

REDES NEURONALES ARTIFICIALES APLICADAS AL RECONOCIMIENTO DE PATRONES

WILMER RIVAS ASANZA / BERTHA MAZÓN OLIVO



Editorial
UTMACH

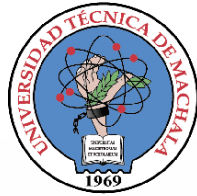
REDES 2017

COLECCIÓN EDITORIAL

Redes neuronales artificiales aplicadas al reconocimiento de patrones

Wilmer Rivas Asanza
Bertha Mazón Olivo

Coordinadores



Primera edición en español, 2018

Este texto ha sido sometido a un proceso de evaluación por pares externos con base en la normativa editorial de la UTMACH

Ediciones UTMACH

Gestión de proyectos editoriales universitarios

199 pag; 22X19cm - (Colección REDES 2017)

Título: Redes neuronales artificiales aplicadas al reconocimiento de patrones. / Wilmer Rivas Asanza / Bertha Mazón Olivo (Coordinadores)

ISBN: 978-9942-24-100-9

Publicación digital

Título del libro: Redes neuronales artificiales aplicadas al reconocimiento de patrones.

ISBN: 978-9942-24-100-9

Comentarios y sugerencias: editorial@utmachala.edu.ec

Diseño de portada: MZ Diseño Editorial

Diagramación: MZ Diseño Editorial

Diseño y comunicación digital: Jorge Maza Córdova, Ms.

© Editorial UTMACH, 2018

© Wilmer Rivas / Bertha Mazón, por la coordinación

D.R. © UNIVERSIDAD TÉCNICA DE MACHALA, 2018

Km. 5 1/2 Vía Machala Pasaje

www.utmachala.edu.ec

Machala - Ecuador

Advertencia: "Se prohíbe la reproducción, el registro o la transmisión parcial o total de esta obra por cualquier sistema de recuperación de información, sea mecánico, fotoquímico, electrónico, magnético, electro-óptico, por fotocopia o cualquier otro, existente o por existir, sin el permiso previo por escrito del titular de los derechos correspondientes".



César Quezada Abad, Ph.D
Rector

Amarilis Borja Herrera, Ph.D
Vicerrectora Académica

Jhonny Pérez Rodríguez, Ph.D
Vicerrector Administrativo

COORDINACIÓN EDITORIAL

Tomás Fontaines-Ruiz, Ph.D
Director de investigación

Karina Lozano Zambrano, Ing.
Jefe Editor

Elida Rivero Rodríguez, Ph.D
Roberto Aguirre Fernández, Ph.D
Eduardo Tusa Jumbo, Msc.
Irán Rodríguez Delgado, Ms.
Sandy Soto Armijos, M.Sc.
Raquel Tinóco Egas, Msc.
Gissela León García, Mgs.
Sixto Chilibingua Villacis, Mgs.

Consejo Editorial

Jorge Maza Córdova, Ms.
Fernanda Tusa Jumbo, Ph.D
Karla Ibañez Bustos, Ing.
Comisión de apoyo editorial

Índice

Capítulo I

Generalidades de las redes neuronales artificiales 11

Wilmer Rivas Asanza; Bertha Mazón Olivo; Edwin Mejía Peñafiel

Capítulo II

Tecnologías utilizadas en el reconocimiento de patrones y clasificación de imágenes 36

Wilmer Rivas Asanza; Bertha Mazón Olivo; Eduardo Tusa

Capítulo III

Reconocimiento de patrones en imágenes 61

Wilmer Rivas Asanza; Bertha Mazón Olivo; Eduardo Tusa

Capítulo IV

Matemáticas aplicadas en la clasificación de patrones 127

Maritza Alexandra Pinta; Mónica Loaiza Loayza; Freddy Jumbo Castillo

Capítulo V

Reconocimiento de patrones en imágenes médicas. Caso de estudio Mamografías.....176

Eduardo Tusa; María Perez

Dedicatoria

A nuestra familia por su apoyo incondicional y comprensión.

A los estudiantes de la Universidad Técnica de Machala,
Carrera de Ingeniería de Sistemas por su aporte valioso

Introducción

Las Redes neuronales artificiales (RNA), es una técnica de la Inteligencia Artificial, debido a que intenta desarrollar soluciones informáticas emulando el comportamiento del cerebro humano. La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos intentar o procurar modelar en el ordenador el funcionamiento del cerebro humano.

En la actualidad son ampliamente usadas en diferentes campos, por ejemplo reconocimiento de imágenes a través de patrones. En este sentido, también los especialistas, expertos, etc., han desarrollado tecnología que permite modelar los diferentes tipos de redes neuronales. Es por ello, que este texto propone abarcar el estudio de las diferentes herramientas tecnológicas, ejemplos prácticos, la relación que existe entre las RNA y las matemáticas y casos prácticos desarrollados por los autores en el ámbito de la medicina y la agronomía.

04 Capítulo Matemáticas aplicadas en la clasificación de patrones

Maritza Alexandra Pinta; Mónica Loaiza Loayza; Freddy Jumbo Castillo

Este capítulo, tiene como objetivo principal presentar la relación existente entre las matemáticas y las redes neuronales artificiales, mediante el uso de los modelos matemáticos en los diferentes tipos de estructura de redes neuronales. La información se estructura en dos secciones.

La primera sección muestra las fórmulas matemáticas utilizadas en los diferentes componentes de las redes neuronales artificiales.

En la segunda sección, se muestra un ejemplo donde se aplica las fórmulas matemáticas.

Maritza Alexandra Pinta: Doctorando de Matemáticas de la Universidad de Almería, España; Ingeniero Civil (Utmach), Licenciado en C.C.E.E. especialidad de Matemáticas (UTPL), Doctor en Ciencias de la Educación (UTPL), Magíster en Desarrollo de la Inteligencia y Educación (UTPL). Actualmente es Profesor Titular en la UTMACH. A la fecha cuenta con varias publicaciones indexadas y libros.

Mónica Loaiza Loayza: Ingeniera en Sistemas, Magíster en Docencia y Gerencia en Educación Superior en la actualidad estoy cursando Máster Universitario en Dirección e Ingeniería de Sitios Web en la Universidad Internacional de la Rioja de España, además de cursar el programa de Doctorado en Informática por la Universidad de Almería en España. Soy docente contratada en la UTMACH

Freddy Jumbo Castillo: Magíster en Ciencias de la Información Geográfica y Sistemas de la University Salzburg (con distinción), Magíster en Educación Superior e Ingeniero en Sistemas de la Universidad San Antonio de Machala. Soy docente contratado en la Universidad Técnica de Machala.

Introducción

Desde los inicios de la ciencia computacional, las matemáticas han estado presente y tomado en consideración para la evolución de las redes neuronales, por ejemplo, en el año 1943, el neurobiólogo Warren McCulloch, y el estadístico Walter Pitts, publicaron el artículo “A logical calculus of Ideas Imminent in Nervous Activity”. El mismo detalla la importancia de las matemáticas en redes neuronales al igual que sirvió como “base y el inicio del desarrollo en diferentes campos como son los Ordenadores Digitales (John Von Neuman), la Inteligencia Artificial (Marvin Minsky con los Sistemas Expertos) y el funcionamiento del ojo (Frank Rosenblatt con la famosa red llamada Perceptron)” (Olabe, 2009, pág. 6).

Desde entonces, las matemáticas cumplen un rol importante en las redes neuronales, ya que, por medio de cálculos, funciones, algoritmos, se logra realizar el proceso de aprendizaje o reconocimiento de patrones de las redes neuronales, que según Marcos Redondo Fonseca, en el libro de “Simulación de redes neuronales como herramienta Big Data en el ámbito sanitario” (Fonseca, 2016), expresa la importancia de las matemáticas citando que “las RN aún están naciendo, de que hacen falta algoritmos mejores (la back propagación funciona pero es muy lenta) sobre todo modelos matemáticos para describir con detalle su dinámica” (Fonseca, 2016). En dicho contexto hace eco a la necesidad de investigar fórmulas matemáticas que permitan tener resultados eficientes.

Además, otro punto a destacar del porque las matemáticas están asociadas a las redes neuronales es que estas “fueron creadas en un principio para conseguir formalizaciones matemáticas acerca de la estructura del cerebro” (Raquel Flores, 2008, pág. 23), pasando por un sin número de experimentos, es así como como “Alan Turing fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación” (Matich, 2001).

Con el pasar del tiempo, se fueron agregando nuevas variaciones matemáticas incorporadas especialmente en el

algoritmo de aprendizaje del perceptrón, dichas variaciones matemáticas fueron realizadas por WIDROW y HOFF (1960), dando como resultado a la denominada “ley de Widrow-Hoff” la misma que dio lugar al modelo ADALINE que constituyó la primera red neuronal artificial aplicada a un problema real, misma que ha venido usándose durante el transcurrir de los años.

Hoy en día las redes neuronales tienen una estrecha relación con distintas ramas de estudios como la informática, biología, estadística, por nombrar algunas de ellas, en la cual cada una de esas ramas, las matemáticas se encuentran inmersas, haciendo que las RN evolucionen de manera impresionante.

Fórmulas matemáticas aplicadas en el proceso de aprendizaje en una de las más importantes tipos de redes neuronales perceptrón multicapa.

Dentro de cada fase del proceso de aprendizaje del perceptrón multicapa, se encuentran distintas fórmulas matemáticas, las cuales se las encuentran desde la propagación de patrones de entrada, pasando por sus actividades tanto neuronales de la capa de entrada oculta y salida.

Propagación de los patrones de entrada

Una de las principales características del perceptrón multicapa, es que las variables empleadas en la capa de entrada, se encuentran relacionadas con las variables de la capa de salida, para lograr esta relación, se ira propagando hacia adelante los valores empleados en la capa de entrada. El funcionamiento de esta propagación se basa en que, cada neurona de la red procesa la información recibida por sus entradas y produce una respuesta o activación que se propaga, a través de las conexiones correspondientes, hacia las neuronas de la siguiente capa.

A continuación, se presenta las expresiones de la propagación, según lo estipulado en el libro de redes neuronales

artificiales (Viñuela, 2004), en la que se define que “sea un Perceptrón multicapa con C capas, $C - 2$ capas ocultas y n_q neuronas en la capa q , $q = 1, 2, 3, 4, \dots, C$. Además, sea $W^q = (w_{ij}^q)$ que va a representar una matriz de pesos que se asocia a las conexiones de la capa q a la $q + 1$ para $q = 1, 2, \dots, C - 1$, en la cual (w_{ij}^q) va a representar el peso que va a tener la conexión de dicha neurona (i) de la capa q hacia la neurona j de la siguiente capa ($q + 1$); y además sea, $U^q = (u_i^q)$ que es el vector que representa a los valores umbrales de las neuronas de la capa q , para $q = 2, 3, \dots, C$. Para la activación de las neuronas i de la capa q , se va a denotar de la siguiente forma: a_i^q .

Para el cálculo de las activaciones, según el libro *Redes Neuronales Artificiales* (Viñuela, 2004), se consideran:

- Activación de las neuronas de la capa de entrada (a_i^1). Las neuronas de la capa de entrada se encargan de transmitir hacia la red las señales recibidas del exterior. Por tanto, la fórmula para esta activación se muestra en la ilustración 1:

$$a_i^1 = x_i \quad \text{para } i=1,2,\dots,n_1$$

Fuente: (Viñuela, 2004, pág. 21)

En donde $X = (x_1, x_2, \dots, x_n)$ la cual es un vector que representa el patrón de entrada de la red.

- Activación de las neuronas de la capa oculta q (a_i^q). Las neuronas ocultas de la red procesan la información recibida aplicando la función de activación f a la suma de los productos de las activaciones que recibe por sus correspondientes pesos, es decir, tal y como se muestra en la ilustración 2:

$$a_i^q = f \left(\sum_{j=1}^{n_{q-1}} w_{ji}^{q-1} a_j^{q-1} + u_i^q \right) \quad \text{para } i = 1, 2, \dots, n_q \text{ y } q = 2, 3, \dots, C - 1$$

Fuente: (Viñuela, 2004, pág. 23)

Donde a_i^{q-1} son las activaciones de las neuronas de la capa $q - 1$.

- Activación de las neurona de la capa de salida (a_i^c). Al igual que en el caso anterior, la activación de estas neuronas viene dada por la función de activación f aplicada a la suma de los productos de las entradas que recibe por sus correspondientes pesos, para esto se usa la fórmula mostrada en la ilustración 3:

$$y_i = a_i^c = f \left(\sum_{j=1}^{nc-1} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \text{ para } i = 1, 2, \dots, nc$$

Fuente: (Viñuela, 2004, pág. 25)

Donde $Y = (y_1, y_2, \dots, y_{nc})$ que representa el vector salida de la red.

Hay que recordar que f representa la función de activación. Dentro de la arquitectura Perceptron multicapa, se destaca 2 funciones de activación, las cuales son: función sigmoideal (ilustración 4 y 5) y la función de la tangente hiperbólica (ilustración 6 y 7). Ambas funciones cuentan con similar “rango continuo de valores dentro de los intervalos $[0, 1]$ y $[-1, 1]$ ” (Viñuela, 2004, pág. 40), la cuales se representan así:

Función sigmoideal

$$f_1(x) = \frac{1}{1 + e^{-x}}$$

Fuente: (Viñuela, 2004, pág. 26)

Función de la tangente hiperbólica

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

Fuente: (Viñuela, 2004, pág. 28)

Ambas funciones son crecientes y poseen 2 niveles de saturación, la primera es un nivel de saturación conocido como máximo y da como resultado 1 y el otro es el mínimo que da como salida 0 en el caso de la función sigmoïdal y -1 para la función tangente hiperbólica, tal y como se muestran en las figuras 6 y 7 respectivamente.

Según (Serrano, 2010, pág. 8), existe otra función de activación la cual es la función gaussiana, esta función “es muy usada en un tipo de red neuronal conocida como función de base radial (RBF)” (Serrano, 2010, pág. 9) . Su fórmula es:

$$f(x) = K_1 \cdot e^{-K_2 x^2}$$

Fuente: (Serrano, 2010, pág. 9)

Además una de las principales características de esta función, a diferencia de las otras, es su localidad, la función tiende a cero a partir de un cierto x de tal forma que solo para un determinado intervalo de entradas a las salidas de la función de activación es diferentes de cero .

Como se mostró anteriormente, existen varias fórmulas de función de activación, pero para la selección de alguna de estas funciones, esta selección se la realiza en base a la elección del diseñador de la red neuronal, que básicamente se fundamentan en el valor que se quiera alcanzar para las neuronas. Hay que destacar que la expresión en ambas funciones están relacionadas, y se representan por: $x_2(x) = 2f_1(x) - 1$, por lo que el uso se lo realiza en base al recorrido que se quiera realizar.

Cálculo de las activaciones para un perceptron multicapa con dos neuronas de entrada, dos ocultas y una salida.

Para demostrar el funcionamiento de las fórmulas de activación de un perceptron multicapa, se tomará el ejemplo de aplicación presentado por el autor (Viñuela, 2004, pág. 45) el cual trata sobre la función XOR, la misma que detalla lo siguiente:

Aplicando las fórmulas de activación descritas anteriormente, se obtiene:

- Activaciones de las neuronas de entrada:

$$a_1^1 = x_1 \text{ y } a_2^1 = x_2$$

- Activaciones de las neuronas ocultas:

$$a_1^2 = f(w_{11}a_1^1 + w_{21}a_2^1 + u_1^2) \text{ y } a_2^2 = f(w_{12}a_1^1 + w_{22}a_2^1 + u_2^2)$$

- Activación de la neurona de salida:

$$y = a_1^3 = f(v_1a_1^2 + v_2a_2^2 + u^3)$$

El patrón de entrada y salida serán los siguientes:

Tabla 1. Función XOR

Entrada	Salida
(0,0)	0
(0,1)	1
(1,0)	1
(1,1)	0

Fuente: (Viñuela, 2004, pág. 46)

Una vez aplicado las fórmulas, el resultado queda en lo siguiente:

Tabla 2. Activaciones del perceptron multicapa para la función XOR

Patrón de entrada	a_1^1	a_2^1	a_1^2	a_2^2	Salida de la red
(0,0)	0	0	0,129781	0,015875	0,128883
(0,1)	0	1	0,972618	0,204662	0,863310
(1,0)	1	0	0,964023	0,202889	0,858615
(1,1)	1	0	0,999843	0,802384	0,154993

Fuente: (Viñuela, 2004, pág. 48)

Realizando un análisis de la ilustración 14, que una vez que se han definido los patrones de entrada de la función XOR (ver ilustración 13), los pesos y umbrales de la red neuronal que se muestran en la ilustración 9, tomarán los valores de $w_{11} = 5,191129$, $w_{12} = 2,758669$, $w_{21} = 5,473012$, $w_{22} = 2,769596$, $v_1 = 5,839709$, $v_2 = -6,186834$, $u^2/1 = -1,90289$, $u^2/1 = -4,127002$ y $u^3 = -2,570539$, todos estos parámetros se los sustituye en las fórmulas anteriores y utilizando la función sigmoideal (ilustración 4), la salida para la red XOR serían los valores que se indican en la Tabla 2.

Proceso de aprendizaje del perceptron multicapa.

El objetivo del perceptron multicapa es de aprender ajustando los parámetros de la red, en otras palabras, ajustando pesos y umbrales, esta tarea se la realiza con el fin de que todas las entradas que se envié a la red de como resultado las salidas deseadas minimizando la función de error E.

Dado esta premisa, se procederá con la explicación del proceso de aprendizaje del perceptron multicapa.

Sea $\{(X(n), S(n)), n=1, \dots, N\}$ un conjunto de patrones que representa el problema que se requiere resolver, en la cual $X(n) = (x_1(n), \dots, x_n(n))$ son todos aquellos patrones que ingresan a la red, es decir todas las entradas, y $S(n) = (s_1(n), \dots, s_n(n))$ son las salidas que se espera obtener de todas las entradas ingresadas y el valor N en cambio es el número de patrones disponibles.

En muchas de las ocasiones, es frecuente que los patrones de entradas y salidas estén normalizados gracias a la utilización de una transformación lineal en los intervalos $[0,1]$ o $[-1,1]$.

Según (Viñuela, 2004), los pasos que componen el proceso de aprendizaje del perceptron multicapa son los siguientes:

1. Se inicializan los pesos y umbrales de la red. Generalmente, esta inicialización es aleatoria y con valores alrededor de cero.
2. Se toma un patron "n" del conjunto de entrenamiento,

$(X(n), S(n))$, y se propaga hacia la salida de la red el vector de entrada $X(n)$ utilizando las siguientes ecuaciones:

- “Activación de las neuronas de la capa de entrada a_i^1 . Las neuronas de la capa de entrada se encargan de transmitir hacia la red las señales recibidas desde el exterior” (Viñuela, 2004). Por tanto:

$$a_i^1 = x_i \text{ para } i = 1, 2, \dots, n_i$$

Donde $X = (x_1, x_2, \dots, x_{n_i})$ representa el vector o patrón de entrada a la red.

Fuente: (Viñuela, 2004, pág. 51)

- “Activación de las neuronas de la capa oculta c (a_i^c). Las neuronas ocultas de la red procesan la información recibida aplicando la función de activación f a la suma de los productos de las activaciones que recibe por sus correspondientes pesos” (Viñuela, 2004), es decir:

$$a_i^c = f \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \text{ para } i = 1, 2, \dots, n_c \text{ y } c = 2, 3, \dots, C - 1$$

donde a_j^{c-1} son las activaciones de las neuronas de la capa $c-1$

Fuente: (Viñuela, 2004, pág. 52)

- “Activación de las neuronas de la capa de salida $a_{(i)}^c$. Al igual que en el caso anterior, la activación de estas neuronas viene dada por la función de activación f aplicada a la suma de los productos de las entradas que recibe por sus correspondientes pesos” (Viñuela, 2004, pág. 56):

$$y_i = a_i^c = f(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^1) \text{ para } i = 1, 2, \dots, n_c$$

Donde $Y = (y_1, y_2, \dots, y_{n_c})$ es el vector de salida de la red.

Fuente: (Viñuela, 2004, pág. 58)

Con la aplicación de estas ecuaciones se obtiene la respuesta de la red para dicho vector de entrada, $Y(n)$.

Se evalúa el error cuadrático cometido por la red para el patrón “n” utilizando la ecuación:

$$e(n) = \frac{1}{n_c} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2$$

Siendo $Y(n) = (y_1(n), \dots, y_{n_c}(n))$ y $S(n) = (s_1(n), \dots, s_{n_c}(n))$ los vectores de salidas de la red y salidas deseadas para el patrón n, respectivamente

Fuente: (Viñuela, 2004, pág. 64)

4. “Se aplica la regla delta generalizada para modificar los pesos y umbrales de la red” (Viñuela, 2004). Para ello se siguen los siguientes pasos:

a “Se calculan los valores δ para todas las neuronas de la capa de salida” (Viñuela, 2004) utilizando la siguiente ecuación:

$$\delta_i^c(n) = -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_c-1} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right)$$

Fuente: (Viñuela, 2004, pág. 68)g

b) “Se calculan los valores δ para el resto de las neuronas de la red utilizando la siguiente ecuación empezando desde la última capa oculta y retropropagando dichos valores hacia la capa de entrada” (Viñuela, 2004, pág. 73):

$$\delta_i^{c+1}(n) = f' \left(\sum_{j=1}^{n_c} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) \sum_{i=1}^{n_c+1} \delta_i^{c+2}(n) w_{ji}^{c-1}$$

Fuente: (Viñuela, 2004, pág. 75)

c) Se modifican los pesos y umbrales de la capa de salida siguiendo las siguientes dos ecuaciones:

Primera ecuación:

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + a\delta_i^{c+1}(n) a_j^{c+1}$$

Para $j = 1, 2, \dots, n_{c-1}$ $i = 1, 2, \dots, n_c$

Fuente: (Viñuela, 2004, pág. 78)

Segunda ecuación:

$$u_i^c(n) = u_i^c(n-1) + a\delta_i^c(n) \text{ para } i = 1, 2, \dots, n_c$$

Fuente: (Viñuela, 2004, pág. 80)

Luego se usarán las siguientes dos ecuaciones para el resto de parámetros de la red:

Primera ecuación:

$$w_{kj}^c(n) = w_{kj}^c(n-1) + a\delta_j^{c+1}(n) a_k^c(n)$$

Para $k = 1, 2, \dots, n_{c-1}$ $j = 1, 2, \dots, n_{c+1}$ y $c = 1, 2, \dots, C-2$

Fuente: (Viñuela, 2004, pág. 82)

Segunda ecuación:

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + a\delta_j^{c+1}(n)$$

Para $j = 1, 2, \dots, n_{c+1}$ y $c = 1, 2, \dots, C-2$

Fuente: (Viñuela, 2004)

5. “Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento, completando así un ciclo de aprendizaje, o epoch” (Viñuela, 2004).

6. Se evalúa el error total E con la siguiente ecuación:

$$E = \frac{1}{N} \sum_{n=1}^N e(n)$$

Donde N es el número de patrones o muestras y e(n) es el error cometido por la red para el patrón dado por:

Fuente: (Viñuela, 2004, pág. 92)

Dicho error también recibe el “nombre de error de entrenamiento, pues se calcula utilizando los patrones de entrenamiento” (Viñuela, 2004, pág. 94).

7. “Se repiten los pasos 2, 3, 4, 5 y 6 hasta alcanzar un mínimo del error de entrenamiento” (Viñuela, 2004), para lo cual se realizan “m” ciclos de aprendizaje.

Ejemplos de red neuronal artificial perceptron multicapa.

Se muestra a continuación la resolución de dos ejercicios que aunque estos han sido subidos y planteados en la nube por otros autores, en este texto se diferencia por dos aspectos (i) las herramientas tecnológicas utilizadas (ii) la relación entre las RNA y las matemáticas, sobre todo el segundo ejercicio ubicado como anexo en el cual se modela la RNA en JOONE y en EXCEL se hace el siguiente de los cálculos matemáticos con sus respectivas formulas.

Problema tabla xor

Planteamiento del problema.

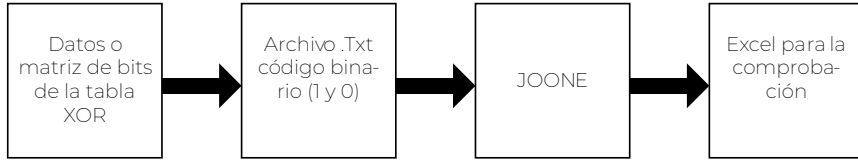
Modelar una red neuronal, de tal manera que mediante el proceso de aprendizaje la aplicación pueda aprender la tabla de verdad de la compuerta lógica XOR, ejemplo que permitirá comprender el funcionamiento de las redes neuronales. Para el objetivo que se plantea, se usa el programa JOONE para automatizar de forma rápida y sencilla los datos que se manipulan en el problema planteado.

Para resolver este ejercicio, primero se modela una red simple en JOONE, donde se destacará el uso de las formulas vistas anteriormente en este capítulo.

Arquitectura de la solución del problema.

La arquitectura a seguir para resolver el problema planteado será el siguiente:

Ilustración 1: Arquitectura de la red neuronal para la tabla XOR

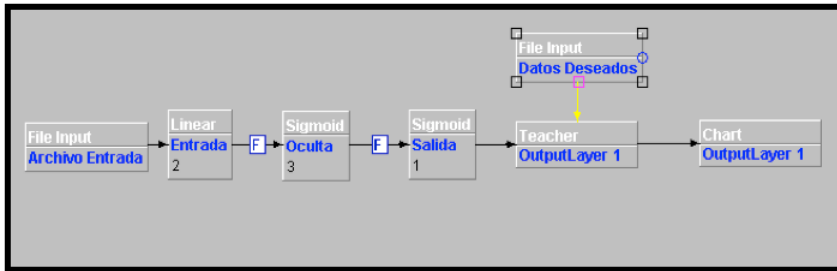


Fuente: Autor

Modelo de la red neuronal artificial en joone

El modelo de la red neuronal en JOONE está compuesto por tres capas: de entrada, oculta y salida de datos como se muestra en la ilustración 2.

Ilustración 2: Modelo de la red neuronal artificial en joone



Fuente: Autor

Modelo matemático de la red neuronal artificial

a) El número de combinaciones para la compuerta XOR son cuatro con sus respectivas salidas como se muestra en la ilustración 3.

Ilustración 3: Tabla lógica de la compuerta XOR

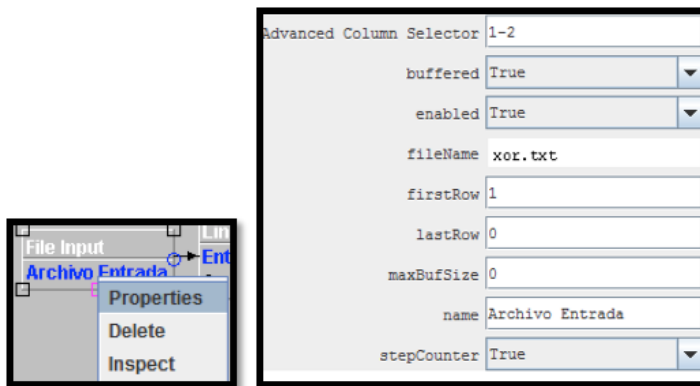
A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	0

Fuente: Autor

d) Luego, seleccionar la capa de file input y dar clic derecho en el mismo y elegir la opción propiedades, deberá aparecer una nueva ventana en la cual deberá modificar los siguientes datos:

Advanced column selector: 1-2 (las columnas 1 y 2 del archivo txt serán los datos de entrada).

FileName: deberá buscar el archivo txt.



e) Después seleccionar a la capa de entrada, dar clic derecho y elegir la opción propiedades, aparecerá una nueva ventana la misma que deberá modificar las siguientes opciones:

Layername: esta opción sirve para cambiar el nombre a la capa, colocar un nombre que identifique a la capa seleccionada.

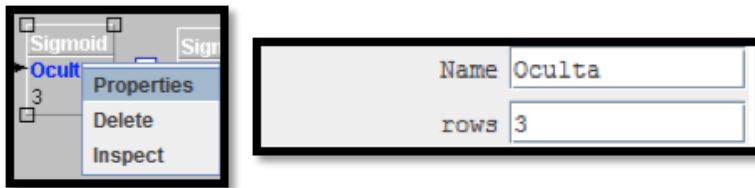
Rows: en esta opción se deberá colocar cuantos datos serán de entrada, para este ejercicio se colocará el número 2, ya que las combinaciones del XOR se realiza con dos datos A y B.



f) Después seleccionar a la capa oculta, dar clic derecho y elegir la opción propiedades, aparecerá una nueva ventana la misma que deberá modificar las siguientes opciones:

Layername: esta opción sirve para cambiar el nombre a la capa, colocar un nombre que identifique a la capa seleccionada.

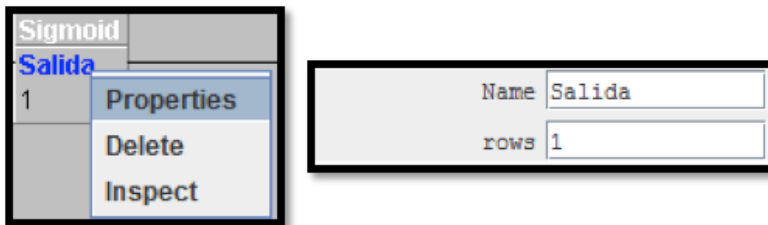
Rows: en esta opción se deberá colocar cuantas capaz usará para la oculta, para este ejercicio se colocará el número 3, pero este número dependerá del diseñador de la red neuronal.



g) Después seleccionar a la capa de salida, dar clic derecho y elegir la opción propiedades, aparecerá una nueva ventana la misma que deberá modificar las siguientes opciones:

Layername: esta opción sirve para cambiar el nombre a la capa, colocar un nombre que identifique a la capa seleccionada.

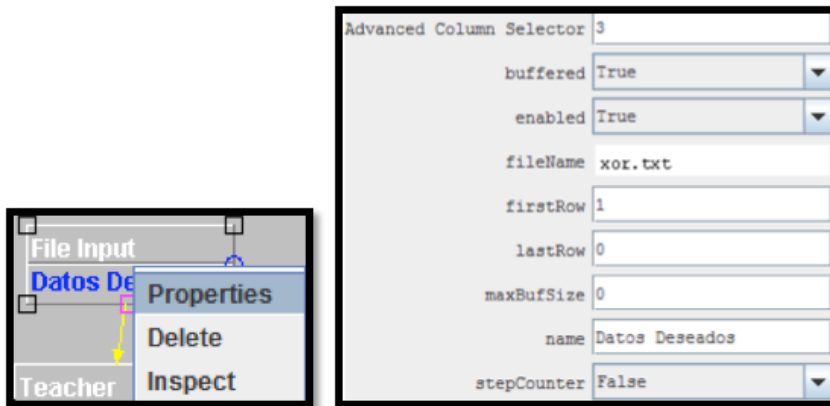
Rows: en esta opción se deberá colocar cuantas capaz usará para la salida, para este ejercicio se colocará el número 1, ya que el XOR solamente tiene una salida.



h) Luego, seleccionar la capa de file input y dar clic derecho en el mismo y elegir la opción propiedades, deberá aparecer una nueva ventana en la cual deberá modificar los siguientes datos:

Advanced column selector: colocar el número 3 (vendría ser la columna 3 del archivo txt que es el dato de salida).

FileName: deberá buscar el archivo txt.

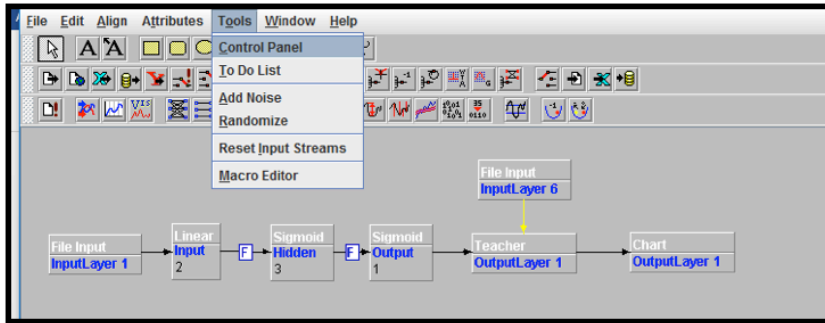


i) En propiedades de la capa Chart (Outlayer 1) se debe tener en cuenta las combinaciones posibles que se deben realizar tal y como se aprecia en la siguiente imagen:

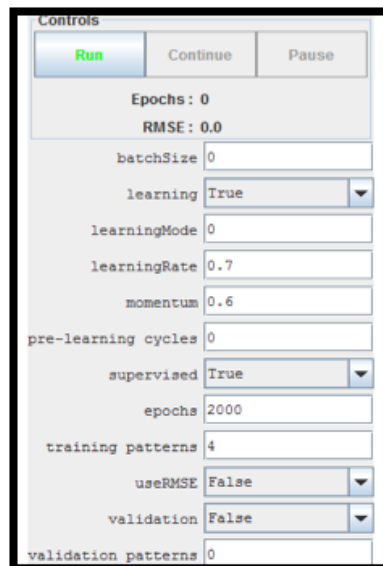


Ejecución de la red neuronal artificial

Para la ejecución del ejercicio realizado en Joone, debe dirigirse al menú Tools y elegir la opción control panel.



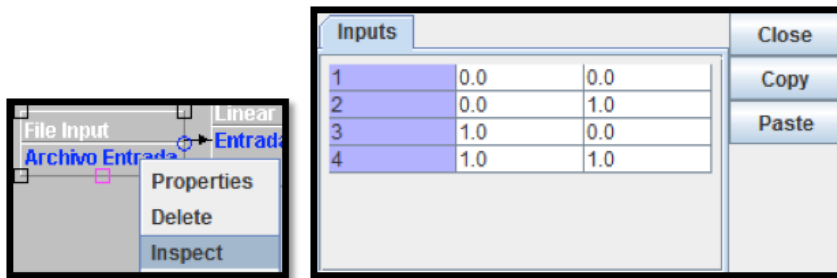
Aparecerá la siguiente imagen donde deberá configurar el número de combinaciones (epochs). Para este ejemplo se utilizará un número de 2000, posteriormente se ejecuta el programa dando clic en run, en donde automáticamente se realizarán los procesos de cálculos, a su vez esto da como resultado los ciclos completados y el error de la red en caso de tenerlo.



Llegado a este punto, el software Joone utilizará las fórmulas de cálculo de errores.

Para la comprobación de los datos se debe inspeccionar cada tabla y las relaciones sinápticas las cuales mostrarán valores como los siguientes:

En la inspección del archivo de entrada se mostrará las combinaciones entre A y B

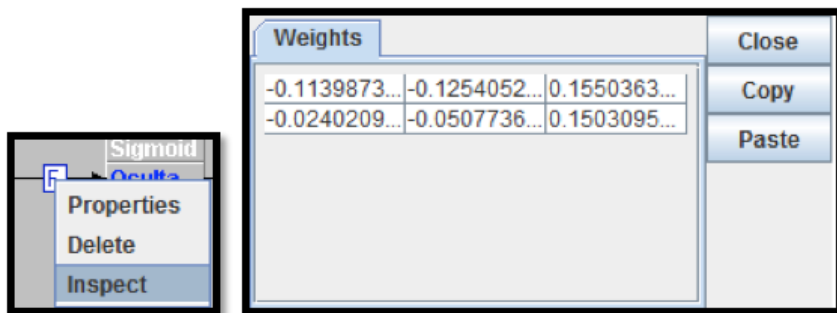


The screenshot shows a context menu for a 'File Input' component with options: Properties, Delete, and Inspect. To the right is a table titled 'Inputs' with the following data:

Inputs		
1	0.0	0.0
2	0.0	1.0
3	1.0	0.0
4	1.0	1.0

En esta parte, el software Joone utilizará la fórmula de activación de las neuronas de la capa de entrada.

En la inspección de la relación sináptica se visualizarán los pesos entre la capa entrada y la capa oculta, la cual otorgará los siguientes valores:

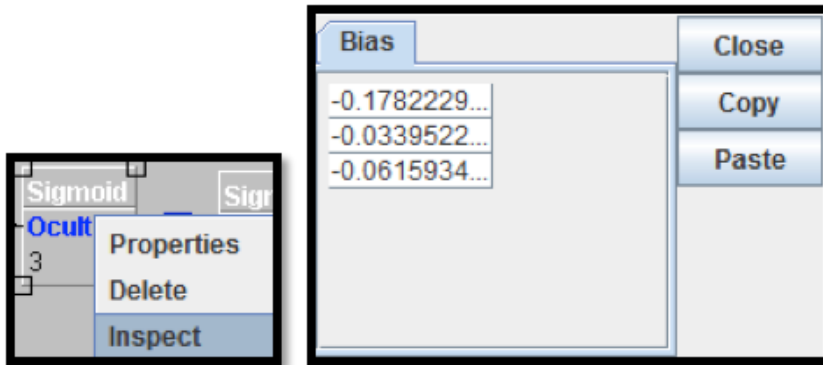


The screenshot shows a context menu for a 'Sigmoid' component with options: Properties, Delete, and Inspect. To the right is a table titled 'Weights' with the following data:

Weights		
-0.1139873...	-0.1254052...	0.1550363...
-0.0240209...	-0.0507736...	0.1503095...

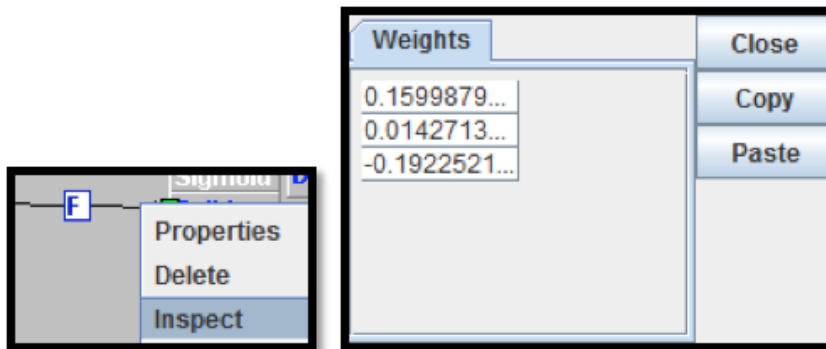
En este lugar, el software Joone utilizará la fórmula de cálculo de los pesos entre la capa de entrada y la capa oculta.

En la inspección de la capa oculta se visualizará los siguientes pesos de la combinación realizada de las tres combinaciones.



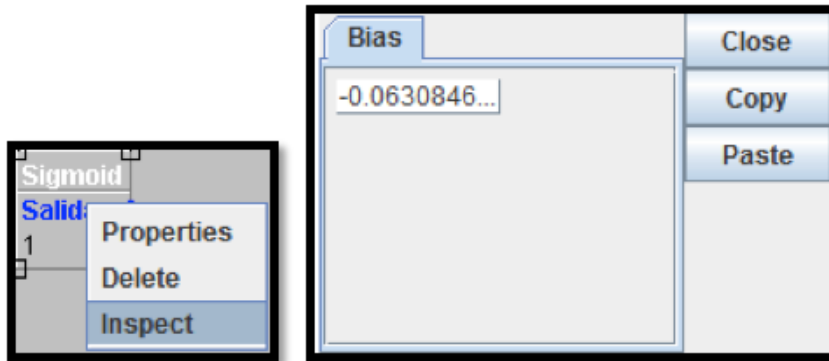
Estando aquí, el software Joone utilizará la fórmula de cálculo de los pesos de la capa oculta.

En la inspección de la relación sináptica visualizamos los pesos entre la capa oculta y la capa salida nos aparecerá los siguientes valores



Una vez en esta parte, el software Joone utilizará la fórmula de cálculo de los pesos entre la capa de entrada y la capa oculta.

En la inspección de la capa salida aparecerá un peso como resultado



Llegado a este punto, el software Joone utilizará la fórmula del cálculo de los pesos de salida.

En los datos Deseados tenemos las salidas o el resultado de la combinación entre A y B

Referencia bibliográfica

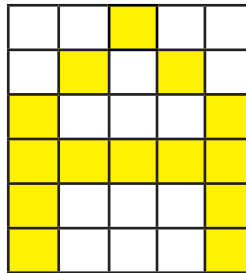
- Fonseca, M. R. (2016). *Simulación de redes neuronales como herramienta Big Data en el ámbito sanitario*. Lulu.com.
- Matich, D. J. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Santa Fe.
- Olabe, X. B. (2009). *Redes neuronales y sus aplicaciones*. Bilbao.
- Raquel Flores, J. F. (2008). *Las redes neuronales artificiales*. España: Gesbiblo S.L.
- Ruiz, C. A. (2012). *Redes Neuronales: conceptos básicos y aplicaciones*. Rosario.
- Serrano, A. J. (2010). *Redes neuronales artificiales*. Valencia: Universida de Valencia.
- Viñuela, P. I. (2004). *Redes Neuronales Artificiales: Un enfoque practico*. Madrd: Pearson Educacion .

Anexos

1. Problema 2. Aprendizaje de vocales

Representar los Patrones en una matriz de 5 x 6 del modo mostrado en la siguiente ilustración:

Ilustración 1: Representación de los patrones en una matriz



		■			
	■		■		
■					■
■	■	■	■	■	■
■					■
■					■

Fuente: Autor

1.1 Representar los Patrones en una Matriz

El artificio para representar un número en dicha matriz, será poner el valor uno (1) en las celdas por donde pasa la marca del número y el valor cero (0) en caso contrario. Por ejemplo, la vocal A se podría representar tal como muestra en la siguiente ilustración.

Ilustración 2: Artificio para Representar los Patrones en una Matriz

	1	2	3	4	5
1	0	0	1	0	0
2	0	1	0	1	0
3	1	0	0	0	1
4	1	1	1	1	1
5	1	0	0	0	1
6	1	0	0	0	1

Fuente: Autor

1.2 Convertir la Matriz en un Vector Lineal

Como se conoce los Datos de Entrada deben estar representados por un vector lineal, por esa razón en nuestro caso pasaremos de la matriz que debe ser 5 x 6 a un vector lineal X de tamaño 30, cuyos elementos serán, X1, X2, X30, según se precisa en la siguiente ilustración.

Ilustración 3: Convertir la Matriz en un Vector Lineal

	1	2	3	4	5
1	0	0	1	0	0
2	0	1	0	1	0
3	1	0	0	0	1
4	1	1	1	1	1
5	1	0	0	0	1
6	1	0	0	0	1

	1	2	3	4	5
1	X1	X2	X3	X4	X5
2	X6	X7	X8	X9	X10
3	X11	X12	X13	X14	X15
4	X16	X17	X18	X19	X20
5	X21	X22	X23	X24	X25
6	X26	X27	X28	X29	X30

Fuente: Autor

Teniendo en cuenta dicha conversión, los 30 valores del vector X, para representar la vocal A de la ilustración anterior, serán mostrados en la siguiente ilustración.

Ilustración 4: Representación de la vocal A usando el vector X

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X30
0	0	1	0	0	0	1	0	1	0		1

Fuente: Autor

1.3 Representar los caracteres que se desean reconocer usando las Estructuras de Datos

A continuación, representaremos, usando las estructuras de datos para las vocales A, E, I, O, U.

Ilustración 5: Representación de Patrones usando Matrices

	1	2	3	4	5		1	2	3	4	5		1	2	3	4	5
1	0	0	1	0	0	1	1	1	1	1	1	1	0	1	1	1	0
2	0	1	0	1	0	2	1	0	0	0	0	2	0	0	1	0	0
3	1	0	0	0	1	3	1	1	1	0	0	3	0	0	1	0	0
4	1	1	1	1	1	4	1	1	1	0	0	4	0	0	1	0	0
5	1	0	0	0	1	5	1	0	0	0	0	5	0	0	1	0	0
6	1	0	0	0	1	6	1	1	1	1	1	6	0	1	1	1	0

	1	2	3	4	5		1	2	3	4	5	
1	1	1	1	1	1	1	1	1	0	0	0	1
2	1	0	0	0	0	1	2	1	0	0	0	1
3	1	0	0	0	0	1	3	1	0	0	0	1
4	1	0	0	0	0	1	4	1	0	0	0	1
5	1	0	0	0	0	1	5	1	0	0	0	1
6	1	1	1	1	1	1	6	1	1	1	1	1

Fuente: Autor

1.4 Estructura de datos para representar los datos de salida

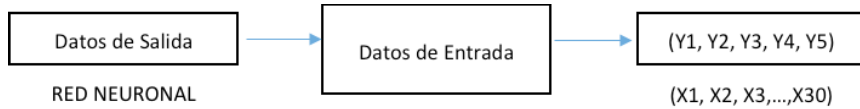
Los Datos de Salida permitirán saber si un elemento de entrada ha sido reconocido o no por nuestra RNA. Por lo tanto, como en nuestro caso solamente deseamos reconocer 5 vocales (A, E, I, O, U), entonces será suficiente un vector de dimensión 5, el cual por corresponder a los Datos de Salida lo representaremos por el vector.

$$Y = [Y_1, Y_2, Y_3, Y_4, Y_5]$$

1.5 Planteamiento del modelo de una RNA para el reconocimiento de las vocales A, E, I, O, U

De acuerdo a todo lo indicado anteriormente, el modelo funcional de la RNA para el reconocimientos de las vocales A, E, I, O y U podría quedar representado por la ilustración 6, donde los Datos de Entrada estarán representados por el vector $X = [X_1, X_2, \dots, X_{30}]$ y los Datos de Salida estarán representados por el vector $(Y = [Y_1, Y_2, Y_3, Y_4, Y_5])$

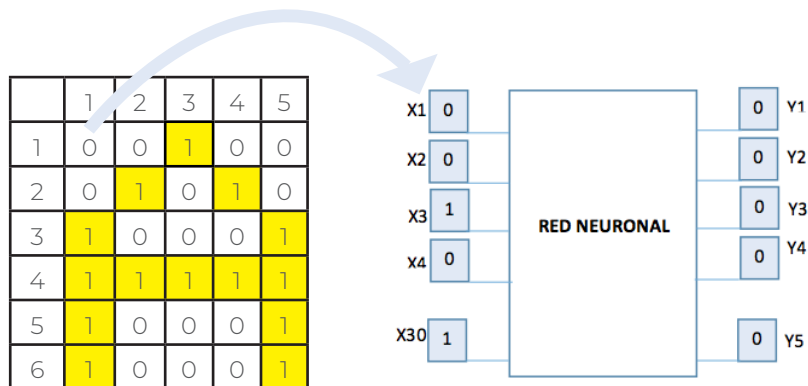
Ilustración 6: Representación de los Datos de Entrada y Salida



Fuente: Autor

Tomando como ejemplo la imagen de la vocal A mostrada en la ilustración 2, una representación gráfica más completa de los Datos de Entrada y los Datos de Salida sería la mostrada en la ilustración 7.

