¿Qué son las redes neuronales?



[Artificial Intelligence](https://www.ibm.com/es-es/think/artificial-intelligence)

* [¿Qué son las redes neuronales?](https://www.ibm.com/es-es/think/topics/neural-networks#%C2%BFQu%C3%A9+son+las+redes+neuronales%3F)
* [¿Cómo funcionan las redes neuronales?](https://www.ibm.com/es-es/think/topics/neural-networks#%C2%BFC%C3%B3mo+funcionan+las+redes+neuronales%3F)
* [tipos de redes neuronales](https://www.ibm.com/es-es/think/topics/neural-networks#tipos+de+redes+neuronales)
* [Redes neuronales frente a deep learning](https://www.ibm.com/es-es/think/topics/neural-networks#Redes+neuronales+frente+a+deep+learning)
* [Historia de las redes neuronales](https://www.ibm.com/es-es/think/topics/neural-networks#Historia+de+las+redes+neuronales)

¿Qué son las redes neuronales?

Una red neuronal es un programa o modelo de [machine learning](https://www.ibm.com/es-es/think/topics/machine-learning) que toma decisiones de forma similar al cerebro humano, al emplear procesos que imitan la forma en que las neuronas biológicas trabajan juntas para identificar fenómenos, sopesar opciones y llegar a conclusiones.

Toda red neuronal consta de capas de nodos o neuronas artificiales: una capa de entrada, una o varias capas ocultas y una capa de salida. Cada nodo se conecta a los demás y tiene su propia ponderación y umbral asociados. Si la salida de cualquier nodo individual está por encima del valor umbral especificado, ese nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, no se pasa ningún dato a la siguiente capa de la red.

Las redes neuronales se basan en datos de entrenamiento para aprender y mejorar su precisión con el tiempo. Una vez perfeccionadas, se convierten en potentes herramientas en informática e [inteligencia artificial](https://www.ibm.com/es-es/think/topics/artificial-intelligence), que nos permiten clasificar y agrupar datos a gran velocidad. Las tareas de reconocimiento de voz o de imágenes pueden llevar minutos frente a horas si se comparan con la identificación manual por parte de expertos humanos. Uno de los ejemplos más conocidos de red neuronal es el algoritmo de búsqueda de Google.

Las redes neuronales a veces se denominan redes neuronales artificiales (ANN) o redes neuronales simuladas (SNN). Son un subconjunto del machine learning y el núcleo de los modelos de [deep learning](https://www.ibm.com/es-es/think/topics/deep-learning).

Boletín del sector

Las últimas tendencias en IA, presentadas por expertos

Obtenga conocimientos organizados sobre las noticias más importantes e intrigantes de la IA. Suscríbase a nuestro boletín semanal Think. Consulte la [Declaración de privacidad de IBM](https://www.ibm.com/es-es/privacy).

Principio del formulario

* Correo electrónico comercial

Suscríbase

Final del formulario

¿Cómo funcionan las redes neuronales?

Piense en cada nodo individual como su propio modelo de [regresión lineal](https://www.ibm.com/es-es/think/topics/linear-regression), compuesto por datos de entrada (o input), ponderaciones, un sesgo (o umbral) y una salida (u output). La fórmula sería la siguiente:

∑wixi + sesgo = w1x1 + w2x2 + w3x3 + sesgo

salida = f(x) = 1 if ∑w1x1 + b>= 0; 0 if ∑w1x1 + b < 0

Una vez determinada la capa de entrada, se asignan las ponderaciones. Estas ponderaciones ayudan a determinar la importancia de cualquier variable, ya que las más grandes contribuyen de forma más significativa a la salida en comparación con otras entradas. A continuación, todas las entradas se multiplican por sus respectivas ponderaciones y se suman. Después, la salida se pasa a través de una función de activación, que determina la salida. Si esa salida supera un umbral determinado, se «dispara» (o activa) el nodo, pasando los datos a la siguiente capa de la red. Esto da como resultado que la salida de un nodo se convierta en la entrada del siguiente nodo. Este proceso de pasar datos de una capa a la siguiente capa define esta red neuronal como una red de proalimentación.

