REVISTA DE INVESTIGACIÓN EN MODELOS MATEMATICOS APLICADOS A LA GESTION Y LA ECONOMIA - AÑO 8 VOLUMEN I (2021-I)

CMA IADCOM - UBA | CENTRO DE INVESTIGACIÓN EN METODOS CUANTITATIVOS APLICADOS A LA ECONOMÍA Y LA GESTIÓN

Universidad de Buenos Aires Facultad de Ciencias Económicas

http://www.economicas.uba.ar/institutos_y_centros/revista-modelos-matematicos/

EL PERCEPTRÓN: UNA RED NEURONAL ARTIFICIAL PARA CLASIFICAR DATOS

García, Roberto

Universidad de Buenos Aires. Facultad de Ciencias Económicas. Centro de Investigación en Métodos Cuantitativos Aplicados a la Gestión y la Economía (CMA). Av. Córdoba 2122 – 1120AAQ. Ciudad Autónoma de Buenos Aires, República Argentina

robertogarcia@economicas.uba.ar

Recibido: 22-11-2021

Aceptado: 07-03-2022

Palabras clave

Neurona artificial-Peso sináptico-Aprendizaje-Clasificación-Separabilidad lineal

Resumen

Las redes neuronales artificiales son modelos computacionales de Machine Learning que están constituidos por unidades elementales de procesamiento, denominadas nodos o neuronas artificiales, dispuestas en capas y con muchas conexiones entre ellas. Estos sistemas aprenden de la experiencia (datos) guardando el conocimiento en la configuración de los valores de ciertos parámetros que los caracterizan, los pesos sinápticos. Se las utiliza, entre otras cosas, para reconocimiento de sonidos, imágenes, clasificación de datos y predicciones. La complejidad de ciertas tareas como el procesamiento de datos no estructurados proveniente de diversas fuentes como imágenes, sonido o texto requiere modelos con múltiples capas en cuyo caso se habla de Deep Learning. Una neurona artificial tiene un funcionamiento sencillo que imita al de una neurona biológica, pudiendo adquirir dos estados de activación, "0" (apagada) y "1" (encendida) dependiendo del valor de las entradas. Cada neurona recibe una o más entradas que combina linealmente para producir una señal neta. Si la entrada neta supera un cierto umbral la neurona se dispara y emite una señal o salida de valor "1" y si no permanece apagada y la salida es de valor "0".

En este artículo se describe el modelo matemático del Perceptrón simple para clasificar datos linealmente separables en dos categorías. Asimismo, se exponen los fundamentos de la regla de aprendizaje para actualizar los pesos sinápticos. También se explica cómo configurar un modelo de Perceptrón multicapa para separar una serie de datos en cuatro clases, siempre que se puedan agrupar en categorías linealmente separables. Finalmente se señalan las limitaciones del método de aprendizaje presentado dando paso a los algoritmos basados en la aplicación del método del gradiente descendente para minimizar el valor de una función de error dependiente de todos los pesos sinápticos de la red.

ISSN (En línea) 2362 3225

THE PERCEPTRON: AN ARTIFICIAL NEURAL NETWORK TO CLASSIFY DATA

Abstract

Artificial neural networks are computational models of Machine Learning that are made up of elementary processing units, called nodes or artificial neurons, arranged in layers and with many connections between them. These systems learn from experience (data) by storing knowledge in the configuration of the values of certain parameters that characterize them, the synaptic weights. They are used, among other things, for recognition of sounds, images, data classification and predictions. The complexity of certain tasks such as the processing of unstructured data from various sources such as images, sound or text requires models with multiple layers, in which case we speak of Deep Learning. An artificial neuron has a simple operation that mimics that of a biological neuron, being able to acquire two activation states, "0" (off) and "1" (on) depending on the value of the inputs. Each neuron receives one or more inputs that it linearly combines to produce a net signal. If the net input exceeds a certain threshold, the neuron fires and emits a signal or output value "1" and if it does not remain off and the output is value "0".

This article describes the simple Perceptron mathematical model to classify linearly separable data into two categories. Likewise, the fundamentals of the learning rule to update the synaptic weights are exposed. It is also explained how to configure a multilayer Perceptron model to separate a series of data into four classes, as long as they can be grouped into linearly separable categories. Finally, the limitations of the learning method presented are pointed out, giving way to algorithms based on the application of the descending gradient method to minimize the value of an error function dependent on all the synaptic weights of the network.

