Redes Neuronales para Aproximación de Funciones

R.O.PUENTE* y J.A.HORAS+

FACULTAD DE CS. FISICO MATEMATICAS Y NATURALES. IMASL
UNIVERSIDAD NACIONAL DE SAN LUIS - CONICET
EJÉRCITO DE LOS ANDES 950, 5700 SAN LUIS. ARGENTINA
e-mail: jhoras@unsl.edu.ar

Resumen

Las redes neuronales multicapas alimentadas hacia adelante, usadas en este trabajo, pueden aproximar funciones continuas con un error dado, provistas del suficiente número de unidades de procesamiento elemental. No existen métodos para determinar a priori el tamaño apropiado de la red ni su conectividad. Redes neuronales con gran número de nodos y conexiones tienen mayor flexibilidad que las pequeñas para ajustarse a los datos usados en el entrenamiento, pero su performance es pobre en cuanto a su capacidad de generalización. Esto es de gran importancia en aproximación de funciones.

En este trabajo se presenta y analiza un método de construcción de redes de mínima complejidad mostrando su uso en aproximación de funciones continuas.

Abstract

A multilayer feedforward neural network can be made to approximate a continue function as much as one likes, if it has a sufficient amount of nodes and connections. There are no methods for to *a priori* determine neither the size nor the connections of such a network. Networks with many connected nodes are more flexible than those with a few, in order to fit the points of a training set, but they have a poorer performance in generalization capability. This point is very important in function approximation.

In this paper a method to construct small complexity neural networks is introduced. It is also shown its use in continuous functions approximation.

Introducción

Las redes neuronales multicapas alimentadas hacia adelante pueden aproximar funciones continuas con un error dado, provistas del suficiente número de nodos (aproximación universal [1]). No existen métodos para determinar a priori el tamaño apropiado de la red ni su conectividad. Redes con gran número de nodos y conexiones tienen mayor flexibilidad para ajustar datos, pero a costa de un mayor tiempo de entrenamiento y logran pobre performance de generalización. El objetivo deseable consiste en la construcción de redes con el menor número posible de nodos y conexiones. En este sentido, el ideal es lograr redes de mínima complejidad, aunque sean obtenidas mediante técnicas heurísticas [2].

En este trabajo se presenta un procedimiento de entrenamiento y construcción simultánea de una red creciente. El método de construcción es iterativo, incorpora un nodo por etapa. Dicha inclusión se determina maximizando la correlación entre el error remanente de la etapa previa y la salida del nodo que se agrega. Se entrena posteriormente sólo la capa de salida de la red aumentada [3].

Se comparan los resultados logrados en esta red con los obtenibles en redes de arquitectura fija de una capa oculta, entrenadas por Back Propagation.

Descripción del algoritmo constructivo

Se describe a continuación una red de n entradas externas, m nodos en la capa oculta y un nodo de salida. En la figura 1 se muestra su arquitectura para $n = 2$, $m = 3$.

Denotamos con P el número de patrones de entrenamiento, $o_m^{(\mu)}$ es la salida de la red para el patrón μ ($\mu = 1, 2, \ldots, P$), v_0 es el peso del elemento bias, v_j es la conexión entre la salida y la entrada externa j ($j = 1, 2, \ldots, n$), v_{n+i} es la

*Departamento de Matemáticas
+Departamento de Física. Investigador CONICET

conexión entre la salida y el nodo i de la capa oculta:

$$o_m^{(\mu)} = \sum_{k=0}^{n+m} v_k \, x_k^{(\mu)} \,. \qquad (1)$$

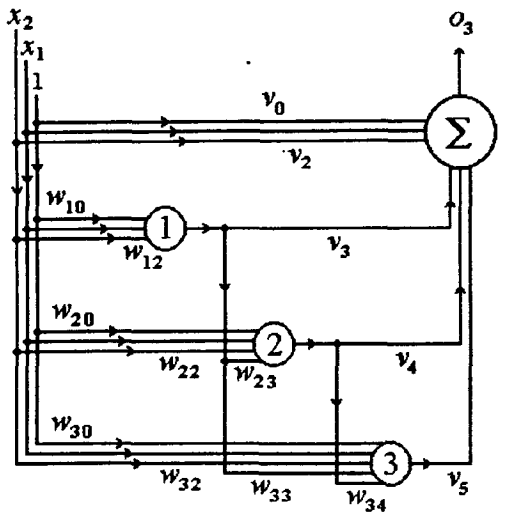

Figura 1: Arquitectura del algoritmo constructivo. (ver texto)