Desglosemos el aspecto de un único nodo utilizando valores binarios. Podemos aplicar este concepto a un ejemplo más tangible, como si deberías ir a hacer surf (Sí: 1, No: 0). La decisión de ir o no ir es nuestro resultado previsto, o y-hat. Supongamos que hay tres factores que influyen en tu decisión:

1. ¿Las olas son buenas? (Sí: 1, No: 0)
2. ¿Está el pico despejado? (Sí: 1, No: 0)
3. ¿Ha habido un ataque de tiburones recientemente? (Sí: 0, No: 1)

Entonces, supongamos lo siguiente, dándonos las siguientes entradas:

* X1 = 1, ya que las olas son buenas
* X2 = 0, ya que está lleno de gente
* X3 = 1, ya que no ha habido un ataque reciente de tiburón

Ahora, tenemos que asignar algunas ponderaciones para determinar la importancia. Unas ponderaciones mayores significan que determinadas variables son más importantes para la decisión o el resultado.

* W1 = 5, ya que las grandes olas no aparecen con frecuencia
* W2 = 2, ya que estás acostumbrado a las multitudes
* W3 = 4, ya que tienes miedo a los tiburones

Por último, también supondremos un valor umbral de 3, lo que se traduciría en un valor de sesgo de –3. Con todas las entradas, podemos empezar a introducir valores en la fórmula para obtener la salida deseada.

Y-hat = (1\*5) + (0\*2) + (1\*4) – 3 = 6

Si utilizamos la función de activación del principio de esta sección, podemos determinar que la salida de este nodo sería 1, ya que 6 es mayor que 0. En este caso, iría a surfear; pero si ajustamos las ponderaciones o el umbral, podemos obtener resultados diferentes del modelo. Cuando observamos una decisión, como en el ejemplo anterior, podemos ver cómo una red neuronal podría tomar decisiones cada vez más complejas en función de la salida de las decisiones o capas anteriores.

En el ejemplo anterior, utilizamos perceptrones para ilustrar algunas de las matemáticas que están en juego aquí, pero las redes neuronales aprovechan las neuronas sigmoidales, que se distinguen por tener valores entre 0 y 1. Dado que las redes neuronales se comportan de forma similar a los árboles de decisión, con datos en cascada de un nodo a otro, tener valores x entre 0 y 1 reducirá el impacto de cualquier cambio dado de una sola variable en la salida de cualquier nodo dado y, posteriormente, en la salida de la red neuronal.

Cuando empecemos a pensar en casos de uso más prácticos para las redes neuronales, como el reconocimiento o la clasificación de imágenes, aprovecharemos el aprendizaje supervisado, o conjuntos de datos etiquetados, para entrenar el algoritmo. Al entrenar el modelo, querremos evaluar su precisión utilizando una función de coste (o pérdida). También se conoce como error cuadrático medio (MSE). En la siguiente ecuación,

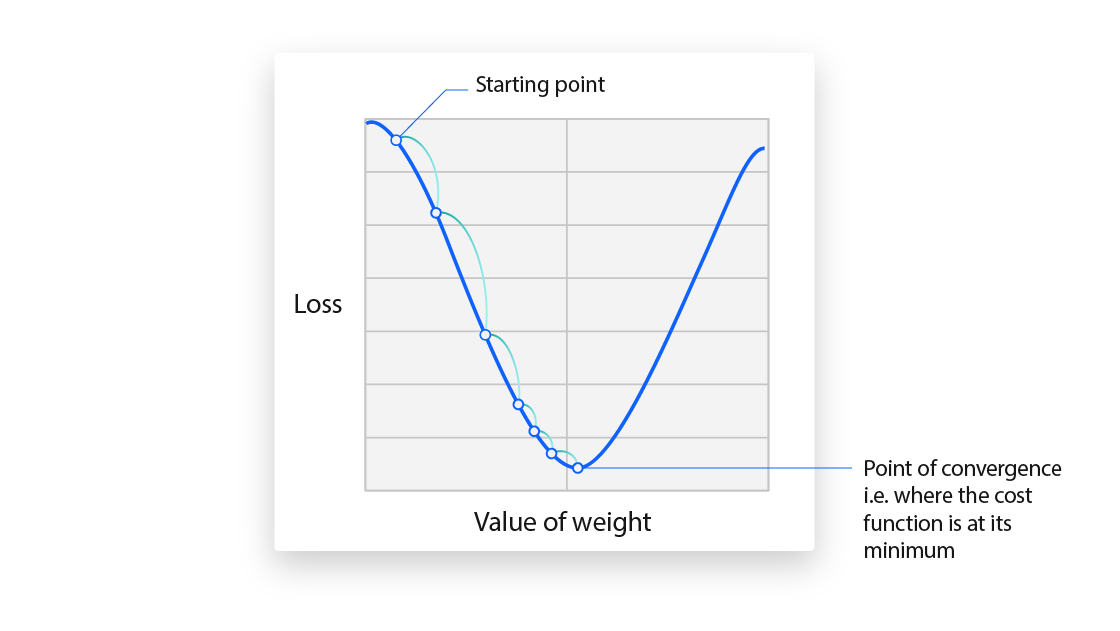
* *i*representa el índice de la muestra,
* y-hat es el resultado previsto,
* y es el valor real y
* *m* es el número de muestras.

