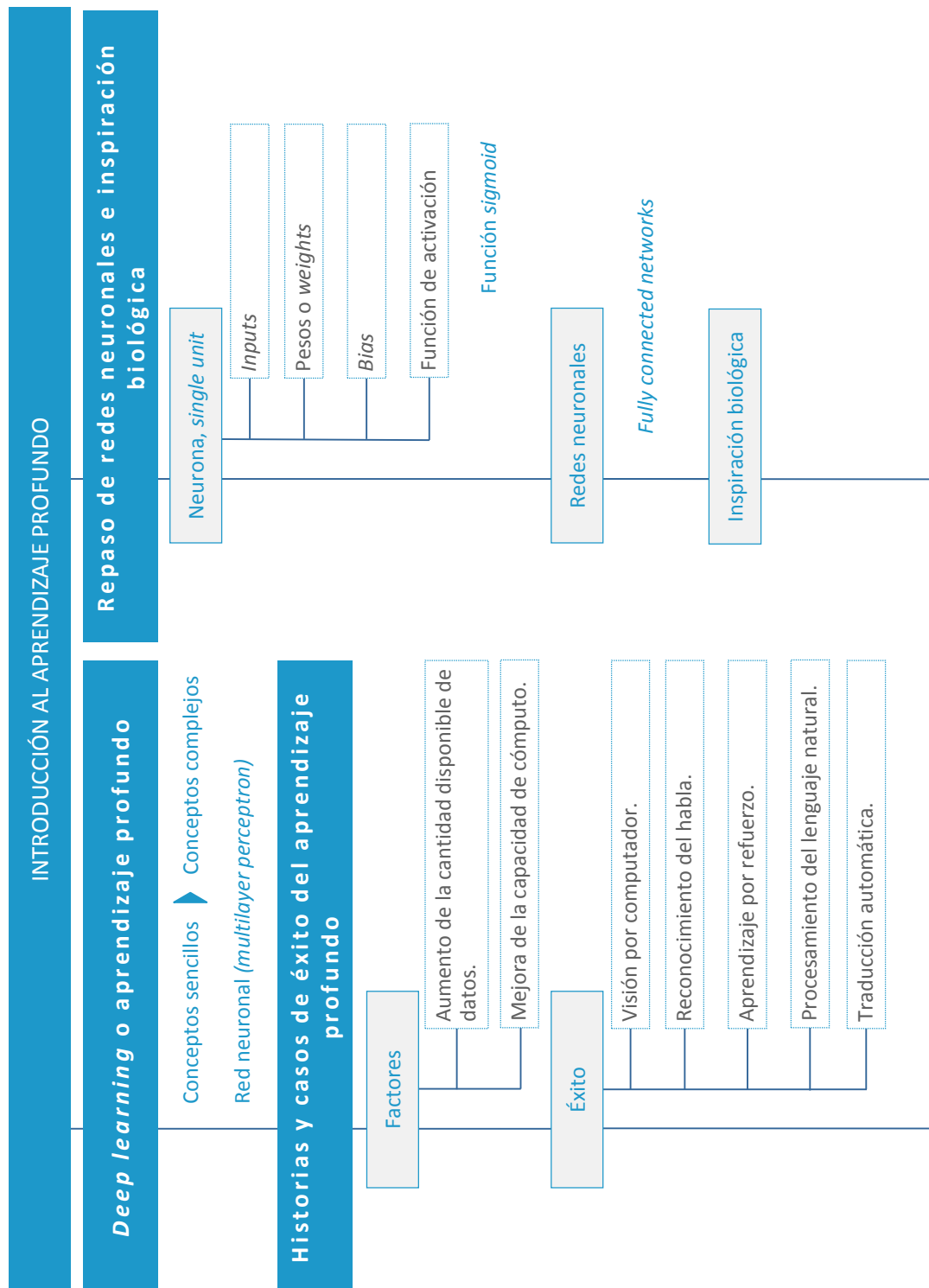


Sistemas Cognitivos Artificiales

Introducción al aprendizaje profundo

Índice

Esquema	3
Ideas clave	4
1.1. ¿Cómo estudiar este tema?	4
1.2. Introducción al aprendizaje profundo	5
1.3. Historia y casos de éxito del aprendizaje profundo	10
1.4. Repaso de redes neuronales e inspiración biológica	15
1.5. Referencias bibliográficas	21
Lo + recomendado	23
Test	24



1.1. ¿Cómo estudiar este tema?

Para estudiar este tema deberás leer las **Ideas clave** que se desarrollan a continuación.

Este tema servirá como introducción al aprendizaje profundo y, en general, a la asignatura. Por ello, primero estudiaremos una primera visión de qué es el aprendizaje profundo o *deep learning* y cómo encaja dentro de la inteligencia artificial (IA) y el aprendizaje automático. A continuación, comentaremos su evolución histórica y cómo este ya ha existido, en cierta manera, desde hace décadas. Finalmente, haremos un breve repaso de qué es una red neuronal y la formulación matemática de una neurona, estableciendo un paralelismo con la inspiración biológica del campo.

Si bien este tema es introductorio, hablaremos de varios conceptos significativos:

- ▶ En el apartado «Introducción al aprendizaje profundo» es importante relacionar dónde se encuentra el *deep learning* dentro de la IA y su relación con el aprendizaje automático que se estudió en el primer trimestre.
- ▶ En el apartado «Historia y casos de éxito del aprendizaje profundo» es importante entender de dónde vienen el aprendizaje profundo y, en especial, los factores que han llevado a su éxito, así como varias de las aplicaciones donde ha triunfado.
- ▶ Finalmente, en el último apartado, es importante comprender la formulación matemática de una neurona y recordar las distintas partes de una red neuronal, lo cual será la base de estudio del siguiente tema.

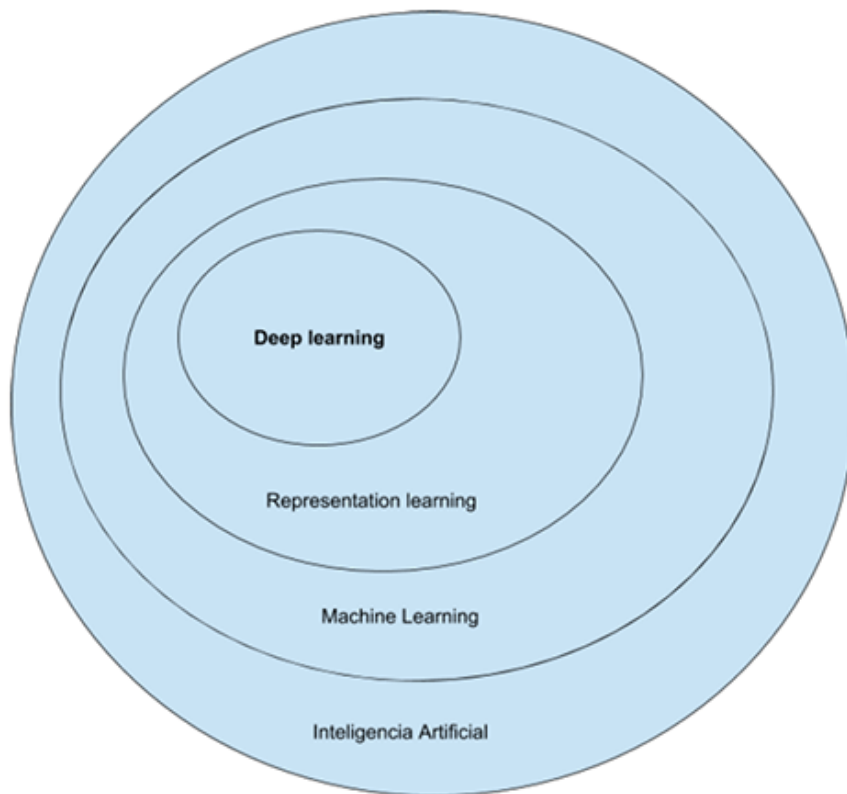


Figura 1. Gráfico de la ubicación del *deep learning* o aprendizaje profundo.
Fuente: adaptado de Goodfellow, Bengio y Courville, 2016.

1.2. Introducción al aprendizaje profundo

Durante años, la inteligencia artificial se fundamentó principalmente en complejos sistemas de reglas y conocimiento definidos por humanos. Estos sistemas eran capaces de resolver problemas basados en reglas formales que resultaban difíciles para las personas, pero relativamente sencillos para un ordenador. Curiosamente, las **tareas abstractas y formales** suelen ser fáciles de resolver para una máquina, mientras que las tareas más sencillas, que resultan intuitivas y directas para una persona (como reconocer una cara) son a su vez las más difíciles de ejecutar para un ordenador.

Consideremos por ejemplo el juego del ajedrez, uno de los primeros éxitos de la IA al ser capaz de vencer a los humanos. Aunque esto fue una gran hazaña, el hecho de

que el ajedrez pueda ser descrito por una serie de reglas formales en un entorno perfectamente definido facilitó su solución mediante la inteligencia artificial.

Sin embargo, mientras los ordenadores eran capaces de vencernos fácilmente al ajedrez, sus capacidades para realizar tareas casi automáticas para nosotros, como entender una frase o describir qué hay en una imagen, eran muy limitadas. Las dificultades en resolver estas tareas que presentaban los sistemas de conocimiento y de reglas provistos por humanos dio paso al *machine learning* o aprendizaje automático, una nueva área dentro de la inteligencia artificial.

En el aprendizaje automático, los sistemas de inteligencia artificial obtienen su propio conocimiento mediante la **extracción de patrones** a partir de la experiencia, dada por datos.

Por ejemplo, un sistema sencillo de aprendizaje automático puede ser entrenado mediante e-mails que son o no *spam*. A partir de los e-mails y de la clase a la que pertenezcan (ser spam o no), un algoritmo puede buscar automáticamente los patrones del lenguaje comunes en los distintos tipos de correo y, por tanto, «aprender» a clasificarlos.

La **representación de los datos** es crucial para el éxito de los algoritmos de *machine learning* —cosa que, curiosamente, también ocurre en la vida real: imagínese el lector el sufrimiento de hacer operaciones aritméticas en números romanos—. Muchas tareas de aprendizaje automático dependen de una buena representación de los datos; una representación que, en muchos de los casos, es de nuevo diseñada por humanos.

Por ejemplo, un sistema de *machine learning* basado en *logistic regression* es capaz de predecir si una parturienta debería recibir una cesárea dada una serie de características seleccionadas por los doctores (Mor-Yosef et al., 1990), utilizando estas características para buscar patrones de correlación con la práctica de una cesárea. Sin embargo, si este algoritmo recibiera directamente los píxeles de la

imagen de resonancia magnética utilizada por el médico para obtener esas características, la capacidad de predicción sería prácticamente nula.

De este modo, muchos problemas de inteligencia artificial pueden solucionarse mediante la **definición de una serie de características en forma de datos (*features*)** que son pasadas a un algoritmo de *machine learning*. No obstante, en muchas ocasiones es complicado saber qué características deberían usarse. Por ejemplo, si queremos detectar un camión en una foto, podríamos utilizar la presencia de ruedas y de un remolque en una foto como *features*. Pero, ¿cómo describimos una rueda o un remolque en términos de píxeles? La rueda podría estar girada, tumbada o tapada; además, los camiones tienen un número distinto de ruedas; igualmente, un camión podría no llevar remolque o el remolque podría ser pequeño o muy grande... Como se ve, esta aproximación al problema se vuelve complicada y requiere de mucho esfuerzo en la definición de un conjunto adecuado de *features*, lo que se conoce como *feature engineering*.

La solución a este problema sería el área denominada ***representation learning***: que el sistema de *machine learning* no solo aprenda la solución de un problema a partir de una representación dada (*features*), sino que también aprenda la representación adecuada a partir de los datos en bruto (por ejemplo, una imagen). Las representaciones aprendidas tendrían un mejor comportamiento que las diseñadas a mano.

Sin embargo, en muchas ocasiones el problema de obtener una representación adecuada parece igual de complicado que la solución del problema en sí. Volviendo al ejemplo de reconocer camiones, podemos pensar que una buena forma de detectar un camión podría ser reconocer la presencia de una o más ruedas, de un remolque o de otros elementos como una cabina, pero la detección de estos elementos parece un problema igual de complejo que la detección de un camión, y requiere a su vez de un nivel de comprensión casi humano. ¿Cómo diseñar un sistema que sea capaz de obtener estas representaciones de manera automática? El ***deep learning, aprendizaje profundo*** en español, intenta solucionar este problema

mediante la creación de representaciones expresadas en términos de otras representaciones más sencillas.

El **aprendizaje profundo** pretende conseguir que los ordenadores sean capaces de construir automáticamente conceptos complejos a partir de conceptos sencillos.

El ejemplo por excelencia de un modelo de *deep learning* es la **red neuronal** o perceptrón multicapa (*multilayer perceptron*). Una red neuronal es básicamente una función matemática definida por nodos en capas donde cada nodo es una función más simple que depende de los nodos de la capa anterior. Puede pensarse, pues, que cada nodo utiliza los conceptos simples aprendidos en capas anteriores para desarrollar un concepto más complejo a partir de ellos.

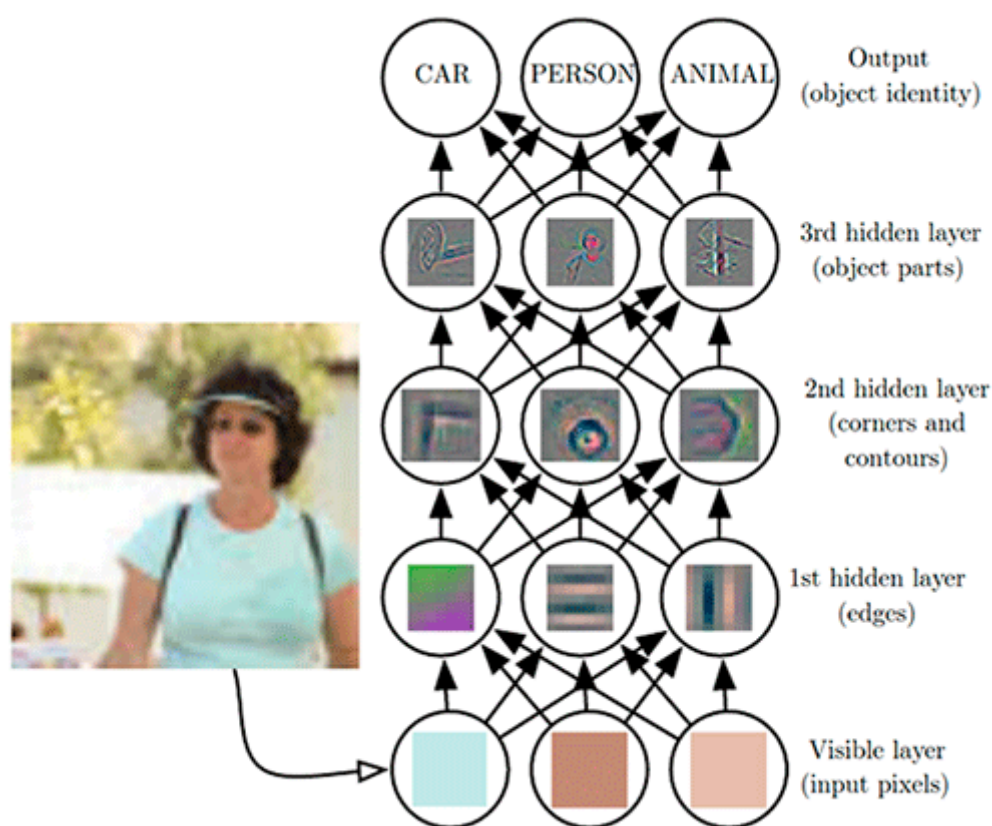


Figura 2. Ejemplo de una red neuronal aprendiendo conceptos complejos.
Fuente: Goodfellow, Bengio y Courville, 2016.

En la imagen anterior (figura 2) puede verse un ejemplo de cómo las redes neuronales aprenden conceptos más abstractos y complejos, aprendiendo efectivamente una representación de los datos adecuada. Resolver el problema en la figura (identificar a una persona, coche o animal) con una función directa a partir de los píxeles de la imagen sería casi imposible. Una red neuronal divide la tarea en una serie de problemas más sencillos a través de sus distintas capas. En el ejemplo, ante la presencia de imágenes:

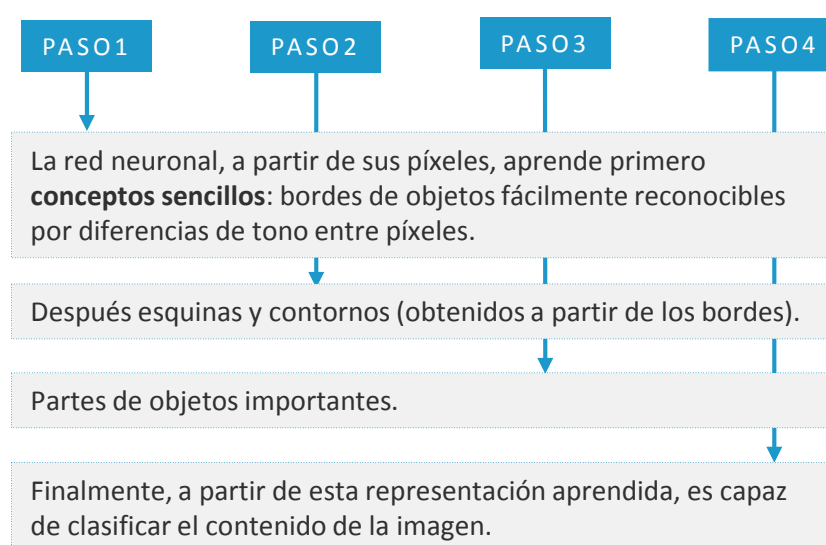


Figura 3. Ejemplo de cómo una red neuronal aprende de conceptos sencillos a complejos.

Esta jerarquía de conceptos aprendida por la red neuronal nos puede ayudar a entender el porqué del nombre de «*deep* o profundo». Para obtener una representación exitosa del problema, se necesita una jerarquía grande o profunda de conceptos. Tradicionalmente se ha asociado esta profundidad con el número de capas en una red neuronal, sin embargo, no hay un número a partir del cual una red neuronal sea considerada «profunda», aunque muchas de las arquitecturas más famosas se han caracterizado por añadir más y más capas.

En resumen, el aprendizaje profundo o *deep learning* es un tipo de aprendizaje automático o *machine learning*, que a su vez es un área dentro de la inteligencia artificial. Dentro del aprendizaje automático, el aprendizaje profundo se basa en la

representación del problema a tratar en una jerarquía de conceptos o abstracciones cada vez más complejos y obtenidos a partir de conceptos más sencillos.

1.3. Historia y casos de éxito del aprendizaje profundo

Si bien el aprendizaje profundo está viviendo su edad dorada ahora mismo, representando los mayores avances de la inteligencia artificial en los últimos tiempos, es importante resaltar que no es un área de estudio nuevo, sino que ha existido en la literatura con distintos nombres ya desde hace varias décadas. Diversos factores han contribuido a su resurgimiento estos últimos años, en especial la disponibilidad de muchos más datos para entrenar modelos y el aumento de la capacidad de cómputo.

Breve historia del *deep learning*

El aprendizaje profundo ha tenido varios nombres a lo largo de la historia, incluyendo *cybernetics*, conexionismo o simplemente redes neuronales artificiales. La inspiración biológica ha sido una constante en el campo, partiendo de la idea de que el cerebro es un ejemplo de aprendizaje e inteligencia: si podemos replicar su funcionamiento con algoritmos, podremos obtener sistemas inteligentes. Si bien la inspiración está ahí, lo cierto es que en la actualidad el campo no está guiado por la neurociencia, nuestra comprensión del cerebro no es lo suficientemente avanzada.

Primeros modelos

La neurona de McCulloch-Pitts (originaria de 1943) y el **perceptrón** (1958) son los primeros ejemplos de modelos que, de una manera muy simplificada, están inspirados en el funcionamiento de una neurona aislada. Del mismo modo, el sistema

ADALINE, *Adaptive Linear Element* (1960), introdujo un sistema de entrenamiento muy parecido al *stochastic gradient descent* que se utiliza ahora para entrenar redes neuronales.

Años 80

En los años 80 surgió una nueva ola de interés en redes neuronales, principalmente gracias a un movimiento interdisciplinar llamado **conexionismo**, bajo el área de la ciencia cognitiva. La idea central del **conexionismo** es que un gran número de pequeñas unidades de cómputo puede alcanzar un comportamiento inteligente mediante su conexión en una red. La idea del algoritmo de *back-propagation*, clave en el entrenamiento de redes neuronales, data de esta época, así como la idea de las representaciones distribuidas, ambos conceptos que veremos durante el curso.

Años 90

Nuevos avances se produjeron en el área de problemas con secuencias. Las redes neuronales secuenciales LSTM, muy utilizadas en la actualidad, datan por ejemplo de 1997 (Hochreiter y Schmidhuber, 1997). Sin embargo, en esta época el interés por las redes neuronales volvió a decaer ante la dificultad de obtener buenos resultados. Otros algoritmos y técnicas de aprendizaje automático, como las *support vector machines*, coparon durante varios años el interés de la comunidad, convirtiéndose en las soluciones estándar a la mayor parte de los problemas tratables por el *machine learning*.

A partir de 2006

Sin embargo, a partir de 2006, varios avances conseguidos por investigadores como Geoffrey Hinton, Yoshua Bengio y Yann LeCun volvieron a situar a las redes neuronales en el centro del tablero. Año a año, estas empezaron a superar en muchas áreas a las otras técnicas de aprendizaje profundo, en ocasiones por un gran margen, y consiguieron resolver problemas que se pensaba llevaría años o décadas solucionar.

El hecho de que fuera posible entrenar con éxito redes más profundas (con más capas) llevó poco a poco al famoso nombre actual de *deep learning*. Sin embargo, ¿cómo es posible que, cuando gran parte de los algoritmos y las técnicas de entrenamiento ya estaban inventados, llevara tanto tiempo al aprendizaje profundo a afianzarse como una parte fundamental y disruptiva del mundo del *machine learning*?

Factores del éxito del *deep learning*

Si bien las redes neuronales eran ya usadas con éxito antes de la explosión del *deep learning*, se consideraba en general que eran muy complicadas de entrenar. Dos factores pueden explicar su resurgimiento en los últimos años: la disponibilidad de muchos más datos y la mejora de la capacidad de cómputo.

El **aumento de la cantidad disponible de datos** en una sociedad tan digital como la nuestra ha permitido contar con más recursos para entrenar los algoritmos de *machine learning*. En los años 80 y 90 los investigadores utilizaban datasets de centenares o varios millares de puntos. En la actualidad, los datasets en los que el aprendizaje profundo está demostrando sus capacidades llegan a constar de millones de puntos. Las redes neuronales son algoritmos complejos, y una de las razones por las que podían no estar alcanzando todo su potencial era por no contar con la cantidad adecuada de datos para su entrenamiento.

Por otro lado, la **mejora de la capacidad de cómputo**, inherente al avance de la informática, ha permitido que la experimentación requiera menos tiempo y que se puedan ejecutar modelos más complicados, así como modelos con más datos. Las redes neuronales con mayor número de neuronas son capaces de solucionar problemas más complejos y desarrollar mejores representaciones de los datos brutos. De manera similar, otros avances en *hardware* como la utilización de tarjetas gráficas para el entrenamiento de redes neuronales han ayudado en gran medida al desarrollo del campo.

Casos de éxito del *deep learning*

El éxito del aprendizaje profundo en la actualidad ha hecho que cada vez se aplique a más problemas, incluso en campos distintos de donde las soluciones de inteligencia artificial se han aplicado tradicionalmente. Sin embargo, hay una serie de casos especiales donde el *deep learning* ha mejorado notablemente, incluso podría decirse revolucionado los resultados, contribuyendo a su éxito presente.

Probablemente el área más famosa donde el aprendizaje profundo ha funcionado de manera espectacular sea la de **visión por computador** (*computer vision*) y, en particular, el reconocimiento de objetos. En 2012, una *convolutional neural network* ganó por primera vez y con gran margen la competición de reconocimiento de objetos ImageNet, consistente en la clasificación de 1000 objetos en un conjunto de más de un millón de imágenes. Durante los años siguientes, nuevas redes profundas aumentaron enormemente la capacidad de clasificar imágenes hasta el punto de que el problema se considera prácticamente resuelto. Otras áreas de la visión por computador donde el *deep learning* ha tenido gran éxito son la detección de objetos en imágenes y la segmentación.

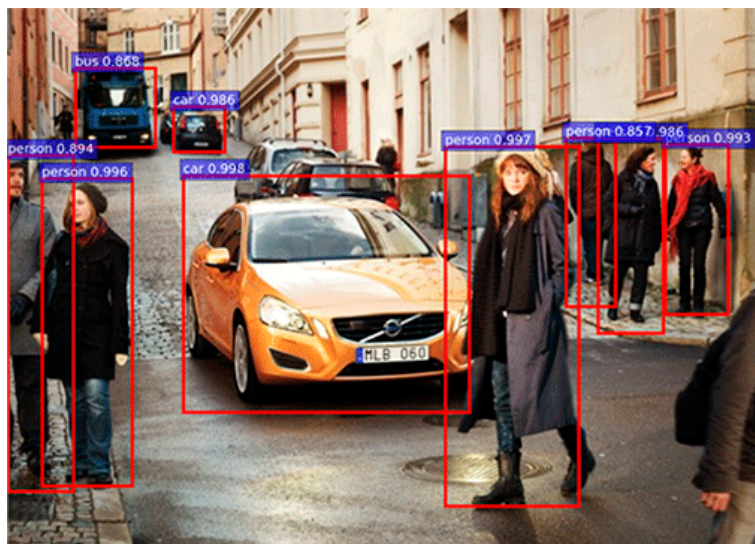


Figura 4. Ejemplo de visión por computador con *deep learning*.

Fuente: <https://bigsnarf.wordpress.com/2016/11/07/faster-r-cnn-pedestrian-and-car-detection/>

En la figura 4 vemos que los sistemas de detección de peatones, desarrollados con tecnologías de visión por computador con *deep learning*, son clave para las nuevas tecnologías de coches sin conductor.

Otra área donde el *deep learning* ha mejorado ostensiblemente los resultados anteriores es el del **reconocimiento de habla** (*speech recognition*). Prácticamente todos los asistentes digitales que utilizamos en nuestros móviles están programados mediante redes neuronales para reconocer lo que decimos.

Asimismo, el área de **aprendizaje por refuerzo** (*reinforcement learning*) ha visto también grandes mejoras de la mano del aprendizaje profundo. DeepMind, parte de Alphabet, demostró cómo un sistema de *reinforcement learning* puede aprender a jugar a videojuegos de Atari directamente a partir de las partidas. En un hito para la historia de la inteligencia artificial, otro sistema desarrollado por esta compañía consiguió derrotar al campeón mundial de *go* en 2016. El *go*, un juego tremendamente complejo, no se consideraba un candidato a ser resuelto por técnicas de inteligencia artificial en el futuro inmediato.

Por último, el **procesamiento del lenguaje natural** (*natural language processing*) es otra área en la que, aunque de manera un poco más tardía, el aprendizaje profundo está empezando a dar buenos resultados. Problemas clásicos como *part of speech tagging* o análisis de sentimientos han obtenido nuevas soluciones de vanguardia basadas en redes neuronales. Igualmente, los sistemas de **traducción automática** como el archiconocido Google Translate se ejecutan ahora sobre soluciones de aprendizaje profundo.

Es importante mencionar que, aunque el *deep learning* está revolucionando la inteligencia artificial y ha adquirido una gran presencia hasta en la prensa y el mundo de los negocios, este no es más que otra técnica dentro del mundo del aprendizaje automático. No todos los problemas son solucionados de manera óptima por redes profundas. Otros algoritmos ofrecen soluciones a veces más efectivas, sencillas y rápidas, si bien es cierto que las redes neuronales profundas son probablemente la

herramienta más versátil y potente en el campo de la inteligencia artificial en la actualidad, esto no quiere decir que el resto hayan quedado automáticamente desfasadas o que incluso en unos años no se encuentre una nueva técnica que adquiera todo el protagonismo.

1.4. Repaso de redes neuronales e inspiración biológica

En este apartado repasaremos las nociones básicas de redes neuronales que ya conocemos de la asignatura de *Aprendizaje Automático*. Nótese que, a partir de aquí, la terminología técnica será escrita mayoritariamente en inglés, ya que este es el idioma que se utiliza normalmente en la literatura y la traducción de muchos términos no está estandarizada, al igual que hace más difícil la búsqueda en Internet de recursos relacionados.

Neurona, *single unit*

Recordemos primero cómo funciona una simple neurona. Tal y como vemos en la figura 5, una neurona o nodo en una red neuronal consiste en:

- ▶ Una serie de **inputs** ($a_1, a_2, a_3 \dots a_n$ en el diagrama).
- ▶ Una serie de pesos o **weights** (w en el diagrama).
- ▶ Un **bias** (b).
- ▶ Una **función de activación** (σ), que se aplica sobre la suma del **bias** con el producto de cada **input** por su peso correspondiente.

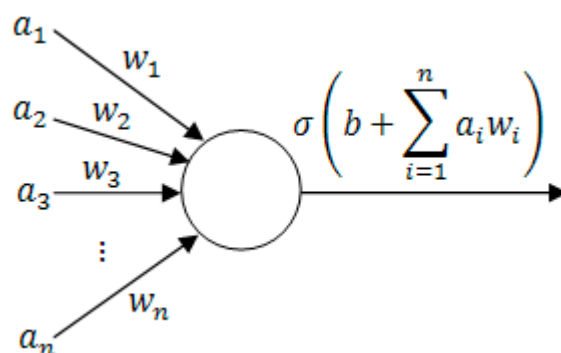


Figura 5. Expresión matemática de una neurona en una red neuronal.

Fuente: <https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc>

La función de activación se conoce también como **non-linearity**. Durante el curso, veremos varias funciones que pueden ser utilizadas aquí; una de las opciones más tradicionales, que asemeja a una neurona al algoritmo de *logistic regression*, es la **función sigmoid** (tradicionalmente escrita, de hecho, como σ).

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

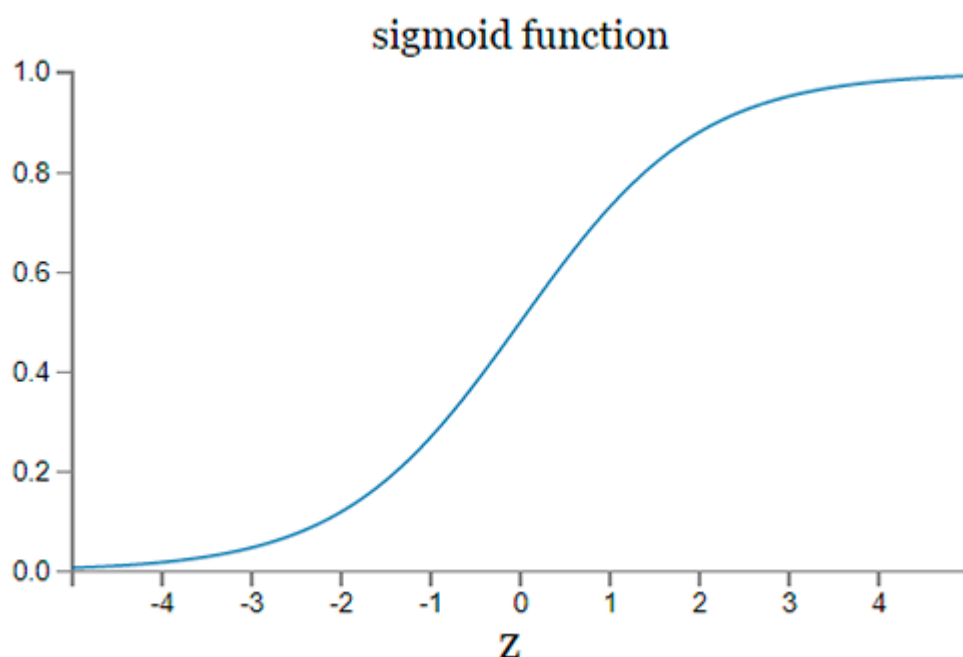


Figura 6. Función *sigmoid*.

Fuente: <http://neuralnetworksanddeeplearning.com/chap1.html>

Como podemos comprobar, la función toma valores entre 0 y 1: tiende a 1 para valores cercanos a infinito y a 0 para valores cercanos a menos infinito. Es por eso que esta función se utiliza normalmente para modelar probabilidades.

Intentemos intuir el funcionamiento de una neurona, en este caso como si de un algoritmo aislado se tratara o, visto de otro modo, como si se tratara de un simple algoritmo de *logistic regression*.

Ejemplo:

Supongamos que, en un ejemplo muy sencillo, queremos que el *output* de nuestra neurona modele la probabilidad de que nos vayamos de vacaciones.

Digamos que las *features* que seleccionamos para modelar este problema como las inputs de la neurona son:

a_1 = cantidad de dinero en el banco en miles de euros (para nuestro ejemplo, 5).

a_2 = número de amigos que nos acompañan (para nuestro ejemplo, 2)

a_3 = horas de trabajo pendiente (para nuestro ejemplo, 1).

Como vemos en la fórmula, cada *input* es multiplicado por su propio *weight*. Podemos entender los pesos o *weights* como cuánto ponderamos ese *input* en la salida final. Asumamos que, en nuestro caso particular, esta **ponderación** es:

$$w_1 = 2$$

$$w_2 = 4$$

$$w_3 = -3$$

En el caso particular del ejemplo descrito, el valor que recibe la **función de activación** sería: $2 \cdot 5 + 4 \cdot 2 - 3 \cdot 1 = 15$, con $\sigma(15)$ valiendo prácticamente 1.

De modo que, para los pesos que modelan nuestro comportamiento y ante los *inputs* de nuestra situación particular, obtenemos que la neurona nos asigna una probabilidad casi 1 de irnos de vacaciones.

No hemos hablado del *bias*. Este término, si bien no es totalmente imprescindible para entender el funcionamiento de una neurona, representa un **valor constante** a tener en cuenta que forma parte del modelado del problema. En nuestro ejemplo, podríamos entenderlo como un valor inherente al modelo que define nuestra predisposición a irnos o no de vacaciones: supongamos que, en general, nos encanta irnos de vacaciones en todo momento independientemente de las condiciones. En ese caso, el *bias* (b) sería un valor positivo que ayudaría a llevar la probabilidad de irnos de vacaciones a valores más altos; en términos estadísticos tendríamos un *bias* (una predisposición) a irnos de vacaciones.

