Entrenamiento de redes neuronales: Análisis de métodos generalizados de minimización por descenso según el gradiente

P.M. GRANITTO, P.F. VERDES, H.D. NAVONE Y H.A. CECCATTO*

IFIR (UNIVERSIDAD NACIONAL DE ROSARIO - CONICET)
BV. 27 DE FEBRERO 210 BIS - (2000) ROSARIO - ARGENTINA
e-mail: ceccatto@ifir.ifir.edu.ar

El entrenamiento de redes neuronales requiere el uso de métodos de minimización sofisticados, de manera de evitar que el proceso de aprendizaje de la información contenida en la base de datos se frustre por la presencia de mínimos locales. En este trabajo se discuten las ventajas relativas de los algoritmos de minimización de Levenberg-Marquardt y gradientes conjugados, y se comparan con el método estándar de retropropagación de errores. Sobre la base de un ejemplo —vinculado a la reconstrucción de la dinámica de un sistema afectado por ruido— se establece la conveniencia de utilizar algoritmos híbridos que combinan los métodos mencionados.

Training a neural network requires sophisticated minimization algorithms to avoid getting stuck in local minima during the process of learning the information contained in the database. In this work we discuss the relative advantages of two simple gradient-descend algorithms, the Levenberg-Marquardt and conjugate gradient methods, and compare them with the standard backpropagation rule. On the basis of an example —related to the dynamics reconstruction of a noisy system— we establish the convenience of using hybrid algorithms which combine the mentioned methods.

PACS: 07.05.Mh

Introducción

Las redes neuronales (RN) constituyen modelos computacionales de significativa importancia para el análisis y predicción de series temporales[1,2]. En los últimos años se han aplicado a distintos sistemas, incluyendo mapas caóticos[3,4] y series de datos reales tales como datos meteorológicos[4] e indicadores económicos[5].

Una vez definidas la arquitectura de una RN (número de capas y cantidad de neuronas por capa) y las funciones de transferencia asociadas a cada unidad de procesamiento, las RN aplican un espacio de entrada o estímulos externos a un espacio de salida o respuesta. Esto es, cada salida $\mathbf{O}$ es una función de la entrada $\mathbf{I}$ y de las conexiones entre neuronas W, $\mathbf{O}=\mathbf{F}(\mathbf{I},W)$. El entrenamiento de una RN consiste en la búsqueda de las interconexiones W que representen más correctamente a los pares entrada-salida $(\mathbf{I},\mathbf{O})$ (patrones de entrenamiento), y que a su vez generalicen esta representación sobre un conjunto de pares definidos para la validación del modelo. Estos últimos patrones no se le presentan a la red durante la fase de aprendizaje.

Dado un conjunto de m patrones de entrenamiento, el proceso de aprendizaje consiste en minimizar el error $E(W)$:

$$\min_W E(W) = \min_W \sum_{i=1}^{m} (\mathbf{F}(\mathbf{I}_i,W) - \mathbf{O}_i)^2$$

Potencialmente, las RN tienen la capacidad de aproximar cualquier mapa, pero este proceso se ve muchas veces dificultado por la complejidad de la superficie de error, con muchos mínimos locales en el espacio de los parámetros. Debido a ello, se han utilizado en la literatura métodos de minimización muy sofisticados (recocido simulado, algoritmos genéticos, etc.), aunque los mismo implican en general un alto costo computacional.

En este trabajo se comparan distintos métodos de minimización de descenso por el gradiente en la reconstrucción de dinámicas

* Autor a quien debe dirigirse la correspondencia.

complejas usando RN. Los métodos considerados son: retroprogagación de errores[6] (BP), gradientes conjugados[7] (GC) y Levenberg-Marquardt[7] (LM), y se aplican a la predicción de una serie temporal generada por ruido. Si bien estos métodos son mucho más sencillos que los mencionados precedentemente, se verá que la utilización conveniente de los mismos permite obtener muy buenos resultados manteniendo los tiempos de cálculo en valores razonables.

