

# ALGORITMO DE ENTRENAMIENTO DE UNA NEURONA ARTIFICIAL PERCEPTRON PARA RECONOCIMIENTO DE CARACTERES, APLICANDO PRINCIPIOS HEURISTICOS

## ALGORITHM OF TRAINING ARTIFICIAL NEURON PERCEPTRON FOR RECOGNITION OF CHARACTERS, APPLYING HEURISTIC PRINCIPLES

Mario G. Borja-Borja<sup>a</sup>, Sally K. Torres-Alvarado<sup>a</sup>

### RESUMEN

En el presente trabajo se aborda el tema de reconocimiento de caracteres impresos aplicando una neurona artificial tipo perceptron para cada letra del abecedario. Generalmente, para la solución de problemas de reconocimiento de caracteres se parte del echo de la complejidad del problema y se crea estructuras muy complejas de redes neuronales artificiales con una gran cantidad de neuronas organizadas en varias capas, dando origen a complejos algoritmos de entrenamiento que debe operar durante largos periodos para encontrar la solución adecuada y por otra lado también el tiempo de reconocimiento aumenta debido a la gran cantidad de calculo que se debe realizar para obtener el resultado. En este trabajo se propone un algoritmo que reduce el tiempo de entrenamiento y a su vez permite resolver el problema de reconocimiento de un carácter en una neurona artificial tipo perceptron aplicando principios heurísticos, los cuales consideran características propias del problema de reconocimiento de letras, características propias de las letras y otras. Los resultados obtenidos nos permiten implementar soluciones de reconocimiento de caracteres más rápidas debido a la gran reducción de cálculo durante el entrenamiento y también durante el funcionamiento, ya que cada carácter solo requiere de una neurona artificial para ser reconocido. Por otro lado este algoritmo puede ser generalizado para plantear algoritmos de entrenamiento para redes neuronales artificiales multicapa.

**Palabras Clave:** neuronas, artificiales, reconocimiento de caracteres, perceptron.

### ABSTRACT

In the present work there is approached the topic of recognition of printed characters applying an artificial neuron type perceptron for every letter of the alphabet. Generally, for the solution of problems of recognition of characters it is considered to be a complex problem and very complex structures of neural artificial networks are created by a great quantity of neurons organized in several caps, giving origin to complex algorithms of training that must operate during long periods to find the suitable solution and for other one side also the time of recognition to increase due to the great quantity of calculation that it is necessary to realize to obtain the result. In this work one proposes an algorithm that reduces the time of training and in turn there allows to solve the problem of recognition of a character in an artificial neuron type perceptron applying heuristic principles, which consider own characteristics of the problem of recognition of letters, own characteristics of the letters and others. The obtained results allow us to implement more rapid solutions of recognition of characters due to the great reduction of calculation during the training and also during the functioning, since every alone character needs of an artificial neuron to be recognized. On the other hand this algorithm can be generalized to raise algorithms of training for neural artificial networks multicaps.

**Keywords:** Neurons, artificial, recognition of characters, perceptron.

### INTRODUCCIÓN

En la actualidad se requiere de nuevas aplicaciones que permitan obtener datos, realizar búsquedas, clasificar, etc documentos impresos para reducir el tiempo de registro manual de información referente, en los sistemas de información computarizados.

La presente investigación propone un algoritmo de entrenamiento de una neurona artificial tipo perceptron en base a principios heurísticos que consideran las características propias de la aplicación de reconocimiento de letras impresas que permite reducir el tiempo de entrenamiento y también permite implementar el reconocimiento de un carácter en una neurona artificial.

### METODOLOGÍA

Una neuronal artificial tipo perceptron se presenta en la Fig 1. En la entrada de la neurona tenemos el vector de entrada  $e$  que representan los puntos negros o blancos que forman una letra representada en forma de matriz. Cada entrada tiene un peso que representa la influencia de dicha entrada en el funcionamiento de la neurona artificial.

