto de variables de entrada y salida que definen el problema.

5.1.2 Aprendizaje Híbrido

- Fase No supervisada: Determinación de parámetros de la capa oculta.
- Fase Supervisada: Determinación de pesos en capa de salida.

5.1.2.1 Fase No Supervisada

Puesto que las neuronas ocultas se caracterizan porque representan zonas diferentes del espacio de entradas, los centros y las desviaciones deben de ser calculados con este objetivo (clasificar el espacio de entradas en diferentes clases).

Determinación de Centros:

- Algoritmo K-medias
- Mapas de Kohonen

Determinación de Desviaciones

Se deben de calcular de manera que cada neurona de la capa oculta se active en una región del espacio de entradas y de manera que el solapamiento de las zonas de activación de una neurona sea lo más ligero posible, para suavizar así la interpolación.

Varias aproximaciones:

- Media Uniforme de las distancias euclídeas del centro C_i a los p centros más cercanos.

$$d_i = \frac{1}{p} \sum_p \|C_i - C_p\|$$

- Media geométrica de la distancia del centro a sus dos vecinos más cercanos.

$$d_i = \sqrt{\|C_i - C_t\| \|C_i - C_s\|} \quad \text{con } C_t \text{ y } C_s \text{ los más cercanos a } C_i$$

5.1.2.2 Fase Supervisada

Se utiliza la técnica de corrección de Error (Adaline, Perceptrón Multicapa.)

Minimización de la función error dada por la salida de la red.

$$E = \frac{1}{N} \sum_{n=1}^N e(n)$$

Donde N es el número de patrones, y $e(n)$ es el error cometido por la red para el patrón X(n), que viene dado por:

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

Donde Y(n) es la salida de la red, S(n) es la salida deseada para un patrón X(n).

5.1.2.3 Determinación de pesos

$$w_{ik}(n) = w_{ik}(n-1) - \mu \frac{\delta e(n)}{\delta w_{ik}}$$

$$u_k(n) = u_k(n-1) - \mu \frac{\delta e(n)}{\delta u_k}$$

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

$$\frac{\delta e(n)}{\delta w_{ik}} = -(s_k(n) - y_k(n)) \frac{\delta y_k(n)}{\delta w_{ik}} \quad \frac{\delta y_k(n)}{\delta w_{ik}} = \Phi_i(n)$$

$$\frac{\delta e(n)}{\delta u_k} = -(s_k(n) - y_k(n)) \frac{\delta y_k(n)}{\delta u_k} \quad \frac{\delta y_k(n)}{\delta u_k} = 1$$

Quedando las ecuaciones de cambio como:

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

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

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

5.2 Aprendizaje Totalmente Supervisado

A diferencia con el método anterior, este tipo de aprendizaje no conserva, en principio, las propiedades o características locales de las redes de base radial. En este caso, todos los parámetros de la red, centros, amplitudes, pesos y umbrales, se determinan de manera completamente supervisada y con el objetivo de minimizar el error cuadrático medio.

En este proceso, en ningún momento se tiene en cuenta que las amplitudes alcancen valores tales que el solapamiento de las activaciones de las neuronas de la capa oculta sea lo más suave posible. Así que en principio, esa característica de localidad se puede perder.

Cálculo de parámetros

$$w_{ik}(n) = w_{ik}(n-1) - \mu \frac{\delta e(n)}{\delta w_{ik}} \quad c_{ij}(n) = c_{ij}(n-1) - \mu \frac{\delta e(n)}{\delta c_{ij}}$$

$$u_k(n) = u_k(n-1) - \mu \frac{\delta e(n)}{\delta u_k} \quad d_i(n) = d_i(n-1) - \mu \frac{\delta e(n)}{\delta d_i}$$

6. EJEMPLO DE UNA RBF

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

$$G(\|\mathbf{x} - \mathbf{c}_i\|) = \exp\left(-\|\mathbf{x} - \mathbf{c}_i\|^2\right), i=1,2.$$

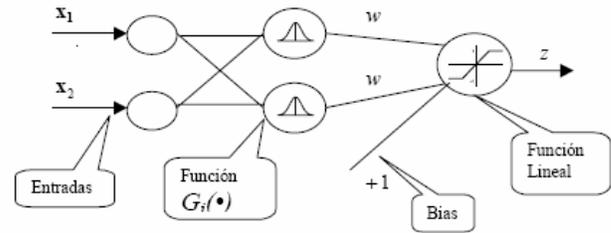
Deben existir dos centros y ellos ya son conocidos. Vienen dados por: $\mathbf{c}_1 = [11]^T$, $\mathbf{c}_2 = [00]^T$.

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

7. CONCLUSIONES

Lo que se puede concluir es que este tipo de modelo de Redes Neuronales es fácil de implementar y presenta buen comportamiento.

Las Funciones de Base Radial presentan una alternativa al tratamiento formal de las Redes Neuronales que permite realizar procedimientos constructivos, es decir determinar la estructura óptima de una red neuronal para una aplicación específica.

Por supuesto aún se debe solucionar el problema de la cantidad de neuronas, sin embargo existen varios procedimientos heurísticos que abordan este problema.

8. REFERENCIAS

- [14] <http://publiespe.espe.edu.ec/investigativas/ciencia/ciencia04.htm>
- [15] <http://icfich.wdfiles.com/local--files/teorias/rbfnn.pdf>
- [16] <http://www.varpa.org/~mgpenedo/cursos/scx/archivospdf/Tema5-6.pdf>
- [17] http://webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf