

Inteligencia Artificial II: Redes Neuronales Artificiales (RNA)

(FUNDAMENTOS, RNA CON CONEXIÓN HACIA ADELANTE, OTROS MODELOS DE
RNA)

∞ cognitio ∞

V. Robles-Bykbaev (vrobles@ups.edu.ec)

<http://catedraunescoinclusion.org/>

<http://giata.blog.ups.edu.ec/>

**Universidad Politécnica Salesiana,
Sede Matriz Cuenca, 2019**



Contenidos

1 Computación Neuronal

2 Modelo biológico

3 Red Neuronal Artificial

Contenidos

1 Computación Neuronal

2 Modelo biológico

3 Red Neuronal Artificial

Introducción

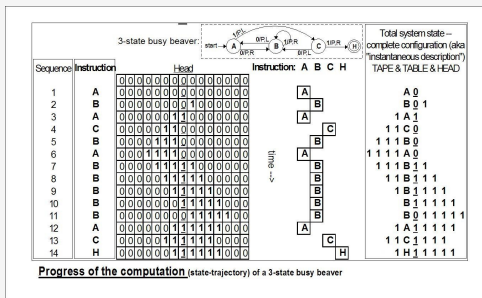
A pesar de que el concepto de Redes Neuronales Artificiales (RNN) se remonta hace más de 50 años atrás, es importante conocer algunos aspectos relacionados con su panorama histórico [Nunes da Silva et al., 2017]:

- Construir una máquina o un mecanismo autónomo dotado de inteligencia ha sido uno de los sueños de los investigadores de diversas áreas de la ingeniería.
- Las primeras explicaciones sobre el cerebro y su funcionamiento fueron propuestas por filósofos griegos (Platón y Aristóteles) [Clarke, 1963].
- Alan Turing, en 1936, fue uno de los primeros científicos en estudiar el cerebro como una forma de ver el mundo de la computación, donde destaca su más importante contribución: *Los números computables, con una aplicación al Entscheidungsproblem* [Copeland, 2004].

Introducción (cont'd)

- Sin embargo, es fundamental tener en cuenta que el cerebro se considera una máquina perfecta: *la capacidad del cerebro humano para pensar, recordar y resolver problemas ha sido fuente de inspiración para muchos científicos* [Muñoz Pérez, 2012]

Figure: Máquina de Turing, fuente: Wikimedia



Una breve introducción sobre los orígenes RNA

A continuación se describen algunos hechos interesantes sobre los orígenes de las RNA [Demuth et al., 2014]:

Inicio de las RNA

- La visión moderna de las RNA inicia en los 40'
- Warren McCulloch y Walter Pitts demuestran que una RNA puede, en principio, calcular cualquier función aritmética o lógica.
- El trabajo de McCulloch y Pitts se considera el origen de las RNA

Aporte de Donal Hebb en 1949

- Propone que el condicionamiento clásico (descubierto por Pavlov) está presente por las propiedades individuales de las neuronas
- Propuso un mecanismo de aprendizaje para las neuronas biológicas

Una breve introducción sobre los orígenes RNA (cont'd)

Primera aplicación práctica de una RNA (década de 1950)

- Se produce cuando Frank Rosenblatt inventa la RNA **Perceptrón** y su **regla de asociación**
- Rosenblatt y sus colegas construyeron y demostraron que el perceptrón podía realizar tareas de reconocimiento de patrones
- Lamentablemente se demostró luego que el perceptrón solo podía resolver ciertos problemas

Aporte de Bernard Widrow y Ted Hoff (1960)

- Proponen un nuevo algoritmo y lo emplean para entrenar redes neuronales lineales adaptativas
- Su regla aún se sigue usando hoy en día

Una breve introducción sobre los orígenes RNA (cont'd)

Limitaciones de las RNA propuestas

- Marvin Minsky y Seymour Papert demuestran en 1969 que las redes propuestas por Rosenblatt y Widrow tenían las mismas limitaciones
- Rosenblatt y Widrow propusieron nuevas redes para superar esas limitaciones, sin embargo, no fueron capaces de modificar con éxito sus algoritmos para entrenar redes más complejas

Década perdida

- La influencia de Minsky y Papert y la falta de computadoras potentes provocaron más de una década en la que se perdió el interés por las RNA
- Durante este periodo se pensó que era el fin de las RNA

Una breve introducción sobre los orígenes RNA (cont'd)

Otros aportes importantes (década de los 70s)

- En 1972 Tuvo Kohonen y James Anderson (de forma independiente) desarrollaron una RNA que podía actuar como una memoria.
- Stephen Grossberg investigó activamente en 1976 las redes auto-organizadas.
- Sin embargo, es importante mencionar que en los 60s decayó el interés por las RNA debido a la falta de capacidad de cómputo. Esta situación se revertió en los 80s, con la llegada de nuevas computadoras.

