

---

---

Red Neuronal ADALINE, análisis como regresor y  
como clasificador.

Julio Cesar Haro Capetillo

Centro de Investigación en Computación  
*Instituto Politécnico Nacional.*

# Red Neuronal ADALINE, análisis como regresor y como clasificador

Julio Cesar Haro Capetillo

<sup>1</sup> Centro de Investigación en Computación IPN.

Received: ... / Accepted: ...

©The Authors 2024

**Resumen** / El presente trabajo es un análisis para la red neuronal ADALINE, se evaluó su capacidad como regresor y como clasificador, como regresor se evaluó su efectividad para aprender para un sistema en donde los objetivos son números binarios como decodificador, resultando como su trabajo como el más eficiente, se reportó que para la clasificación de 2 clases, la red neuronal clasifica con un  $\alpha$  de 0.1, pero para un  $\alpha$  de 0.0001, la tendencia de las fronteras de decisión parecieran aproximarse a la clasificación, pero es demasiado lento, lo suficientemente lento como para no poder clasificarlo en 800 épocas, las épocas fueron incrementadas para la clasificación para poder llegar a un aprendizaje, hay evidencia de que la función de costes de ésta red es una función cuadrática a la que le corresponde un punto mínimo, sin embargo, no es alcanzado para la clasificación de 4 clases.

**Abstract** / The present work is an analysis for the ADALINE neural network, its capacity as a regressor and as a classifier was evaluated, as a regressor its effectiveness in learning for a system where the objectives are binary numbers as a decoder was evaluated, resulting in its work as the most efficient, it was reported that for 2-class classification, the neural network classifies with an  $\alpha$  of 0.1, but for a  $\alpha$  of 0.0001, the trend of the decision boundaries seems approach the classification, but it is too slow, slow enough to not be able to classify it in 800 epochs, the epochs were increased for the classification to be able to achieve learning, there is evidence that the cost function of this network is a function quadratic to which a minimum point corresponds, however, it is not reached for the 4-class classification.

**Keywords** / ADALINE, Clasificación, Regresor, Entrenamiento, Función de costes, redes neuronales.

## 1. Introducción

Bernard Widrow y Ted Hoff son famosos en el mundo del procesamiento de señales por publicar su trabajo fundamental llamado *Adaptive Linear Neural Network* (Adaline) y el algoritmo de (*Least-Means-Squared*) (LMS, por sus siglas en inglés) para la reducción del error y optimización del aprendizaje, algoritmo que permitió el progreso del Machine Learning (Cheong-Tock *et al.* 2021). Adaline puede ilustrarse como un "sistema nervioso" constituido por varios nodos. Cada nodo requiere de varias entradas a su vez, que producen una simple salida. Puede comenzar como una simple neurona. El algoritmo de Adaline es utilizado para obtener un sistema artificial de red neuronal (Siswiraptini *et al.* 2020). La función que describe la red neuronal es la siguiente (Alcantara-Benjumea *et al.* 2020).

$$f(t) = \mathbf{W}^T \cdot x(t) \dots (1)$$

Una de las consideraciones más importantes a resaltar de la red ADALINE, es su modelo de aprendizaje, que a diferencia del perceptrón, se le

considera como un aprendizaje de tipo continuo, siendo la actualización de los pesos proporcional al error cometido por la neurona (Jara-Ruiz *et al.* 2018).

Considerando la práctica anterior, el modelo del perceptrón, de manera particular, se obtuvieron resultados satisfactorios para clasificar con el perceptrón para modelos de 2 y 4 clases.

En la literatura, se ha reportado con anterioridad, que entre el modelo del perceptrón y el modelo de ADALINE, ambos pueden llegar a clasificar satisfactoriamente para dos clases, sin embargo, se ha resaltado el hecho de que para clasificación, la red del perceptrón ha resultado ser superior en cuanto a su desempeño que la red de ADALINE (Jara-Ruiz *et al.* 2018).