Ilustración 7: Representación de los datos de entrada y salida para el reconocimiento para la vocal A



Fuente: Autor

Se puede apreciar que el vector X almacena la representación gráfica bidimensional de la vocal A contenida en una matriz de 5 x 6 mientras que el vector Y tiene el valor [1, 0, 0, 0, 0] que indica que la imagen reconocida por la RNA corresponde al primer elemento en nuestro caso a la vocal A.

1.6 Proceso de entrenamiento de la RN y el reconocimiento de los caracteres numéricos 3, 4 y 5 utilizando un software

1.6.1 Preparar la Base de Conocimientos para el Entrenamiento de la RN

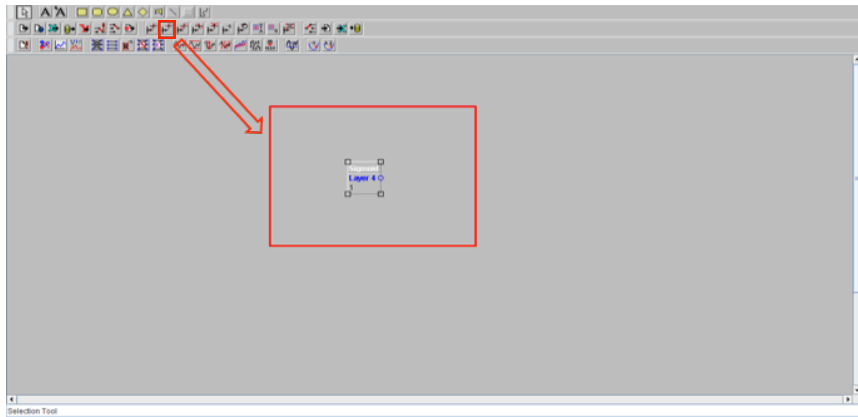
La base de conocimientos llamados también DATOS DE ENTRENAMIENTO de la RNA contiene en forma de una tabla todos los diferentes valores que se puedan dar a los vectores X e Y de modo que representen diversos modelos de los elementos que deseamos reconocer. Por ejemplo, en la ilustración 8 hemos considerado dos modelos para cada número que deseamos reconocer, utilizando dichos ejemplos, la tabla que contiene los datos de entrada X y los datos de salida Y sería la mostrada en la ilustración 8. Se puede apreciar claramente que en cada línea, el vector X representa la forma gráfica de cada número mientras que el vector Y identifica el número que se está reconociendo.

Ilustración 8: Datos de entrenamiento-Base de Conocimientos para el reconocimiento de las vocales A,E,I,O, U

	X 1	X 2	X 3	X 4	X 5	X 6	X 7	X 8	X 9	X 10	X 11	X 12	X 13	X 14	X 15	X 16	X 17	X 18	X 19	X 20	X 21	X 22	X 23	X 24	X 25	X 26	X 27	X 28	X 29	X 30	Y 1	Y 2	Y 3	Y 4	Y 5	
A	0	0	1	0	0	0	1	0	1	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	
E	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0
I	1	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	0	0	1	0	0	0
O	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0
U	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	0

Fuente: Autor

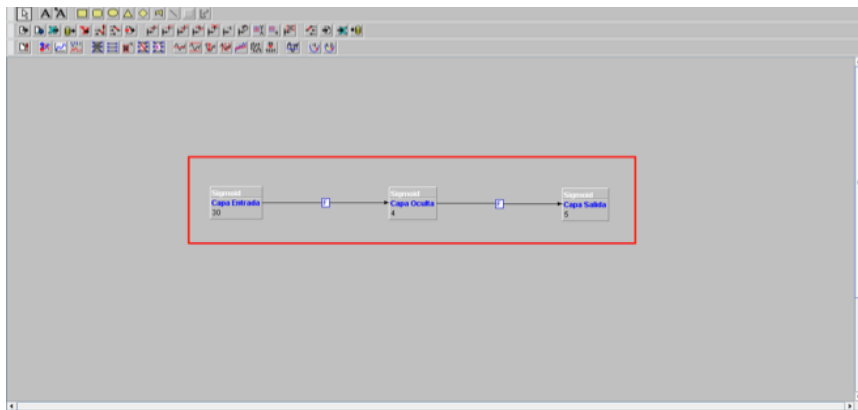
Ilustración 10: Agregación de capa sigmoid



Fuente: Autor

Seguidamente agregamos 2 capas sigmoid, quedándonos en total 3 capas, en donde establecemos las relaciones entre sí, como se muestra en la ilustración 11.

Ilustración 11: Definición de tres capas para RNA



Fuente: Autor

A continuación, definimos la capa de entrada, capa oculta y capa de salida, con sus respectivos valores de trabajo, como se muestra en la ilustración 12

Ilustración 12: Capa entrada, capa oculta y capa salida

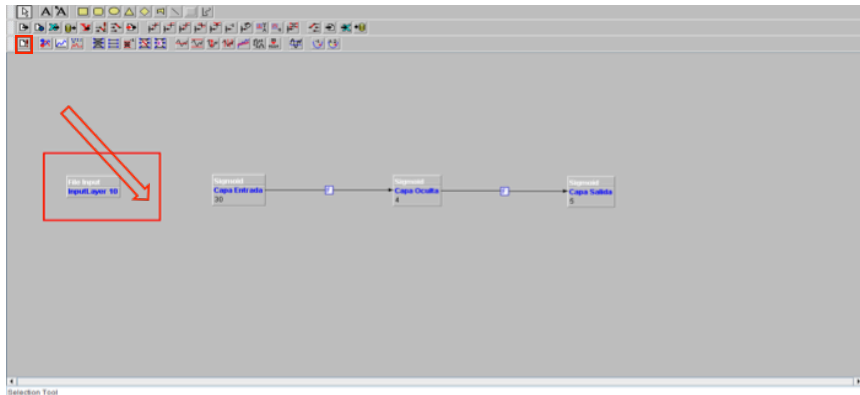
Name	Capa Entrada	Name	Capa Oculta
rows	30	rows	4

Name	Capa Salida
rows	5

Fuente: Autor

Seguidamente agregamos un nuevo archivo de entrada para nuestra RNA, como se muestra en la ilustración 13.

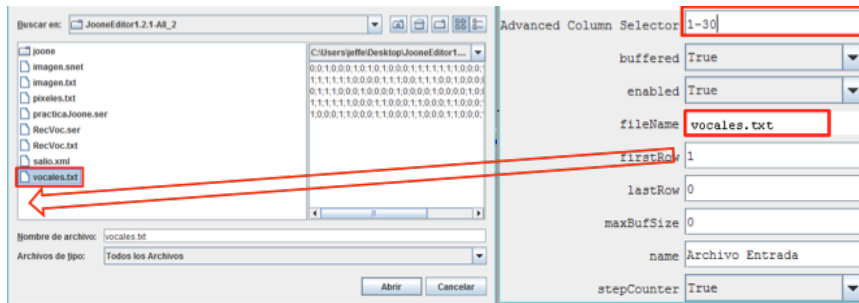
Ilustración 13: Archivo de entrada



Fuente: Autor

A continuación, elegimos el archivo que contiene nuestra base de conocimiento, además especificamos el rango de las columnas que debemos considerar como entrada, como se muestra en la ilustración 10.

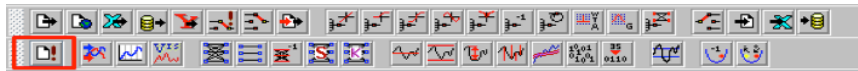
Ilustración 14: Selección de archivo base de conocimiento.



Fuente: Autor

Posteriormente se agrega un archivo de salida en donde identificamos los datos de salida para nuestra RNA, como se muestra en la ilustración 11.

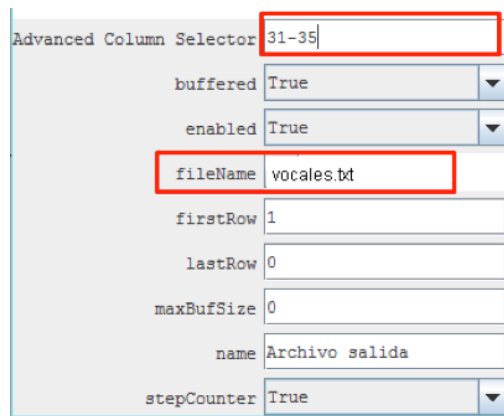
Ilustración 15: Archivo de salida para RNA



Fuente: Autor

A continuación detallamos nuestra archivo de base de conocimiento en donde detallamos la columnas que representan los datos de salida, como se muestra en la ilustración.

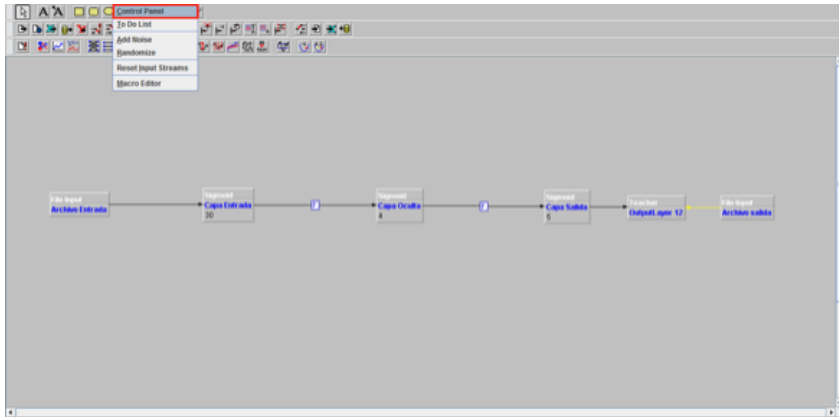
Ilustración 16: Archivo de salida RNA



Fuente: Autor

A continuación, procedemos a realizar el aprendizaje a nuestra red neuronal, para lo cual nos dirigimos a la barra de herramientas “Tools”, y escogemos la opción “Control Panel”, como se muestra en la ilustración.

Ilustración 17: Selección del panel de control



Fuente: Autor

Establecemos las siguientes configuraciones, donde iniciaremos el proceso de aprendizaje de nuestra RNA, además establecemos el número de iteraciones, en nuestro caso 1000, finalmente damos clic en “Run”, como se muestra en la siguiente ilustración.

Ilustración 18: Configuraciones para el proceso de aprendizaje

Controls	
Run	Continue Pause
Epochs: 0 RMSE: 0.0	
batchSize	0
learning	True
learningMode	0
learningRate	0.7
momentum	0.6
pre-learning cycles	0
supervised	True
epochs	1000
training patterns	5
useRMSE	True
validation	False
validation patterns	0

Fuente: Autor

Continuamos ejecutando el proceso de aprendizaje hasta que el valor del “RMSE” sea menor que “0,01”, lo cual nos indica que nuestra RNA aprendió, como se muestra en la ilustración.