KEYWORDS

Artificial neuron-synaptic weight-learning-classification-linear separability

ISSN (En línea) 2362 3225

INTRODUCCIÓN

Las redes neuronales artificiales son sistemas computacionales capaces de realizar tareas como reconocer imágenes, sonidos, identificar una huella digital, clasificar un objeto a partir de sus propiedades o hacer predicciones como del valor de un índice financiero entre otras. Se trata de un caso especial de modelos de aprendizaje automático (Machine Learning) que permiten que las computadoras aprendan a realizar tareas sin programarlas, es decir sin indicarle las reglas a seguir para realizarlas. Estos sistemas aprenden de la experiencia (datos) guardando el conocimiento en la configuración de los valores de ciertos parámetros que los caracterizan. Los métodos de aprendizaje son, en general, de tres tipos: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. Están constituidas por unidades elementales de procesamiento denominadas nodos o neuronas, dispuestas por capas muy interconectadas. La cantidad de capas (profundidad de la red) requerida o de neuronas por capa como así también las conexiones entre neuronas aumentan a medida que se torna más compleja la tarea que la red debe aprender a realizar. Se utiliza el término Deep Learning para referirse al aprendizaje automático basado en modelos compuestos por múltiples capas de neuronas artificiales. El modelo más simple, denominado Perceptrón, fue el primer modelo de red neuronal artificial desarrollado por Rosenblatt en 1958. Su estructura consta de una capa de entrada de varias neuronas lineales que transmiten la información sin procesarla y una capa de salida formada por una única neurona que procesa las señales de entrada, permitiendo clasificar datos en dos categorías según los patrones de salida u outputs que ellos producen.

Una neurona artificial está inspirada en el funcionamiento de una neurona biológica. Esta última es una célula nerviosa con un cuerpo o soma, rodeado de una serie de ramificaciones o dendritas. Unido al cuerpo existe un filamento conocido como axón que se extiende y ramifica en terminales axónicos. Cada neurona recibe señales eléctricas a través de las dendritas. Las señales recibidas son procesadas o integradas en el cuerpo. Cuando la señal eléctrica resultante supera un potencial o voltaje umbral se dispara un potencial de acción a través del axón. El potencial de acción llega a los terminales axónicos y mediante un mecanismo químico en el que participan sustancias denominadas neurotransmisores se establecen conexiones o sinapsis con las dendritas de otra neurona generándose un potencial postsináptico excitatorio o inhibitorio cuya magnitud depende de la intensidad o peso sináptico con que se conecten ambas neuronas. Este proceso de ponderación de señales suma y activación se repite en las demás neuronas y se recupera para modelizar el funcionamiento de una neurona artificial.

1. MODELO DE NEURONA ARTIFICIAL. EL PERCEPTRÓN SIMPLE.

El equivalente artificial de una neurona biológica es el nodo o neurona artificial U_j cuyo esquema se muestra en la figura 1. Se trata de una unidad de procesamiento que combina linealmente n señales de entrada x_i multiplicando cada una de ellas por un peso sináptico w_{ij} para producir una entrada neta v_j

$$v_j = w_{1j}.x_1 + w_{2j}.x_2 + \cdots + w_{nj}.x_n$$

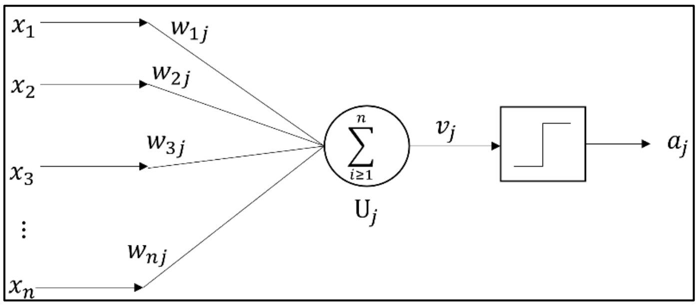
que utilizando el símbolo de sumatoria queda expresada como:

$$v_j = \sum_{i=1}^{i=n} w_{ij}.x_i$$

Si la entrada neta iguala o supera un cierto potencial umbral θ, entonces la neurona dispara un potencial o activación a_j de valor 1 y si no lo alcanza, la neurona permanece inactiva y la activación se considera de valor 0. Este comportamiento corresponde al modelo de una neurona binaria y se describe matemáticamente mediante una función de activación escalón, del siguiente modo:

$$a_j = f(v_j) = \begin{cases} 1 & \text{si } v_j \geq \theta \\ 0 & \text{si } v_j < \theta \end{cases}$$

Figura 1: Modelo de neurona artificial



Fuente: Elaboración propia

Definiendo el vector de entradas o inputs como aquel cuyas componentes son las señales de entrada, es decir:

$$\mathrm{X}=(x_1, x_2, \dots, x_n)$$

Y el vector de pesos sinápticos como:

$$\mathrm{W}_j = (w_{1j}, w_{2j,} \dots, w_{nj})$$

la entrada neta v_j se obtiene como resultado del producto escalar entre los vectores X y W_j

$$v_j = \mathrm{W}_j \cdot \mathrm{X}$$

Luego, la activación o salida de la neurona está dada por

$a_j = 1$ si $\mathrm{W}_j \cdot \mathrm{X} \geq \theta$

$a_j = 0$ si $\mathrm{W}_j \cdot \mathrm{X} < \theta$

Teniendo en cuenta que el producto escalar se puede calcular multiplicando las normas de los vectores $\|\mathrm{W}_j\|$ y $\|\mathrm{X}\|$ por el coseno del ángulo comprendido entre ellos del siguiente modo

$$\mathrm{W}_j \cdot \mathrm{X} = \|\mathrm{W}_j\|.\|\mathrm{X}\|.\cos\varphi$$

donde $0 \leq \varphi \leq \pi$ es el ángulo comprendido entre los vectores W_j y X, y que la proyección ortogonal de X sobre W_j está dada por

$$x_W = \|X\|.\cos\varphi$$

Resulta válida la siguiente igualdad

$$W_j \cdot X = \|W_j\|.\,x_W$$

La condición que permite separar los patrones de entrada en dos clases, según que la activación de la neurona sea $a_j = 0$ o $a_j = 1$, se denomina frontera de decisión y queda expresada matemáticamente por la siguiente ecuación

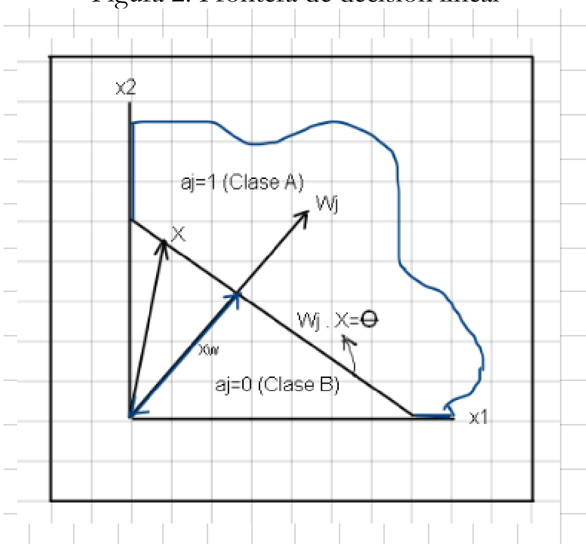
$$W_j \cdot X = \theta$$

O su equivalente

$$x_W = \frac{W_j \cdot X}{\|W_j\|} = \frac{\theta}{\|W_j\|}$$

Cada patrón de entradas a la neurona U_j es un vector X que se puede representar en un espacio n dimensional o espacio patrón de entradas. Si se supone, sin perder generalidad, que la neurona tiene dos entradas, cada vector X tiene dos componentes y el espacio de entradas es bidimensional. Una vez elegidos los valores de los pesos sinápticos y del umbral, $\|W_j\|$ y θ quedan fijos, y por lo tanto la frontera de decisión está determinada por todos los vectores X que tienen proyección ortogonal sobre W_j de valor $x_W = \frac{\theta}{\|W_j\|}$ y su representación gráfica es una recta perpendicular al vector de pesos sinápticos W_j como puede apreciarse en la figura 2.

