

Redes neuronales y Deep Learning.



Caso práctico



[@casfatesvano \(CC BY-SA\)](#)

Miguel está terminando sus estudios sobre Inteligencia Artificial, y al igual que sus compañeras y amigas Max y Eva, se dispone a realizar prácticas en una empresa puntera en el sector.

En concreto en dicha empresa se han especializado en desarrollar modelos con las técnicas más actuales en Inteligencia Artificial: redes neuronales y deep learning.

Una de las cosas que más ha impresionado a Miguel los primeros días que ha acudido a las oficinas de esta empresa son los potentes ordenadores y servidores propios que poseen para poder procesar todos los datos que los modelos de aprendizaje profundo requieren.

Su tutora de prácticas le ha dicho que tiene que revisar bien todo lo que le han explicado sobre algoritmos.

A lo largo de esta unidad vas a conocer cómo funciona a grandes rasgos una red neuronal. Profundizarás en las diversas técnicas y algoritmos que más se utilizan en el Aprendizaje Profundo (Deep Learning). Y entenderás cuál es la principal característica que diferencia este tipo de Inteligencia Artificial respecto a las que has estudiado previamente.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

**Materiales formativos de FP Online propiedad del Ministerio de
Educación y Formación Profesional.**
[Aviso Legal](#)

1.- Definición y esquema general de una red neuronal.



Caso práctico

Miguel va a participar en el desarrollo de una Inteligencia Artificial basada en redes neuronales que debe ser capaz de leer textos manuscritos, es decir, escritos a mano.

Antes de iniciar el entrenamiento, van a hacer algunas pruebas con muestras de datos más concretas: números. Han logrado hacerse con multitud de cuadernos de ejercicios de matemáticas de antiguos alumnos de primaria, con numerosas páginas llenas de cuentas y operaciones con números.

Miguel está fascinado por cómo el ordenador, conectado a una cámara, va a ser capaz de identificar cada cifra...



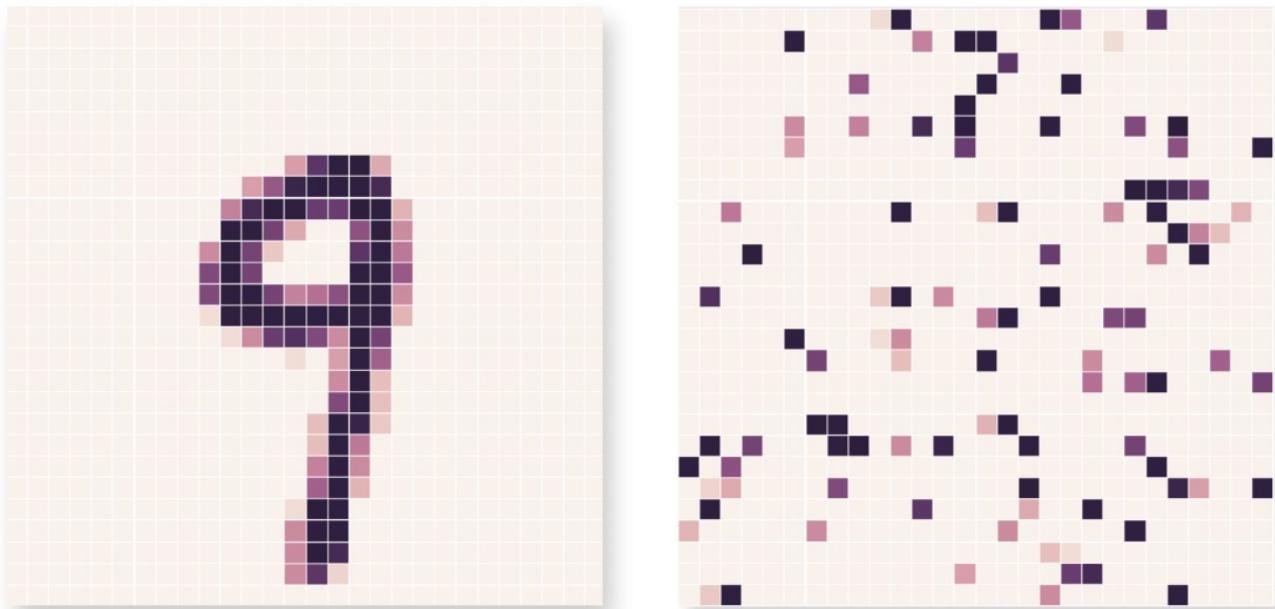
[@Casfatesvano \(CC BY-SA\)](#)

Algoritmos de redes neuronales

El nombre viene de la idea de imitar el funcionamiento de las redes neuronales de los organismos vivos: un conjunto de neuronas conectadas entre sí y que trabajan en conjunto, sin que haya una tarea concreta para cada una. Con la experiencia, las neuronas van creando y reforzando ciertas conexiones para "aprender" algo que se queda fijo en el tejido.

Ahora bien, en el desarrollo de la Inteligencia Artificial, el enfoque biológico no ha sido especialmente útil. De hecho, a medida que se han ido poniendo en práctica las teorías sobre las redes neuronales dichos conceptos biológicos han ido descartándose para concentrarse más bien en operaciones matemáticas y estadística. Hoy en día las redes neuronales se basan en una idea sencilla: **dados unos parámetros hay una forma de combinarlos para predecir un cierto resultado**. Por ejemplo, conociendo los píxeles de una imagen habrá una forma de saber qué número hay escrito. El problema, claro está, es que no sabemos cómo combinarlos.

Ejemplo de píxeles ordenados o desordenados

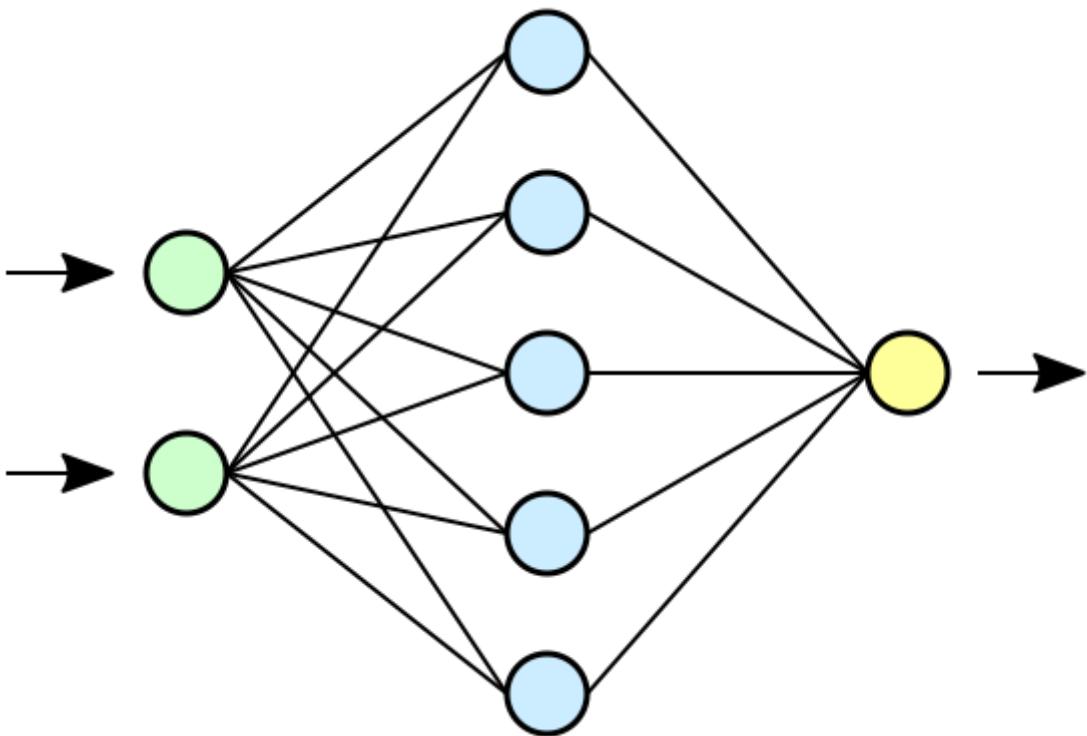


Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Las redes neuronales son un modelo para **encontrar esa combinación de parámetros** y aplicarla al mismo tiempo. En el lenguaje propio, encontrar la combinación que mejor se ajusta es "**entrenar**" la red neuronal. Una red ya entrenada se puede usar luego para hacer predicciones o clasificaciones, es decir, para "aplicar" la combinación. A medida que aumenta la complejidad de la aplicación para la que se entrena al modelo, es necesario aumentar el número de capas.

Nos podemos encontrar varias capas intermedias con varias neuronas cada una, llegando a lo que llaman "redes neuronales profundas". La idea es que con más capas con más neuronas cada una se pueden mejorar las predicciones en conjuntos de datos más complicados.

