

Neuronas artificiales con wavelets paramétricos

Oscar Herrera-Alcántara¹, Miguel González-Mendoza²

¹ Universidad Autónoma Metropolitana,
México

² Instituto Tecnológico y de Estudios Superiores de Monterrey,
México

oha@azc.uam.mx, mgonza@itesm.mx

Resumen. Presentamos neuronas artificiales con wavelets paramétricos con las cuales es posible representar las compuertas lógicas AND, OR, NOT, NAND, NOR, y XOR con una única neurona. Esta es una ventaja respecto a otro tipo de neuronas, incluido el modelo de McCulloch-Pits y del perceptrón, para las cuales se requieren conectar más de una de ellas para expresar la compuerta XOR que representa un problema no linealmente separable. Los resultados experimentales muestran propiedades y ventajas de las neuronas con wavelets paramétricos en problemas de clasificación y aproximación de funciones. En particular, comparamos el desempeño de redes de neuronas con wavelets Haar y wavelets paramétricos. Las redes neuronales con wavelets paramétricos lograron clasificaciones del 100 % en un conjunto de datos tomados de la base de datos del UCI.

Palabras clave: Wavelets, neurona artificial, compuertas lógicas, aproximación de funciones.

Artificial Neurons with Parameterized Wavelets

Abstract. We present artificial neurons with parameterized wavelets from which is possible to represent the AND, OR, NOT, NAND, NOR, and XOR logic gates with a single neuron. This is an advantage with respect to other kind of neurons including the McCulloch-Pits and perceptron models, for which is required more than a single one to express the XOR gate that represents a not linearly separable problem. Experimental results show properties and advantages of parameterized wavelet neurons in classification and approximation problems. In particular, we compare the performance of neural networks with the Haar wavelet and parametric wavelets. The neural networks with parameterized wavelets achieved 100 % when classifying data taken from the UCI database.

Keywords: Wavelets, artificial neurons, logic gates, function approximation.

1. Introducción

En [15] fueron presentadas redes neuronales con wavelets como una alternativa a las redes de retropropagación para aproximar funciones no lineales. Sin embargo, a más de dos décadas, no ha proliferado su uso, al menos en comparación con otros tipos de redes neuronales que involucran funciones crecientes, continuas, y acotadas [2].

Aunque existe evidencia de la eficiencia de aproximación con redes de wavelets [1,11], en muchas aplicaciones se ha preferido el uso de funciones con propiedades como la derivabilidad, que facilitan la aplicación de métodos de optimización basados en gradiente, como es el caso del algoritmo de retropropagación para perceptrones multicapa (MLPBP) con funciones tangente hiperbólico o sigmoidal.

En el caso de wavelets, también existen funciones como el “sombrero mexicano” que son derivables, y que permiten usar métodos de gradiente en el entrenamiento de redes que conectan varios “wavelons” o unidades de procesamiento en redes wavelet.

En varias arquitecturas de redes neuronales, incluidas las MLPBP, se plantea el número de neuronas y el número de capas ocultas como los factores primordiales en la mejora de la precisión de la aproximación. Sin embargo, no se considera la opción de adaptar las funciones de activación (funciones base). Si acaso, se limitan a elegir de entre un número limitado de funciones predeterminadas.

Las funciones sigmoidal, tangente hiperbólico, lineal, o escalón usadas ampliamente en MLPBP, o las funciones wavelet en los wavelons, tienen una “forma fija”, que no se puede adaptar para ajustarse a los datos de entrada, así que la aproximación depende de otros parámetros.

Nuestra idea principal es promover el uso de funciones base adaptables, bajo la hipótesis de que el uso de funciones construidas *ad-hoc* permite minimizar el número de neuronas requeridas para aproximar la función que mejor se ajuste a los datos de entrada. La idea anterior está fundamentado en la teoría de aproximación con wavelets [3] y su relación con las parametrizaciones de filtros de reconstrucción perfecta [5,6,16,12,13].