Ilustración 19: Proceso de aprendizaje

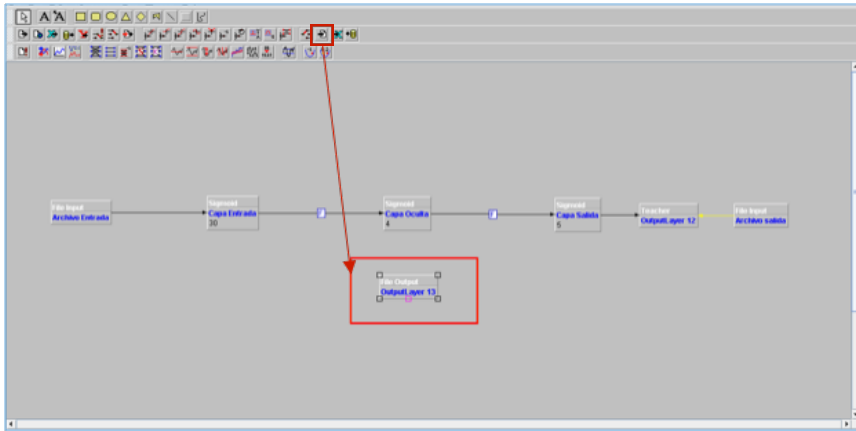
Controls	
Run	Continue Pause
Epochs: 1000 RMSE: 0.04062977239329988	
batchSize	0
learning	True
learningMode	0
learningRate	0.7
momentum	0.6
pre-learning cycles	0
supervised	True
epochs	1000
training patterns	5
useRMSE	True
validation	False
validation patterns	0

Controls	
Run	Continue Pause
Epochs: 1000 RMSE: 0.008518496393833...	
batchSize	0
learning	True
learningMode	0
learningRate	0.7
momentum	0.6
pre-learning cycles	0
supervised	True
epochs	1000
training patterns	5
useRMSE	True
validation	False
validation patterns	0

Fuente: Autor

A continuación, agregamos un archivo de salida, como se muestra en la ilustración.

Ilustración 20: Archivo de salida



Fuente: Autor

Escribimos el nombre que tendrá nuestro archivo de salida, como se muestra en la ilustración.

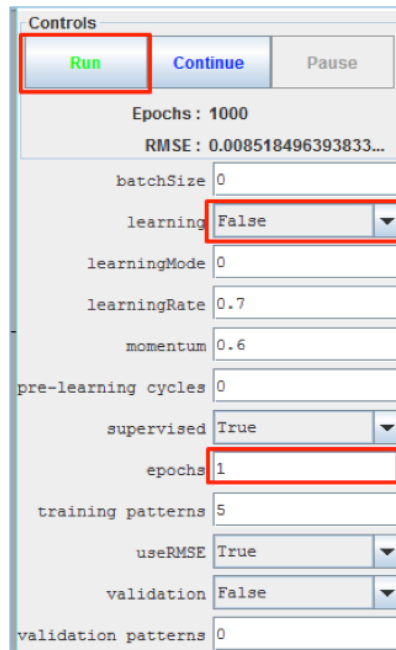
Ilustración 21: Definición de nombre archivo de salida

append	False
buffered	True
enabled	True
fileName	salida_vocales.txt
name	Salida Informacion

Fuente: Autor

Nuevamente nos dirigimos al panel de control, en donde establecemos en “False” el campo de aprendizaje, y el número de iteraciones a 1, como se muestra en la ilustración

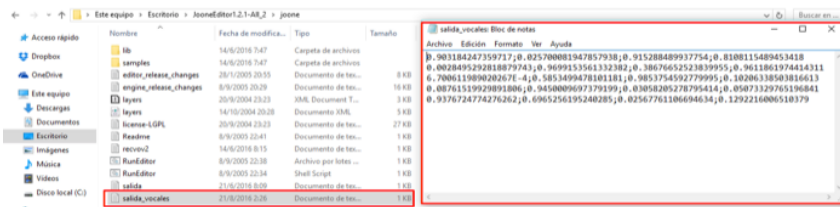
Ilustración 22: Proceso de salida de datos



Fuente: Autor

Finalmente nos genera un archivo, como se muestra en la ilustración.

Ilustración 23: Generación de archivo de salida



Fuente: Autor

1.8 Creación de interfaz para RNA

Crearemos una interfaz para nuestra RNA, utilizando Excel, previamente debemos configurar para que nos reconozca los decimales con “.” En vez decimal, debido a que Joone, devuelve valores en donde él “.” representa el decimal, de tal forma que nos quedaría tal como se muestra en la ilustración

Ilustración 24: Configuración de números

The image shows a dialog box titled 'Números' with tabs for 'Moneda', 'Hora', and 'Fecha'. Under the 'Ejemplo' section, there are two input fields: 'Positivo:' containing '123.456.789,00' and 'Negativo:' containing '-123.456.789,00'. Below this, several settings are listed as dropdown menus:

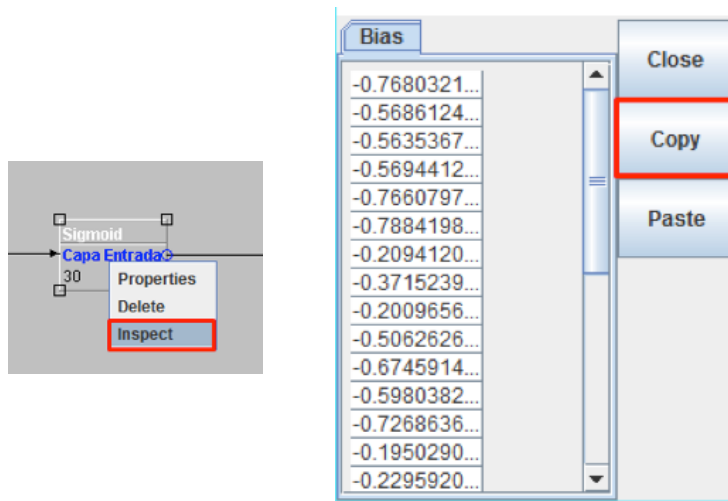
- Símbolo decimal: . (highlighted with a red box)
- Número de dígitos decimales: 2
- Símbolo de separación de miles: | (highlighted with a red box)
- Número de dígitos en grupo: 123.456.789
- Símbolo de signo negativo: =
- Formato de número negativo: -1,1
- Mostrar ceros a la izquierda: 0,7
- Separador de listas: ;
- Sistema de medida: Métrico
- Dígitos estándar: 0123456789
- Usar dígitos nativos: Nunca

At the bottom, there is a note: 'Haga clic en Restablecer para restaurar la configuración predeterminada de números, moneda, hora y fecha.' followed by a 'Restablecer' button. At the very bottom are three buttons: 'Aceptar', 'Cancelar', and 'Aplicar'.

Fuente: Autor

A continuación, creamos un archivo de Excel, donde procedemos a colocar los valores generados en el aprendizaje de nuestra RNA, empezando por la capa de entrada, damos clic derecho y seleccionamos “Inspect”, a continuación nos muestra una ventana donde se visualiza los valores de las Bías de capas de entradas, damos clic en “Copy” como se muestra en la ilustración.

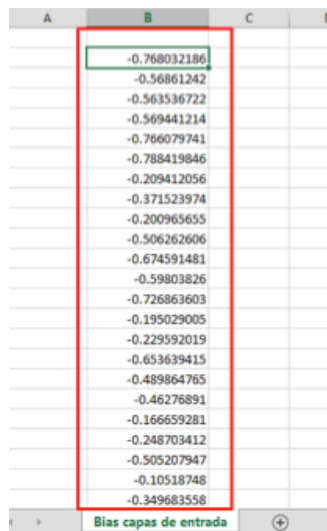
Ilustración 25: Bias de capa de entrada



Fuente: Autor

Nos dirigimos al archivo de Excel, definimos una hoja denominada "Bias capas de entrada" y pegamos los valores correspondientes, como se muestra en la ilustración 22.

Ilustración 26: Bias de capas de entrada



Fuente: Autor

A continuación, realizar el mismo proceso, como se muestra en la ilustración.

Ilustración 27: Bias de capa de entrada

The image shows a neural network configuration window. On the left, a diagram shows a 'Capa Oculta' (Hidden Layer) with 4 nodes, connected to a 'Capa Entrada' (Input Layer) with 30 nodes. A red box highlights the 'Properties' button for the hidden layer. On the right, a 'Weights' table is displayed, showing a grid of numerical values. A red box highlights the 'Copy' button in the table's interface, and a red arrow points from this button towards the right.

Weights			
0.0955136	2.1810530	-2.9113807	0.6621999
-7.1889792	3.2859035	-0.3104014	-1.8493560
-6.2663789	1.4395611	1.1788427	-0.6817686
-7.2025393	3.5028118	-0.1335322	-1.6431727
0.1163106	2.2069795	-2.9552358	0.7273540
-0.1087234	2.3970353	-2.6851017	0.5791115
1.4456359	-2.4007325	1.8022606	0.7009930
-0.6238603	-1.5846341	2.6393635	-2.0525608
1.5299448	-2.1396459	1.7982120	0.6802822
1.4995130	0.7091749	-2.3840447	-3.6200843
1.3322478	0.6467342	-1.4963491	1.5990513
-1.2156092	1.1910515	0.1369734	3.9174694
-2.1321783	0.2673702	2.1369827	2.3307445
0.3064483	-0.5590411	0.4079338	-0.2606341
2.9403843	-1.4887771	-0.8980500	-2.6413316
1.2801213	0.6100856	-1.5104145	1.4789591
0.0052789	-0.8507583	1.3819289	4.4552925
-1.0475909	-1.7805112	3.6568971	2.9818308
1.4211303	-2.0560747	2.0156871	0.6993677
2.7913887	-1.4139469	-0.9221289	-2.7104044
1.2175036	0.2955890	-1.0708414	1.4288684
0.1197623	-0.7173114	0.3869347	-0.3737985
-0.3605743	-1.5022899	2.6974148	-2.0243089
0.1471638	-0.6539085	0.5534681	-0.2000973
2.8999719	-1.4254810	-0.7831133	-2.5735590
1.1660082	0.6300199	-1.4410246	1.6248189
-0.9925034	1.3319910	-0.6695016	-1.0901345
-0.7879675	1.0183812	-0.6565106	-1.3160346
-0.9631244	1.1819339	-0.8140737	-0.9519210
1.0742500	0.3264618	-1.3873869	1.5267742

Fuente: Autor

Definimos una hoja denominada “BiasCapasEntrada” y pegamos los valores correspondientes, como se muestra en la siguiente ilustración.

Definimos una hoja denominada “BiasCapasEntrada” y pegamos los valores correspondientes, como se muestra en la siguiente ilustración.