Las funciones que describen el modelo, citados en Jara-Ruiz *et al.* (2018), son los siguientes:

### 1. Salida de la red ADALINE:

$$y = \mathbf{w}^T \mathbf{x} + b$$

### 2. Función de error:

$$E = \frac{1}{2} \sum_{i=1}^N (d_i - y_i)^2$$

### 3. Regla de actualización de pesos:

$$w_j^{(t+1)} = w_j^{(t)} + \alpha \cdot (d_i - y_i) \cdot x_j$$

### 4. Y para el sesgo o bias:

$$b^{(t+1)} = b^{(t)} + \alpha \cdot (d_i - y_i)$$

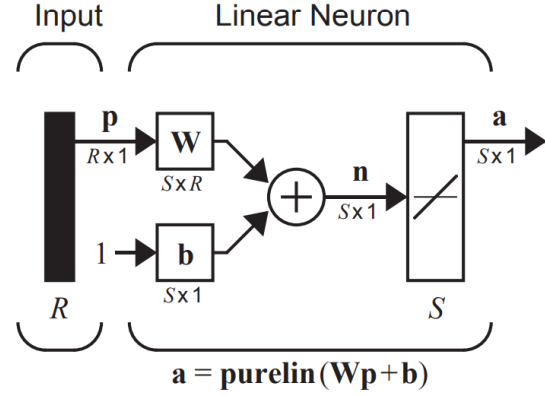
En el caso de la presente práctica, se utilizan los símbolos presentados por Hagan *et al.* (2014).

El modelo a analizar en este trabajo es el propuesto por Hagan (2014) (figura 3), en el cual se puede observar que la función de activación es la función de línea recta, cuya función en matlab es llamada `purelin`, la cuál regresa el valor del producto de los pesos con los vectores de entrada mas un sesgo, lo que, a diferencia del perceptrón y su clasificación binaria (0 ó 1), no sólo permite la clasificación sino también permite a la red servir como regresor, ya que se ha utilizado de esta manera en diversos estudios de manera satisfactoria, incluso con resultados superiores al método de regresión lineal múltiple (Ouorou-Ganni *et al.* 2021).

En su artículo original, Widrow y Hoff (1960) mencionan la utilidad de redes neuronales de 2 neuronas para la clasificación de sistemas y filtrados de ruidos de señales, actualmente sigue siendo considerada como buena opción para el aprendizaje en cuanto a clasificación, series de tiempo y como filtrado de señales (Suárez-Castro y Landino-Vega 2023).

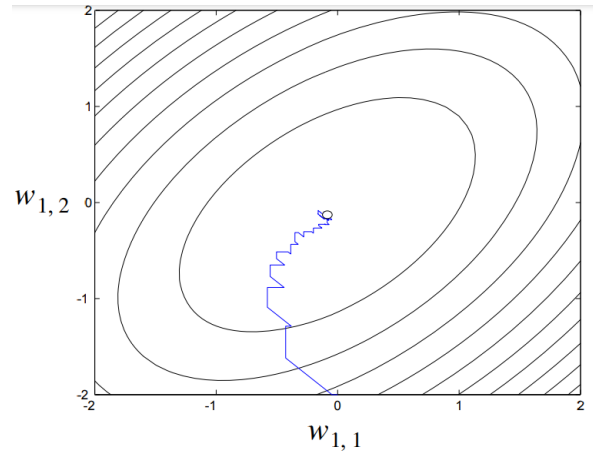
ADALINE ha sido utilizada con buenos resultados en la evaluación de la calidad de Energía Eléctrica Trifásico (Álcántara-Benjumea *et al.* 2022), permitiendo la optimización de un circuito integrador, en ya que su modo regresor generó resultados aproximados a sus valores teóricos de los armónicos de las señales de las tensiones e intensidades del circuito. También podemos resaltar su utilidad en el filtrado para la cancelación de ruido, mostrando resultados importantes cuando se presenta un  $\alpha = 0.1$  y un  $\alpha = 0.001$ , en donde el mejor resultado fue con el  $\alpha$  más pequeño (Hernández-Santiago *et al.* 2015).

La red ADALINE es muy similar a la red del perceptrón, la única diferencia es que ésta tiene una función de transferencia lineal.



**Fig. 1.** Modelo descrito en el libro de Hagan *et al.* (2014)

A continuación, Hagan *et al.* (2014) nos dan una representación gráfica del comportamiento de una red cuando la función de costes es cuadrática, y es posible aproximarse mediante un valor de  $\alpha = 0.1$  a lo que viene siendo el valor del punto mínimo de la función cuadrática a la que pertenece, según Nielsen (2015), ésto es válido cuando los valores se ajustan a funciones sin mínimos locales y sólo un mínimo global. Ésto nos da pauta para poder analizar el comportamiento de ADALINE, cuya función de costes es cuadrática.