Con fines explicativos, considérese que quisieramos rellenar un rectángulo con círculos, requeriríamos un número infinito de ellos aún cuando podamos colocar sus centros en diferentes posiciones y con diferentes radios. Por otro lado, si se tuviera la oportunidad de elegir cuadrados en lugar de círculos, es fácil pensar que con un menor número de ellos se pueda rellenar el rectángulo en cuestión.

Para mostrar lo anterior, presentamos la aproximación de funciones booleanas (compuertas lógicas) mediante neuronas con wavelets, mismas que se comparan con la arquitectura presentada para una MLP de la función XOR (ver [4]) que representa un problema no linealmente separable.

El artículo está organizado de la siguiente manera, en la Sección 2 presentamos la metodología seguida en el presente trabajo de investigación, en la Sección 3 presentamos la arquitectura de una neurona con wavelet paramétrico, en la Sección 4 presentamos varios experimentos y sus resultados, y finalmente, en la Sección 5 presentamos conclusiones y trabajos futuros.

2. Metodología

En el estudio de las neuronas con wavelets paramétricos seguimos los siguientes pasos:

1. Revisamos la arquitectura de una neurona artificial básica, y proponemos una que involucre funciones wavelet con parámetros.
2. Elegimos un conjunto de funciones a aproximar, que permitan medir la eficiencia de aproximación de las neuronas con wavelets.
3. Realizamos experimentos con un programa de computadora para medir la capacidad de aproximación de neuronas con wavelets.

3. Arquitectura de una neurona con wavelets paramétricos

El término “neurona” se atribuye a la inspiración en el modelo biológico. Se usa neurona artificial para hacer énfasis en que se trata de modelos matemáticos que asemejan el funcionamiento de la neurona biológica, al menos como fue planteado hace varias décadas en trabajos pioneros en este tema [9].

En ocasiones, se da poca importancia a la justificación del por qué las redes neuronales funcionan. En particular, para redes de perceptrones, en [2] se puede revisar uno de los trabajos que sustenta y da pauta a la elección de las funciones que pueden usarse en redes neuronales, y de hecho a la justificación de que una sola capa oculta es suficiente para aproximar cualquier función de energía finita.

En cuanto a los wavelets, la fórmula de reconstrucción de la transformada wavelet continua para una función $f(x)$ está dada por la ecuación (1) [3,10]:

$$f(x) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \langle f(x), \psi\left(\frac{x-b}{a}\right) \rangle \frac{1}{\sqrt{|a|}} \psi\left(\frac{x-b}{a}\right) \frac{1}{a^2} da db, \quad (1)$$

donde $\psi(x)$ es la función wavelet, con término de traslación $b \in \mathbb{R}$, y factor de escalamiento $a \neq 0$. C_ψ es una constante de admisibilidad, y $\langle f(x), \psi\left(\frac{x-b}{a}\right) \rangle$ representa la transformada wavelet (coeficientes de transformación) dada por la ecuación (2):

$$\langle f(x), \psi\left(\frac{x-b}{a}\right) \rangle = \int f(x) \frac{1}{\sqrt{|a|}} \bar{\psi}\left(\frac{x-b}{a}\right) dx, \quad (2)$$

Las ecuaciones (1) y (2) representan la piedra angular de la arquitectura que define una red neuronal con “átomos” de descomposición en el espacio wavelet.

Entonces, en forma análoga al modelo matemático del perceptrón, se propone una neurona wavelet como se ilustra en la Figura 1.

Su descripción matemática esta dada por la ecuación (3):

$$Y = w_s g(u), \quad (3)$$

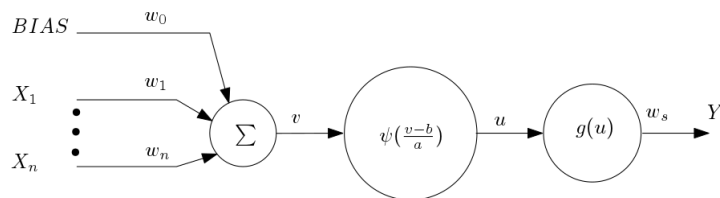


Fig. 1. Neurona con wavelet paramétrico.

donde w_s es el peso de la capa de salida. La función wavelet $\psi(\cdot)$ es trasladada por el término b y escalada por el factor $a \neq 0$, entonces se tiene la ecuación (4) para:

$$u = \psi\left(\frac{v - b}{a}\right), \quad (4)$$

con $g(u)$ como función de la capa de salida. El potencial de activación v está dado por la ecuación (5):

$$v = \sum_n w_i X_i, \quad (5)$$

donde X_i es el dato de un vector de dimensión n en la capa de entrada, X_0 es el *BIAS* unitario, y w_i son los pesos sinápticos.