𝐶𝑜𝑠𝑡 𝐹𝑢𝑛𝑐𝑡𝑖𝑜𝑛= 𝑀𝑆𝐸=1/2𝑚 ∑129\_(𝑖=1)^𝑚▒(𝑦 ̂^((𝑖) )−𝑦^((𝑖) ) )^2

En última instancia, el objetivo es minimizar nuestra función de coste para garantizar la corrección del ajuste para cualquier observación dada. A medida que el modelo ajusta sus ponderaciones y sesgos, utiliza la función de coste y el aprendizaje por refuerzo para alcanzar el punto de convergencia o el mínimo local. El proceso mediante el cual el algoritmo ajusta sus ponderaciones se basa en el descenso del gradiente, lo que permite al modelo determinar la dirección que debe tomar para reducir los errores (o minimizar la función de coste). Con cada ejemplo de entrenamiento, los parámetros del modelo se ajustan gradualmente hasta alcanzar el mínimo.

Consulte este [artículo de IBM Developer para obtener una explicación más profunda de los conceptos cuantitativos implicados en las redes neuronales.](https://developer.ibm.com/articles/l-neural/).

La mayoría de las redes neuronales profundas son alimentadas, lo que significa que fluyen en una sola dirección, de la entrada a la salida. Sin embargo, también puede entrenar su modelo mediante retropropagación; es decir, moverse en la dirección opuesta, de la salida a la entrada. La retropropagación nos permite calcular y atribuir el error asociado a cada neurona, lo que nos permite ajustar y encajar adecuadamente los parámetros del modelo o modelos.



tipos de redes neuronales

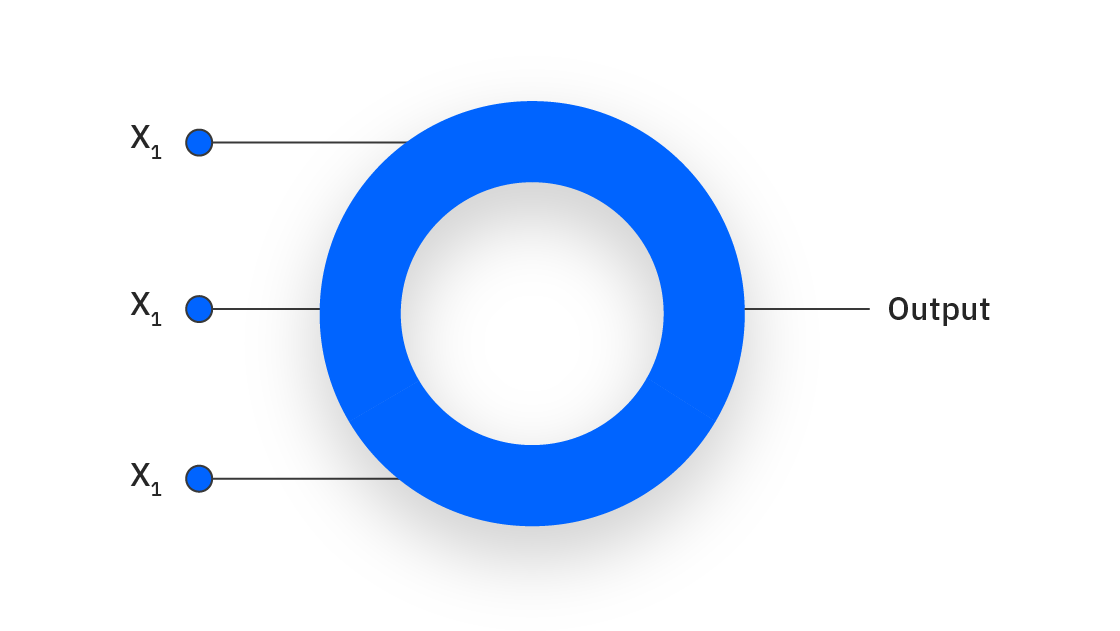
Las redes neuronales se pueden clasificar en distintos tipos, que se utilizan para fines diferentes. Aunque no se trata de una lista exhaustiva de tipos, la siguiente sería representativa de los tipos más comunes de redes neuronales que encontrará para sus casos de uso habituales:

