

Optimización en la solución del problema inverso en geofísica usando el algoritmo de entrenamiento supervisado de Levenberg-Marquardt

YAQUELINE FIGUERO BAÉZ

Estudiante de Maestría en Geofísica, Universidad Nacional de Colombia

RAMIRO LEÓN DIAZ CAMPOS

Profesor asociado. Departamento de Geociencias, Universidad Nacional de Colombia

e-mail: ramdiaz@ciencias.unal.edu.co

RESUMEN

En este trabajo se presenta una metodología para el entrenamiento supervisado de redes neuronales, utilizando el algoritmo de Levenberg-Marquardt. Este método se aplica en gravimetría para optimizar la convergencia en el problema de inversión.

ABSTRACT

In this work it is introduced a methodology for the supervised training of the Neural Networks, using Levenberg Marquardt's algorithm. This method is applied in gravimetry for the optimization in the convergence at the inverse problem.

INTRODUCCIÓN

El método de inversión en geofísica es un problema de estimación y optimización; existen varios métodos para solucionarlo. En este artículo se usa el algoritmo de entrenamiento de redes neuronales Levenberg - Marquardt (ALM) que, apoyado en la información suministrada, permite estimar y optimizar parámetros físicos y geométricos de una estructura.

MÉTODO DIRECTO

Mediante el algoritmo de Talwani se puede calcular el campo gravitacional de un cuerpo irregular a lo largo de un perfil que corta la estructura perpendicularmente. La sección transversal se aproxima a un polígono con un número finito de vértices y una densidad de masa ρ (Talwani, 1959).

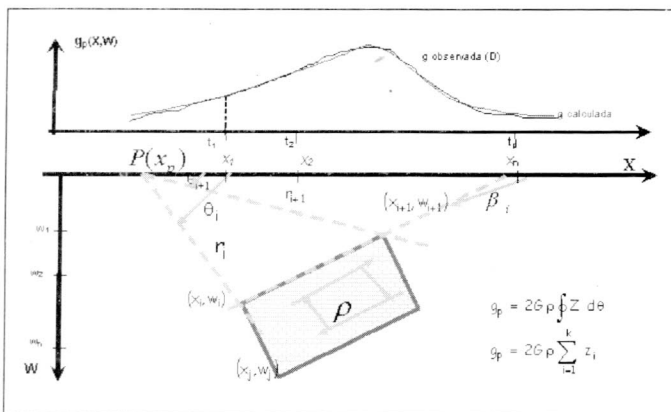


Figura 1. Método de Talwani para gravimetría.

ρ : Densidad de masa.

$$z_i = A_i \text{sen} \beta_i [(\theta_i - \theta_{i+1}) + \text{tg} \beta_i \text{Ln}(C)]$$

$$A_i = \frac{x_{i+1} + w_{i+1}(x_{i+1} - x_i)}{(w_i - w_{i+1})} \quad \beta_i = \arctan \left(\frac{w_{i+1} - w_i}{x_{i+1} - x_i} \right)$$

$$\theta_i = \arctan \left(\frac{w_i}{x_i} \right) \quad \theta_{i+1} = \arctan \left(\frac{w_{i+1}}{x_{i+1}} \right)$$

$$C = \frac{\cos \theta_i (\text{tg} \theta_i - \text{tg} \beta_i)}{\cos \theta_{i+1} (\text{tg} \theta_{i+1} - \text{tg} \beta_i)}$$

ALGORITMO DE LEVENBERG-MARQUARDT (ALM)

En este algoritmo se determina el error de retropropagación con base en el error cuadrático.

$$F(W^k) = \sum_{k=1}^p (D - g_p^k(X, W))^2 \quad (1)$$

$F(W^k)$ = Función por optimizar. Es una función estado que durante la evolución dinámica se aproxima a un estado mínimo estable y óptimo.

k = identifica el número de salida de la red y se relaciona con el proceso iterativo.

$k = 1, 2, 3, \dots, m$ m = número máximo de iteraciones.

$D = [d_1, d_2, d_3, \dots, d_p]^T$ representa el patrón en el proceso de entrenamiento y corresponde a los valores del campo gravitacional observados en los puntos $t_1, t_2, t_3, \dots, t_p$

p = Número de observaciones

(X, W) : Son los vértices del polígono.

$X = [x_1, x_2, x_3, \dots, x_j]$ son las abscisas de los vértices del polígono.

$W = [w_1^m, w_2^m, w_3^m, \dots, w_j^m]^T$ es el vector de pesos o profundidades de los vértices del polígono al que aproxima la sección transversal. El proceso se inicia con una primera aproximación.