Las funciones wavelet ψ son generadas mediante el algoritmo en cascada [8], y dependen de parámetros de filtros de reconstrucción perfecta que determinan su forma. Algunas parametrizaciones de filtros que se pueden usar para dar lugar a la aproximación de estas funciones wavelet se pueden revisar en [5]. En este trabajo, se implementaron las parametrizaciones con un parámetro α para filtros de longitud 4, y α, β para filtros de longitud 6.

Cabe mencionar que la forma del wavelet está determinada por los valores de los parámetros. En [5] pueden revisarse distintas combinaciones de parámetros asociados a diferentes wavelets. La longitud del filtro determina el soporte compacto del wavelet (el dominio del wavelet a partir del cual su valor, tanto a la izquierda o derecha del eje real, es cero). Así, por ejemplo, un valor de $\alpha = \frac{\pi}{4}$ de un filtro de longitud 4, genera un wavelet tipo Haar, con soporte en el intervalo $[0, \frac{3}{2}]$. En forma más general, un filtro de longitud L tiene un soporte compacto $L - 1$, y en el caso anterior, se tiene que la mitad de ese intervalo (de $\frac{3}{2}$ a 3) es cero.

4. Experimentos y resultados

Acorde con la metodología de la Sección 2, realizamos varios experimentos por computadora con los siguientes pasos:

1. Elegir un conjunto de compuertas lógicas y datos de clasificación para verificar el modelo propuesto de neurona con wavelet paramétrico

2. Aplicar el modelo de neurona propuesto y optimizar los parámetros libres w_i , a , b , w_s , y α o β según corresponda a la longitud del filtro. Para ello se aplicó un algoritmo genético con población P , cruzamiento anular, selección directa con pares $(0, P - 1), (1, P - 2), (2, P - 3), \dots$, probabilidad de cruzamiento P_c , y probabilidad de mutación P_m .
3. Medir el error de aproximación con el RMS, y el porcentaje de clasificación.

4.1. Experimento 1

En un primer experimento, se aprovechó que entre los pocos wavelets conocidos con expresión analítica, está el wavelet de Haar dado por la ecuación (6):

$$\psi(x) = \begin{cases} 1 & \text{Si } 0 \leq x < \frac{1}{2}, \\ -1 & \text{Si } \frac{1}{2} \leq x < 1, \\ 0 & \text{caso contrario,} \end{cases} \quad (6)$$

el cual, como ya se mencionó, representa un caso especial dentro de los wavelets paramétricos, que puede aproximarse con $\alpha = \frac{\pi}{4}$ y el algoritmo en cascada. El wavelet Haar de la ecuación (6) tiene un soporte compacto de $[0, 1)$.

Así que, para facilitar la evaluación de la función wavelet de Haar sin necesidad de ejecutar el algoritmo en cascada, se optó por fijar el parámetro $\alpha = \frac{\pi}{4}$ y usar la ecuación (6) con la debida adaptación al soporte compacto.

Las funciones de entrada seleccionados fueron las compuertas AND, OR, NAND, NOR y XOR, como se especifican en la Tabla 1, que representan problemas de clasificación binaria. También se probó con la función NOT que invierte las entradas -1 y 1 .

Tabla 1. Tablas de verdad para las compuertas AND, OR, NAND, NOR y XOR.

X1	X2	AND	OR	NAND	NOR	XOR
-1	-1	-1	-1	1	1	-1
-1	1	-1	1	1	-1	1
1	-1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1

La función de salida utilizada fue la función signo, dada en la ecuación (7):

$$g(u) = \begin{cases} 1 & \text{Si } 0 \leq u, \\ -1 & \text{Si } u < 0, \end{cases} \quad (7)$$

Una de las ventajas de usar wavelets con soporte compacto, es que permiten acotar los valores de los parámetros a y b , toda vez que fuera del soporte el valor del wavelet es cero. En los experimentos, el valor absoluto máximo de a y b fue 32 que es casi 10 veces el valor del soporte.