Se denota como $x_j^{(\mu)}$ la componente j de la entrada externa, $x_{n+i}^{(\mu)}$ es la salida del nodo i de capa oculta, $x_0^{(\mu)} = 1$. Con pesos w_{ik} en las conexiones de entrada al nodo i y función de transferencia sigmoidal $\Phi(t) = \tanh(t/2)$, se tiene:

$$x_{n+i}^{(\mu)} = \Phi\left(\sum_{k=0}^{n+i-1} w_{ik} \, x_k^{(\mu)} \right). \qquad (2)$$

Si $y^{(\mu)}$ es el valor correcto de la salida,

$$E_m^{(\mu)} = o_m^{(\mu)} - y^{(\mu)}, \quad EC_m = \sum_{\mu=1}^{M} \left(E_m^{(\mu)} \right)^2 . \qquad (3)$$

La red inicial ($m = 1$) es entrenada para minimizar el error cuadrático EC_1 con el algoritmo usual del Perceptron [4]. Suponiendo realizadas m iteraciones, para la siguiente se procede así:

a) Se construye el nodo m+1, con n entradas externas, 1 bias y m entradas provenientes de la capa oculta de la red de tamaño m, y función de transferencia sigmoidal. Los pesos de estas conexiones $w_{m+1\,k}$ ($k = 0, 1, \ldots, n+m$) son los que resultan de maximizar la correlación (corr) entre el error remanente y la salida del nuevo nodo:

$$\max_{w_{m+1\,k}} \left\{ \left| \, \mathrm{corr}(E_m, x_{n+m+1}) \, \right| \right\}. \qquad (4)$$

b) Se instala el nodo (m+1) con las conexiones obtenidas en (4), y conexión de salida dada por:

$$v_{n+m+1} = \frac{(E_m \cdot x_{n+m+1}) - P\,\overline{E_m}\;\overline{x_{n+m+1}}}{(x_{n+m+1} \cdot x_{n+m+1}) - P\,(\overline{x_{n+m+1}})^2} \qquad (5)$$

$$\overline{E_m} = \frac{1}{P} \sum_{\mu=1}^{P} E_m^{(\mu)}, \quad \overline{x_k} = \frac{1}{P} \sum_{\mu=1}^{P} x_k^{(\mu)}. \qquad (6)$$

Además, se cambia v_0 (bias de salida final):

$$v_{0\,\text{nuevo}} = v_{0\,\text{viejo}} + \overline{E_m} - v_{n+m+1}\,\overline{x_{n+m+1}} \,. \qquad (7)$$

c) Se reentrena la red aumentada para minimizar el error cuadrático EC_{m+1} con el algoritmo del Perceptron [4] modificando únicamente las conexiones v_k ($k = 0, 1, 2, \ldots, n+m+1$).

Obsérvese que en la etapa a), cuando se entrena el nodo (m+1), la red (de tamaño m) permanece fija, y cuando se modifican las conexiones de salida (etapa c) los demás pesos se mantienen fijos.

Análisis de resultados

El esquema anterior es aplicable para aproximar una función f de n variables reales en una región acotada $D \subset \mathbf{R}^n$. El conjunto de entrenamiento: $\{x^{(\mu)} : \mu = 1, 2, \ldots, P\} \subset D$ se considera fijo.

En particular, se utilizó la función $f(x,y) = 0.8 \cos x \, \mathrm{sen}\, 5y$ en el cuadrado $D = [-1,1]\mathrm{x}[-1,1]$ (ver Ref. [5]), tomando como conjunto de entrenamiento una malla rectangular regular de 100 puntos. Para maximizar la correlación (etapa a)) se usó gradiente ascendente, iniciando con pesos aleatorios. Se seleccionó la mejor correlación de 100 ensayos.

Para comparación se usaron redes de arquitectura fija con 2 entradas externas, una salida y m nodos ocultos, para m = 5, 10, 15, 20, 25 y 30, función de transferencia $\Phi(t) = \tanh(t/2)$ en la capa oculta, $\Phi(t) = t$ en la salida, con bias en las dos capas.

En ambos casos el parámetro de aprendizaje fue h = 0.1 y el criterio de detención fue la variación relativa del ECM_m promedio de 50 épocas consecutivas menor que 10^{-3}, donde

$$ECM_m = \left[\sum_{\mu=1}^{P} \left(y^{(\mu)} - o_m^{(\mu)} \right)^2 \right]^{1/2} . \qquad (8)$$

Se realizaron 20 ensayos y los resultados se muestran en la figura 2.