De este modo, los *bias* permiten a una neurona modelar un valor constante e independiente de los *inputs* que influye en la salida de la misma.

Este ejemplo nos da una noción de cómo funciona una neurona por medio de la ponderación de sus *inputs* y de asignar un *output* o valor de salida. Como vemos, a valores positivos obtenemos salidas cercanas a 1, mientras que a valores negativos, salidas cercanas a 0. Por supuesto, la idea aquí es que los pesos (w) y el *bias* (b) sean aprendidos a partir de los datos y no tengan que conocerse de antemano.

Podríamos recolectar los datos de momentos en los que nos fuimos o no de vacaciones en el pasado y permitir que la neurona asignara esos pesos de manera automática. Es importante destacar también que, aunque en este caso la salida de la neurona puede entenderse como una probabilidad, esto no es una norma. De hecho, veremos funciones de activación distintas de *sigmoid* con valores fuera del intervalo $[0, 1]$.

Redes neuronales

Una sola neurona representa una **función lineal**. Si bien los modelos lineales son bastante poderosos, tienen ciertas limitaciones, la más clara de ellas es que no permiten la interacción de distintos *inputs*. Como se ve en la ecuación, cada input está multiplicado por su respectivo peso, pero no interactúan entre ellos.

En la introducción histórica ya vimos que la corriente conexionista defendía que un gran número de pequeñas unidades de cómputo pueden alcanzar un comportamiento inteligente mediante su conexión en una red. De manera similar, juntando varias neuronas podemos crear una red neuronal, capaz de resolver problemas más complejos y de permitir interacciones entre los diversos *inputs* mediante la conexión de distintas neuronas.

Una red neuronal, también llamada *multilayer perceptron* (MLP), es un conjunto de neuronas simples situadas en capas.

Normalmente, al hablar de redes neuronales nos referimos a *fully connected networks*: redes donde todas las neuronas o nodos de una capa están conectados con todas las neuronas de la capa anterior.

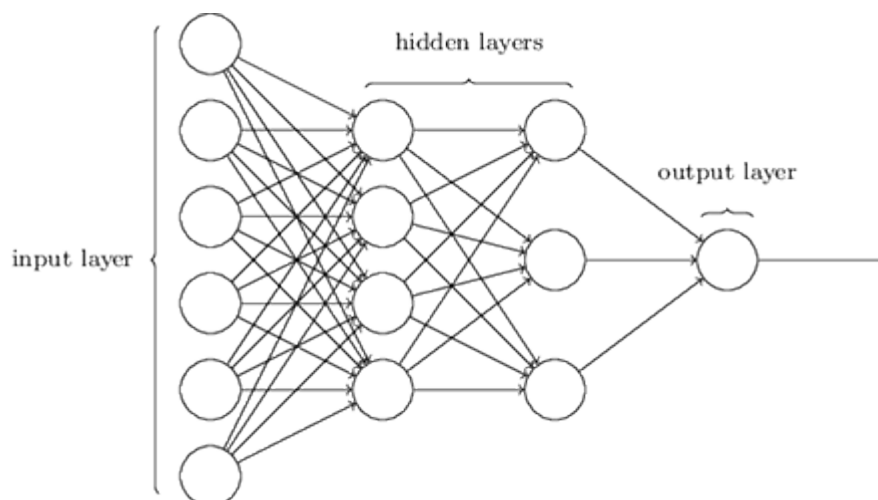


Figura 7. Red neuronal.

Fuente: <http://neuralnetworksanddeeplearning.com/chap1.html>

Como se ve en imagen anterior (figura 7), las distintas neuronas se colocan en capas y reciben como *inputs* la salida de todas las neuronas de la capa anterior. Capa por capa, los *outputs* de una neurona son una fuente de datos para las neuronas de la siguiente capa. Vemos aquí claramente la idea clave del *deep learning*: obtener representaciones cada vez más complejas y basadas en conceptos simples.

Una red neuronal tiene varios tipos distintos de capas:

- ▶ La primera capa, la *input layer*, es la capa que recoge los datos a tratar.
- ▶ La última capa, *output layer*, representa la salida de la red.
- ▶ Finalmente, las capas interiores se denominan *hidden layers*.

Entrenar una red neuronal significa encontrar los valores óptimos de los pesos (w) y *bias* (b) de todas las neuronas que mejor describen los datos en términos del *output* deseado. Por ejemplo, y como ya hemos visto: el *output* podría ser la probabilidad de que una imagen sea de un camión y el *input*, todos los píxeles que componen esa imagen.

Otro punto a tratar aquí es que una red del tipo de la figura 7 se denomina *feed-forward network*. Esto es debido a que no hay bucles en la red y la información siempre fluye hacia delante. Durante el curso, veremos otro tipo de redes que no siguen este paradigma, llamadas *recurrent neural networks*.

Inspiración biológica

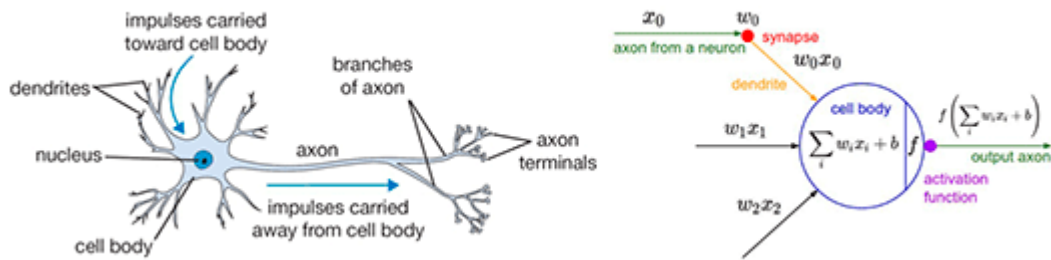


Figura 8. Neurona real y neurona de una red neuronal.