Tabla 2. Parámetros para las compuertas AND, OR, NAND, NOR y XOR.

Función	w_0	w_1	w_2	a	b	Generaciones
AND	-8.547	1.574	1.255	-8.047	-0.141	4
OR	-17.070	5.063	4.773	-24.0472	-0.141	91
NAND	8.548	10.187	7.773	28.848	-16.554	6
NOR	8.548	10.187	7.773	28.848	-16.554	6
XOR	-1.0629	4.063	4.773	-8.047	-0.141	10
NOT	8.548	10.187	4.773	28.848	-16.554	2

Por simplicidad, se fijó $w_s = 1,0$. Los demás valores de los parámetros libres de la neurona wavelet se optimizaron evolutivamente y los resultados se muestran en la Tabla 2.

En la Tabla 2 se puede apreciar, en la última columna, el número de generaciones que requirió el algoritmo genético para alcanzar el 100% de precisión, es decir, el error igual a cero. Se logró el empate idéntico entre los valores de entrada y la salida en parte debido a la apropiada elección de la función de salida descrita en la ecuación (7) que incluye los valores -1 y 1 .

Nótese que la función que más generaciones requirió fue la función lógica OR.

Para la función XOR, que se sabe representa un problema no linealmente separable, se puede tener una solución usando una sola capa oculta con dos neuronas N_1 y N_2 del modelo McCulloch-Pitts, con entradas 0 y 1 (ver [4,14]). En tal caso, los parámetros solución son: $w_{11} = w_{12} = 1$, $b_1 = -\frac{3}{2}$, $w_{21} = w_{22} = 1$, $b_2 = -\frac{1}{2}$, $w_{31} = -2$, $w_{32} = 1$, y $b_3 = -\frac{1}{2}$ como se ilustra en la Figura ???. La capa de salida contiene la neurona N_3 .

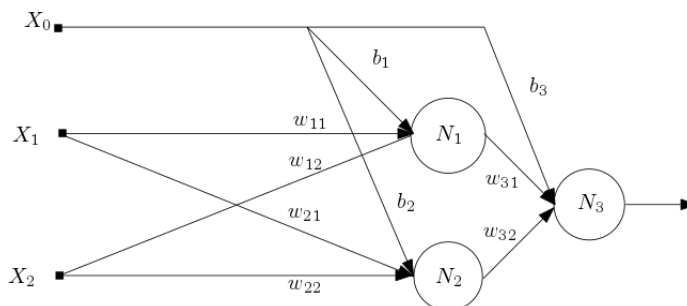


Fig. 2. Red de neuronas McCulloch-Pitts que resuelven el problema del XOR.

4.2. Experimento 2

En un segundo experimento se usaron 26 neuronas. Cada una de las 26 es una instancia de la neurona wavelet descrita en el Experimento 1. De esta forma,

se contruyó una red neuronal con arquitectura “tipo MLP” con una capa de entrada, una capa oculta con neuronas wavelets, y una capa de salida con función signo.

Los datos de prueba fueron los de clasificación de vinos, que considera 3 tipos de vinos [7]. Entonces, se dividieron en dos experimentos: Caso 1. Los de la primera clase con la tercera clase, y un Caso 2: Los de la primera clase con los de la segunda clase. En ambos casos se tiene una clasificación binaria, así que los valores de salida fueron etiquetados como -1 y 1 . Ahora se describen los experimentos con ambos casos.

Experimento del caso 1. Con $G = 314$ generaciones, $P = 40$, $P_{cruz} = 0,97$ y $P_{mut} = 0,01$, se logró el máximo éxito de clasificación posible, es decir el 100 % para los 107 datos de entrada, con un valor de parámetro óptimo $\alpha = 4,517208$. En tanto que, con un filtro Haar (equivalente al caso especial de $\alpha = \frac{\pi}{4}$) apenas se logró, en las mismas $G = 314$ generaciones, una clasificación de 92,52 %. El tiempo promedio de ejecución en estos experimentos del Caso 1, fue de 173,2 segundos.