Serie temporal estudiada

A los efectos de comparar los distintos métodos de minimización mencionados se propone la siguiente serie temporal no lineal generada por la presencia de ruido[8]:

$$x_{t+1} = 0.2 - 3x_t + 3\sqrt{x_t} + \rho_{t+1} = f(x_t) + \rho_{t+1}$$

donde ρ_{t+1} es un término de ruido blanco. La parte determinista $f(x_t)$ de esta ecuación tiene un ciclo límite de período dos. Cuando se le adiciona ruido se genera una señal compleja con distintas concentraciones de puntos en el mapa de las fases (Figura 1). Esto dificulta el proceso de aprendizaje de su dinámica.

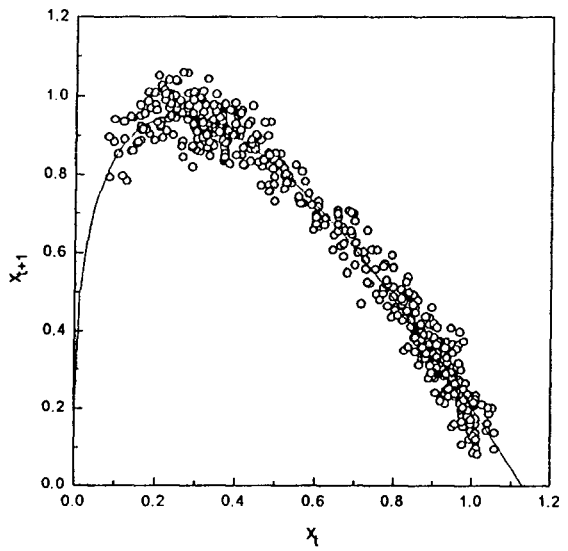

Figura 1: Mapas de las fases correspondientes a la serie ruidosa estudiada (círculos abiertos) y a la ley determinista $f(x_t)$ subyacente (línea llena).

El conjunto de entrenamiento utilizado consiste en 500 iteraciones a partir de $x_0 = 0.3$ del mapa definido en la ecuación anterior, con un ruido blanco de desvío standard $\sigma_\rho = 0.05$. Los siguientes 500 registros se utilizan para evaluar la capacidad de generalización lograda usando los distintos métodos. Además, y dado que la ley $f(x_t)$ subyacente es conocida (línea llena en la Figura 1), se evalúa la capacidad de modelización de la misma que adquiere la red sobre 500 valores equiespaciados en el intervalo [0,1].

Evaluación de los procesos de aprendizaje

La eficiencia de los distintos métodos se caracteriza mediante la variancia relativa promedio (*ARV*) definida por la siguiente expresión:

$$ARV(S) = \frac{\sum_{t \in S} (x_t - p_t)^2}{\sum_{t \in S} (x_t - \bar{x}_t)^2}$$

donde S indica alternativamente el conjunto de entrenamiento o validación usado, p_t es la predicción de la red neuronal para x_t, y $\bar{x}_t$ es el promedio de x_t en S. Esta cantidad es 1 para una predicción constante igual al promedio de las iteraciones y 0 para una predicción perfecta.

Para reconstruir la dinámica de la serie temporal estudiada se utilizaron redes de arquitectura 1:2:1, fijando en cada neurona de la capa oculta una función de transferencia sigmoide y una unidad lineal en la capa de salida. Para cada método de minimización estudiado se tomó un conjunto de 10 matrices de pesos inicializados con valores al azar. En cada proceso de aprendizaje se evaluó la evolución del *ARV* sobre los conjuntos de validación a los efectos de evitar el posible sobreentrenamiento de la RN. En la Tabla 1 se muestran los mejores valores de *ARV* correspondientes al intervalo de validación, conseguidos a partir de las distintas matrices de pesos al azar. A su vez, y a los efectos de evaluar la capacidad de reconstrucción de la dinámica intrínseca de la serie, también se muestran los *ARV* calculados respecto de la ley determinista sobre el intervalo [0,1].