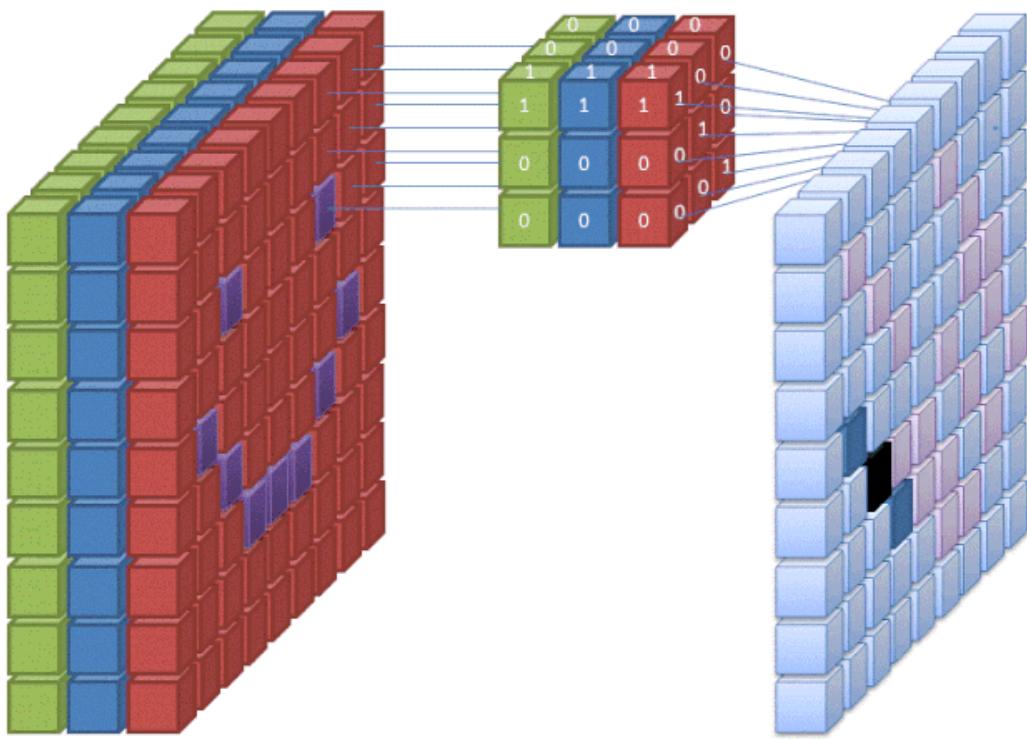
Red neuronal sencilla



[Dake, Mysid \(CC BY-SA\)](#)

El siguiente paso son las redes convolucionales, que están funcionando muy bien en reconocimiento de voz y procesamiento de imágenes. En una red neuronal como las que hemos visto antes, tendríamos una neurona para cada píxel de una imagen y después pondremos varias capas con varias neuronas, todas conectadas entre sí, para tratar de encontrar un número en una foto, por ejemplo. El problema es que no es demasiado efectivo (imaginaos todos los pesos que habría computar para una red que acepte imágenes de 1920x1080 píxeles).

Red neuronal convolucional

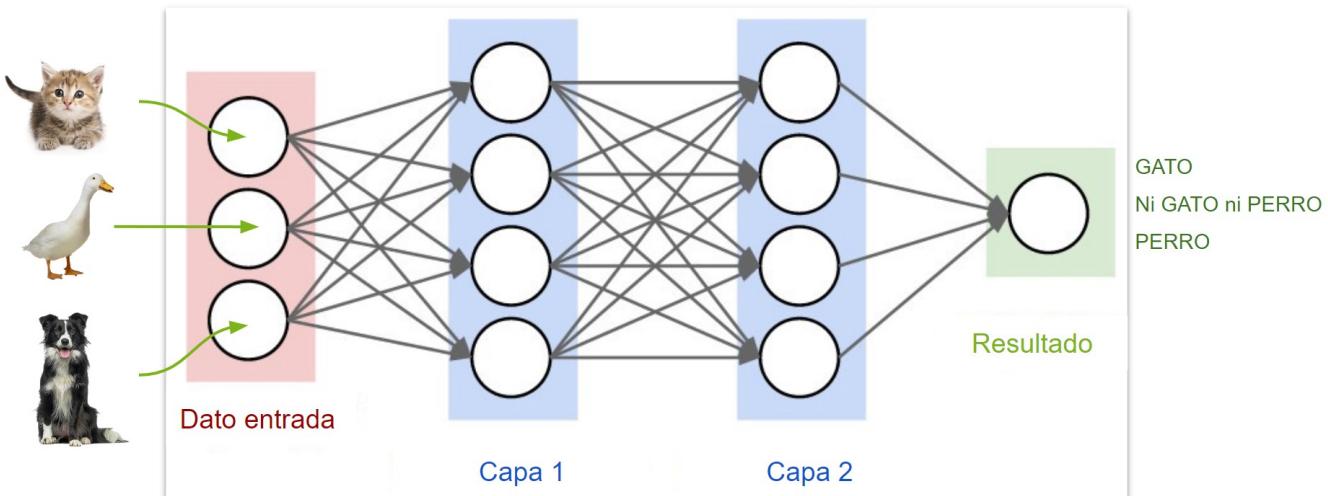


[Cecbur \(CC BY-SA\)](#)

La idea de las redes convolucionales es tratar de buscar características locales en pequeños grupos de entradas (en el caso de las imágenes, de píxeles), como puedan ser bordes o colores más o menos homogéneos. Es la misma idea que comentábamos cuando introducimos las capas unos párrafos más arriba, pero con una variación: buscamos características no en toda la imagen sino sólo en pequeñas regiones. Además, buscamos siempre detectar la misma característica en todos los grupos, por lo que podemos repetir esa estructura y reducir los ajustes que tenemos que hacer.

Para llevar a cabo esta idea, ponemos un mismo grupo de neuronas por cada grupo de entradas (por ejemplo, un cuadrado de 3x3 píxeles en una imagen o una secuencia de 4 mediciones en un archivo de sonido). La idea es que todos los elementos que metamos en la capa (llamada capa de convolución) tienen los mismos pesos por cada entrada, y se reduce considerablemente el número de parámetros. Si metemos más capas, la red neuronal podrá descubrir más y más complejas características de la imagen: se puede empezar por colores o bordes orientados y acabar con capas que se activan con formas circulares o cuadradas, por poner un ejemplo.

Aplicación práctica de red neuronal para clasificar animales



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Después de las capas de convolución se suele poner otra red neuronal "tradicional", que ahora tendrá más fácil el trabajo: no tiene que valorar cada píxel por separado sino que mira a un conjunto de características de alto nivel de la imagen. Ya no se trata de decidir si la imagen es un gato, perro o pato sabiendo que el píxel 1208 es amarillo y el 1209 es verde, sino quizás sabiendo que hay una forma puntiaguda o redondeada en la imagen con dos formas circulares en la parte inferior. De nuevo, se trata de extraer la información "oculta" en la entrada para tratar de encontrar qué es lo que define esos datos.

En definitiva, aunque como en otros campos de la Inteligencia Artificial puede haber cambios y matices en próximos años, podríamos considerar esta definición:

Una red neuronal es un procesador distribuido en paralelo de forma masiva con una propensión natural a almacenar conocimiento experimental y convertirlo en disponible para su uso. Asemeja al cerebro en dos aspectos:

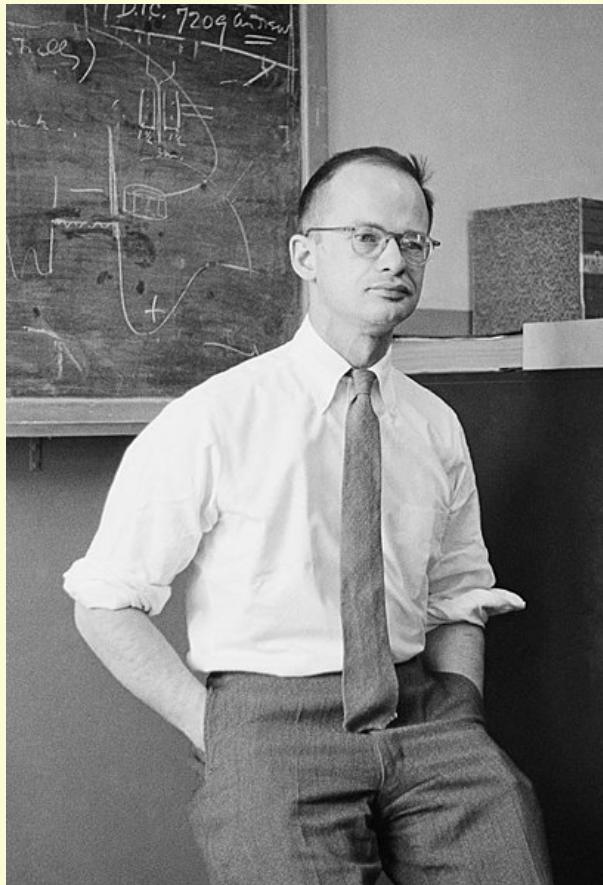
- ✓ El conocimiento se adquiere por la red mediante un proceso de aprendizaje.
- ✓ Las fuerzas de conexión interneuronal, conocidas como ponderaciones sinápticas, se utilizan para almacenar el conocimiento.



Para saber más

Las redes neuronales no son una idea nueva. Datan de los años 40 y 50 del Siglo XX, cuando se empezaron a publicar los primeros conceptos acerca de cómo deberían funcionar. Sin embargo en los años posteriores no llegaron a tener mucho éxito, pues era preciso contar con una cantidad importante de recursos de un ordenador para entrenar y ejecutar dichas redes neuronales con buenos resultados. Y en los ordenadores tardaron más años en disponer de dichos recursos.

En 1943 los investigadores **Warren McCulloch** y **Walter Pitts** crearon **Neurona**, el primer modelo informático basado en las redes neuronales. A partir de aquí, se establecieron dos caminos de investigación. Por un lado, en



[Desconocido](#) (Dominio público)

los procesos biológicos del cerebro y, por otro lado, la aplicación de estas redes en la Inteligencia Artificial.

Posteriormente, surgió el aprendizaje de **Hebb**, desarrollado por el psicólogo del mismo nombre, que se basaba en un mecanismo de aprendizaje no supervisado.

Finalmente, cuando **Turing** creó las *máquinas tipo B*, basó el funcionamiento de las mismas en los estudios de neurología y psicología que se centraban en el proceso de estudio de la plasticidad neuronal. A medida que se conocían más detalles sobre el funcionamiento de las mismas, se ampliaban los estudios de las redes neuronales artificiales. De esta forma los estudios realizados en las redes neuronales humanas, ayudaban a desarrollar el funcionamiento de las artificiales. El objetivo de

estas primeras investigaciones era profundizar en los usos del *Machine Learning*.

Desde los inicios del desarrollo de las primeras redes neuronales, hasta hoy en día, las investigaciones han pasado por varios hitos y autores que han ido sentando cátedra y aportando diferentes posibilidades.

Entre ellos, podemos destacar el algoritmo **perceptrón**, desarrollado por **Frank Rosenblatt** en 1958. Esta red se basaba en el aprendizaje de redes neuronales por computadora, a través de dos capas que utilizaban sistemas de adición y sustracción simples.

Gran parte del trabajo importante en la definición y estudio de las redes neuronales se produjo en los años 80 y 90 (puedes buscar más información sobre el Neocognitron de Fukushima), pero aún en ese momento las computadoras eran lentas y las bases de datos disponibles muy pequeñas. La investigación no encontró muchas aplicaciones en el mundo real (la propuesta de LeCun en 1998 con su reconocimiento de dígitos escritos a mano fue uno de las pocas propuestas en esa época). A efectos prácticos, en la primera década del siglo XXI en los proyectos de aprendizaje automático no se utilizaron las redes neuronales como algoritmo de entrenamiento. Trabajar en redes neuronales fue definitivamente marginal.

Hasta la finalización de esta primera década del siglo XXI no empezaron a surgir proyectos basados en redes neuronales: primero en reconocimiento de voz alrededor de 2009, luego en reconocimiento de imágenes (Alexnet de Krizhevsky) alrededor de 2012 o de traducción simultánea en 2014. Se produjo entonces un resurgimiento de las redes neuronales en buena medida gracias al crecimiento de la **cantidad y calidad de las bases de datos**, el abaratamiento y mejora de la **capacidad de cálculo**, en concreto por el aprovechamiento que

podemos hacer de las GPU (que se desarrollaron especialmente para mejorar las tarjetas gráficas para videojuegos).

Las redes neuronales hoy en día están en todas partes. Prácticamente todos los proyectos e investigaciones más ambiciosos que están en marcha ahora mismo usan redes neuronales y Deep Learning.

2.- Deep Learning y arquitectura por capas.



Caso práctico



[@casfatesvano \(CC BY-SA\)](#)

El primer experimento de reconocimiento de texto escrito a mano en el que iba a participar Miguel dentro de sus prácticas de empresa, no ha salido muy bien. Dice su tutor que no han sabido elegir correctamente la rede neuronal profunda adecuada. Y le ha facilitado a Miguel documentación con los principales algoritmos de redes neuronales más utilizadas, clasificadas según el tipo de aplicación práctica en la que es está usando en otros proyectos.

¡Menuda cantidad de algoritmos! Miguel va a tener que ponerse al día y entender lo mejor posible cómo funciona cada uno, y, sobre todo, para qué casos son más apropiados. A ver si dan con algún algoritmo que les proporcione mejores resultados en su experimento.

El *Deep Learning* o Aprendizaje Profundo es una de las principales tecnologías del aprendizaje automático, que se aplica tanto en entrenamientos supervisados como no supervisados. Utiliza algoritmos capaces de imitar hasta cierto punto las acciones del cerebro humano mediante redes neuronales artificiales. Estas redes se componen de docenas o incluso cientos de “capas” de neuronas, cada una de las cuales recibe e interpreta información de la capa anterior.

Esquema de una Red Neuronal Profunda

¿Esto es GATO o PERRO?



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Para comprender bien qué son y cómo funcionan las capas de una red neuronal es necesario conocer la unidad básica de funcionamiento, que llamamos Perceptrón. Lo definió Frank Rosenblatt en 1957.

Perceptrón

El perceptrón es una neurona artificial (una unidad de red neuronal) que realiza ciertos cálculos para detectar capacidades de los datos de entrada. Desde el punto de vista matemático se trata de un algoritmo de aprendizaje de clasificación binaria supervisado. Este algoritmo permite que las neuronas aprendan elementos y los procese uno por uno durante la preparación.

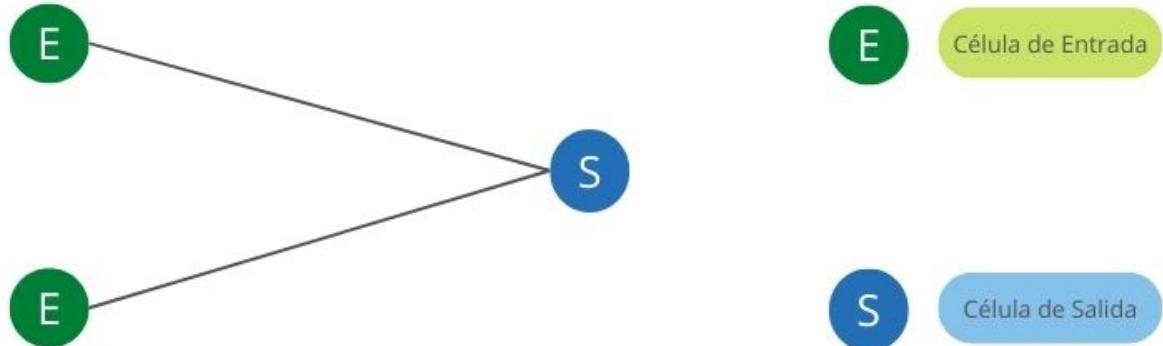
Al principio, el perceptrón fue planteado por Rosenblatt como una máquina y no tanto como un programa. Fue más adelante cuando su concepto se transformó al ámbito del software.

Desde el punto de vista biológico (recuerda que inicialmente los expertos en Inteligencia Artificial pretendían emular con máquinas el modo de funcionar del cerebro humano), el Perceptrón sería equivalente a la neurona. Así como la neurona es en biología la unidad básica del cerebro, el perceptrón es el modelo matemático más simple de una red neuronal.

De hecho, al perceptrón se lo conoce como la red neuronal mínima, con una capa de entrada y otra de salida. Y en ocasiones se puede utilizar para modelos de clasificación, codificación de bases de datos, o monitorización de datos de acceso.

Perceptrón

Perceptron (P)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))



Autoevaluación

El perceptrón es una artificial que realiza ciertos cálculos para detectar capacidades de los datos de

2.1.- Topologías de Redes Neuronales.

Red Neuronal de Avance - *Feed Forward (FF)*

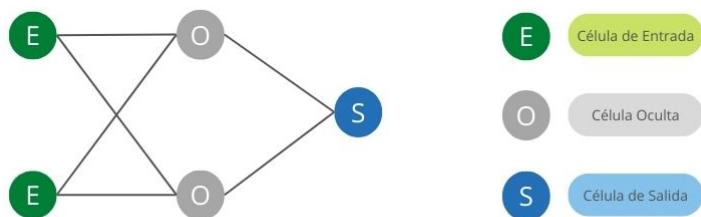
Una red neuronal de retroalimentación es una red neuronal artificial en la que los nodos nunca forman un ciclo. En esta red neuronal, todos los perceptrones están dispuestos en capas donde la capa de entrada recibe la entrada y la capa de salida genera la salida. Las capas ocultas no tienen conexión con el mundo exterior; por eso se les llama capas ocultas. En una red neuronal de retroalimentación, cada perceptrón en una capa está conectado con cada nodo en la siguiente capa. Por lo tanto, todos los nodos están completamente conectados. Otra cosa a tener en cuenta es que no hay una conexión visible o invisible entre los nodos de la misma capa. No hay bucles de retroceso en la red de avance. Por lo tanto, para minimizar el error en la predicción, generalmente usamos el algoritmo de retropropagación para actualizar los valores de peso.