Ilustración 28: Bias Capas de entrada

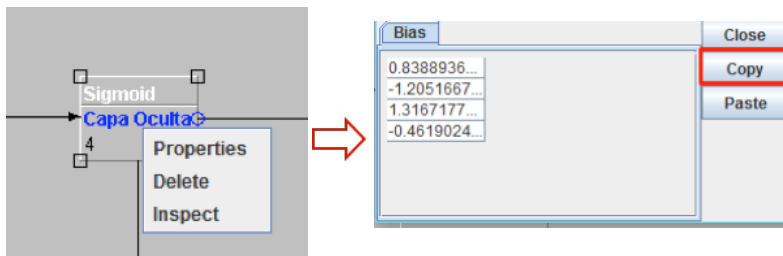
0.095513675	2.181053003	-2.91138079	0.662199906
-7.188979221	3.285903515	-0.310401419	-1.849356095
-6.26637694	1.439561146	1.178842772	-0.681768606
-7.202539393	3.50281185	-0.133532289	-1.643172777
0.116310665	2.206979586	-2.955235896	0.727354061
-0.108723433	2.397035374	-2.685101727	0.57911154
1.445535963	-2.400732501	1.802260614	0.700993024
-0.623860376	-1.584634199	2.63936354	-2.052560829
1.529944888	-2.13964597	1.798212059	0.680282299
1.49951307	0.709174913	-2.384044718	-3.620084384
1.332247895	0.646734265	-1.496349121	1.599051319
-1.215609206	1.191051541	0.136973429	3.917469444
-2.132178382	0.267370269	2.136982785	2.330744561
0.306448359	-0.559041106	0.407933883	-0.260634151
2.940384347	-1.488777176	-0.898050083	-2.641331671
1.280121345	0.610085614	-1.510414578	1.478959157
0.005276938	-0.650758308	1.381928965	4.456292585
-1.047590919	-1.78051124	3.656697159	2.981830896
1.421130365	-2.056074702	2.015687108	0.69936773
2.791388789	-1.413946907	-0.922128965	-2.710404416
1.217503667	0.295589094	-1.070841475	1.426868427
0.119762306	-0.717311449	0.386934726	-0.373798534
-0.360574384	-1.502289998	2.697414816	-2.024308967

Bias capas de entrada Hoja2 +

Fuente: Autor

Continuamos con el mismo procedimiento, para la capa oculta, como se muestra en la siguiente ilustración.

Ilustración 29: Valores Capa de salida



Fuente: Autor

Definimos una hoja denominada “Bias capa oculta” y pegamos los valores correspondientes, como se muestra en la siguiente ilustración.

Ilustración 30: Bias de capa oculta

B
0.838893688
-1.205166705
1.316717756
-0.461902442

Fuente: Autor

Continuamos, con la arista entre la capa oculta y capa de salida, como se muestra en la siguiente ilustración.

Ilustración 31: Bias de capa de oculta

Weights				Close
4.2904188...	-8.7002591...	-8.4272895...	-8.7518589...	11.113989...
-10.883338...	1.7201547...	-2.1657035...	6.0269365...	-0.9739274...
2.9680666...	-1.4214019...	9.1793096...	-8.4636389...	-9.1249120...
0.9361960...	10.364748...	-4.6157263...	-8.4289770...	-3.0848401...

Fuente: Autor

Definimos una hoja denominada “Bias capa oculta traspuesta” y pegamos los valores correspondientes, de estos valores, extraemos su traspuesta, como se muestra en la siguiente ilustración.

Ilustración 32: Traspuesta Bias capas ocultas

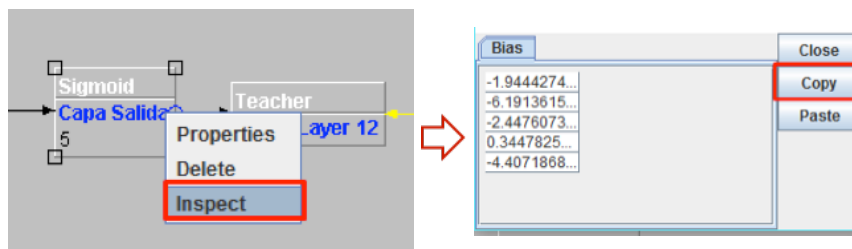
4.29041889	-8.70025912	-8.42728959	-8.75185893	11.1139896
-10.8833384	1.72015473	-2.16570351	6.02693654	-0.97392745
2.96806669	-1.42140193	9.17930962	-8.46363895	-9.124912
0.93619607	10.3647489	-4.61572634	-8.42897703	-3.08484015

4.29041889	-10.8833384	2.96806669	0.93619607
-8.70025912	1.72015473	-1.42140193	10.3647489
-8.42728959	Opciones de pegado:		
-8.75185893	123	123	51572634
11.1139896	-0.97392745	9.124912	42897703
			-3.08484015

Fuente: Autor

Finalmente realizamos con las capas de salida, como se muestra en la siguiente ilustración.

Ilustración 33: Bias capa de salida



Fuente: Autor

Definimos una hoja denominada “Bias capa salida” y pegamos los valores correspondientes, como se muestra en la siguiente ilustración.

Ilustración 34: Hoja Bias capas salida

-1.944427428
-6.191361583
-2.447607355
0.344782561
-4.407186811

Fuente: Autor

A continuación, definimos una nueva hoja la cual denominaremos principal, es donde ubicaremos nuestra interfaz para nuestro modelo de RNA, para poner en práctica el reconocimiento de las vocales, para lo cual definimos nuestra matriz que representa la estructura de las vocales, como se muestra en la ilustración 30.

Ilustración 35: Matriz estructurada de las vocales

A	B	C	D	E	F	G	H	I	J
MATRIZ							RESULTADO		
		1	2	3	4	5			
1		0	0	0	0	0	A		
2		0	0	0	0	0	E		
3		0	0	0	0	0	I		
4		0	0	0	0	0	O		
5		0	0	0	0	0	U		
6		0	0	0	0	0			

Fuente: Autor

Seguidamente traspasamos los valores de la capa de entrada, como se muestra en la siguiente ilustración.

Ilustración 36: Valores de la capas de entradas

B	C
-0.768032186	
-0.56861242	
-0.563536722	
-0.569441214	
-0.766079741	
-0.788419846	
-0.209412056	
-0.371523974	
-0.200965655	
-0.506262606	
-0.674591481	
-0.59803826	
-0.726863603	
-0.195029005	
-0.229592019	
-0.653639415	
-0.489864765	
-0.46276891	
-0.166659281	
-0.248703412	
-0.505207947	
-0.10518748	
-0.349683558	
-0.211357127	
-0.223684916	
-0.65306437	
-0.926349918	
-0.753240509	
-0.916437179	
-0.592555562	

L	M	N	O	P
Bias de Capa de Entrada				
-0.76803219	-0.56861242	-0.56353672	-0.56944121	-0.76607974
-0.78841985	-0.20941206	-0.37152397	-0.20096566	-0.50626261
-0.67459148	-0.59803826	-0.7268636	-0.19502901	-0.22959202
-0.65363941	-0.48986477	-0.46276891	-0.16665928	-0.24870341
-0.50520795	-0.10518748	-0.34968356	-0.21135713	-0.22368492
-0.65306437	-0.92634992	-0.75324051	-0.91643718	-0.59255556

Fuente: Autor

	A	B	C	D	E
1		0.095513675	2.181053003	-2.91138079	0.662199906
2		-7.18897922	3.285903515	-0.31040142	-1.8493561
3		-6.26637694	1.439561146	1.178842772	-0.68176861
4		-7.20253939	3.50281185	-0.13353229	-1.64317278
5		0.116310665	2.206979586	-2.9552359	0.727354061
6		-0.10872343	2.397035374	-2.68510173	0.57911154
7		1.445535963	-2.4007325	1.802260614	0.700993024
8		-0.62386038	-1.5846342	2.63936354	-2.05256083
9		1.529944888	-2.13964597	1.798212059	0.680282299
10		1.49951307	0.709174913	-2.38404472	-3.62008438
11		1.332247895	0.646734265	-1.49634912	1.599051319
12		-1.21560921	1.191051541	0.136973429	3.917468444
13		-2.13217838	0.267370269	2.136982785	2.330744561
14		0.306448359	-0.55904111	0.407933883	-0.26063415
15		2.940384347	-1.48877718	-0.89805008	-2.64133167
16		1.280121345	0.610085614	-1.51041458	1.478959157
17		0.005276938	-0.65075831	1.381928965	4.456292585
18		-1.04759092	-1.78051124	3.656697159	-2.981830896
19		1.421130365	-2.0560747	2.015687108	0.69926773
20		2.791388789	-1.41394691	-0.92212896	-2.71040442
21		1.217503667	0.295589057	-1.07084147	1.426868427
22		0.119762306	-0.71731145	0.386934726	-0.37379853
23		-0.36057438	-1.50229	2.697414816	-0.02430897
24		0.147163872	-0.65390857	0.553468121	-0.20009739
25		2.899971984	-1.42548105	-0.78311332	-2.5735591
26		1.166008216	0.630019949	-1.4410247	1.624818931
27		-0.99250345	1.331991067	-0.66950165	-1.09013452
28		-0.78796751	1.018381217	-0.65651063	-1.31603462
29		-0.9631244	1.181933926	-0.81407375	-0.95192106
30		1.074250001	0.32646181	-1.38738691	1.526774212

Peso para la Neurona 2 de la Capa Oculta				
2.181053	3.28590351	1.43956115	3.50281185	2.20697959
2.39703537	-2.4007325	-1.5846342	-2.13964597	0.70917491
0.64673427	1.19105154	0.26737027	-0.55904111	-1.48877718
0.61008561	-0.65075831	-1.78051124	-2.0560747	-1.41394691
0.29558909	-0.71731145	-1.50229	-0.65390857	-1.42548105
0.63001995	1.33199107	1.01838122	1.18193393	0.32646181

	A	B	C	D	E	F
1		0.095513675	2.181053003	-2.91138079	0.662199906	
2		-7.18897922	3.285903515	-0.31040142	-1.8493561	
3		-6.26637694	1.439561146	1.178842772	-0.68176861	
4		-7.20253939	3.50281185	-0.13353229	-1.64317278	
5		0.116310665	2.206979586	-2.9552359	0.727354061	
6		-0.10872343	2.397035374	-2.68510173	0.57911154	
7		1.445535963	-2.4007325	1.802260614	0.700993024	
8		-0.62386038	-1.5846342	2.63936354	-2.05256083	
9		1.529944888	-2.13964597	1.798212059	0.680282299	
10		1.49951307	0.709174913	-2.38404472	-3.62008438	
11		1.332247895	0.646734265	-1.49634912	1.599051319	
12		-1.21560921	1.191051541	0.136973429	3.917468444	
13		-2.13217838	0.267370269	2.136982785	2.330744561	
14		0.306448359	-0.55904111	0.407933883	-0.26063415	
15		2.940384347	-1.48877718	-0.89805008	-2.64133167	
16		1.280121345	0.610085614	-1.51041458	1.478959157	
17		0.005276938	-0.65075831	1.381928965	4.456292585	
18		-1.04759092	-1.78051124	3.656697159	-2.981830896	
19		1.421130365	-2.0560747	2.015687108	0.69926773	
20		2.791388789	-1.41394691	-0.92212896	-2.71040442	
21		1.217503667	0.295589057	-1.07084147	1.426868427	
22		0.119762306	-0.71731145	0.386934726	-0.37379853	
23		-0.36057438	-1.50229	2.697414816	-0.02430897	
24		0.147163872	-0.65390857	0.553468121	-0.20009739	
25		2.899971984	-1.42548105	-0.78311332	-2.5735591	
26		1.166008216	0.630019949	-1.4410247	1.624818931	
27		-0.99250345	1.331991067	-0.66950165	-1.09013452	
28		-0.78796751	1.018381217	-0.65651063	-1.31603462	
29		-0.9631244	1.181933926	-0.81407375	-0.95192106	
30		1.074250001	0.32646181	-1.38738691	1.526774212	

Peso para la Neurona 3 de la Capa Oculta				
-2.91138079	-0.31040142	1.17884277	-0.13353229	-2.9552359
-2.68510173	1.80226061	2.63936354	1.79821206	-2.38404472
-1.49634912	0.13697343	2.13698278	0.40793388	-0.89805008
-1.51041458	1.38192896	3.65669716	2.01568711	-0.92212896
-1.07084147	0.38693473	2.69741482	0.55346812	-0.78311332
-1.4410247	-0.66950165	-0.65651063	-0.81407375	-1.38738691