Fuente: <https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc>

La inspiración biológica de las redes neuronales es clara como vemos en la figura 8. Una neurona común obtiene impulsos de otras neuronas a través de las dendritas y transmite a su vez un impulso a otras neuronas a través del axón. Esto es equivalente a los *inputs* y *outputs* que hemos visto. De manera similar al cerebro, las redes neuronales se componen de muchas neuronas y sus conexiones, pero es importante volver a recalcar que las similitudes acaban pronto.

La realidad es que hay muchos tipos distintos de neuronas en el cerebro y que estas no actúan con las funciones de activación o las arquitecturas de red normalmente utilizadas en el campo del *deep learning*. Por ello, pese a que la neurociencia ha servido de clara inspiración al campo del aprendizaje profundo, ambas disciplinas científicas siguen caminos y objetivos separados.

1.5. Referencias bibliográficas

Goodfellow, I., Bengio, Y. y Courville, A. (2016). *The Deep Learning Book*. Cambridge (Estados Unidos): The MIT Press.

Hochreiter, S. y Schmidhuber, J. (1997). Long Short-Term Memory. *Journal Neural Computation*, 9(8), 1735-1780.

Mor-Yosef, S. et al. (1990). Vaginal Delivery following One Previous Cesarean Birth: Nation Wide Survey. *The Journal of Obstetrics and Gynaecology Research*, 16(1), 33-37. Recuperado de <https://obgyn.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1447-0756.1990.tb00212.x>

Lo + recomendado

No dejes de leer

The Deep Learning Book

Goodfellow, I., Bengio, Y. y Courville, A. (2016). *The Deep Learning Book*. Cambridge (Estados Unidos): The MIT Press.

Parte de este tema está basado en este libro, escrito por varios reconocidos expertos del campo, incluyendo a uno de sus «padres», Yoshua Bengio. Este documento puede ser muy útil para los alumnos interesados en profundizar en muchos de los aspectos que veremos en este curso. Nótese, sin embargo, que es un recurso un poco complejo de leer: su profundidad y las descripciones completas lo llevan a ser muy riguroso y algo denso en cuanto a matemática se refiere.

Accede al libro a través del aula virtual o desde la siguiente dirección web:

<http://www.deeplearningbook.org/>

1. El aprendizaje profundo (marca todas las respuestas correctas):
 - A. Es una técnica dentro del aprendizaje automático.
 - B. Es un área de la inteligencia artificial.
 - C. Está basado en reglas formales diseñadas por humanos.
 - D. Depende de bases de conocimiento diseñadas por humanos.
 - E. Se basa en entender conceptos cada vez más complejos.

2. ¿Qué factores pueden explicar el concepto «profundo» en el término de aprendizaje profundo? (Marca todas las respuestas correctas):
 - A. Los algoritmos de aprendizaje profundo aprenden con una profundidad nunca vista hasta ahora.
 - B. Las redes neuronales se hacen profundas con la introducción de múltiples capas.
 - C. Las redes neuronales pueden aprender jerarquías complejas de conceptos.
 - D. Los problemas resueltos por el aprendizaje profundo son más complejos y complicados para el hombre.
 - E. El *deep learning* es capaz de realizar razonamientos como una persona.

3. En un problema de clasificación de personas a partir de imágenes con aprendizaje profundo, ¿cuál es la entrada que recibiría la red neuronal?
 - A. Una serie de *features* diseñadas por humanos en la imagen. Por ejemplo, color de cabello, uso de gafas o no, color de ojos, color de piel...
 - B. Una representación de la imagen dada por otros algoritmos.
 - C. Los píxeles de la imagen.

4. El *deep learning* (marca todas las respuestas correctas):
- A. Ha tenido varios nombres a lo largo de la historia.
 - B. Consiste en una serie de técnicas y avances muy recientes.
 - C. Ha tenido siempre una posición predominante en el mundo de la inteligencia artificial.
 - D. Es, tras todos sus avances, la mejor solución para todos los problemas de aprendizaje automático.
5. ¿Qué cantidad de datos es necesaria para entrenar con éxito una red neuronal profunda?
- A. Entre 100 y 1000 *training examples*.
 - B. Entre 1000 y 100 000 *training examples*.
 - C. Más de 100 000 *training examples*.
 - D. La cantidad depende del problema a tratar.
6. ¿Cuáles son algunas de las aplicaciones del *deep learning*? (Marca todas las respuestas correctas):
- A. Visión por computador.
 - B. Traducción automática.
 - C. Agentes que aprenden a jugar a juegos.
 - D. Reconocer el habla.
7. Señala las afirmaciones correctas:
- A. Una red neuronal es un cerebro artificial.
 - B. El aprendizaje profundo está inspirado en el funcionamiento del cerebro, pero las similitudes entre los algoritmos de *deep learning* y el cerebro acaban pronto.
 - C. El aprendizaje profundo está inspirado en el funcionamiento del cerebro y cada nuevo avance en el área lo acerca más a este.

8. En una red neuronal, si una neurona tiene 5 *inputs*, ¿cuántos parámetros la modelan?
- A. 5
 - B. 6
 - C. 7
 - D. 10
 - E. 11
 - F. 12.
9. En una red neuronal *fully connected* con una *input layer* con 10 nodos, una *hidden layer* con 5 nodos y una *output layer* con 10 nodos, ¿cuántos parámetros tiene en total la red neuronal?
- A. 25.
 - B. 110.
 - C. 115.
 - D. 125.
 - E. 220.
10. Marca todas las respuestas correctas acerca del funcionamiento de una red neuronal:
- A. La salida de una neurona es una representación de una probabilidad.
 - B. La *input layer* es una representación de los datos de entrada y no conlleva cálculos.
 - C. Las neuronas se activan con un valor definido por la función de activación.
 - D. En una *feed-forward network*, la información siempre fluye hacia delante.