En la Figura 3 se puede observar que, durante la ejecución del algoritmo genético, la red neuronal con wavelets paramétricos tiene una convergencia competitiva con la red neuronal con wavelet Haar, pero a partir de la generación 49 la neurona con filtro wavelet paramétrico converge mucho más rápido a la solución óptima ($RMS = 0$, clasificación del 100 %) en 313 generaciones, pero no así la red neuronal con wavelet Haar para la cual con 2000 generaciones logra un $RMS = 0,193$.

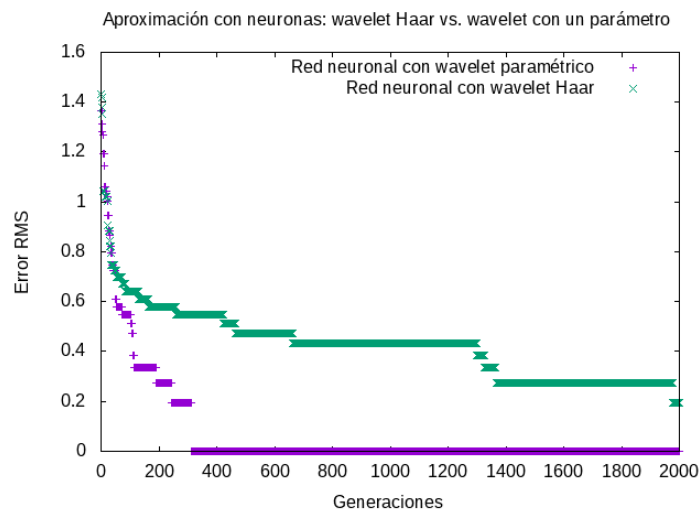


Fig. 3. Desempeño de dos redes neuronales en la clasificación de 107 muestras de dos tipos de vinos: i) red de neuronas wavelet Haar ii) red de neuronas wavelet con parámetro α .

Experimento del caso 2. Con $G = 500$ generaciones, $P = 40$, $P_{cruz} = 0,97$ y $P_{mut} = 0,02$, se logró una clasificación del 100% ($RMS = 0$) para 130 datos de entrada con $\alpha = 4,0535$ y $\beta = 1,855$. Esto representa un desempeño superior al obtenido con $\alpha = \frac{\pi}{4}$ (filtro Haar) que logró una clasificación de 93,84% y un $RMS = 0,4961$ en 2000 generaciones. En la Figura 4 se ilustra la ejecución del algoritmo genético en donde se aprecia que una red neuronal con wavelet Haar no logra reducir el error RMS aún durante un gran número de $G = 2000$ generaciones. En contraste, una red neuronal con wavelet de dos parámetros logra en $G = 500$ generaciones la aproximación perfecta para 130 datos de entrada.

El tiempo promedio de ejecución en estos experimentos del Caso 2, fue de 265,2 segundos.

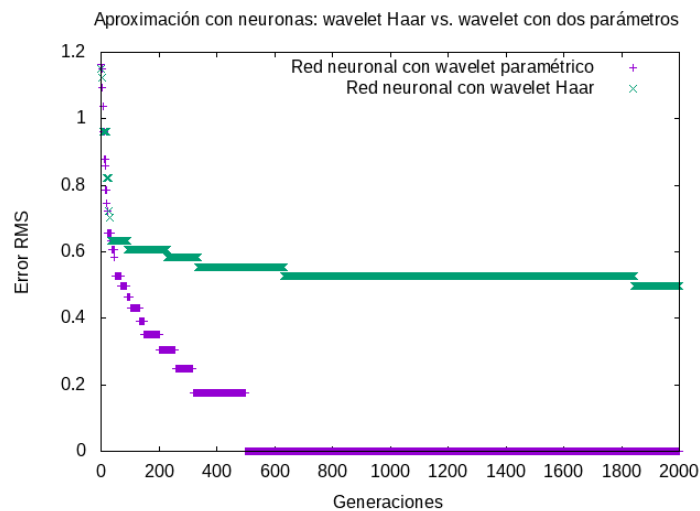


Fig. 4. Desempeño de dos redes neuronales en la clasificación de 130 muestras de dos tipos de vinos: i) red de neuronas wavelet Haar ii) red de neuronas wavelet con dos parámetros α y β .