**Fig. 2.** Acercamiento de los pesos al mínimo local con un  $\alpha = 0.1$

## 2. Marco teórico

### Modelo de la red ADALINE según Hagan

El perceptrón es un modelo básico de red neuronal que clasifica patrones linealmente separables. Su salida se calcula utilizando la siguiente ecuación:

$$a = \text{purelin}(\mathbf{W}\mathbf{x}^T + b), \quad (1)$$

donde:

- $\mathbf{W}$  es el vector de pesos de la red de dimensión  $1 \times n$  (con  $n$  características de entrada),
- $\mathbf{x}$  es el vector de entrada de dimensión  $1 \times n$ ,
- $b$  es el sesgo,
- $\text{purelin}(\mathbf{W} \cdot \mathbf{x} + b)$  es la función de activación definida como:

$$\text{purelin}(\mathbf{W} \cdot \mathbf{x} + b) = \begin{cases} a \geq 0, & \text{si } \mathbf{W} \cdot \mathbf{x} + b \geq 0, \\ a < 0, & \text{si } \mathbf{W} \cdot \mathbf{x} + b < 0. \end{cases} \quad t_i = \begin{cases} \{[0, 0], [0, 0]\}, & \text{Clase 1,} \\ \{[1, 0], [1, 0]\}, & \text{Clase 2,} \\ \{[0, 1], [0, 1]\}, & \text{Clase 3,} \\ \{[1, 1], [1, 1]\}, & \text{Clase 4.} \end{cases}$$

El modelo se entrena utilizando un conjunto de datos  $\{(\mathbf{x}_i, t_i)\}_{i=1}^m$ , donde  $\mathbf{x}_i$  son los vectores de entrada y  $t_i$  los objetivos deseados. El proceso de ajuste de los pesos se basa en la minimización del error:

$$E = \frac{1}{2} \sum_{i=1}^N (d_i - y_i)^2 \quad (2)$$

donde  $e_i$  es el error correspondiente a la muestra  $i$ .

### Algoritmo Least-Mean-Square algorithm (LMS)

El algoritmo LMS permite obtener un valor mínimo óptimo del comportamiento de la función de costes (2), o función del error, el cuál, considerando las derivadas parciales de los términos con respecto a  $d_i$ ,  $y_i$  se obtienen las siguientes funciones, donde se multiplica la derivada con un valor arbitrario  $\alpha$ , el cuál permite acelerar el gradiente, el cuál se considera es necesario tener la precaución para que el resultado de los parámetros se vayan al sobreentrenamiento (Nielsen 2015).

Los pesos y el sesgo se actualizan usando las siguientes reglas:

$$w_j^{(t+1)} = w_j^{(t)} + \alpha \cdot (d_i - y_i) \cdot x_j \quad (3)$$

$$b^{(t+1)} = b^{(t)} + \alpha \cdot (d_i - y_i) \quad (4)$$

### La red neuronal como clasificador

Clasificación para 2 clases.

Para el caso de dos clases, se utiliza una única neurona. Los objetivos están codificados como  $t_i \in \{0, 1\}$ . El objetivo del perceptrón es encontrar un hiperplano que divida el espacio de entrada en dos regiones:

$$\mathbf{W}\mathbf{x}^T + b = 0. \quad (5)$$

Durante el entrenamiento:

1. Se calcula la salida  $a$  utilizando la función de activación.
2. Se evalúa el error  $e_i = (t - a)^2$ .
3. Si  $e \neq 0$ , se ajustan los pesos y el sesgo según las ecuaciones (3) y (4).

Clasificación para 4 clases Para cuatro clases, se requiere una red con múltiples neuronas (una para cada clase). Los objetivos se representan utilizando

un codificado de una sola salida activa (*one-hot encoding*), es decir:

### La red neuronal en modo regresor

Para el modo regresor, simplemente se generaron números binarios correspondientes a un máximo de bits proporcionado por el usuario del programa, en donde se evaluó para 3 bits. Durante el entrenamiento:

1. Se calcula la salida  $a$  utilizando la función de activación.
2. Se evalúa el error  $e_i = (t - a)^2$ .
3. Si  $e \neq 0$ , se ajustan los pesos y el sesgo según las ecuaciones (3) y (4).