El funcionamiento de la neurona artificial se puede representar matemáticamente con las ecuaciones (1) y (2):

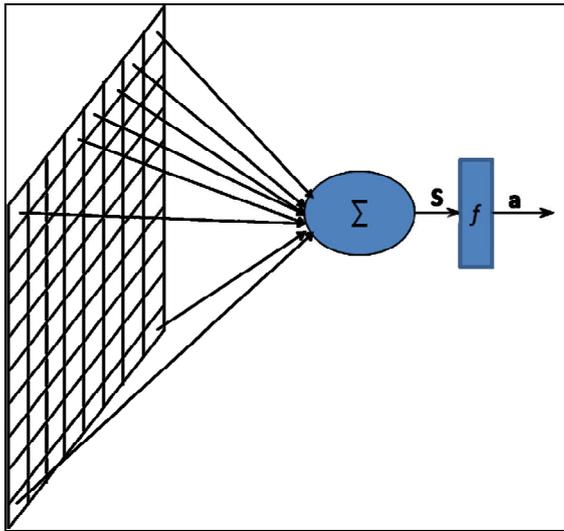
$$S = \sum_{i=1}^n e_{(i)} * p_{(i)} \quad (1)$$

<sup>a</sup> Facultad de Ingeniería de Sistemas, Computo y Telecomunicaciones, Universidad Inca Garcilazo de La Vega, Perú.

$$a = \begin{cases} 1 & \text{Si } S \geq U \\ 0 & \text{Si } S < U \end{cases} \quad (2)$$

**S** – Sumatoria  
**e** – Vector de entradas  
**p** – Vector de pesos  
**U** – Umbral  
**a** – Resultado

Según (1) para calcular la **S** debemos encontrar la sumatoria del producto de las entradas por los pesos y el resultado de la neurona la cual tiene dos estados activado 1 y desactivado 0 se calcula con (2) en donde si **S** sobre pasa el umbral se activa, en caso contrario de desactiva.



**Fig. 1: Neuronal artificial**

Como podemos apreciar el funcionamiento de la neurona artificial no es complejo sin embargo si tenemos los pesos adecuados puede resolver problemas complejos como el reconocimiento de caracteres.

Un algoritmo común [1] de entrenamiento durante el cual se actualiza los pesos tenemos en (3) y (4):

$$P_i = P_{i-1} + k \cdot e_i \cdot \text{Err} \quad (3)$$

$$\text{Err} = (U - S) \quad (4)$$

Donde:

$P_i$  – Peso nuevo  
 $P_{i-1}$  – Peso anterior  
 $k$  - Velocidad de aprendizaje  
 $\text{Err}$  – Diferencia entre el valor de Umbral me la sumatoria

Realizando un análisis minucioso del proceso de reconocimiento de caracteres desde un punto de vista heurístico, para garantizar la convergencia y rapidez del algoritmo de entrenamiento (3) se debe tener las siguientes consideraciones:

1. Para la actualización de los pesos solo se debe considerar los puntos que representen el carácter, más no el fondo.
2. Para que se distribuya el conocimiento en todos los

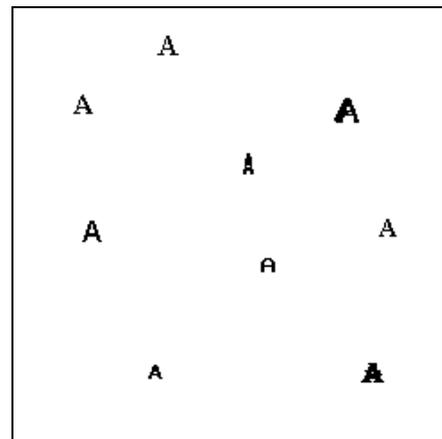
pesos de las entradas se debe seleccionar un valor de  $k$  pequeño.

3. Para obtener pesos heterogéneos actualizar cada peso calculando un nuevo error.
4. Ingresar los patrones de entrenamiento en orden empezando por aquellos que están formados por menos puntos negros a aquellos que contienen una gran cantidad de puntos negros.
5. Durante el entrenamiento con cada patrón solo actualizar los pesos de aquellos que no han sido actualizados anteriormente.
6. En caso que se active la neuronal artificial con carácter semejante, ajustar los pesos en lo que dicho carácter es diferente para acentuar la diferencia.