$g_p(X, W) = [g_1, g_2, \dots, g_p]^T$ es la función de activación o modelo que permite calcular la gravedad mediante el algoritmo de Talwani.

$$D - g^k(X, W) = E^k \tag{2}$$

$E = [e_1, e_2, \dots, e_p]^T$ es el vector de desviaciones de los datos de gravedad generados por el modelo con respecto a las observaciones.

Remplazando la ecuación (2) en la ecuación (1), la función de estado se convierte en:

$$F(W^k) = \sum_{k=1}^p (E^k)^2 \tag{3}$$

El ALM se puede aplicar utilizando diferentes desarrollos, entre ellos se pueden mencionar: ajuste por mínimos cuadrados con información a priori para el problema inverso no lineal (Meju, 1994), expansión de orden uno de la serie de Taylor para el problema inverso no lineal, donde el gradiente se aplica a la función teórica (Mirko, 2000). En este trabajo se utiliza el procedimiento planteado por Serdar Iplikci (2000), que se desarrolla a partir del método de Newton.

La ecuación de recurrencia en un proceso iterativo es:

$$W^{k+1} = W^k + \Delta W^k \tag{4}$$

El término de ajuste en el método de Newton se calcula de la siguiente manera:

$$\Delta W^k = [H(W^k)]^{-1} A(W^k) \tag{5}$$

Si el sistema converge en un proceso iterativo, ΔW^k debe tender a cero, siendo $H(W^k)$ la matriz Hesiana y $A(W^k)$ el vector gradiente.

$$H(W^k) = \begin{bmatrix} \frac{\partial^2 F}{\partial w_1^2} & \frac{\partial^2 F}{\partial w_1 \partial w_2} & \dots & \frac{\partial^2 F}{\partial w_1 \partial w_n} \\ \frac{\partial^2 F}{\partial w_2 \partial w_1} & \frac{\partial^2 F}{\partial w_2^2} & \dots & \frac{\partial^2 F}{\partial w_2 \partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 F}{\partial w_n \partial w_1} & \frac{\partial^2 F}{\partial w_n \partial w_2} & \dots & \frac{\partial^2 F}{\partial w_n^2} \end{bmatrix} \tag{6}$$

$$A(W^k) = \left[\frac{\partial F}{\partial w_1} \dots \frac{\partial F}{\partial w_n} \frac{\partial F}{\partial p} \right]^T \tag{7}$$

El vector gradiente y la matriz Hesiana se pueden escribir en términos de la matriz Jacobiana de la siguiente manera:

$$A(W^k) = 2J(W^k)^T E^k \tag{8}$$

$$H(W^k) = 2J(W^k)^T J(W^k) + \text{Términos omitidos} \tag{9}$$

La matriz Jacobiana $J(W^k)$ es:

$$J(W^k) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_n}{\partial w_1} & \frac{\partial e_n}{\partial w_2} & \dots & \frac{\partial e_n}{\partial w_n} \end{bmatrix} \tag{10}$$

Después de sustituir las ecuaciones (8) y (9) en (5), se genera la actualización por el método de Gauss-Newton, y el factor de corrección es:

$$\Delta W^k = -[J(W^k)^T J(W^k)]^{-1} J(W^k)^T E^k \tag{11}$$

En el método de ALM, el posible problema de singularidad de la matriz $[J(W^k)^T J(W^k)]$ se resuelve introduciendo un término adicional que corresponde al factor de entrenamiento en redes neuronales.

$$\Delta W^k = -[J(W^k)^T J(W^k) + \beta^k]^{-1} J(W^k)^T E^k \tag{12}$$

SISTEMA DE ENTRENAMIENTO SUPERVISADO

En redes neuronales, un sistema de entrenamiento se puede ver como una transformación lineal, no lineal o un enfoque de sistemas. En este sistema de entrenamiento supervisado, los vectores que corresponden a los pesos de entrada W_m y de salida W_{m+1} se usan para calcular mediante el modelo de Talwani, un nuevo campo gravitacional. El sistema de entrenamiento supervisado, es un proceso iterativo que, partiendo de una primera aproximación, va ajustando los pesos de la red hasta llegar a sus valores óptimos.

La arquitectura está constituida por 3 capas (figura 2). La capa de entrada suministra información de las variables a los neurones de la segunda capa. La segunda capa se identifica con el perfil gravitacional, que permite –con los datos observados– calcular de nuevo los pesos para convertirse luego en la capa de salida.