Figura 2: Frontera de decisión lineal



Fuente: Elaboración propia

Cualquier patrón de entradas caracterizado por un vector X cuya proyección ortogonal sea $x_W > \frac{\theta}{\|W_j\|}$ producirá una activación de la neurona de valor $a_j = 1$ y será clasificado como perteneciente a la Clase A, mientras que aquellos patrones representados por vectores X que satisfagan la condición $x_W < \frac{\theta}{\|W_j\|}$ producirán una activación de valor $a_j = 0$ y serán ubicados en la Clase B.

Si los patrones de entradas son de dimensión 3, la frontera de decisión es un plano cuya normal tiene la dirección del vector de pesos sinápticos. Para casos de dimensión $n > 3$ la frontera que separa las clases o categorías es un hiperplano y no admite representación gráfica.

Por lo visto, una neurona binaria como la de la figura 1 o TLU (Threshold Linear Unit) con una función de activación escalón que produzca una salida binaria puede clasificar en dos categorías aquellos datos o patrones de entrada siempre que se puedan separar por fronteras de decisión lineales. Tales conjuntos de datos se dice que son linealmente separables.

Un ejemplo de separabilidad lineal en el plano es el que corresponde a los patrones que caracterizan a la función lógica AND que se muestran en la Figura 3:

Figura 3: función AND

x_1	x_2	OUT
0	0	0
0	1	0
1	0	0
1	1	1

Es posible elegir una configuración de valores de pesos sinápticos y umbral que definan una frontera de decisión que separe los patrones en dos clases, aquella a la que pertenezcan los patrones cuya salida es 0 y la que contiene el único patrón con salida 1. Dichos valores podrían ser $w_1 = 1, w_2 = 1$ y $\theta = 1{,}5$

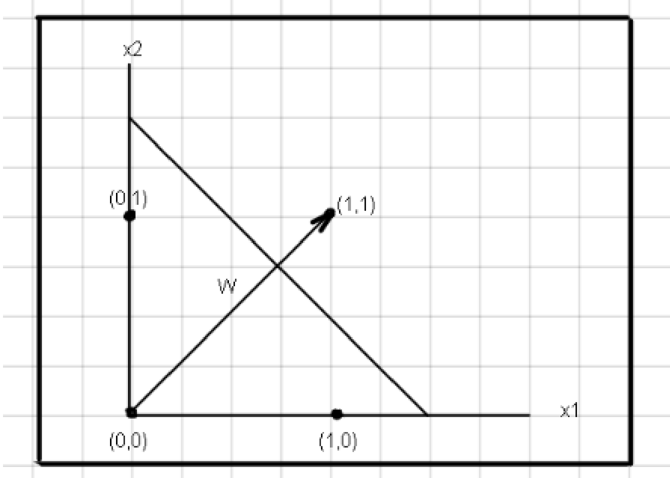
La condición de separación para dicho conjunto de valores está dada por la ecuación

$$1. x_1 + 1. x_2 = 1{,}5$$

la que claramente define una recta, la frontera de decisión.

La representación gráfica de los patrones de entrada, la frontera de decisión y el vector de pesos sinápticos en el espacio de dos dimensiones se muestra en la figura 4.

Figura 4: Separabilidad lineal



Fuente: Elaboración propia

Encontrar los pesos sinápticos y el umbral para conseguir que una neurona binaria de dos entradas clasifique en forma adecuada patrones que sean linealmente separables es tarea fácil y puede resolverse gráficamente. Sin embargo, si se trata de patrones de entrada de dimensión mayor, encontrar los parámetros mencionados para lograr la clasificación requiere de un procedimiento iterativo conocido como algoritmo de entrenamiento. Antes de describir dicho procedimiento, resulta conveniente mostrar una transformación que permitirá ajustar el umbral de la neurona como un peso sináptico más. Para ello partiendo de la condición de decisión:

$$\sum_{i=1}^{n} w_{ij}.x_i = \theta$$

Es posible escribir

$$\sum_{i=1}^{n} w_{ij}.x_i - \theta = 0$$

O bien

$$\sum_{i=1}^{n+1} w_{ij}.x_i = 0$$

Donde en la última expresión se ha considerado $w_{n+1,j} = \theta$ y $x_{n+1} = -1$