B	C	D	E	F
0.095513675	2.181053003	-2.91138079	0.662199906	
-7.18897922	3.285903515	-0.31040142	-1.8493561	
-6.26637694	1.439561146	1.178842772	-0.68176861	
-7.20253939	3.50281185	-0.13353229	-1.64317278	
0.116310665	2.206979586	-2.9552359	0.72735406	
-0.10872343	2.397035374	-2.68510173	0.57911154	
1.445535963	-2.4007325	1.802260614	0.700993024	
-0.62386038	-1.5846342	2.63936354	-2.05256083	
1.529944888	-2.13964597	1.798212059	0.680282299	
1.49951307	0.709174913	-2.38404472	-3.62008438	
1.332247895	0.646734265	-1.49634912	1.599051319	
-1.21560921	1.191051541	0.136973429	3.917469444	
-2.13217838	0.267370269	2.136982785	2.330744561	
0.306448359	-0.55904111	0.407933883	-0.26063415	
2.940384347	-1.48877718	-0.89805008	-2.64133167	
1.280121345	0.610085614	-1.51041458	1.478959157	
0.005276938	-0.65075831	1.381928965	4.456292585	
-1.04759092	-1.78051124	3.656697159	2.981830896	
1.421130365	-2.0560747	2.015687108	0.69936773	
2.791388789	-1.41394691	-0.92212896	-2.71040442	
1.217503667	0.295589094	-1.07084147	1.426868427	
0.119762306	-0.71731145	0.386934726	-0.37379853	
-0.36057438	-1.50229	2.697418416	-2.02430897	
0.147163872	-0.65390857	0.553468121	-0.20009779	
2.899971984	-1.42548105	-0.78311332	-2.5735491	
1.166008216	0.630019949	-1.4410247	1.624816931	
-0.99250345	1.331991067	-0.66950165	-1.09013453	
-0.78796751	1.018381217	-0.65651063	-1.31603442	
-0.9631244	1.181933926	-0.81407375	-0.95192706	
1.074250001	0.32646181	-1.38738691	1.526794212	

Peso para la Neurona 4 de la Capa Oculta				
0.66219991	-1.8493561	-0.68176861	-1.64317278	0.72735406
0.57911154	0.70099302	-2.05256083	0.6802823	-3.62008438
1.59905132	3.91746944	2.33074456	-0.26063415	-2.64133167
1.47895916	4.45629258	2.9818309	0.69936773	-2.71040442
1.42686843	-0.37379853	-2.02430897	-0.20009739	-2.5735591
1.62481893	-1.09013452	-1.31603462	-0.95192106	1.52677421

Fuente: Autor

Posteriormente realizamos una multiplicación entre las tablas creadas anteriormente con la tabla transferencias entre entrada y capa de entrada, seguidamente realizamos la suma de los valores de las filas, y después sumamos estos valores, como se muestra en la ilustración 34, el mismo procedimiento lo realizamos con cada una de las demás tablas.

Ilustración 39: Sumatoria entre la transferencia y peso 1

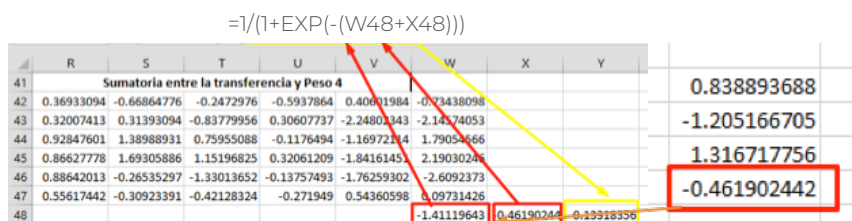
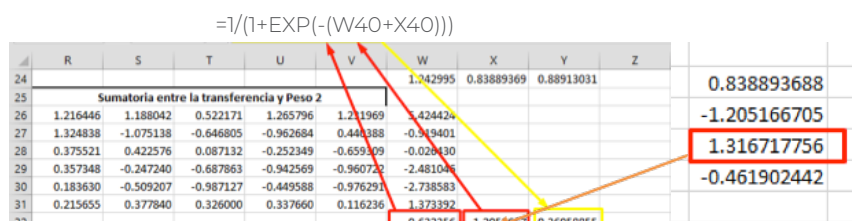
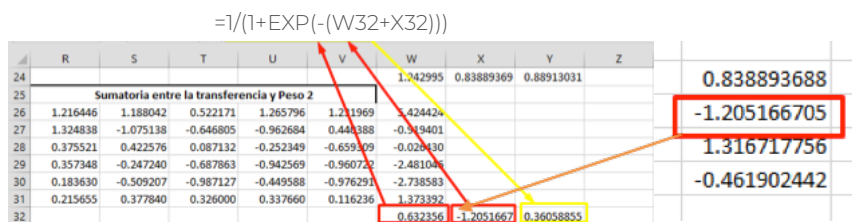
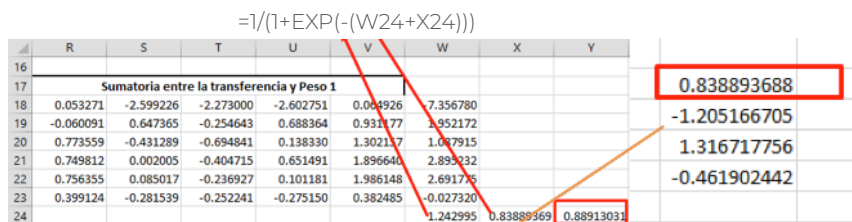
$$=L10*L18$$

K	L	M	N	O	P	Q	R	S	T	U	V	W
	-0.65306477	-0.57634992	-0.75324051	-0.91643718	-0.59255556							
	Transferencia entre entrada y Capa de Entrada											
	0.557733	0.361557	0.362730	0.361366	0.558215							
	0.552699	0.447837	0.408173	0.449927	0.620986							
	0.580642	0.354793	0.325883	0.451397	0.442853							
	0.585735	0.379925	0.386329	0.458431	0.679461							
	0.621235	0.709882	0.657082	0.687540	0.684885							
	0.342299	0.283666	0.320116	0.285684	0.356049							
	Peso para la Neurona 1 de la Capa Oculta											
	0.09551367	-7.18897922	-6.26637694	-7.20253939	0.11631066							
	-0.10872343	1.44553596	-0.62386038	1.52994489	1.49951307							
	1.3322479	-1.21560921	-2.13217838	0.30644836	2.94038435							
	1.28012135	0.00527694	-1.04759092	1.42113036	2.79138879							
	1.21750367	0.11976231	-0.36057438	0.14716387	2.89997198							
	1.16600822	-0.99250345	-0.78796751	-0.9631244	1.07425							
	Sumatoria entre la transferencia y Peso 1											
	0.053271	-2.599226	-2.273000	-2.602751	0.064926						-7.356780	
	-0.060091	0.647365	-0.254643	0.688364	0.931177						1.952172	
	0.773559	-0.431289	-0.694841	0.138330	1.302157						1.087915	
	0.749812	0.002005	-0.404715	0.651491	1.896640						2.895232	
	0.756355	0.085017	-0.236927	0.101181	1.986148						2.691775	
	0.399124	-0.281539	-0.252241	-0.275150	0.382485						-0.027320	
											1.242995	

Fuente: Autor

Seguidamente aplicamos la función simoidal entre el valor de la suma obtenida anteriormente con los valores de la capa oculta, como se muestra en la siguiente ilustración.

Ilustración 40: Función simoidal



Fuente: Autor

Posteriormente realizamos una multiplicación de los 4 valores obtenidos anteriormente con los valores de las columnas respectivamente, como se muestra en la ilustración.

Ilustración 41: Proceso de multiplicación

$$=L51*Y\$24$$

	L	M	N	O
49				
50	Pesos de Capa Oculta y Capa de Salida			
51	4.29041885	-10.8833384	2.96806669	0.93619607
52	-8.70025912	1.72015473	-1.42140193	10.3647489
53	-8.42728959	-2.16570351	9.17930962	-4.61572634
54	-8.75185893	6.02693654	-8.46363895	-8.42897703
55	11.1139896	-0.97392745	-9.124912	-3.08484015
56				
57	3.81474147	-3.9244072	0.61525783	0.12468593
58	-7.73566407	0.6202681	-0.2946459	1.3804142
59	-7.49295859	-0.78092789	1.90280163	-0.61473888
60	-7.78154303	2.1732443	-1.7544485	-1.1226012
61	9.88178498	-0.35118708	-1.89152542	-0.41085001

Fuente: Autor

Finalmente realizamos la suma de los valores de la fila y aplicamos la función simoidal con los valores de la capa de salida, después asociamos las respuestas con nuestra tabla resultado, por ende, si formamos una vocal dentro de la matriz, nuestro modelado, reconocer el patrón e identificara de que vocal se asemeja, como se muestra en la ilustración.

Ilustración 42: Reconocimiento de vocales

$$1/(1+\text{EXP}(-(Q57+R57)))$$

	L	M	N	O	P	Q	R	S	T
56						Sumatoria	Bias C.Salida	Funcion Simoidal	
57	3.81474147	-3.9244072	0.61525783	0.12468593		0.63027803	-1.94442743	0.21179333	
58	-7.73566407	0.6202681	-0.2946459	1.3804142		-6.02962767	-6.19136158	4.9259E-06	
59	-7.49295859	-0.78092789	1.90280163	-0.61473888		-6.98582373	-2.44760736	7.9998E-05	
60	-7.78154303	2.1732443	-1.7544485	-1.1226012		-8.48534844	0.34478256	0.00029139	
61	9.88178498	-0.35118708	-1.89152542	-0.41085001		7.22822247	-4.40718681	0.94380202	
62									

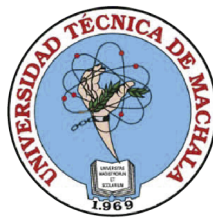
	A	B	C	D	E	F	G	H	I	J
1		MATRIZ						RESULTADO		
2			1	2	3	4	5			
3		1	1	0	0	0	1		A	0
4		2	1	0	0	0	1		E	0
5		3	1	0	0	0	1		I	0
6		4	1	0	0	0	1		O	0
7		5	1	1	1	1	1		U	1
8										

Fuente: Autor

*Redes neuronales artificiales aplicadas al
reconocimiento de patrones*
Edición digital 2017-2018.
www.utmachala.edu.ec

Redes

Redes es la materialización del diálogo académico y propositivo entre investigadores de la UTMACH y de otras universidades iberoamericanas, que busca ofrecer respuestas glocalizadas a los requerimientos sociales y científicos. Los diversos textos de esta colección, tienen un espíritu crítico, constructivo y colaborativo. Ellos plasman alternativas novedosas para resignificar la pertinencia de nuestra investigación. Desde las ciencias experimentales hasta las artes y humanidades, Redes sintetiza policromías conceptuales que nos recuerdan, de forma empeñosa, la complejidad de los objetos construidos y la creatividad de sus autores para tratar temas de acalorada actualidad y de demanda creciente; por ello, cada interrogante y respuesta que se encierra en estas líneas, forman una trama que, sin lugar a dudas, inervará su sistema cognitivo, convirtiéndolo en un nodo de esta urdimbre de saberes.



UNIVERSIDAD TÉCNICA DE MACHALA
Editorial UTMACH
Km. 5 1/2 Vía Machala Pasaje
www.investigacion.utmachala.edu.ec / www.utmachala.edu.ec

ISBN: 978-9942-24-100-9