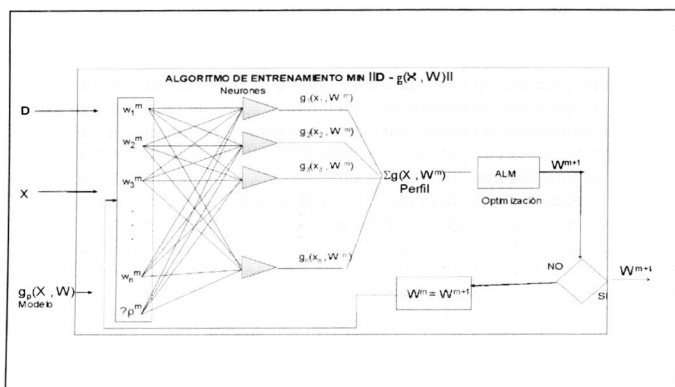


Figura 2. Sistema de entrenamiento supervisado.

FACTOR DE ENTRENAMIENTO

El método de regularización bayesiana, es una generalización del Algoritmo de Levenberg – Marquardt (Tarantola, 1987) usa la matriz de covarianza para los datos y la matriz de covarianza para los parámetros del modelo; se caracteriza por producir una buena pero lenta convergencia. En este proceso se utilizó la matriz de covarianza para los parámetros (C_m), como factor de entrenamiento, ya que refleja su verdadera dispersión.

La ecuación (12) se transforma en:

$$\Delta W^k = -[J(W^k)^T J(W^k) + C_m^k]^{-1} J(W^k)^T E^k \quad (13)$$

CURVA DE APRENDIZAJE

La curva de aprendizaje se obtiene al graficar el número de iteraciones contra la función estado. Utilizando el factor de entrenamiento C_m , se garantiza una rápida convergencia a una solución óptima y no presenta mínimos ni máximos locales. Ver figura 3.

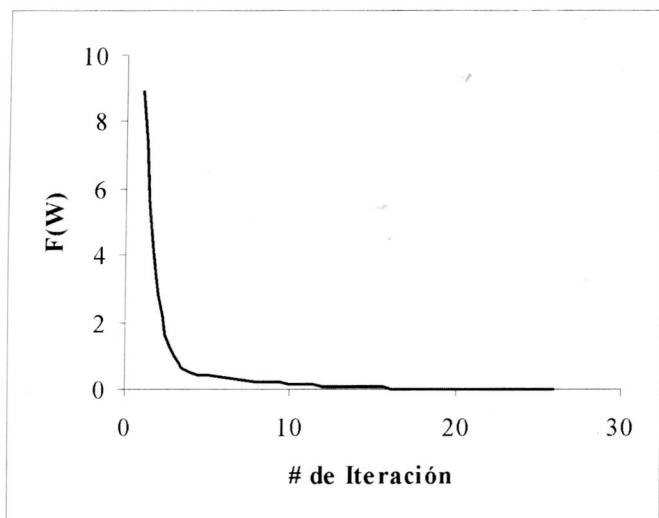


Figura 3. Curva de aprendizaje.

VENTAJAS DE ALM

- Este algoritmo se ajusta con información a priori, que regulariza el cálculo de los pesos a través de la penalización de pendientes, rugosidad o derivadas de alto orden en el modelo (términos omitidos). Es útil para la obtención de una solución suavizada (Scalles, 2001).
- Ajusta los valores absolutos de las perturbaciones de los parámetros durante sucesivas aproximaciones de Taylor (Meju, 1994).
- Combina la velocidad del algoritmo de Newton con la estabilidad del método de gradiente descendente (Meju, 1994).
- Usando el factor de entrenamiento C_m , este algoritmo permite una rápida convergencia en pocas iteraciones.

ESTADO DE LA INFORMACIÓN Y DENSIDAD DE PROBABILIDAD

La aproximación bayesiana combina los estados de la información teórica con la información obtenida de medidas (datos) a través de funciones de probabilidad gaussiana. Diferentes configuraciones de los parámetros generan diferentes funciones de probabilidad gaussiana que, agrupadas, definen la densidad de probabilidad asociada a la solución del problema de inversión.

En la distribución de densidad de probabilidad se puede definir una frontera razonable límite del conjunto de soluciones que mejor ajusta los datos, permitiendo el reconocimiento de patrones.

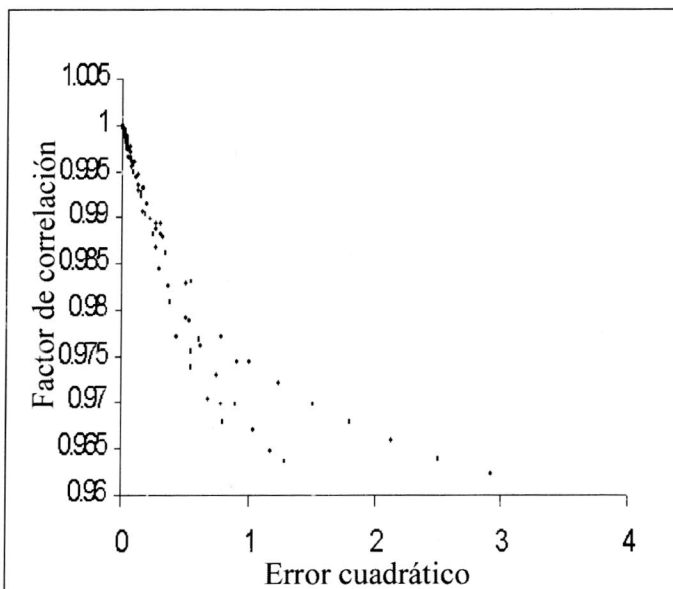


Figura 4. Esta curva representa la densidad de probabilidad obtenida de un proceso de entrenamiento.