La condición puede escribirse, utilizando vectores, como

$$W \cdot X = 0$$

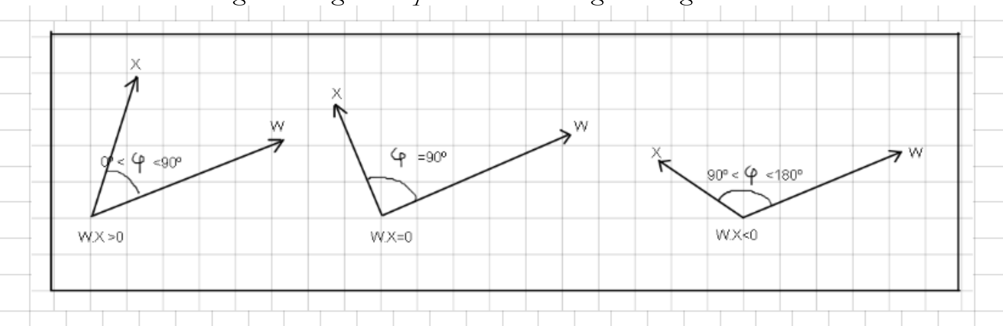
Donde W es ahora un vector de dimensión $n + 1$ denominado vector de pesos aumentado, cuya $(n + 1)\_ésima$ componente es igual a θ y X= $(x_1, x_2, \cdots, x_n, -1)$

Por lo tanto

Si W·X >0 entonces $a = 1$ y si W·X <0 entonces $a = 0$

El signo del producto escalar está determinado por el ángulo entre los vectores W y X como se muestra en la Figura 5:

Figura 5: Signos del producto escalar según el ángulo entre los vectores



Fuente: Elaboración propia

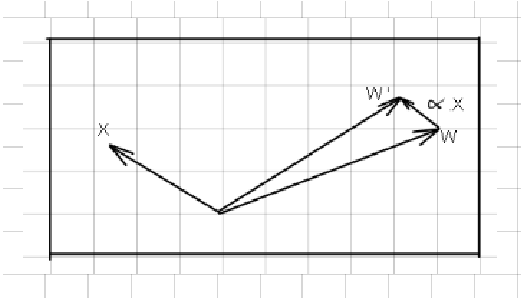
2. ¿CÓMO APRENDE EL PERCEPTRÓN SIMPLE?

El conocimiento que aprende una red queda guardado en el valor de los pesos sinápticos. Estos se van modificando gradualmente por medio de un proceso iterativo, hasta obtener los valores que configuran la red para que las salidas producidas no difieran de las conocidas o etiquetas de los datos utilizados para el entrenamiento (aprendizaje supervisado). La diferencia entre las salidas reales y las deseadas para un patrón de entrada determina los cambios a introducir en los pesos sinápticos. Cualquier fórmula para dichas modificaciones se conoce como regla de aprendizaje. Se denomina algoritmo de entrenamiento o aprendizaje al proceso completo que incluye hasta el orden en que se presentan los datos de entrenamiento, el criterio para decidir la terminación del proceso, etc.

Supóngase tener que entrenar una neurona binaria utilizando un conjunto de entrenamiento consistente en pares $\{X, y\}$ donde X es el vector que recoge las entradas e y es la salida deseada. La neurona puede tener n entradas, aunque resultará conveniente considerar el caso de dos entradas para poder seguir razonamientos e interpretaciones gráficas en el plano. Se trabajará con el vector aumentado de pesos W.

Supóngase ahora que al vector X del conjunto de entrenamiento le corresponda salida deseada $y = 1$, en tanto que para el vector actual W la salida o activación real sea $a = 0$. Es evidente que la neurona clasificó mal a este patrón y por lo tanto habrá que hacer una modificación de los pesos. Para haber ubicado al patrón en la clase o categoría correcta la activación de la neurona debió haber sido 1, en cuyo caso el producto interno debió satisfacer W · X>0 cuando en verdad se satisfizo W · X<0. Para corregir esta situación debe rotarse W en el sentido en que disminuya el ángulo φ sin provocar un cambio drástico, lo que puede conseguirse sumando a W un múltiplo escalar de X para producir una actualización de pesos W '=W+α.X con $0 < \alpha < 1$ como ilustra la Figura 6.