El perceptrón es la red neuronal más antigua, creada por Frank Rosenblatt en 1958.

En este artículo nos hemos centrado principalmente en las redes neuronales de avance o perceptrones multicapa (MLP). Se componen de una capa de entrada (o input), una o varias capas ocultas y una capa de salida (u output). Aunque estas redes neuronales también suelen denominarse MLP, es importante señalar que en realidad están formadas por neuronas sigmoidales, no por perceptrones, ya que la mayoría de los problemas del mundo real no son lineales. Los datos generalmente se introducen en estos modelos para entrenarlos, y son la base para la visión artificial, el [procesamiento del lenguaje natural](https://www.ibm.com/es-es/think/topics/natural-language-processing) y otras redes neuronales.

Las [redes neuronales convolucionales (CNN)](https://www.ibm.com/es-es/think/topics/convolutional-neural-networks) son similares a las redes de propagación hacia adelante (feedforward), pero generalmente se utilizan para el reconocimiento de imágenes, el reconocimiento de patrones y/o la visión artificial. Estas redes aprovechan principios del álgebra lineal, en particular la multiplicación de matrices, para identificar patrones dentro de una imagen.

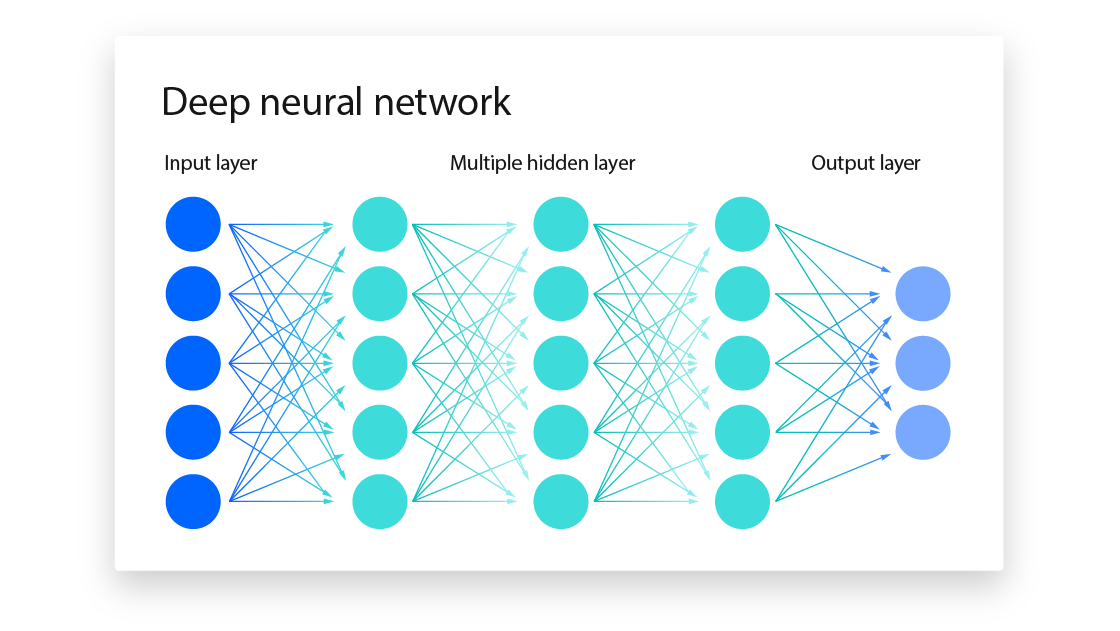
[Las redes neuronales recurrentes (RNN)](https://www.ibm.com/es-es/think/topics/recurrent-neural-networks) se identifican por sus bucles de feedback. Estos algoritmos de aprendizaje se utilizan principalmente cuando se emplean datos de series temporales para hacer predicciones sobre resultados futuros, como predicciones bursátiles o previsiones de ventas.