### 3. Diagrama de Flujo

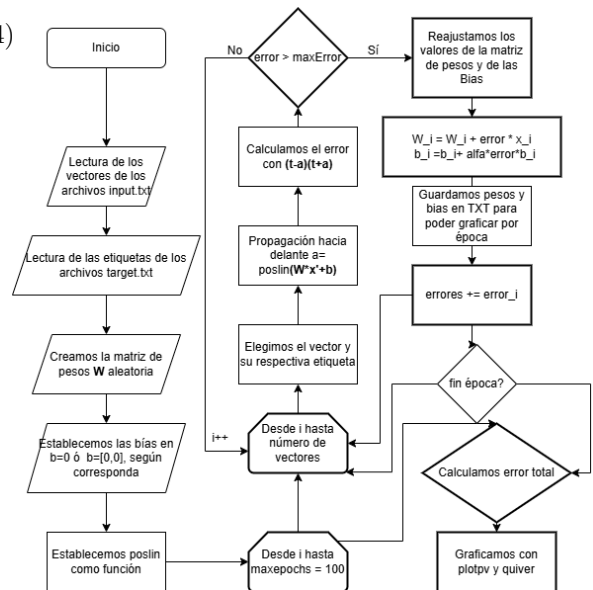


Fig. 3. Diagrama de Flujo.

## 4. Pseudocódigo

---

**Algorithm 1:** Entrenamiento de la red ADALINE
 

---

**Data:** Entradas  $X$ , objetivos  $D$ , máximo de épocas  $max\_epochs$ , error máximo  $max\_error$ , bias  $b$

**Result:** Pesos entrenados  $W$ , sesgo entrenado  $b$

```

1 Inicializar pesos  $W$  y sesgo  $b$ ;
2 for cada época hasta  $max\_epochs$  do
3   Inicializar contador de errores
    $errors \leftarrow 0$ ;
4   for cada muestra  $x$  y objetivo  $t$  en los
   datos do
5     Calcular salida  $a \leftarrow poslin(W \cdot x^T)$ ;
6     Calcular error  $error \leftarrow (t - a)^2$ ;
7     if  $error \neq 0$  then
8       Actualizar pesos
        $W \leftarrow W + \alpha \cdot (t - a) \cdot x$ ;
9       Actualizar sesgo
        $b \leftarrow b + \alpha \cdot (t - a)$ ;
10      Incrementar  $errors$ ;
11   if  $errores < maxError$  then
12     break;
  
```

---

## 5. Experimentos (Resultados)

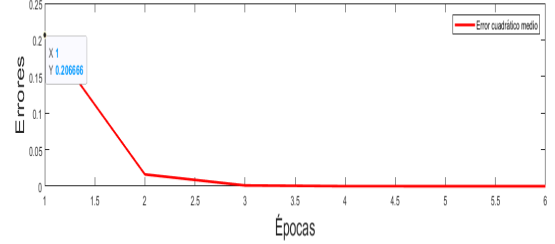
Se reporta que se creó un sistema interactivo con el usuario a través de consola, en donde el usuario puede elegir entre modo regresor y modo clasificador, para clasificador de 4 clases se reporta que se evaluaron dos valores de  $\alpha$  distintos, uno de  $\alpha = 0.1$  y el otro de  $\alpha = 0.00001$ , para los cuales no pudo obtenerse un resultado satisfactorio para las fronteras de decisión. Sin embargo, con un  $\alpha = 0.1$  se pudo observar que la red de ADALINE puede clasificar satisfactoriamente para 2 clases, así como se reporta que con un  $\alpha = 0.1$  fue suficiente para converger en modo regresor, cuyo error se va a cero en unas cuantas épocas, a diferencia de las épocas, que, variantemente y dependientemente de los valores aleatorios originales de los pesos, se pudo observar que fueron incluso necesarias más de las 100 épocas para poder generar una clasificación para las 2 clases.

También fue posible observar que para la clasificación de 4 clases, al disminuir el  $\alpha$ , también se aproximan las fronteras de decisión a la clasificación más próxima. Sin embargo, no es suficiente para que la red pueda clasificar las clases correctamente.

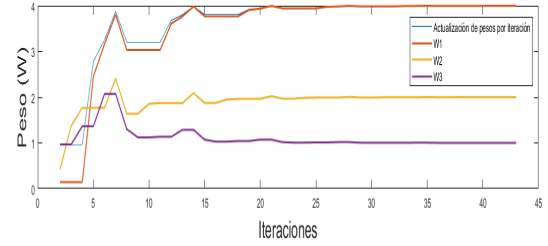
En la figura 3 se presentan los resultados del modo regresor con un  $\alpha = 0.1$ , en donde se ve que la reducción del error se aproxima al cero rápidamente después de la época 2, obteniendo un valor cercano al mínimo, que vendría siendo cercano al mínimo

local para la función de costes  $E$  presentada en el marcoteórico como (2). Se puede observar como a partir de cierto número de iteraciones, aproximadamente la iteración 22, los pesos se ajustan y casi no varían.