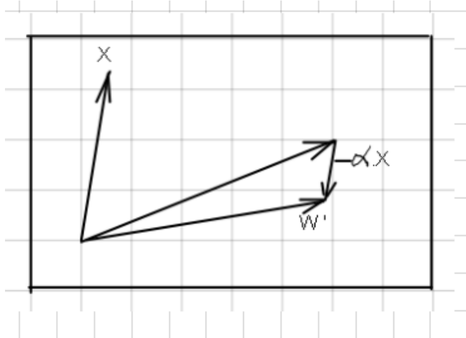
Figura 6: Actualización de pesos en caso de que sea $y - a_j = 1$



Fuente: Elaboración propia

Contrariamente, si el error de clasificación hubiese sido producir una activación $a_j = 1$ para una salida deseada $y = 0$, el producto interno W·X tuvo signo positivo cuando debió ser negativo. En tal caso, la actualización del vector de pesos se consigue rotando W en el sentido en que aumente el ángulo φ haciendo W '=W-α.X como se muestra en la Figura 7.

Figura 7: Actualización de pesos en caso de que sea $y - a_j = -1$



Fuente: Elaboración propia

Ambas actualizaciones se combinan en una única regla del siguiente modo:

$$W' = W + \alpha.(y - a).X$$

La variación del vector de pesos en cada actualización será:

$$\Delta W = \alpha.(y - a).X$$

Y para cada componente:

$$\Delta w_i = \alpha.(y - a).x_i \qquad \forall i = 1 \text{ a } n + 1$$

Con $w_{n+1} = \theta$ y $x_{n+1} = -1$ siempre. El parámetro α se denomina tasa o razón de aprendizaje ya que controla la magnitud del cambio de los pesos en cada iteración y en consecuencia mide la velocidad con que la unidad de procesamiento o neurona aprende.

Este algoritmo de aprendizaje es conocido como la *Regla de entrenamiento del Perceptrón.* Se trata de un procedimiento iterativo que puede sintetizarse en los siguientes pasos:

Repetir hasta haber utilizado todos los datos del set de entrenamiento

1. Definir W.
2. Tomar un par $\{X, y\}$ del set de entrenamiento.
3. Evaluar la activación a de la neurona cuando la entrada es X.
4. Si $a \neq y$ entonces actualizar los pesos según W '=W+α.$(y - a)$.X y pasar a 5, caso contrario volver a 2.
5. Hacer W=W ' y volver a 2.

Una vez procesados todos los datos del conjunto de entrenamiento se dice que se ha cumplido una época.

3. EL PERCEPTRÓN MULTICAPA

Las configuraciones de las redes presentan aspectos muy diferentes, pero tienen un aspecto común, el ordenamiento de las neuronas en capas o niveles como se muestra en la Figura 8.

A partir de su situación dentro de la red se pueden distinguir tres tipos de capas:

a) De entrada: es la capa que recibe directamente la información de las fuentes externas de la red.

b) Ocultas: son internas a la red y no tienen contacto directo con el exterior. Una red puede no tener capas ocultas, tener algunas o el número de ellas puede ser elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, la topología de la red neuronal.

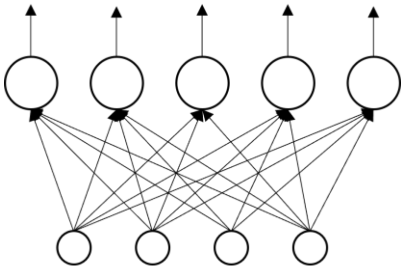
c) De salida: transfieren información de la red hacia el exterior.

Se dice que una red es totalmente conectada si todas las salidas desde un nivel llegan a todos y cada uno de los nodos del nivel siguiente.

Las redes multicapa se forman con un grupo de capas en cascada. El número de capas de una red está determinado por las capas ocultas y de salida. Así una red formada por una capa de entrada y una de salida se considera monocapa. Esto se debe a que las neuronas de la capa de entrada transmiten la información sin procesarla, es decir son lineales y la función de activación es la función identidad $a_j = f(v_j) = v_j$ En Las llamadas redes feedforward la salida de cada una de las neuronas de una capa es entrada de cada una de las neuronas de la siguiente capa y ninguna neurona recibe señales de otra de su misma capa o de capas posteriores. Se ha demostrado que las redes multicapa presentan cualidades y aspectos por encima de las redes de una capa simple.