Aplicaciones:

- ✓ Compresión de datos.
- ✓ Reconocimiento de patrones.
- ✓ Visión por computador.
- ✓ Reconocimiento de objetivo de sonda.
- ✓ Reconocimiento de voz.
- ✓ Reconocimiento de caracteres manuscritos.

Red neuronal de retroalimentación

Feed Forward (FF)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Base Radial - *Radial Basis Network (RBN)*

Las redes de funciones de base radial se utilizan generalmente para problemas de aproximación de funciones. Se pueden distinguir de otras redes neuronales por su velocidad de aprendizaje más rápida y su aproximación universal. La principal diferencia entre las

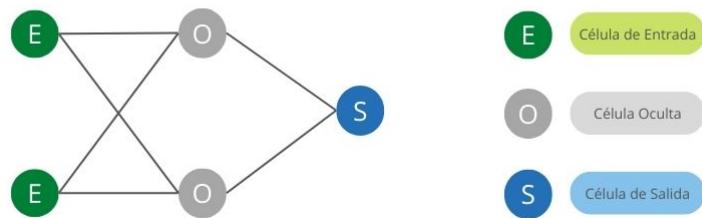
redes de base radial y las redes de retroalimentación directa es que los RBN utilizan una función de base radial como función de activación . Una función logística (función sigmoidea) da una salida entre 0 y 1, para encontrar si la respuesta es sí o no. El problema con esto es que si tenemos valores continuos, entonces no se puede usar un RBN. Las carreras impulsadas determinan lo lejos que está nuestra salida generada de la salida objetivo. Estos pueden resultar muy útiles en caso de valores continuos. En resumen, las RBN se comportan como redes FF utilizando diferentes funciones de activación.

Aplicaciones:

- ✓ Aproximación de funciones.
- ✓ Predicción de series temporales.
- ✓ Clasificación.
- ✓ Control de sistema.

Redes de funciones de Base Radial

Radial Basis Network (RBN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

2023

En resumen, las RBN se comportan como redes FF utilizando diferentes funciones de activación.

Aplicaciones:

- ✓ Aproximación de funciones.
- ✓ Predicción de series temporales.
- ✓ Clasificación.
- ✓ Control de sistema.

Red de Alimentación Directa Profunda - Deep Feed Forward (DFF)

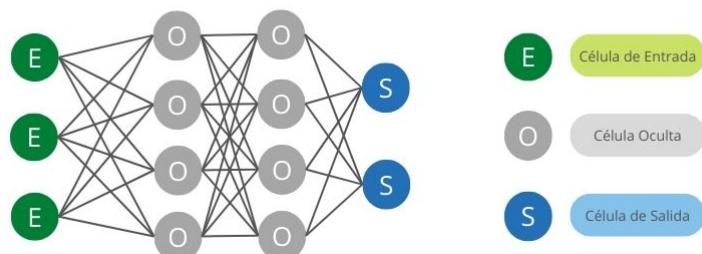
Una red de alimentación directa profunda es una red de retroalimentación directa que utiliza más de una capa oculta. El principal problema de usar solo una capa oculta es el de sobreajuste, por lo que al agregar más capas ocultas, podemos lograr (no en todos los casos) un sobreajuste reducido y una generalización mejorada.

Aplicaciones:

- ✓ Compresión de datos.
- ✓ Reconocimiento de patrones.
- ✓ Visión por computador.
- ✓ Filtrado de ruido de ECG.
- ✓ Predicción financiera.

Redes de alimentación directa profunda

Deep Feed Forward (DFF)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

2023

Red Neuronal Recurrente - *Recurrent Neural Network (RNN)*

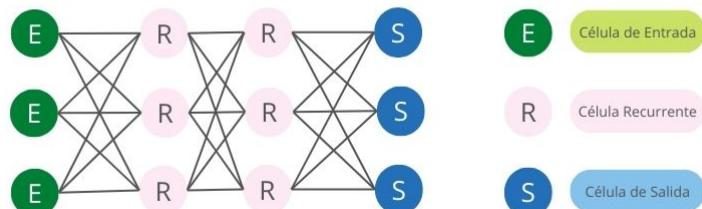
Las redes neuronales recurrentes (RNN) son una variación de las redes de retroalimentación directa (FF). En este tipo, cada una de las neuronas en capas ocultas recibe una entrada con un retraso específico en el tiempo. Usamos este tipo de red neuronal donde necesitamos acceder a información previa en iteraciones actuales. Por ejemplo, cuando intentamos predecir la siguiente palabra en una oración, primero necesitamos conocer las palabras utilizadas anteriormente. Los RNN pueden procesar entradas y compartir cualquier longitud y peso a lo largo del tiempo. El tamaño del modelo no aumenta con el tamaño de la entrada y los cálculos en este modelo tienen en cuenta la información histórica. Sin embargo, el problema con esta red neuronal es la baja velocidad computacional. Además, no puede considerar ninguna entrada futura para el estado actual. No puede recordar información de hace mucho tiempo.

Aplicaciones:

- ✓ Máquina traductora.
- ✓ Control de robot.
- ✓ Predicción de series temporales.
- ✓ Reconocimiento de voz.
- ✓ Síntesis de voz.
- ✓ Detección de anomalías de series temporales.
- ✓ Aprendizaje del ritmo.
- ✓ Composición musical.

Redes Neuronales Recurrentes

Recurrent Neural Network (RNN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red con Memoria a Corto y Largo Plazo - *Long / Short Term Memory (LSTM)*

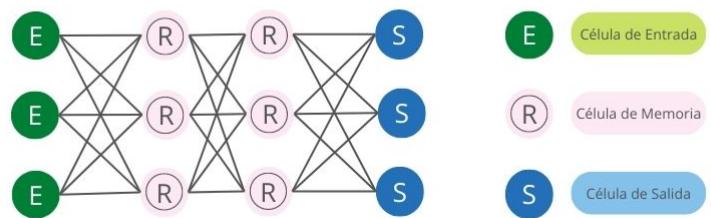
Las redes LSTM introducen una celda de memoria. Pueden procesar datos con espacios de memoria. Arriba, podemos notar que podemos considerar el retraso de tiempo en los RNN, pero si nuestro RNN falla cuando tenemos una gran cantidad de datos relevantes y queremos encontrar datos relevantes de ellos, entonces LSTM es el camino a seguir. Además, los RNN no pueden recordar datos de hace mucho tiempo, a diferencia de los LSTM.

Aplicaciones:

- ✓ Reconocimiento de voz.
- ✓ Reconocimiento de escritura.

Redes con memoria a corto y largo plazo

Long / Short Term Memory (LSTM)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Unidades Periódicas Cerradas - *Gated Recurrent Unit (GRU)*

Las unidades periódicas cerradas son una variación de las LSTM porque ambas tienen diseños similares y en su mayoría producen resultados igualmente buenos. Los GRU solo tienen tres puertas y no mantienen un estado de celda interno.

Puerta Update: determina la cantidad de conocimientos pasados que se transmiten al futuro.

Puerta Reset: determina la cantidad de conocimientos pasados que se deben olvidar.

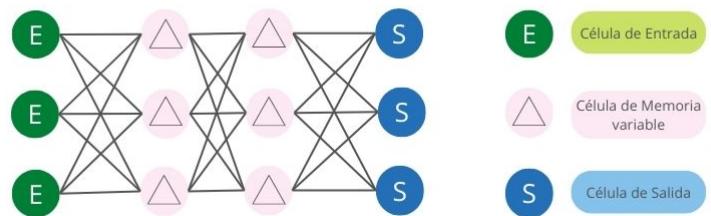
Puerta de memoria actual: subparte del destino de reinicio.

Aplicaciones:

- ✓ Modelado de música polifónica.
- ✓ Modelado de señales de voz.
- ✓ Procesamiento natural del lenguaje.

Red de Unidad Recurrente Cerrada

Gated Recurrent Unit (GRU)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

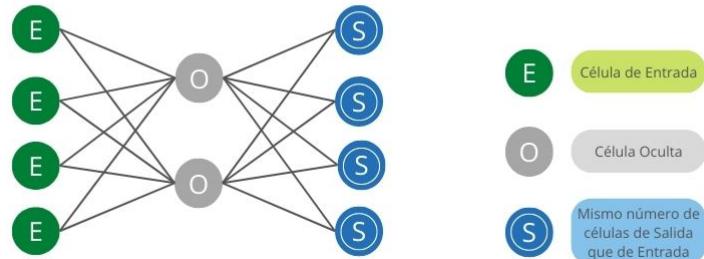
Red Neuronal de Codificador Automático - *Auto Encoder (AE)*

Una red neuronal de codificador automático es un algoritmo de aprendizaje automático no supervisado. En un codificador automático, la cantidad de celdas ocultas es menor que las celdas de entrada. El número de celdas de entrada en los codificadores automáticos es igual al número de celdas de salida. En una red AE, la entrenamos para que muestre la salida, que es tan cercana como la entrada alimentada, lo que obliga a las AE a encontrar patrones comunes y generalizar los datos. Usamos codificadores automáticos para la representación más pequeña de la entrada. Podemos reconstruir los datos originales a partir de datos comprimidos. El algoritmo es relativamente simple ya que AE requiere que la salida sea la misma que la entrada.