Conclusiones

El error cuadrático medio ECM logrado con la red en cascada es menor al obtenible con redes de arquitectura fija para tamaños comparables, y presenta menor dispersión (esto no sucede en redes pequeñas, pero en estos casos el error es inaceptable para ambos métodos). El error continúa descendiendo con el crecimiento de la red en cascada, mientras permanece relativamente estacionario en el otro caso. Desde el punto de vista del error, el algoritmo constructivo es superior.

El tiempo de aprendizaje de las redes de tamaño similar, es comparable. Sin embargo, con redes de arquitectura fija se requiere además algún tipo de experiencia o bien múltiples ensayos para determinar su mejor tamaño y conectividad. Por el contrario, con el algoritmo expuesto ambos se autoajustan, obteniéndose una importante reducción de costo computacional.

Los métodos de "poda" actúan después de entrenar una red grande eliminando un cierto número de conexiones o nodos en la arquitectura inicial (ver Refs. en [6]). El algoritmo presentado incorpora nodos completos y puede pensarse que la red final es el resultado de no incluir nodos innecesarios, logrando una arquitectura más simple.

Por último, el algoritmo constructivo produce una buena aproximación a la red de mínima complejidad.

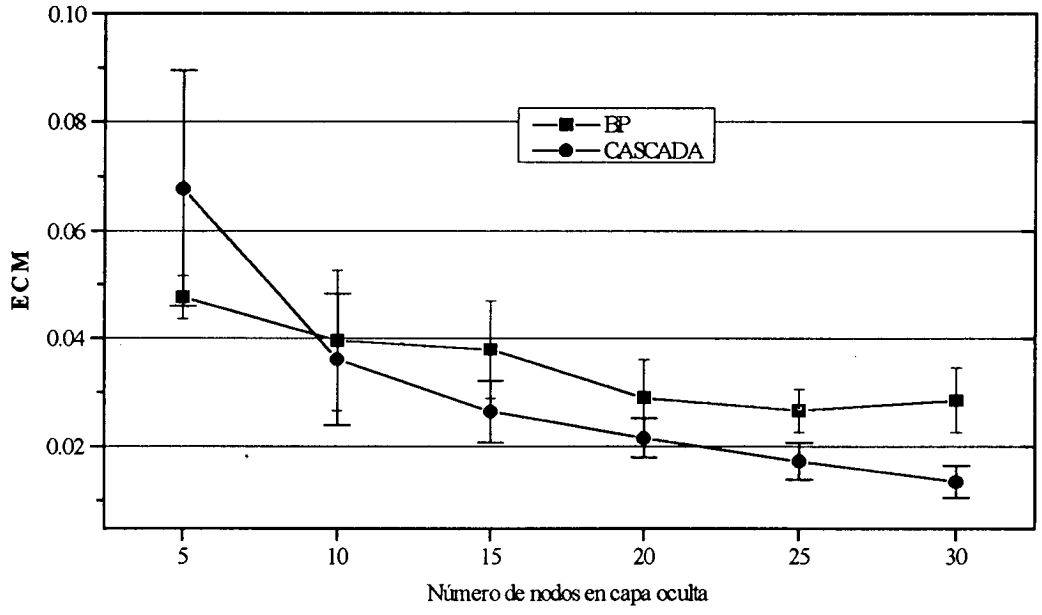

Figura 2: Performance de redes de arquitectura fija (BP) y algoritmo constructivo (Cascada). Se grafica el ECM promedio vs. número de nodos y la dispersión denotada por las barras verticales.

Referencias

1 - G. Cybenko, *Mathematics of Control Signals and System* **2**, 303-314, (1989)

2 - A. Blum and R.L. Rivest, *Proceedings of the First Workshop on Computational Learning Theory* (Morgan Kaufmann 1988) p. 9

3 - S.E. Fahlman and C. Lebiere, *Advances in Neural Information Processing System 2* (Morgan Kaufmann 1990), pp. 524-532

4 - J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the Theory of Neural Computation*, (Santa Fe Institute studies in the sciences of complexity. Lecture notes: v. I), Addison-Wesley P.C., 1991, pp. 102-107

5 - J. Mitchell, *Network* **3**, 19-25 (1992)

6 - A.U. Levin, T.K. Leen and E. Moody, *Advances in Neural Information Processing Systems 6* (Morgan Kaufmann 1994), pp. 35-42