Utilizando el algoritmo visto anteriormente es posible entrenar múltiples nodos o neuronas formando una red monocapa como la de la figura 8, para clasificar datos linealmente separables. Esto podría ocurrir si quisiéramos clasificar caracteres alfabéticos escritos a mano donde se requieren 26 dicotomías, cada una separando una clase de letra del resto del alfabeto

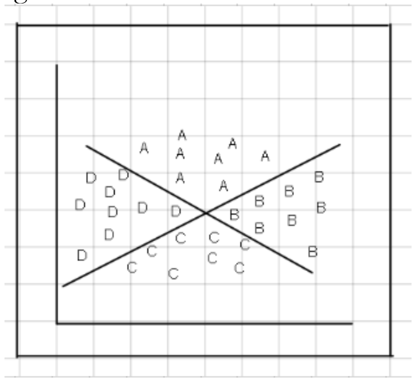
Figura 8: Red neuronal monocapa



Fuente: Elaboración propia

Supongamos ahora tener cuatro clases A, B, C y D que se pueden separar mediante dos fronteras de decisión lineales en el espacio de patrones de entrada como se muestra en la figura 9. La representación es esquemática de un espacio de entradas de alta dimensión. Sería inútil utilizar una red monocapa para separar estas clases, ya que cada una de ellas no es linealmente separable del resto de ellas consideradas en forma conjunta. Si bien las cuatro clases no son linealmente separables, si lo son ciertas clases que podrían formarse agrupándolas de a dos. Por ejemplo, la clase de mayor orden a la que pertenecen los patrones del tipo A y B, denominada AB, es linealmente separable de aquella formada por los patrones de tipo C y D juntos, llamada CD. De la misma manera las clases AD y BC son linealmente separables.

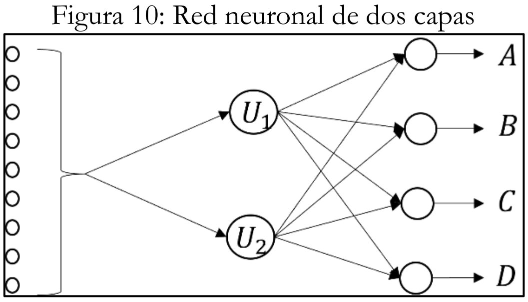
Figura 9: Clasificación de cuatro clases



Fuente: Elaboración propia

Se puede configurar una red de dos capas como la de la Figura 10.

Figura 10: Red neuronal de dos capas



Fuente: Elaboración propia

La primera de las capas de la red consta de dos neuronas U_1 y U_2. Estas unidades con salidas a_1 y a_2 se entrenan para producir las dicotomías mostradas en la Figura 11.

Figura 11: Salidas de la primera capa

U_1	
Clase	a_1
AB	1
CD	0

U_2	
Clase	a_2
AD	1
BC	0

Fuente: elaboración propia

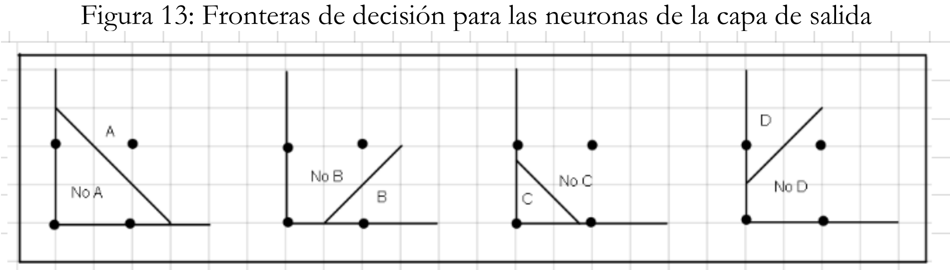
Supóngase que la entrada de las unidades U_1 y U_2 sea un miembro de la clase A. De las tablas de la figura 11 se desprende que los outputs de ambas neuronas serán $a_1 = a_2 = 1$. Contrariamente, si las salidas que producen ambas unidades frente a un input X de clase desconocida es 1, según la salida de la unidad U_1 el patrón pertenece a la clase AB y según la salida de la unidad U_2 el patrón es de la clase AD. La única manera de que el patrón X pertenezca a ambas clases es que sea de la categoría A. Se concluye entonces que $a_1 = 1$ y $a_2 = 1$ si y solo si X está en la categoría A. Procediendo en forma análoga con las otras tres posibilidades se obtiene un código en función de las activaciones a_1 y a_2 que se muestra en la Figura 12.