Gráfica de errores para el entrenamiento de ADALINE en modo regresor

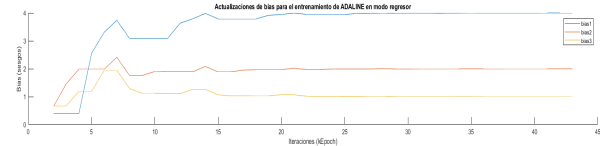


Actualizaciones de pesos para el entrenamiento de ADALINE en modo regresor



**Fig. 4.** Gráfica 1: Reducción del error por épocas, gráfica 2: aprendizaje de los pesos por iteraciones

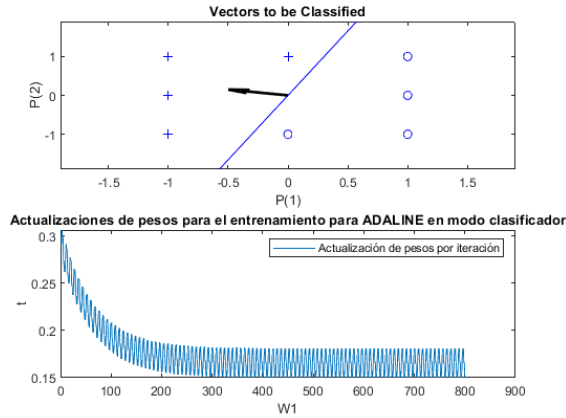
En la figura 4, se puede observar que los sesgos tienen un comportamiento muy similar al comportamiento de los pesos. En los entregables se puede observar que no son los mismos datos, por lo que se puede entender que varían de manera similar.



**Fig. 5.** Comportamiento de los sesgos por iteración

La figura 5 es una representación de los datos de la primera clasificación presentada en la práctica anterior para la clasificación del perceptrón. Podemos observar que este mismo dataset, ahora clasificado con ADALINE, tuvo resultados positivos, aunque se reportan 800 iteraciones para su correcta clasificación, se puede observar ambos pesos subiendo y bajando, éstos pesos están representados por los picos, sin embargo, es posible enfatizar que los pesos tienen un comportamiento similar, ya que van disminuyendo del 0.5 a valores incluso

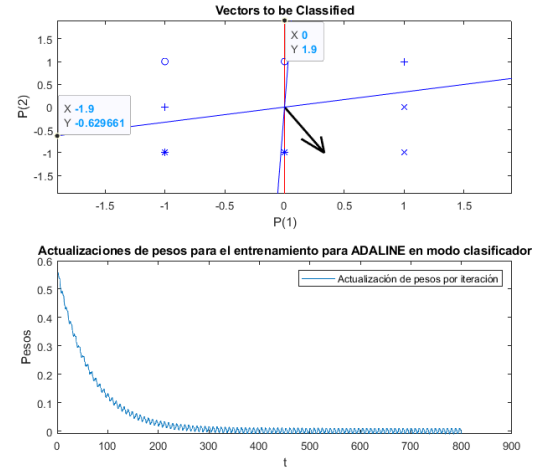
negativos, y cuyo comportamiento en general para ambos es una función logarítmica negativa.



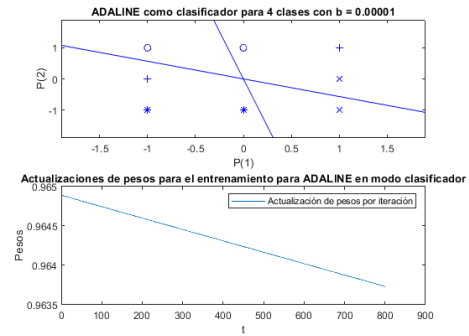
**Fig. 6.** Gráfica 1: Se puede observar como la frontera de decisión y los pesos, clasifican adecuadamente para ambas clases. Gráfica 2: Se puede observar un comportamiento general del aprendizaje similar a una función logarítmica negativa.

En las figuras 7 y 8 podemos observar el comportamiento para la clasificación de 4 clases, para la figura 7 lo que variamos es el valor de  $\alpha = 0.1$ , y para la figura 8 el valor de  $\alpha$  es de 0.0001, cabe resaltar que no logra la clasificación, sin embargo se puede apreciar que las fronteras se aproximan.