- ✓ Codificador: convierte los datos de entrada en dimensiones inferiores.
- ✓ Decodificador : reconstruye los datos comprimidos.
- ✓ Clasificación.
- ✓ Agrupación.
- ✓ Compresión de funciones.

Red Neuronal de Codificador Automático

Auto Encoder (AE)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Un Autoencoder Variacional (VAE) utiliza un enfoque probabilístico para describir observaciones. Muestra la distribución de probabilidad para cada atributo en un conjunto de características.

Red de Autoencoder Variacional - Variational Auto Encoder (VAE)

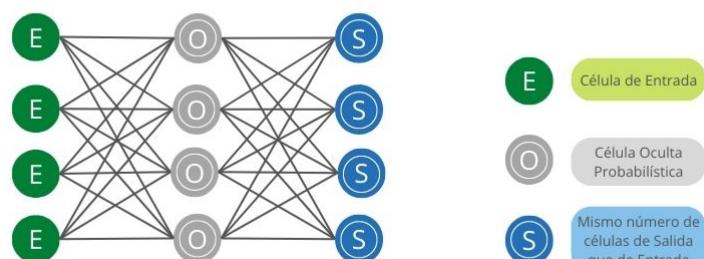
Un Autoencoder Variacional (VAE) utiliza un enfoque probabilístico para describir observaciones. Muestra la distribución de probabilidad para cada atributo en un conjunto de características.

Aplicaciones:

- ✓ Interpolación entre oraciones.
- ✓ Generación automática de imágenes.

Red de Autoencoder Variacional

Variational AE (VAE)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Codificador Automático de Eliminación de Ruido - *Denoising Auto Encoder (DAE)*

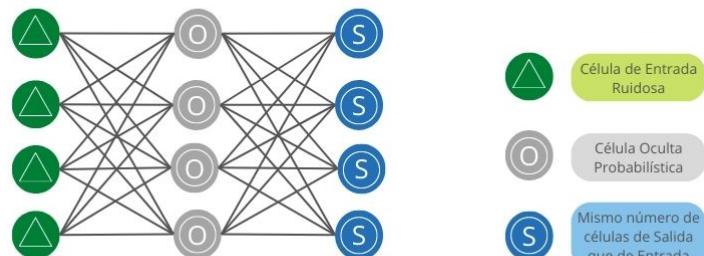
En este codificador automático, la red no puede simplemente copiar la entrada a su salida porque la entrada también contiene ruido aleatorio. En DAE, lo estamos produciendo para reducir el ruido y dar como resultado datos significativos dentro de él. En este caso, el algoritmo fuerza a la capa oculta a aprender características más sólidas para que la salida sea una versión más refinada de la entrada ruidosa.

Aplicaciones:

- ✓ Extracción de características.
- ✓ Reducción de dimensionalidad.

Red de Codificador Automático de Eliminación de Ruido

Denoising AE (DAE)



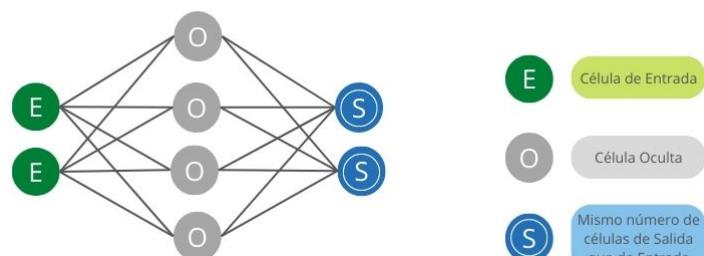
Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Autoencoder Dispersa - *Sparse Auto Encoder (SAE)*

En redes dispersas de autocodificador, construiríamos nuestra función de pérdida penalizando las activaciones de capas ocultas para que solo se activen unos pocos nodos cuando una sola muestra cuando la alimentamos a la red. La intuición detrás de este método es que, por ejemplo, si una persona afirma ser experta en los temas A, B, C y D, entonces la persona podría ser más generalista en estos temas. Sin embargo, si la persona solo afirma estar dedicada al tema D, es probable que anticipe conocimientos del conocimiento de la persona sobre el tema D.

Red de Autoencoder Dispersa

Sparse AE (SAE)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Aplicaciones:

- ✓ Extracción de características.
- ✓ Reconocimiento de dígitos escritos a mano.

Red de Cadenas de Markov - *Markov Chain (MC)*

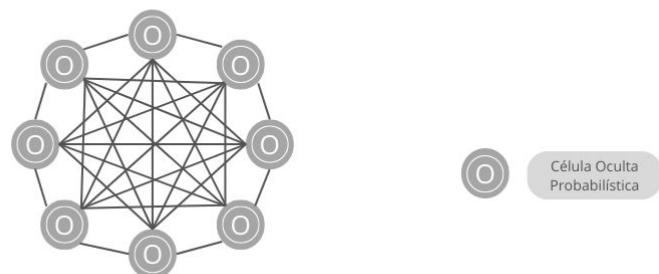
Una cadena de Markov es un sistema matemático que experimenta la transición de un estado a otro basándose en algunas reglas probabilísticas. La probabilidad de pasar a cualquier estado en particular depende únicamente del estado actual y del tiempo transcurrido.

Por ejemplo, algunos conjuntos de estados posibles pueden ser:

- ✓ Letras.
- ✓ Números.
- ✓ Las condiciones climáticas.
- ✓ Resultados de béisbol.
- ✓ Rendimiento de acciones.
- ✓ Reconocimiento de voz.
- ✓ Sistema de información y comunicación.
- ✓ Teoría de colas.
- ✓ Estadísticas.

Red de Cadenas de Markov

Markov Chain (MC)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red Hopfield - *Hopfield Network (HN)*

En una red neuronal de Hopfield, cada neurona está conectada directamente con otras neuronas. En esta red, una neurona está encendida o apagada. El estado de las neuronas puede cambiar al recibir información de otras neuronas. Generalmente usamos redes Hopfield para almacenar patrones y recuerdos. Cuando entrenamos una red neuronal en un conjunto de patrones, entonces puede reconocer el patrón incluso si está algo distorsionado o incompleto. Puede reconocer el patrón completo cuando lo alimentamos con una entrada incompleta, lo que devuelve la mejor suposición.

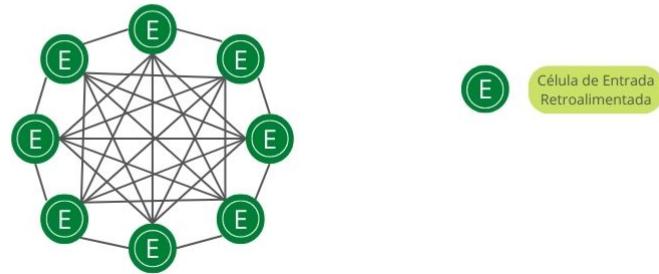
Aplicaciones:

- ✓ Problemas de optimización.
- ✓ Detección y reconocimiento de imágenes.

- ✓ Reconocimiento de imágenes médicas.
- ✓ Mejora de las imágenes de rayos X.

Red Hopfield

Hopfield Network (HN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Máquinas Boltzmann - *Boltzmann Machine (BM)*

Una red de máquinas de Boltzmann implica aprender una distribución de probabilidad de un conjunto de datos original y usarla para hacer inferencias sobre datos no vistos. En los BM, hay nodos de entrada y nodos ocultos, tan pronto como todos nuestros nodos ocultos cambian de estado, nuestros nodos de entrada se transforman en nodos de salida. Por ejemplo: supongamos que trabajamos en una planta de energía nuclear, donde la seguridad debe ser la prioridad número uno. Nuestro trabajo es garantizar que todos los componentes de la central sean seguros de usar, habrá estados asociados con cada componente, utilizando valores booleanos para simplificar, 1 para utilizable y 0 para inutilizable. Sin embargo, también habrá algunos componentes para los que nos será imposible medir los estados con regularidad.

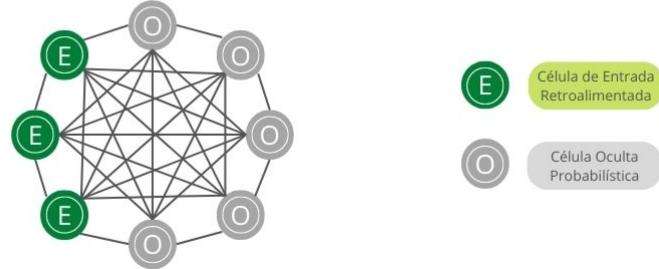
Además, no tenemos datos que nos digan cuándo explotará la planta de energía si el componente oculto deja de funcionar. Entonces, en ese caso, construimos un modelo que se da cuenta cuando el componente cambia de estado. Entonces, cuando lo haga, se nos notificará para verificar ese componente y garantizar la seguridad del conjunto.