El algoritmo propuesto a sido implementado en un prototipo desarrollado en Visual C++, el cual toma una imagen de mapa de bits, realiza la binarización, segmenta los caracteres, expande o reduce el carácter según sea el caso a una dimensión de estándar de 20x16 puntos dando como resultado 320 entradas a la neurona artificial la cual realiza el reconocimiento.

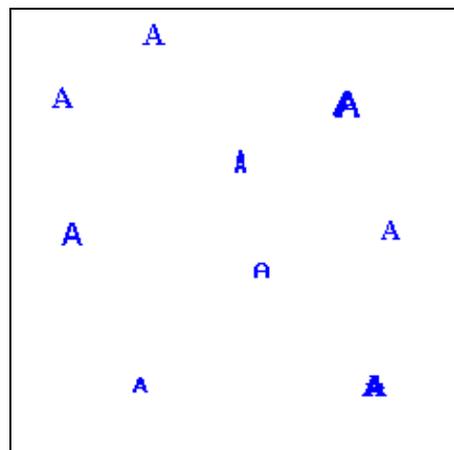
## RESULTADOS

Al prototipo ingresa una imagen como la que se presenta en la Fig. 2., en la cual se encuentra diversos tipo de de la letra A mayúscula, esta imagen es binarizada y se realiza la segmentación de cada carácter como se muestra en la Fig. 3.



**Fig. 2. Imagen digitalizada**

En la Fig. 4. se muestra la imagen del carácter segmentado y en la Fig. 5. se muestra la imagen de un carácter expandido.



**Fig. 3. Imagen binarizada y segmentada**

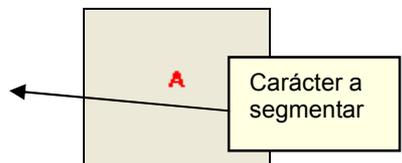


Fig. 4. Segmento de documento con carácter

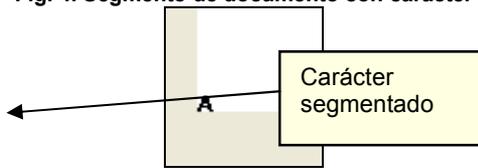


Fig. 6. Carácter segmentado

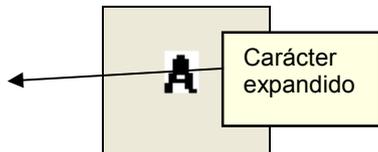


Fig. 7. Carácter expandido

Cada carácter expandido se pasa por la red neuronal ajustando los pesos para que la sumatoria resultante tienda a un valor de un umbral como se muestra en la Fig. 8. aplicando el algoritmo propuesto en la presente investigación.

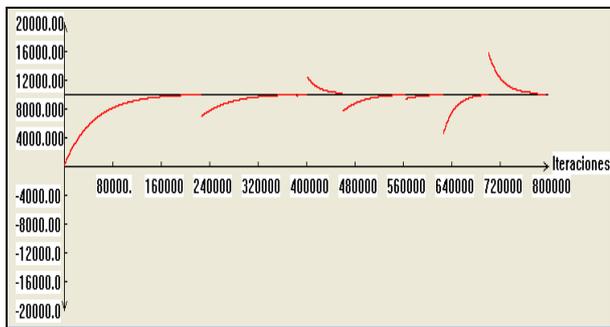


Fig. 8. Dependencia de S(iteraciones)

Como se puede apreciar en la Fig. 8. se muestra para cada una de los caracteres el ajuste de los pesos para que la sumatoria S se acerque al valor de umbral.

Los pesos que resultan del proceso de entrenamiento son utilizados para el funcionamiento de la neurona artificial durante el proceso de reconocimiento.