APLICACIÓN

El ALM fue desarrollado en plataforma Visual Basic y C++, en el cual se puede ver el proceso de entrenamiento, así como la respectiva convergencia a través de cada una de las iteraciones.

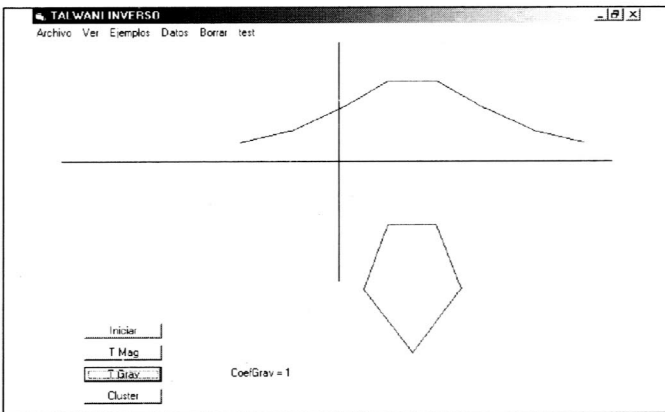
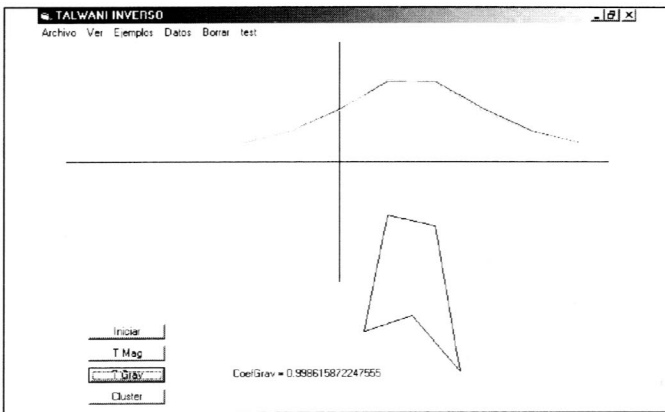
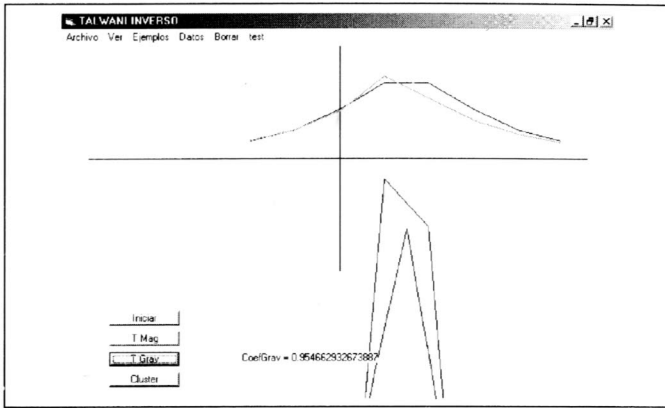


Figura 5. En esta figura se observa la optimización de la sección transversal.

CONCLUSIONES

El Algoritmo de Levenberg-Marquardt permite determinar en forma eficiente los parámetros considerados en el problema de inversión.

Esta metodología no solamente se puede enfocar a problemas de estimación y optimización, sino que también podría aplicarse en el análisis de las deformaciones corticales a partir de las variaciones del campo gravitacional.

BIBLIOGRAFÍA

- MEJU, M. A. 1994. *Geophysical Data Analysis: Understanding Inverse Problem Theory and Practice*. Society of Exploration Geophysicists.
- MIRKO VAN DER BAAN, and Ch. JUTTEN. 2000. Neural networks in geophysical applications. *Geophysics* 65 (4): 1032-1047.
- SERDAR, I. 2000. IEEE NNC Student Summer Research Support Program Report. www.ieee-nns.org/edu/research/reports-2000/.
- TARANTOLA A. 1987. *Inverse Problem Theory*. Elsevier Science B.V.
- TALWANI, L. 1959. Rapid gravity computation for two-dimensional bodies with application to the Mendocino submarine fracture zone. *Journal Geophysical Research* 64: 49-59.