Aplicaciones:

- ✓ Reducción de dimensionalidad.
- ✓ Clasificación.
- ✓ Regresión.
- ✓ Filtración colaborativa.

Red de Máquinas Boltzmann

Boltzmann Machine (BM)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

- ✓ Aprendizaje de funciones.

Red Restringida de Máquinas de Boltzmann - *Restricted Boltzmann Machine (RBM)*

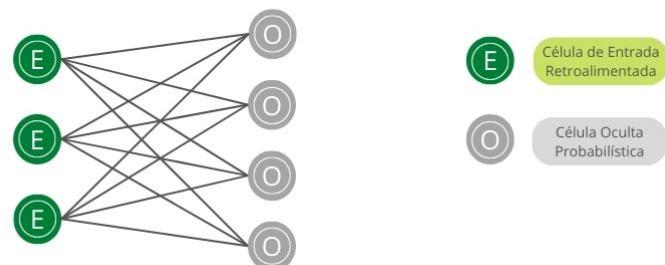
Las redes restringidas de Máquinas de Boltzmann son una variante de las Máquinas de Boltzmann. En este modelo, las neuronas de la capa de entrada y la capa oculta pueden tener conexiones simétricas entre ellas. Una cosa a tener en cuenta es que no hay conexiones internas dentro de cada capa. Por el contrario, las máquinas de Boltzmann pueden tener conexiones internas en la capa oculta. Estas restricciones en BM permiten un entrenamiento eficiente del modelo.

Aplicaciones:

- ✓ Filtración.
- ✓ Aprendizaje de funciones.
- ✓ Clasificación.
- ✓ Detección de riesgos.
- ✓ Análisis económico y empresarial.

Red Restringida de Máquinas de Boltzmann

Restricted BM (RBM)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Creencias Profundas - *Deep Belief Network (DBN)*

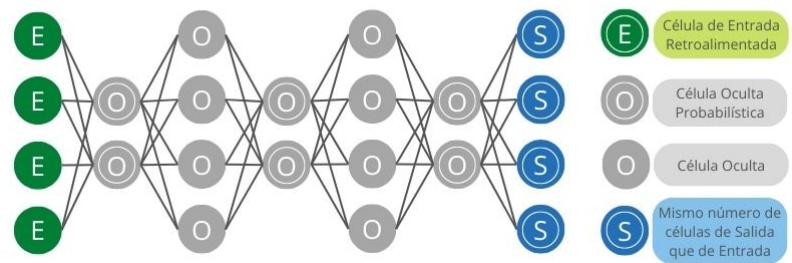
Las redes de creencias profundas contienen muchas capas ocultas. Podemos llamar a los DBN con un algoritmo no supervisado, ya que aprende por primera vez sin supervisión. Las capas en un DBN actúan como un detector de características. Después de un entrenamiento sin supervisión, podemos entrenar nuestro modelo con métodos de supervisión para realizar la clasificación. Podríamos representar los DBN como una composición de Máquinas de Boltzmann restringidas y Autoencoders, los últimos DBN utilizan un enfoque probabilístico hacia sus resultados.

Aplicaciones:

- ✓ Recuperación de documentos / imágenes.
- ✓ Reducción de dimensionalidad no lineal.

Red de Creencias Profundas

Deep Belief Network (DBN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red Neuronal Convolucional Profunda - Deep Convolutional Network (DCN)

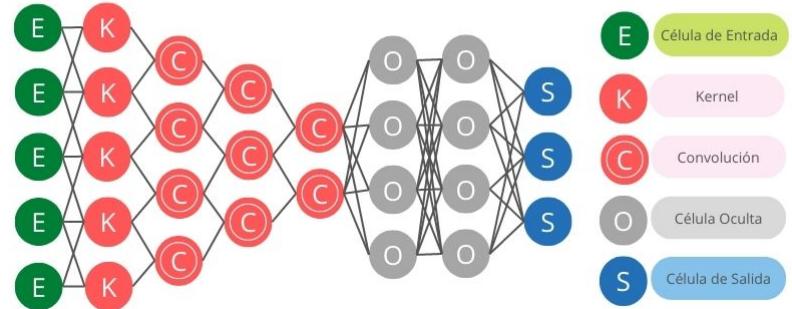
Las redes neuronales convolucionales son redes neuronales que se utilizan principalmente para la clasificación de imágenes, la agrupación de imágenes y el reconocimiento de objetos. Los DCN permiten la construcción sin supervisión de representaciones de imágenes jerárquicas. Los DCN se utilizan para agregarle características mucho más complejas para que pueda realizar la tarea con mayor precisión.

Aplicaciones:

- ✓ Identificar caras, letreros callejeros, tumores.
- ✓ Reconocimiento de imagen.
- ✓ Análisis de video.
- ✓ PNL.
- ✓ Detección de anomalías.
- ✓ Descubrimiento de medicamento.
- ✓ Juego de damas.
- ✓ Pronóstico de series de tiempo.

Red Neuronal Convolucional Profunda

Deep Convolutional Network (DCN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red Neuronal Deconvolucional - *Deconvolutional Network (DN)*

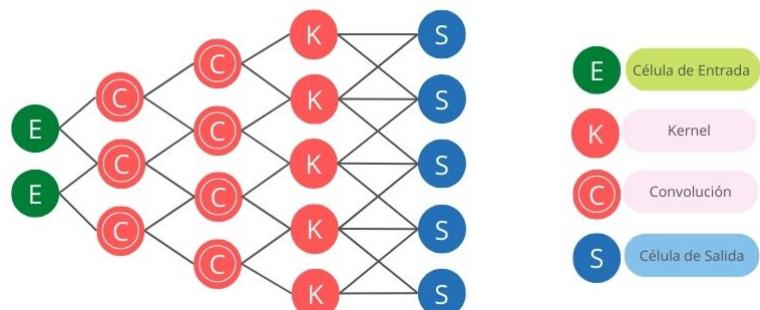
Las redes deconvolucionales son redes neuronales convolucionales que funcionan en un proceso inverso. Aunque una red neuronal deconvolucional es similar a una DCN en la naturaleza del trabajo, su aplicación en Inteligencia Artificial es muy diferente. Las redes deconvolucionales ayudan a encontrar características o señales perdidas en redes que antes consideraban útiles. Una red deconvolucional puede perder una señal debido a que se ha enredado con otras señales. Una red deconvolucional puede tomar un vector y hacer una imagen de él.

Aplicaciones:

- ✓ Súper resolución de imagen.
- ✓ Estimación de la profundidad de la superficie a partir de una imagen.
- ✓ Estimación de flujo óptico.

Red Neuronal Deconvolucional

Deconvolutional Network (DN)



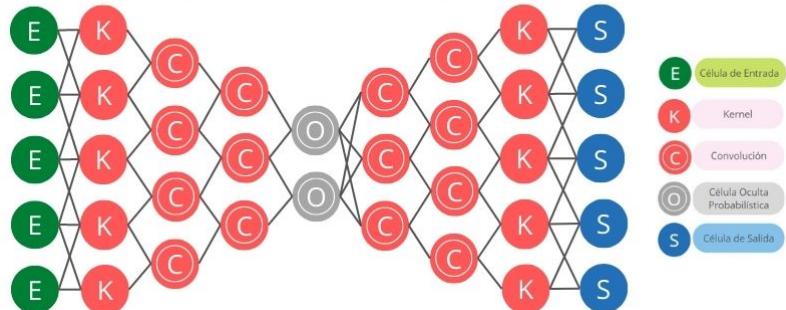
Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Gráficos Inversa Convolucional Profunda - *Deep Convolutional Inverse Graphics Network (DCIGN)*

Las redes gráficas inversas convolucionales profundas tienen como objetivo relacionar las representaciones gráficas con las imágenes. Utiliza elementos como iluminación, ubicación de objetos, textura y otros aspectos del diseño de imágenes para un procesamiento de imágenes muy sofisticado. Utiliza varias capas para procesar la entrada y la salida. La red de gráficos inversos convolucionales

Red de Gráficos Inversa Convolucional Profunda

Deep Convolutional Inverse Graphics Network (DCIGN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

profundos usa capas iniciales para codificar a través de varias convoluciones, utilizando la agrupación máxima, y luego usa capas posteriores para decodificar con desenrollado.

Aplicaciones:

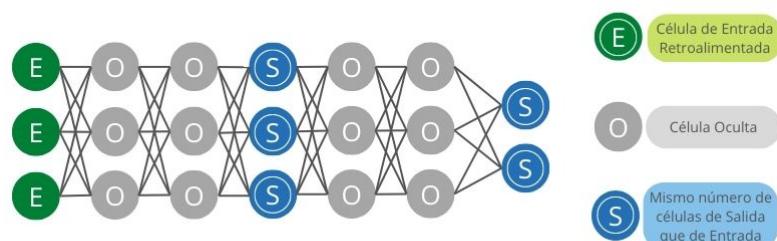
- ✓ Manipulación de rostros humanos.