Una breve introducción sobre los orígenes RNA (cont'd)

Dos nuevos conceptos que ayudan al renacimiento de las RNA (década de los 80s)

- 1 El físico John Hopfield usa mecanismos estadísticos para explicar la operación de cierto tipo de RNA recurrente, que podía usarse como memoria asociativa.
- 2 Creación del algoritmo de **retropropagación de error** por parte de diversos investigadores (de forma independiente).

El artículo que mayor influencia tuvo del algoritmo de retropropagación fue el publicado por David Rumelhart y James McClelland (1986) y constituía una respuesta a el estudio crítico planteado en los 60s por Minsky y Papert.

Una breve introducción sobre los orígenes RNA (cont'd)

En años recientes se han logrado importantes avances en el ámbito de las RNA [Nunes da Silva et al., 2017]:

- Hagan y Menhaj (1994) desarrollaron un interesante algoritmo de aprendizaje basado en el método **Levenberg-Marquart**, que ha demostrado su eficiencia en diversas aplicaciones.
- Las RNA basadas en **Máquinas de Soporte Vectorial** que han sido usadas para tareas de regresión lineal y clasificación (Vapnik, 1998)
- Desarrollo de circuitos integrados neuronales (Beiu et al., 2003)

Ejercicio en clase No 001

Enunciado

- Objetivo: *entender el problema de la decisión, sus implicaciones, elementos, estructuras y propuestas existentes para su solución*
- Desarrolle un cuadro sinóptico, un mapa mental o una estructura similar del problema **Entscheidungsproblem**
- El cuadro deberá indicar los elementos más importantes del problema como: descripción, estructura, fundamento (lógica de primer orden), etc.
- Cada elemento del cuadro deberá estar acompañado con una referencia a un paper, libro o algún documento científico

Ejercicio en clase No 001 (cont'd)

Instrucciones de entrega

- El trabajo es individual y puede emplearse cualquier herramienta (por ejemplo: Wise Mapping)
- Debe contener la revisión de al menos 5 papers (de cualquier fecha)
- La entrega se realizará en el AVAC (Ejercicio en Clase 1 _3-4-2018_: Problema Entscheidungsproblem)
- El archivo a cargar deberá ser guardado como una imagen (PNG, JPG, etc) y deberá subirse hasta las 19H00

Contenidos

1 Computación Neuronal

2 Modelo biológico

3 Red Neuronal Artificial

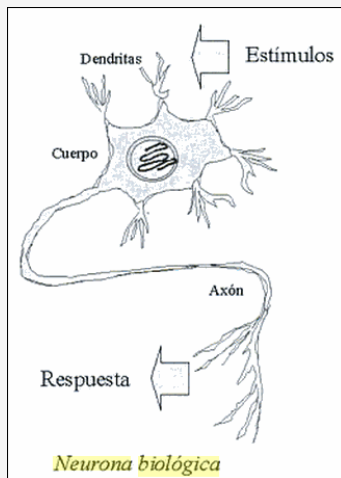
Introducción

El cerebro del ser humano es una máquina con excepcionales características [Nunes da Silva et al., 2017]:

- Contiene un gran número de elementos (10^{11}) altamente interconectado (aproximadamente 10^4) conexiones por cada elemento. Dichos elementos se conocen con el nombre de **neuronas**.
- Fundamentalmente, las neuronas poseen tres elementos principales:
 - 1 *Dendritas*: constituyen el canal por el cual entra la información a la neurona.
 - 2 *Soma*: órgano de cómputo central. También se conoce como *cuerpo celular*.
 - 3 *Axón*: canal de salida de la neurona.

Introducción (cont'd)

Figure: Modelo de Neurona Biológica [Banchs, 2005]

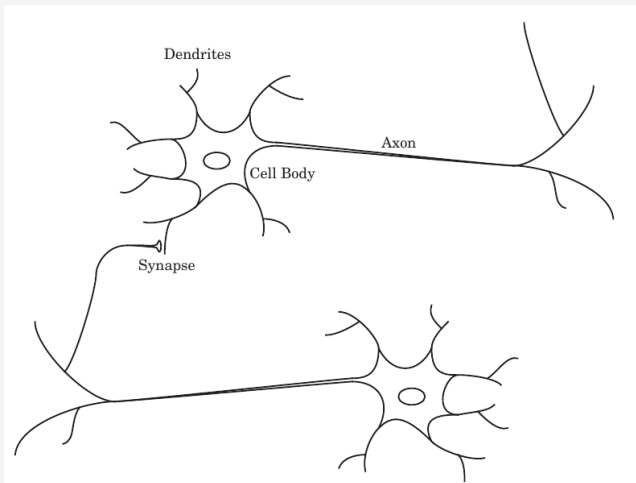


Introducción (cont'd)

- Las dendritas son como redes receptivas de fibras nerviosas que transportan señales eléctricas al cuerpo celular.
- El cuerpo celular o soma suma las señales y las limita.
- El axón es una fibra larga que lleva la señal del cuerpo celular de la neurona fuera, hacia otras neuronas.
- El punto de contacto entre el axón y de una neurona y la dendrita de otra se conoce como **sinapsis**

Introducción (cont'd)

Figure: Sinapsis entre 2 neuronas [Nunes da Silva et al., 2017]



Introducción (cont'd)

- Las **sinapsis alteran la efectividad** con que la señal pasa de una neurona a otra (a través de un parámetro: **peso**).
- Con ello, es el el arreglo de neuronas y los pesos de las sinapsis individuales (determinadas por un complejo proceso químico) lo que establece la función de la red neuronal.
- Es importante destacar que una parte de la estructura neuronal es definida al nacer. Otras partes **se desarrollan a través del aprendizaje**, como la creación nuevas conexiones o la eliminación de otras tantas. Este desarrollo es mucho más significativo en etapas tempranas de la vida.

Introducción (cont'd)

- Algunos datos interesantes señalan los siguientes [Nunes da Silva et al., 2017]:
 - Los lingüistas han determinado que aquellos infantes mayores de 6 meses ya no podrán discriminar ciertos sonidos si no han sido expuestos a ellos antes.
 - Los neurocientíficos han descubierto que el hipocampo de los conductores de taxi de Londres son significativamente más grandes que del promedio de gente. Esto se debe a que deben memorizar grandes volúmenes de datos de navegación (miles de calles) a través de un proceso que toma 2 años.

Contenidos

1 Computación Neuronal

2 Modelo biológico

3 Red Neuronal Artificial

Esquema de una RNA

Perceptrón simple

Las estructuras de una red neuronal artificial se diseñaron a partir de los modelos del sistema nervioso central y el cerebro humano. Algunas de las características más importantes de los mismos son las que detallan a continuación [Nunes da Silva et al., 2017]:

- Las neuronas artificiales (componentes computacionales o unidades de procesamiento) constituyen modelos simplificados de una neurona biológica.
- Estos modelos fueron inspirados por el análisis sobre cómo el cuerpo celular o soma de una neurona genera y propaga impulsos eléctricos.

Esquema de una RNA (cont'd)

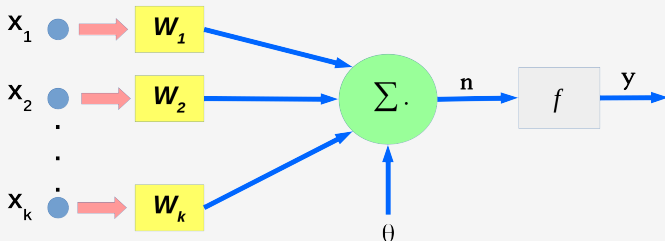
Perceptrón simple

- Las neuronas artificiales usadas en RNA se caracterizan por ser no-lineales.
- Estas neuronas por lo común producen salidas continuas y realizan operaciones simples como recoger señales de las entradas y ensamblarlas de acuerdo a sus funciones de transferencia u operación.
- El modelo más simple de neurona artificial incluye las características principales de una red neuronal biológica: **paralelismo** y **alta conectividad**.
- Este modelo fue propuesto por McCulloch y Pitts (1943) y se puede apreciar en la Figura 4.

Esquema de una RNA (cont'd)

Perceptrón simple

Figure: Estructura de una neurona artificial



Esquema de una RNA (cont'd)

Perceptrón simple

- Como podemos apreciar, se tiene los siguientes elementos [Nunes da Silva et al., 2017]:
 - 1 *Señales de entrada*: vienen del entorno externo y se representan por el conjunto: $\{x_1, x_2, \dots, x_k\}$. Normalmente se debe **normalizar** las entradas para potenciar la capacidad de la red.
 - 2 *Pesos sinápticos*: se aplican para ponderar cada una de las entradas. Se representan por como $\{w_1, w_2, \dots, w_k\}$.
 - 3 *Sumador lineal* \sum : recoge todas las entradas ponderadas (por los pesos) a fin de producir un voltaje de activación.
 - 4 *Umbral de activación o bias* (θ): variable empleada para aplicar la umbralización producida por el sumador lineal.

Esquema de una RNA (cont'd)

Perceptrón simple

- 5 *Potencial de activación (n)*: resultado de la diferencia entre el sumador lineal y el umbral de activación. Si es positivo $n \geq \theta$, la neurona producirá potencial excitatorio, caso contrario será inhibitorio.
- 6 *Función de activación (f)*: limita la salida de la neurona en un rango de valores razonables.
- 7 *Señal de salida (y)*: valor final producido por la neurona dado un patrón de entrada o un conjunto de señales de entrada. Se puede usar para alimentar otras neuronas.

Funcionamiento de una RNA

Perceptrón simple

El proceso de operación de una RNA se lleva a cabo a través de la siguiente ecuación:

$$n = \sum_{i=1}^k w_i \cdot x_i + \theta \quad (1)$$
$$y = f(n)$$

Con ello, podemos resumir el proceso de operación en los siguientes pasos [Nunes da Silva et al., 2017]:

- 1 Presentar un conjunto de valores a la neurona, donde dichos valores representan las variables de entrada.
- 2 Multiplicar cada entrada de la neurona por su peso sináptico correspondiente.

Funcionamiento de una RNA (cont'd)

Perceptrón simple

- 3 Obtener el potencial producido por la suma ponderada de las señales de entrada (considerando el valor de umbral).
- 4 Aplicar la función de activación o transferencia a fin de limitar la salida de la neurona
- 5 Compilar la salida empleando la función de activación neuronal dentro del potencial de activación.

Funciones de transferencia

Parcialmente derivables

Son aquellas funciones con puntos para los que las derivadas de primer orden no existen. Fundamentalmente existen 3 funciones de este tipo

[Nunes da Silva et al., 2017]:

- 1 **Función escalón (Hard limiter):** el resultado que entrega esta función serán valores positivos de 1 cuando el potencial de activación de la neurona sea igual o mayor que cero. Se define a través de la siguiente especificación:

$$f(n) = \begin{cases} 1, & \text{si } n \geq 0 \\ 0, & \text{si } n < 0 \end{cases}$$

Funciones de transferencia (cont'd)

Parcialmente derivables

- 2** *Función escalón bipolar (Symmetric hard limiter):* el resultado que entrega esta función serán valores positivos de 1 cuando el potencial de activación de la neurona sea igual o mayor que cero y valores de -1 cuando el potencial de activación es menor que 0. Se define a través de la siguiente especificación:

$$f(n) = \begin{cases} 1, & \text{si } n \geq 0 \\ -1, & \text{si } n < 0 \end{cases}$$

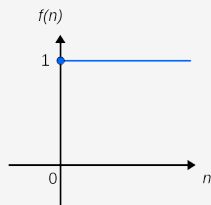
- 3** *Función de rampa simétrica:* los valores que devuelve esta función son los mismos que el potencial de activación en sí mismo, definido en el rango $[-a, a]$:

$$f(n) = \begin{cases} a, & \text{si } n > a \\ n, & \text{si } -a \leq n \leq a \\ -a, & \text{si } n < -a \end{cases}$$

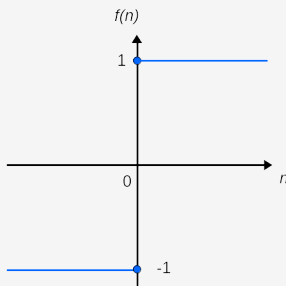
Funciones de transferencia (cont'd)

