

Nuevas Tendencias en Redes Neuronales Artificiales: Extreme Learning Machine

Pedro José García Laencina, Rafael Verdú Monedero, Jorge Larrey Ruiz,
Juan Morales Sánchez, José Luis Sancho Gómez.

Departamento de Tecnologías de la Información y las Comunicaciones
Universidad Politécnica de Cartagena. Plaza del Hospital 1, 30202 Cartagena
Teléfono: (+34) 968326542. E-mail: pedroj.garcia@upct.es

Resumen. Las redes neuronales artificiales han sido ampliamente utilizadas para resolver problemas de diagnosis médica, reconocimiento de voz, predicción de índices bursátiles, etc. A pesar de ello, presentan como claros inconvenientes el elevado tiempo de cálculo necesario y la convergencia a mínimos locales. Este artículo analiza un novedoso, rápido y eficiente método para el entrenamiento de redes tipo “feed-forward” conocido como Extreme Learning Machine.

1. Introducción

Las Redes Neuronales Artificiales (RNA) han sido empleadas con éxito en múltiples aplicaciones durante las últimas décadas [1]. No obstante, surgen diversos inconvenientes durante el proceso de optimización de los hiper-parámetros (i.e., pesos y número de neuronas) como son el elevado tiempo de cálculo necesario y la convergencia a mínimos locales, debido a la no-unicidad de solución para un determinado problema al usar métodos de entrenamiento basados en gradiente [1]. Muchos trabajos han sido desarrollados para obtener rápidos y precisos algoritmos de entrenamiento supervisado para esquemas del tipo “feed-forward”, siendo uno de los más eficientes el método conocido como *Extreme Learning Machine* (ELM) [2], [3]. Este trabajo analiza este nuevo algoritmo, junto con un método para la selección óptima del número de neuronas. Finalmente, se muestran los resultados obtenidos en dos problemas de regresión.

2. Redes Neuronales Artificiales

Las RNA del tipo “feed-forward”, o Perceptrones Multicapa (Multi-Layer Perceptron, MLP), constan de múltiples capas de unidades computacionales (neuronas) interconectadas, pudiendo aproximar cualquier función continua seleccionando los hiper-parámetros adecuadamente [1]. Un MLP estándar está compuesto de una capa oculta de H neuronas. Los pesos de la capa de entrada conectan las n variables de entrada con las H neuronas, mientras que una segunda capa de pesos conecta las salidas de las H neuronas con las m unidades de salida del MLP. Considerar un vector de entrada $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$, con “target” $\mathbf{t} = [t_1, t_2, \dots, t_m]^T \in \mathbb{R}^m$. Así, la salida del MLP viene dada por

$$\mathbf{y} = \sum_{j=1}^H \beta_j f(\mathbf{w}_j \cdot \mathbf{x} + b_j), \quad (1)$$

donde $\mathbf{y} = \{y_k\}_{k=1}^m$, $f(\cdot)$ son las funciones de activación de las neuronas ocultas, $\beta_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jm}]^T$ son los pesos de salida asociados a la neurona oculta j -ésima, mientras que $\mathbf{w} = [w_{j1}, w_{j2}, \dots, w_{jn}]^T$ y b_j representa el sesgo de entrada asociado a dicha neurona. Las unidades de salida tienen funciones de activación lineales, mientras que para $f(\cdot)$ se suelen emplear funciones de tipo sigmoide. Como se puede comprobar, los hiper-parámetros a optimizar durante el entrenamiento son los pesos y el número de neuronas. Generalmente, se establece un número de neuronas H , los pesos son inicializados aleatoriamente, y a partir de un conjunto de entrenamiento asociado al problema a resolver, la red es entrenada mediante el algoritmo “back-propagation”, junto con métodos de optimización basados en gradiente. El número óptimo de neuronas es seleccionado por validación cruzada. Este tipo de procedimiento conlleva un gran número de iteraciones de entrenamiento, y una búsqueda exhaustiva del número de neuronas, lo que es puede suponer un gran inconveniente en aplicaciones de tiempo real. Para más detalles de implementación/diseño, ver [1].

3. Extreme Learning Machine

El algoritmo *Extreme Learning Machine* (ELM) está fundamentado en que un MLP compuesto por H neuronas, cuyos pesos de entrada están inicializados aleatoriamente, pueden “aprender” N distintos casos de entrenamiento produciendo un error cero, siendo $N \geq H$, y aproximar cualquier tipo de función continua [2]. Tras inicializar de manera aleatoria los pesos de entrada, un MLP puede ser considerado como un sistema lineal y los pesos de salida pueden obtenerse de manera analítica mediante un simple cálculo de la pseudo-inversa de la matriz de las salidas de las H neuronas ocultas para un determinado conjunto de entrenamiento. Así, dado un conjunto de N vectores de entrada, un MLP con H neuronas

ocultas puede aproximar estos N casos con error nulo ($\sum_{i=1}^N \|y_i - t_i\| = 0$), i.e., existen β_j , w_j y b_j tal que,

$$\sum_{j=1}^N \beta_j f(w_j \cdot x_i + b_j) = t_i, \quad i = 1, \dots, N. \quad (2)$$