Red Generativa de Adversarios - *Generative Adversarial Network (GAN)*

Dados los datos de entrenamiento, las redes generativas adversarias aprenden a generar nuevos datos con las mismas estadísticas que los datos de entrenamiento. Por ejemplo, si entrenamos nuestro modelo GAN en fotografías, entonces un modelo entrenado podrá generar nuevas fotografías que se vean auténticas para el ojo humano. El objetivo de estas redes es distinguir entre resultados reales y sintéticos para que puedan generar resultados más auténticos.

Red Generativa de Adversarios

Generative Adversarial Network (GAN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Aplicaciones:

- ✓ Genera nuevas poses humanas.
- ✓ Fotos a Emojis.
- ✓ Envejecimiento facial.
- ✓ Súper resolución.
- ✓ Traducción de ropa.
- ✓ Predicción de video.

Máquina de Estado Líquido - *Liquid State Machine (LSM)*

Una máquina de estado líquido es un tipo particular de red neuronal de picos. Consta de una amplia colección de neuronas. Aquí cada nodo recibe entradas de una fuente externa y otros nodos, que pueden variar según el tiempo. Es importante entender que los nodos de estas máquinas se conectan aleatoriamente entre sí. En estos casos las funciones de activación se reemplazan por niveles de umbral. Solo cuando las máquinas alcanzan el nivel de umbral,

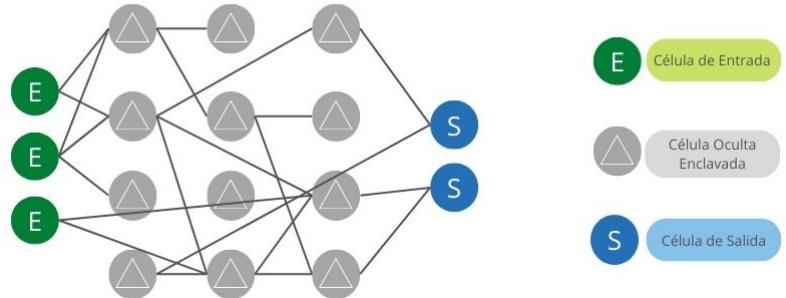
una neurona en particular emite su salida.

Aplicaciones:

- ✓ Reconocimiento de voz.
- ✓ Visión por computador.

Máquina de estado líquido

Liquid State Machine (LSM)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Máquinas de Aprendizaje Extremo - *Extreme Learning Machine (ELM)*

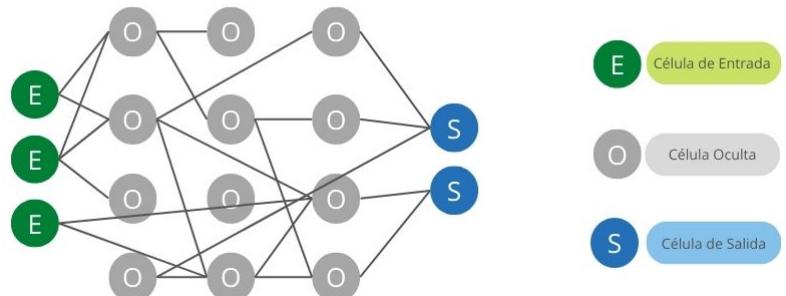
En general es útil para conjuntos de datos pequeños. Pero presenta inconvenientes para conjuntos de datos masivos, como por ejemplo se reduce la velocidad de aprendizaje y ajusta todos los parámetros de forma iterativa.

Aplicaciones:

- ✓ Clasificación.
- ✓ Regresión.
- ✓ Agrupación.
- ✓ Aproximación escasa.
- ✓ Aprendizaje de funciones.

Red de Máquinas de Aprendizaje Extremo

Extreme Learning Machine (ELM)



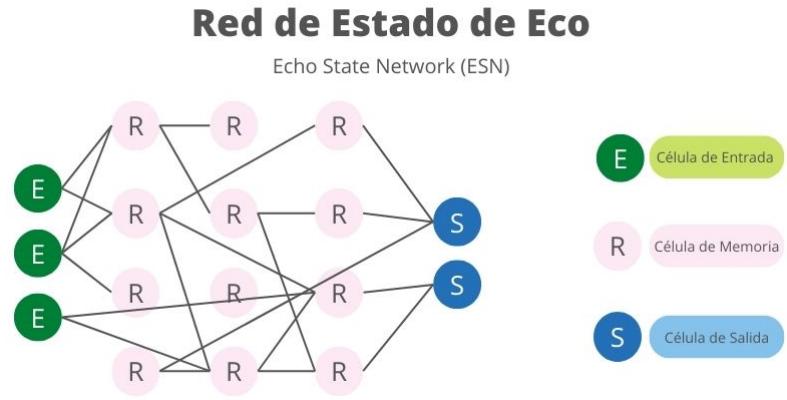
Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red de Estado de Eco - *Echo State Network (ESN)*

Las redes de estado de Eco son un subtipo de las redes neuronales recurrentes. En este caso cada nodo de entrada recibe una señal no lineal. Los nodos ocultos están escasamente conectados. De hecho la conectividad y los pesos de los nodos ocultos se asignan al azar. En este tipo de redes los pesos de salida finales se pueden entrenar y actualizar.

Aplicaciones:

- ✓ Predicción de series temporales.
- ✓ Procesamiento de datos.



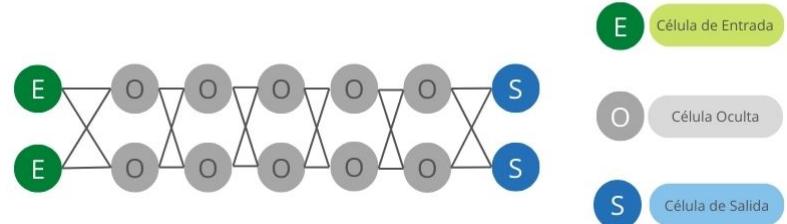
Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Red Residual Profunda - *Deep Residual Network (DRN)*

Las redes neuronales profundas con muchas capas pueden ser difíciles de entrenar y necesitar mucho tiempo durante la fase de entrenamiento. También puede darnos resultados degradados. De manera que las redes residuales profundas evitan la degradación de los resultados, aunque también tengan muchas capas. En este tipo de redes algunas partes de sus entradas pasan a la siguiente capa. Por tanto, estas redes pueden ser bastante profundas (puede contener alrededor de 300 capas).

Red Residual Profunda

Deep Residual Network



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Aplicaciones:

- ✓ Clasificación de imágenes.
- ✓ Detección de objetos.
- ✓ Segmentación semántica.
- ✓ Reconocimiento de voz.
- ✓ Reconocimiento de idiomas.

Red de Kohonen - *Kohoner Network (KN)*

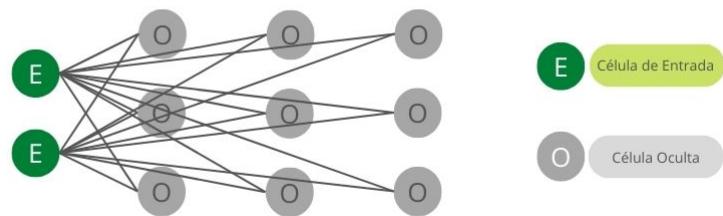
Una red de Kohonen es un algoritmo no supervisado. Estas redes también se conocen como mapas autoorganizados, pues resultan muy útiles cuando tenemos nuestros datos dispersos en muchas dimensiones y los queremos en una o dos dimensiones solamente. Se puede considerar como un método de reducción de dimensionalidad. Usamos redes de Kohonen para visualizar datos de alta dimensión. Utilizan el aprendizaje competitivo en lugar del aprendizaje de corrección de errores.

Varias topologías:

- ✓ Topología de cuadrícula rectangular.
- ✓ Topología de cuadrícula hexagonal.
- ✓ Reducción de dimensionalidad.
- ✓ Evaluación y predicción de la calidad del agua.
- ✓ Gestión de aguas costeras.

Red de Kohonen (KN)

Kohonen Network (KN)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Máquina de Vector Soporte - *Support Vector Machine (SVM)*

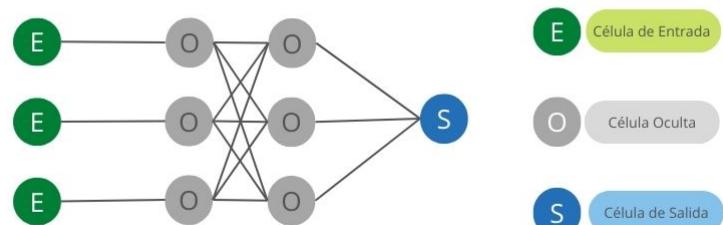
La red neuronal de vector soporte es un algoritmo híbrido de máquinas vectoriales de soporte y redes neuronales. Para un nuevo conjunto de ejemplos, siempre intenta clasificarlos en dos categorías: Sí o No (1 o 0). Las máquinas de vector soporte se utilizan generalmente para clasificaciones binarias. Por lo general, no se consideran redes neuronales.