Parcialmente derivables

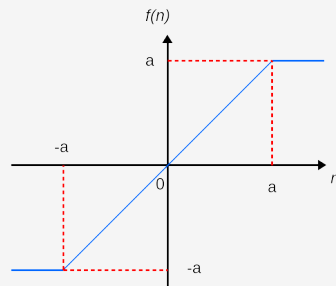
Figure: Funciones parcialmente derivables [Nunes da Silva et al., 2017]



*Función Escalón
(Hard Limiter)*



*Función Escalón Bipolar
(Symmetric Hard Limiter)*



Función de Rampa Simétrica

Funciones de transferencia

Totalmente derivables

Son aquellas funciones para las que la primera derivada existe para todos los puntos del dominio en que se define. Fundamentalmente existen 4 funciones de este tipo que se pueden emplear en RNA [Nunes da Silva et al., 2017]:

- 1 *Función logística*: el resultado que entrega esta función siempre será un valor contenido entre 0 y 1. Se define a través de la siguiente función matemática, donde β es una constante real que especifica el punto de inflexión:

$$f(n) = \frac{1}{1 + \exp(-\beta \cdot n)} \quad (2)$$

Funciones de transferencia (cont'd)

Totalmente derivables

- 2 *Función tangencial hiperbólica:* el resultado que entrega esta función siempre será un valor contenido entre -1 y 1. Se define a través de la siguiente función matemática, donde β es una constante real que especifica el punto de inflexión:

$$f(n) = \frac{1 - \exp(-\beta \cdot n)}{1 + \exp(-\beta \cdot n)} \quad (3)$$

- 3 *Función Gaussiana:* con esta función la neurona producirá resultados iguales para aquellos valores potenciales de activación (n) que estén ubicados a la misma distancia del centro (media). La curva es simétrica y está dada por la siguiente ecuación, donde σ denota la desviación estándar y c el centro de la Gaussiana:

$$f(n) = \exp\left(-\frac{(n - c)^2}{2 \cdot \sigma^2}\right) \quad (4)$$

Funciones de transferencia (cont'd)

Totalmente derivables

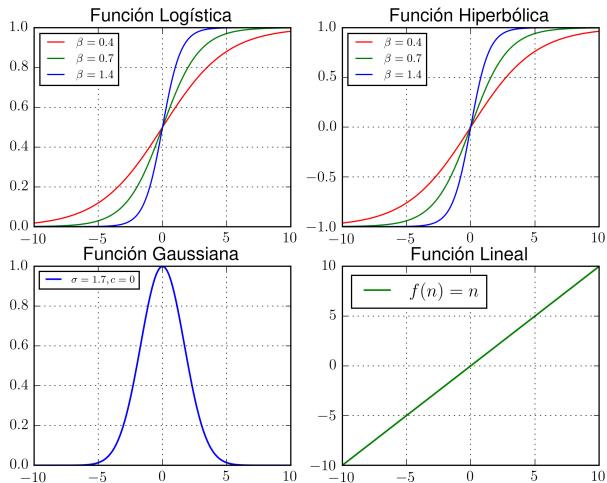
- 4 *Función Lineal*: conocida también como función identidad, produce un resultado igual al potencial de activación (n). Se define a través de la siguiente expresión matemática:

$$f(n) = n \quad (5)$$

Figure: Funciones totalmente derivables

Funciones de transferencia (cont'd)

Totalmente derivables



Regla de aprendizaje del perceptrón simple

A fin de ajustar los pesos y el bias del perceptrón, de modo que se puedan clasificar patrones que pertenecen a una de dos posibles clases, se emplea la regla de aprendizaje de Hebb (1949)

[Nunes da Silva et al., 2017, Demuth et al., 2014]. Por ello, previo a conocer los detalles de la regla de aprendizaje del perceptrón, es importante conocer dos aspectos: el modo de funcionamiento de una RNA y los tipos de aprendizaje.

- **Modo de funcionamiento de una RNA:** existen dos modos, uno es el de **entrenamiento**, donde la RNA aprende a clasificar o identificar determinados patrones (dada una entrada), mientras que en el modo de **ejecución** realiza el proceso de clasificación o identificación de los patrones. Es fundamental considerar que cuando se entrena a la red, esta **ajusta** los pesos y el bias, mientras que en el otro caso solo calcula la salida.