Para la verificación de la efectividad del funcionamiento de la neurona artificial utilizando los pesos obtenidos durante el entrenamiento se utiliza la imagen de entrada que se muestra en la Fig. 9.

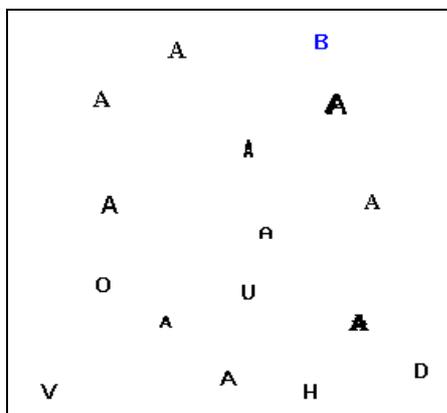


Fig. 9. Imagen con diversas letras

En esta imagen se incluye diversos caracteres semejantes a la al letra A para verificar la validez de los pesos obtenidos.

En las Fig. 10, Fig. 11. y Fig. 12 se presenta imágenes de resultados de funcionamiento de la neurona artificial utilizando los pesos obtenidos durante la el entrenamiento.



Fig. 10. Letra B no reconocida



Fig. 11. Letra A reconocida



Fig. 12. Letra A reconocida

## CONCLUSIONES

En base a los experimentos realizados utilizando el prototipo de software desarrollado en Visual C++, el cual implementa el algoritmo de entrenamiento propuesto en el presente trabajo podemos concluir que la convergencia del algoritmo es garantizada y rápida, como se puede ver en la Fig. 8., en donde se observa que en aproximadamente 8,000.000 iteraciones (actualizaciones de cada peso) se logra obtener los pesos adecuados para los nueve caracteres de entrenamiento.

Y se comprueba que los pesos obtenidos permiten reconocer otros tipos letra A mayúscula además de los tipos para los cuales fue entrenada la neurona artificial y detecta la diferencia con caracteres parecidos como la B, O, etc.

Los resultados de la presente investigación demuestran que si es posible implementar el reconocimiento de un carácter en una neurona artificial tipo perceptron con un limitador fuerte.

Adicionalmente se demuestra que un elemento de cálculo sencillo como es el perceptron permite resolver problemas complejos si se tiene los algoritmos apropiados para encontrar los pesos adecuados, lo cuales representa el conocimiento de la neurona artificial.

Por otro lado, se concluye que la efectividad de las soluciones en donde se utilizan neuronas artificiales depende estrictamente de los algoritmos para encontrar los pesos adecuados y también que dichos algoritmos basados en principios heurísticos son muy efectivos ya que las neuronas son elementos heurísticos y crean dicho comportamiento en las personas.

## REFERENCIAS BIBLIOGRÁFICAS

- [1]. Stuart Russell, Peter Norving. Inteligencia Artificial un enfoque moderno. Prentice Hall 1995. Segunda Edición.
- [2]. K.S. Fu; R.C. Gonzales; C.S.G. Lee. Robotics: Control, Sensing, Vision and Intelligence. McGraw – Hill Book Company. 1989
- [3]. Nils J. Nilsson. Artificial Intelligence. Hardcover - 536 pages (April 1998) Morgan Kaufmann Publishers.
- [4.] Abhijit S. Pandya; Robert B. Macy Pattern Recognition with Neural Networks in C++. IEEE PRESS. 1995.
- [5.] Jang, J. -S. R. Sun, C. -T. Mizutami, E. Neuro – Fuzzy and Soft Computing A Computational Approach to Learning and Machine Intelligence, Upper Saddle River: Prentice Hall INC., 1997.
- [6.] Fayin, L. Wechsler, H. Open Set Face Recognition Using Transduction. IEEE Transactions on Pattern Analysis and Machine Intelligence, November 2005, v.27 n.11, p.1686-1697.
- [7.] Torres, S. Reconocimiento Facial para Identificación de Personas mediante Redes Neuronales. Informe Técnico de Investigación RES. REC. N° 372.07-RR – FIIS - UNAC, Setiembre, 2008.