5. Conclusiones y trabajo futuro

Se presentó un modelo de neurona artificial con una capa de entrada, una capa de procesamiento con función wavelet paramétrica de soporte compacto y una capa de salida. Para probar su validez, se usó la compuerta NOT y compuertas lógicas AND, OR, NAND, y NOR con dos entradas, con las cuales se logró un error de aproximación de cero, con una sola neurona de este tipo. También se aplicó a la función lógica XOR, que suele requerir más neuronas de otros modelos incluido el de McCulloch-Pitts y el perceptrón. En el caso de la neurona

con wavelet, fue posible resolver el problema del XOR con una única neurona y wavelet Haar visto como un caso especial de wavelet paramétrico.

Aunque no se conocen las funciones analíticas de los wavelets paramétricos, es posible aplicar métodos evolutivos para optimizar los parámetros y ajustar la red neuronal a los datos de entrada. Los experimentos complementan los conceptos teóricos que afirman que la transformada wavelet (y por ende las redes neuronales basadas en wavelets) requieren un menor número de coeficientes (neuronas) para concentrar la energía de una función y mejorar la aproximación.

Los resultados obtenidos son alentadores y sugieren proponer más arquitecturas basadas en funciones paramétricas, que provean alternativas a otros métodos que usan funciones “de forma fija” primordialmente derivables, cuyo comportamiento no permite capturar cambios abruptos en las funciones a aproximar. Por otro lado, las funciones wavelet han mostrado una superioridad ya que permiten detectar fenómenos transitorios, y en nuestro caso al usar neuronas con wavelets Haar en problemas de clasificación se lograron desempeños cercanos al 100 %, y al optimizar parámetros con un algoritmo genético se lograron clasificaciones del 100 % en un menor número de generaciones.

Como trabajo futuro se plantea proponer técnicas heurísticas para mejorar los tiempos de ejecución, y aplicarlas a problemas de clasificación y aproximación de funciones más complejos.

Referencias

1. Candès, E.J., Donoho, D.L.: Ridgelets: a key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 357(1760), 2495–2509 (1999)
2. Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function. *Math. Control Signal Systems* 24 (1989), <https://doi.org/10.1007/BF02551274>
3. Daubechies, I.: *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1992)
4. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice Hall (1998)
5. Herrera-Alcántara, O., González-Mendoza, M.: *Inverse formulas of parameterized orthogonal wavelets*. Computing (2018)
6. Lai, M.J., Roach, D.W.: *Parameterizations of Univariate Orthogonal Wavelets With Short Support*. Vanderbilt University Press (2002)
7. Lichman, M.: *UCI Machine Learning Repository* (2013)
8. Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic Press (1999)
9. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 51(115) (1943)
10. Navarro, J., Elizarraraz, D.: *Introducción a la Transformada de Wavelet Continua*. Reverte, Mexico (2010)
11. Pati, Y.C., Krishnaprasad, P.S.: Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. *IEEE Transactions on Neural Networks* 4(1), 73–85 (Jan 1993)
12. Roach, D.W.: Frequency selective parameterized wavelets of length ten. *Journal of Concrete and Applicable Mathematics* 8(1), 1675–179 (2010)
13. Roach, D.: *The Parameterization of the Length Eight Orthogonal Wavelets with No Parameter Constraints* (2008)

Oscar Herrera-Alcántara, Miguel González-Mendoza

14. Touretsky, D.S., Pomerleau, D.A.: What is hidden in the hidden layers? *Byte* 14, 227–233 (1989)
15. Zhang, Q., Benveniste, A.: Wavelet networks. *IEEE Transactions on Neural Networks* 3(6), 889–898 (Nov 1992)
16. Zou, H., Tewfik, A.: Parametrization of compactly supported orthonormal wavelets. *IEEE Trans. Signal Process.* 41(3), 1428–1431 (1993)