Se puede observar que en ambos casos existe un aprendizaje en donde los valores de los pesos de a cuerdo a las épocas es similar a una función logarítmica negativa en el caso del  $\alpha = 0.1$ , en donde podemos ver que sí se van ajustando a un valor en particular, y para el caso de  $\alpha = 0.0001$ , se observa un comportamiento más similar a una recta con pendiente negativa.



**Fig. 7.** Gráfica 1: Se observan las fronteras de decisión en azul, éstas no clasifican, Gráfica 2: Se observa el comportamiento de los pesos por iteración.



**Fig. 8.** Gráfica 1: fronteras de decisión en azul, Gráfica 2: Se observa un comportamiento casi recto.

## 6. Discusión

Los resultados obtenidos por la práctica concuerdan con lo proporcionado por (Jara-Ruiz *et al.* 2018; Suárez-Castro *et al.* 2023) es decir, se puede apreciar que la red neuronal ADALINE funciona mejor como regresor que como clasificador. Cabe mencionar que los resultados reportados por Jara-Ruiz *et al.* (2018) fueron resultados de clasificación para 2 clases y no para 4.

Se puede observar un comportamiento logístico negativo en el aprendizaje para la clasificación de 2 clases, incluso para el de una clase, eso concuerda como consecuencia de que la función de costes es una función cuadrática como lo expresa Nielsen (2015).

La diferencia entre las salidas entre una red neuronal de perceptron y la salida de ADALINE es continua en cada iteración para ADALINE, tal

como fue reportado en la literatura (Jara-Ruiz *et al.* 2018).

## 7. Conclusiones

- (i) La principal diferencia de la red de ADALINE comparada con la red del perceptrón de la práctica pasada es que la red de ADALINE es útil para procesos de regresión, sin embargo, no es tan eficiente al momento de clasificar, ya que la clasificación se adapta más a salidas de funciones de umbral como la hardlim ya que el resultado es discreto, en el caso de las salidas de poslin o de la red ADALINE, el resultado siempre va a ser continuo, la única manera en que una salida sea 0 es si el vector es 0 o los pesos valen 0. Cosa que difícilmente va a ocurrir en la realidad.
- (ii) El comportamiento que tiene el aprendizaje en todos los casos, excepto para el modo clasificador con  $\alpha = 0.0001$ , puede hacernos notar que es similar a una función logarítmica negativa, lo que concuerda con el comportamiento de una función de costes cuyo comportamiento es de una función cuadrática, ya que una es la función inversa de la otra, visto como un límite la podemos entender como una función que converge en un punto, al ser un resultado numérico, es dependiente de un error máximo, no podemos asegurar que el resultado final es el valor del punto mínimo de la función de costes pero podemos estar seguros que la tendencia del aprendizaje conduce a un punto mínimo.
- (iii) Para la clasificación de 4 clases, no es posi-

ble asegurar que va a converger, sin embargo, el comportamiento del aprendizaje puede ser señal de que pudieran haber sido necesarias más épocas para su convergencia para un  $\alpha$  tan pequeño. Sin embargo, no podemos decir que para estos casos la red ADALINE no es segura que va a poder clasificar para 4 clases.

( ) (Santiago et al., 2015) (Benjumea et al., 2022) (Took et al., 2021) (Guera et al., 2021) (Jara-Ruiz et al., 2018) (Nielsen, 2015) (Siswipraptini et al., 2020) (Suárez-Castro & Ladino-Vega, 2023) (Widrow & Hoff, 1960)

## References

- Benjumea F.J.A., et al., 2022, Ingeniare. Revista chilena de ingeniería, 659
- Guera O.G.M., et al., 2021, Revista Forestal Mesoamericana Kurú, 29
- Jara-Ruiz R., et al., 2018, Revista de Innovación Sistemática
- Nielsen M., 2015, Determination Press. <http://neuralnetworksanddeeplearning.com/>
- Santiago K.A.H., Ríos A.A., Godoy R.S., 2015, Boletín No 50
- Siswipraptini P.C., et al., 2020, Telkomnika (Telecommunication Comput. Electron. Control.)
- Suárez-Castro R., Ladino-Vega I., 2023, Tecnura, 72
- Took C.C., et al., 2021, IEEE Signal Processing Magazine, 51
- Widrow B., Hoff M.E., 1960, IRE WESCON Convention Record, 96