Regla de aprendizaje del perceptrón simple (cont'd)

- **Tipos de aprendizaje:** básicamente existen 3 formas o tipos en los que una RNA y otros sistemas inteligentes pueden aprender. A continuación se describe brevemente cada uno [Demuth et al., 2014]:
 - 1 *Aprendizaje supervisado:* la regla de aprendizaje se establece a partir de un conjunto de muestras o patrones. Con ello, la RNA debe ajustar los pesos y bias a fin de acercar las salidas actuales a las salidas deseadas.
 - 2 *Aprendizaje por refuerzo:* es similar al supervisado, salvo que en lugar de indicar cuál debería ser la salida correcta dada una muestra o patrón, el algoritmo indica un grado con el cual la salida es la apropiada.
 - 3 *Aprendizaje no supervisado:* en este caso los pesos y bias son ajustados solo en respuesta a las entradas de la red. En este caso no se cuenta con “salidas deseadas”, y generalmente estos algoritmos realizan alguna operación de agrupación (*clustering*).

A continuación se presenta el proceso para realizar el entrenamiento de un perceptrón simple:

- 1 Inicializar los pesos y bias con valores pequeños en el rango: $[-1, 1]$

Regla de aprendizaje del perceptrón simple (cont'd)

- 2 Presentar una nueva muestra o patrón a la red $\vec{X} = \{x_1, x_2, \dots, x_k\}$ de acuerdo a las salidas deseadas $\vec{\delta} = \{\delta_1, \delta_2, \dots, \delta_s\}$, donde k indica el total de entradas que tiene un patrón, y s cuántos elementos tiene una salida deseada para un patrón de entrada dado.
- 3 Calcular la salida actual empleando la ecuación $y = f\left(\sum_{i=1}^k w_i \cdot x_i + \theta\right)$, donde f es la función de transferencia *escalón*.
- 4 Cuando ocurre un error, es decir, cuando la RNA devuelve una salida distinta de la deseada, debe ajustar los pesos a través de la siguiente fórmula:

$$w_i(t+1) = w_i(t) + \alpha \cdot (\text{error}) \cdot x \quad (6)$$

Donde:

- w_i representa el peso que se da a la entrada i
- δ representa la salida deseada para un patrón dado
- t indica la iteración

Regla de aprendizaje del perceptrón simple (cont'd)

- α es la tasa de aprendizaje que puede tener valores en el rango $0.0 < \alpha < 1$
- El *error* se calcula como: $error = \delta - y$

De igual forma, se deberá actualizar también el bias:

$$\theta(t+1) = \theta(t) + \alpha \cdot error \quad (7)$$

- 5 Repetir desde el paso 2, hasta que al presentar todas las muestras o patrones a la RNA no se obtenga error en ninguna de las salidas.

Regla de aprendizaje del perceptrón simple (cont'd)

Ejercicio de cálculo

Dados los siguientes valores, calcule los pesos y bias de la RNA a fin de que la red converja y pueda realizar correctamente la clasificación:

$$\vec{W} = [0.23, -0.1]$$

$$\vec{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \delta \\ 1.53 & 0.53 & 1 \\ 1.5 & 1.0 & 1 \\ 2.0 & 1.73 & 0 \\ 3.0 & 1.5 & 0 \end{bmatrix}$$

$$\theta = 1.47$$

$$\alpha = 1.0$$

Referencias



Banchs, R. (2005).

Redes neuronales para la caracterización de yacimientos: del mito a las aplicaciones prácticas.

tekhne, (8).



Clarke, E. (1963).

Aristotelian concepts of the form and function of the brain.

Bulletin of the History of Medicine, 37:1.



Copeland, B. J. (2004).

The Essential Turing.

Clarendon Press.



Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. (2014).

Neural Network Design.

Martin Hagan, USA, 2nd edition.

Referencias (cont'd)



Muñoz Pérez, J. (2012).

Redes neuronales artificiales: el cerebro como fuente de inspiración.



Nunes da Silva, I., Hernane Spatti, D., Andrade Flauzino, R.,
Bartocci Liboni, L. H., and dos Reis Alves, S. F. (2017).

Artificial Neural Networks : A Practical Course.

Springer International Publishing, 1 edition.