Las anteriores N ecuaciones se pueden expresar:

$$\mathbf{H}\beta = \mathbf{T}, \quad (3)$$

donde $\mathbf{H} \in \mathbb{R}^{N \times H}$ es la matriz de salidas de la capa oculta de neuronas del MLP, $\beta \in \mathbb{R}^{H \times m}$ es la matriz de pesos de salida, y $\mathbf{T} \in \mathbb{R}^{N \times m}$ es la matriz de “targets” de los N casos de entrenamiento. De esta forma, el entrenamiento del MLP viene dada la solución del problema de mínimos cuadrados establecido en (3), es decir, los pesos óptimos de la capa de salida son $\hat{\beta} = \mathbf{H}^\dagger \mathbf{T}$, donde \mathbf{H}^\dagger es la pseudo-inversa de Moore-Penrose [4]. ELM proporciona un entrenamiento rápido y eficiente para MLPs [2], aunque es necesario pre-establecer el número de neuronas ocultas. A continuación se describe un procedimiento para la selección automática del número de neuronas.

3.1. Selección Óptima de Neuronas

El método *Optimal Pruned ELM* (OP-ELM) [3] establece un número inicial muy elevado de neuronas ocultas ($H \gg N$), y mediante el algoritmo *Least Angle Regression* (LARS) [5] elimina/poda aquellas variables (neuronas) que no son útiles para resolver el problema de mínimos cuadrados. Para ello ordena el conjunto posible de neuronas, conforme a su importancia para resolver (3). La solución obtenida por LARS es única cuando el problema es lineal [5]. La poda de neuronas es realizada mediante validación cruzada del tipo Leave-One-Out (LOO), escogiendo aquella combinación de neuronas (que han sido previamente ordenadas mediante el algoritmo LARS) que proporciona menor error de validación [3]. Por tanto únicamente son seleccionadas las neuronas más importantes según LARS, y junto con el entrenamiento ELM, se obtiene una solución única para el diseño de un MLP.

4. Experimentos y Resultados

Inicialmente, vamos a emplear un problema sencillo de regresión unidimensional. Se han generado artificialmente 1000 casos de entrenamiento, según la suma de dos señales sinusoidales y ruido aleatorio gaussiano. La Figura 1 muestra con puntos azules el conjunto de datos de entrenamiento. A continuación se ha entrenado un MLP mediante OP-ELM, empleando un tiempo de cálculo de 5,47 s (segundos). La Figura 1 muestra el modelo obtenido por OP-ELM en rojo. Podemos comprobar que el resultado obtenido es muy preciso.

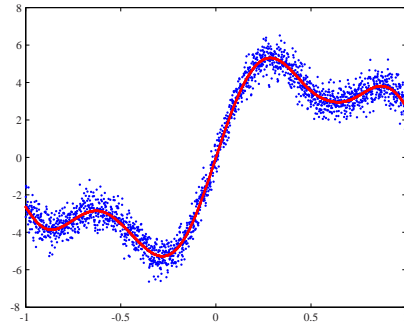


Fig. 1. Regresión unidimensional mediante OP-ELM.

A continuación, se ha empleado un problema real de regresión, *Abalone*¹. Consta de 4177 casos (2784 para entrenamiento y 1393 para test) y 8 características de entrada. Los hiper-parámetros del MLP han sido seleccionados mediante validación cruzada (10 “folds”), y para el entrenamiento se ha usado el Neural Network Toolbox de MATLAB. Para minimizar el efecto de los mínimos locales, se han realizado 10 inicializaciones aleatorias de pesos, y se ha seleccionado el número de neuronas mediante búsqueda entre 1 y 20 neuronas. El entrenamiento estándar del MLP obtiene un error de test igual a 4,34 tras 2640 s, mientras que OP-ELM obtiene un error de test igual a 4,50 tras emplear 25 s. Podemos comprobar como OP-ELM obtiene resultados prácticamente similares al MLP estándar, siendo el tiempo de cálculo 100 veces más rápido.

5. Conclusiones

Este trabajo analiza un rápido y eficiente método para el entrenamiento óptimo de MLPs llamado *Extreme Learning Machine* (ELM), además de describir un procedimiento para selección automática del número de neuronas. Este método obtiene prestaciones similares al entrenamiento estándar de un MLP en términos de error, y teniendo como mayor ventaja la gran velocidad de convergencia que presenta.

Agradecimientos

Este trabajo está parcialmente financiado por el MEC a través del proyecto TEC2006-13338/TCM.

Referencias

- [1] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, USA: Oxford University Press (1995).
- [2] G.-B. Huang, Q.-Y. Zhu, C.-k. Siew, “Extreme Learning Machine: Theory and Applications”, *Neurocomputing*, 70(1-3), pp. 489-501 (2006).
- [3] Y. Miche, P. Bias, C. Jutten, O. Simula, A. Lendasse, “A methodology for Building Regression Models using Extreme Learning Machine”, *16th European Symposium in Artificial Neural Networks*, pp. 247-252, Bruges, Belgium, 2008.
- [4] A. Björk. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.
- [5] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, “Least Angle Regression”, *Annal of Statistics*, vol. 32, pp. 407-499, (2004).

¹<http://www.liaad.up.pt/~ltorgo/Regression/DataSets.html>