Redes neuronales frente a deep learning

El deep learning y las redes neuronales tienden a utilizarse indistintamente en la conversación, lo que puede ser confuso. Por lo tanto, cabe señalar que el término "deep" (profundo) en el deep learning se refiere únicamente a la profundidad de las capas de una red neuronal. Una red neuronal que conste de más de tres capas, que incluirían las entradas y la salida, puede considerarse un algoritmo de deep learning. Una red neuronal que solo tiene dos o tres capas es una red neuronal básica.

Si quiere obtener más información sobre las diferencias entre las redes neuronales y otras formas de inteligencia artificial, como el machine learning, lea la entrada del blog "[IA, machine learning, deep learning y redes neuronales: ¿cuál es la diferencia?](https://www.ibm.com/es-es/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks)"



Historia de las redes neuronales

La historia de las redes neuronales es más larga de lo que la mayoría de la gente cree. Aunque la idea de "una máquina que piensa" se remonta a los antiguos griegos, nos centraremos en los acontecimientos clave que condujeron a la evolución del pensamiento en torno a las redes neuronales, cuya popularidad ha sufrido altibajos a lo largo de los años:

**1943:** Warren S. McCulloch y Valeriano Pitts publicaron "[Cálculo lógico de las ideas inherentes a la actividad nerviosa](https://web.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf)" ["A logical calculus of the ideas immanent in nervous activity"] (enlace externo a ibm.com). El objetivo de esta investigación era comprender cómo el cerebro humano podía producir patrones complejos a partir de células cerebrales conectadas, es decir, neuronas. Una de las principales ideas que surgieron de este trabajo fue la comparación de las neuronas con un umbral binario de la lógica booleana (es decir, 0/1 o verdadero/falso).

**1958:** Se atribuye a Frank Rosenblatt el desarrollo del perceptrón, que se documenta en su investigación "[El perceptrón: Un modelo probabilístico de almacenamiento y organización de la información en el cerebro](https://www.academia.edu/60542953/The_perceptron_a_probabilistic_model_for_information_storage_and_organization_in_the_brain)" ["The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain"] (enlace externo a ibm.com). Su trabajo sobre McCulloch y Pitt supuso un avance al introducir pesos en la ecuación. Gracias a un IBM 704, Rosenblatt consiguió que un ordenador aprendiera a distinguir las cartas marcadas a la izquierda de las marcadas a la derecha.

**1974:**Aunque numerosos investigadores contribuyeron a la idea de la retropropagación, Paul Werbos fue la primera persona en EE. UU. en incluir su aplicación en redes neuronales dentro de su [tesis doctoral](https://www.researchgate.net/publication/35657389_Beyond_regression_new_tools_for_prediction_and_analysis_in_the_behavioral_sciences) (enlace externo a ibm.com).

**1989:** Yann LeCun publicó un [artículo](https://direct.mit.edu/neco/article-abstract/1/4/541/5515/Backpropagation-Applied-to-Handwritten-Zip-Code) (enlace externo a ibm.com) en el que se ilustra cómo el uso de restricciones en la retropropagación y su integración en la arquitectura de la red neuronal pueden servir para entrenar algoritmos. Esta investigación aprovechó con éxito una red neuronal para reconocer dígitos de códig