Figura 12: Código de salidas de la primera capa

a_1	a_2	Clase
0	0	C
0	1	D
1	0	B
1	1	A

La segunda capa de la red contiene cuatro neuronas, todas ellas conectadas a cada una de las neuronas de la primera capa. Para señalar la clase A la primera neurona de la segunda capa se configura para producir un output de valor “1” cuando el patrón de entrada sea (1,1) y “0” para cualquier otra entrada. La segunda neurona se configura para que el output sea “1” si el vector de inputs es (1,0) y “0” para cualquier otro patrón de entradas y así para C y D. Cabe destacar que los patrones de entrada que llegan a cada una de las neuronas de la segunda capa son linealmente separables como se puede visualizar gráficamente en la Figura 13.

Figura 13: Fronteras de decisión para las neuronas de la capa de salida



Fuente: Elaboración propia

Hay que observar que el éxito de la clasificación requirió contar previamente con la siguiente información sobre los patrones de entrada:

1. Las cuatro clases se pueden separar mediante dos fronteras lineales de decisión

2. Las categorías AB y CD son linealmente separables, lo mismo que AD y BC.

En efecto, si se hubieran agrupado las clases formando los grupos AC y BD el procedimiento hubiera fallado. Obviamente resulta conveniente poder entrenar todos los pesos de una red multicapa prescindiendo del conocimiento a priori de información acerca de los patrones de entrada como la citada en (2). Una alternativa de entrenamiento supervisado consiste en definir una función de pérdida basada en la suma de los errores debidos a la diferencia entre la salida real y la salida deseada de todos los patrones del conjunto de entrenamiento. El objetivo es minimizar el valor de esta función de error actualizando iterativamente los pesos de la red mediante el método del gradiente descendente. Este método requiere utilizar funciones de activación que sean diferenciables como la función sigmoidea $f(v_j) = \frac{1}{1+e^{-kv_j}}$ una de las más ampliamente utilizadas en los modelos de redes neuronales. Entre los métodos mencionados pueden citarse la regla delta y al algoritmo de retropropagación del error o backpropagation que no son tratados en este artículo.

CONCLUSIÓN

El perceptrón simple es una red neuronal muy sencilla con una única neurona binaria (función de activación escalón) en la capa de salida que puede entrenarse para clasificar un conjunto de datos de entrada en dos categorías. Como se ha visto el entrenamiento es exitoso si los datos son linealmente separables, es decir pertenecen a uno u otro de los semiespacios delimitados por algún hiperplano cuya orientación y desplazamiento están determinados por el conjunto de valores que toman los pesos sinápticos de las conexiones entre las neuronas de la capa de entrada y la única neurona de la capa de salida y del umbral de ésta última. La red aprende a partir de un conjunto de datos etiquetados (de clases conocidas) mediante un procedimiento iterativo que actualiza los pesos sinápticos y el umbral en cada paso hasta lograr clasificar los datos correctamente. En caso de que no haya separabilidad lineal o para clasificación en más de dos categorías se utilizan redes con arreglos multicapa recurriéndose a métodos de entrenamiento basados en el método del gradiente descendente para encontrar el mínimo de una función de error.

BIBLIOGRAFÍA

Gurney K. (2004). *An introduction to neural networks.* UCL Press, Taylor & Francis e-Library. London.

Haykin, S. (2009). *Neural Networks and Learning Machines.* PEARSON Prentice Hall, 3rd.Edition. USA

Torres, J. (2020). *Python Deep Learning. Introducción práctica con Keras y TensorFlow 2.* Marcombo. Spain.

García Martínez, R.; Servente, M. y Pasquini, D. (2003). *Sistemas inteligentes.* Nueva Librería S.R.L. Buenos Aires, Argentina.