TABLA 1: ARV SOBRE EL CONJUNTO DE VALIDACION DE LA SERIE ESTUDIADA Y POR LA LEY DINAMICA

Métodos	ARV Datos	ARV Ley	CPU
BP	0.03090	0.00340	17'42''
GC	0.03116	0.03927	17''
LM	0.03077	0.00266	6''

El método BP converge a valores razonables de *ARV*, tanto respecto al conjunto de datos usados como validación como a los valores generados

por la ley $f(x_t)$, para 9 conjuntos de pesos iniciales del total de 10 analizados. En el caso restante, el proceso de minimización queda estancado en un mínimo local. El método GC converge en todos los casos a valores aceptables del error de validación sobre la serie. Sin embargo, no logra modelizar correctamente la dinámica subyacente en los datos, originando un *ARV* muy alto para los datos generados directamente aplicando la ley. La metodología de LM queda atrapada en mínimos locales muy frecuentemente, convergiendo en sólo 1 de los 10 casos analizados. Sin embargo, en este caso los valores de *ARV* obtenidos para la validación sobre los datos y respecto de la ley son los mejores respecto de los métodos anteriores.

La convergencia o no hacia valores razonables de *ARV* depende de cada método y de la inicialización particular de los parámetros de la red. En base a las características particulares de cada técnica se combinaron las metodologías empleadas. Los mejores resultados obtenidos para cada caso se resumen en la Tabla 2.

TABLA 2: METODOS COMBINADOS

Métodos	ARV Datos	ARV Ley	CPU
BP-LM	0.03065	0.00220	17'43''
GC-LM	0.03055	0.00214	45''
BP-GC	0.03048	0.00238	17'48''

Como se aprecia en dicha tabla, la combinación de métodos de minimización permite obtener valores de *ARV* menores tanto para la validación sobre los datos como sobre la ley. Las metodologías de LM y GC, como se explicara anteriormente, aportan nuevos elementos al proceso de minimización por gradientes y ambas permiten mejorar la generalización alcanzada por el BP. Si bien todos los valores finales son similares, los mejores resultados sobre los datos de validación se obtienen combinando BP y GC, mientras que sobre la ley determinista surgen de la combinación GC y LM. En este último caso, de los 10 entrenamientos analizados 9 convergieron aceptablemente, mientras que en el restante el proceso de minimización condujo a un mínimo local. Notar, sin embargo, la economía de tiempo de CPU con respecto al uso del BP combinado con los otros métodos.

Los tiempos de CPU indicados en las Tablas 1 y 2 corresponden a implementadaciones de las distintas metodologías en una computadora SUN Ultra 1 con procesador de 167 MHz..

Conclusiones

Los distintos métodos empleados para el entrenamiento de RN quedan atrapados frecuentemente en mínimos locales. Las metodologías de GC y LM son las más afectadas por la estructura compleja de la función error, dado que minimizan *E(W)* en forma global. En el BP, en cambio, esta problemática es mucho menor, dado que explora el espacio de parámetros corrigiendo el error cometido sobre un solo patrón de entrenamiento por vez. El mismo resulta por ello un método más costoso desde el punto de vista computacional.

La combinación de las distintas metodologías de entrenamiento permite aprovechar las características particulares de cada una de ellas, mejorando considerablemente las generalizaciones obtenidas y optimizando los tiempos de procesamiento.

Referencias

1- Elsner, J.B. & Tsonis, A.A. *Bulletin American Meteorological Society*, **73**, 49-60 (1992).

2- Weigend, A.S., Huberman, B.A., & Rumelhart, D.E. *International Journal of Neural Systems*, **1**, 193-209 (1990).

3- Elsner, J.B. *Journal of Physics A: Mathematical and General*, **25**, 843-850 (1992).

4- Navone, H.D. & Ceccatto, H.A. *Climate Dynamics*, **10**, 305-312 (1994).

5- *Neural networks in the capital markets*, Ed. Apostolos-Paul Refenes (Wiley finance editions, 1995).

6- *Parallel distributed processing*, Rumelhart D.E., J.L. McClelland and the PDP Research Group (The MIT Press, 1986).

7- *Numerical Recipes*, Press W.H., Flannery B.P., Teukolsky S. A. and Vetterling W.T. (Cambridge University Press, 1990).

8- Saxén H., *International Journal of Neural Systems*, 7, 195-201 (1996).