Aplicaciones:

- ✓ Detección de rostro.
- ✓ Categorización de texto.
- ✓ Clasificación.
- ✓ Bioinformática.
- ✓ Reconocimiento de escritura a mano.

Máquina de Vector Soporte

Support Vector Machine



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Máquina Neuronal de Turing - Neural Turing Machine (NTM)

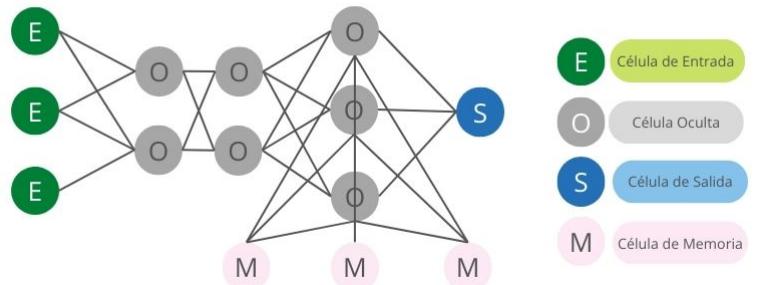
La arquitectura de una Máquina Neuronal de Turing se compone de dos elementos principales: un controlador de red neuronal y un banco de memoria.

Aplicaciones:

- ✓ Robótica.
- ✓ Construyendo un cerebro humano artificial.

Máquina neuronal de Turing

Neural Turing Machine (NTM)



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

3.- Proceso de entrenamiento de una Red Neuronal Profunda.



Caso práctico

El equipo de trabajo con el que colabora Miguel en sus prácticas de empresa por fin han conseguido resultados aceptables en su prototipo de Inteligencia Artificial capaz de reconocer y leer texto manuscrito.

A Miguel le ha llamado la atención especialmente que el prototipo va siendo a cada iteración más preciso: Tarda menos tiempo en dar una respuesta y cada vez lo hace con un porcentaje de confiabilidad más elevado. ¡Eso no pasaba con modelos sencillos de Aprendizaje Automático!

El modelo que han entrenado sabe reconocer ya números escritos casi de cualquier manera.

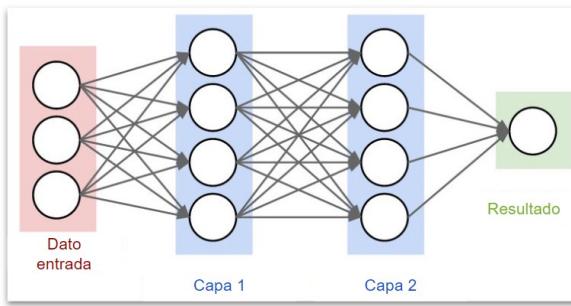


[@Casfatesvano \(CC BY-SA\)](#)

Funcionamiento práctico de una Red Neuronal

DATOS DE ENTRADA

- Imágenes
- Texto
- Email
- Español
- Precios históricos
- Imagen satelite
- Datos usuario
- Datos cliente
- ...



DATOS DE SALIDA

- Qué hay en la imagen
- Sentimiento del texto
- Normal/Spam
- Inglés
- Compra/Venta
- Localización
- Probabilidad click ad
- Probabilidad de éxito
- ...

En definitiva, el proceso de entrenamiento de una Red Neuronal Profunda, nos permite encontrar un Resultado a partir de uno o varios datos de entrada. Por ejemplo identificar qué hay en una serie de imágenes, qué sentimiento tienen ciertos textos, si un email es spam o no, traducir un texto de español a otro idioma...

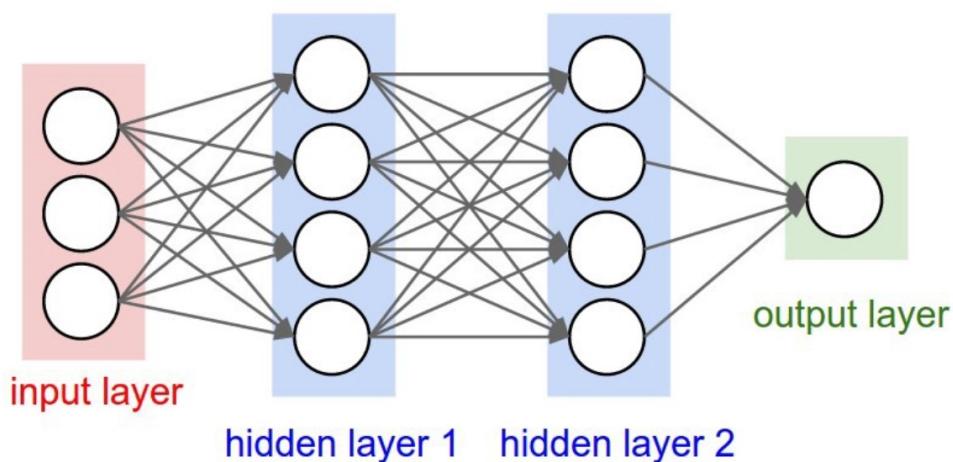
Y la principal característica de este tipo de aprendizaje es que a diferencia del Aprendizaje Automático convencional (Machine Learning), que toma decisiones según lo aprendido con los datos de entrenamiento sin ir más allá, en el caso del Aprendizaje Profundo (Deep Learning) el modelo sigue aprendiendo (y mejorando su precisión) cada vez que se utiliza.

Es decir, que para mejorar un modelo de Aprendizaje Automático es necesario que el ser humano intervenga ajustando o mejorando el código del programa, o depurando la base de datos, o ampliando dicha base de datos... en definitiva, que tiene que "volver" a empezar. Mientras que en el Aprendizaje Profundo el modelo se mejora a sí mismo de manera automática, sin intervención directa humana.

Fases del entrenamiento de una Red Neuronal Profunda.

✓ Fase 1 - Forward Pass

Se procesa la información de entrada y se genera un resultado.

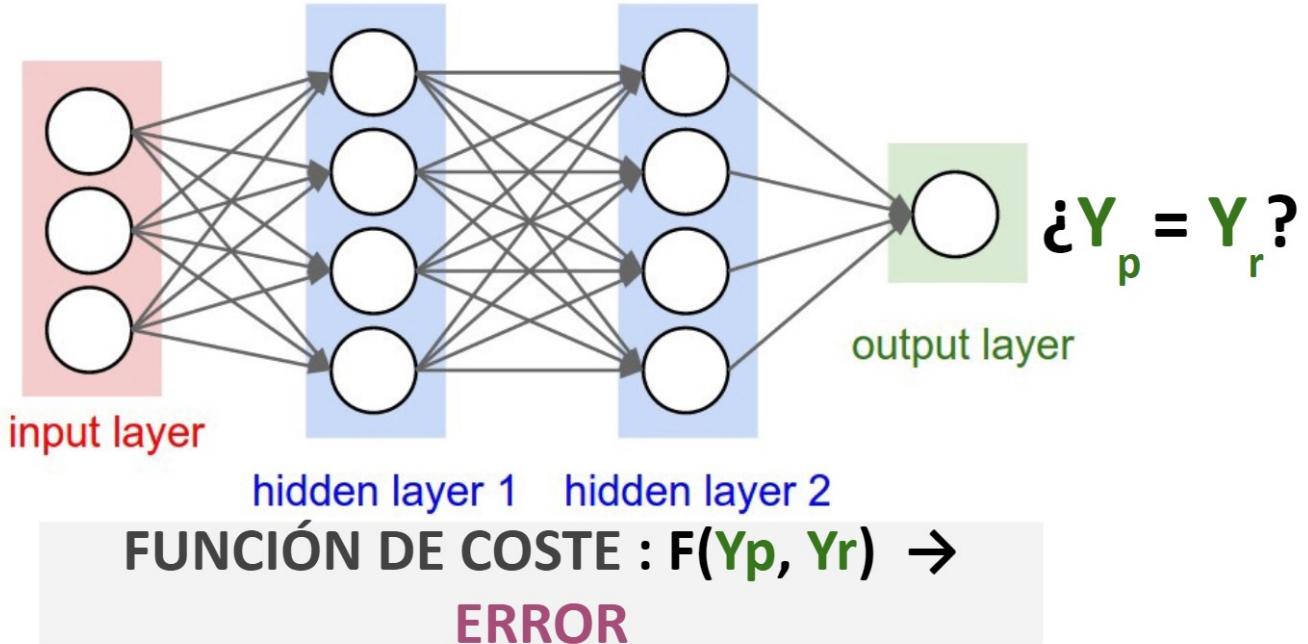


$$X \rightarrow (\text{Suma ponderada} \rightarrow \text{Función de activación}) \times \text{Nº Capas} \rightarrow Y_p$$

Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

✓ Fase 2 - Función de coste

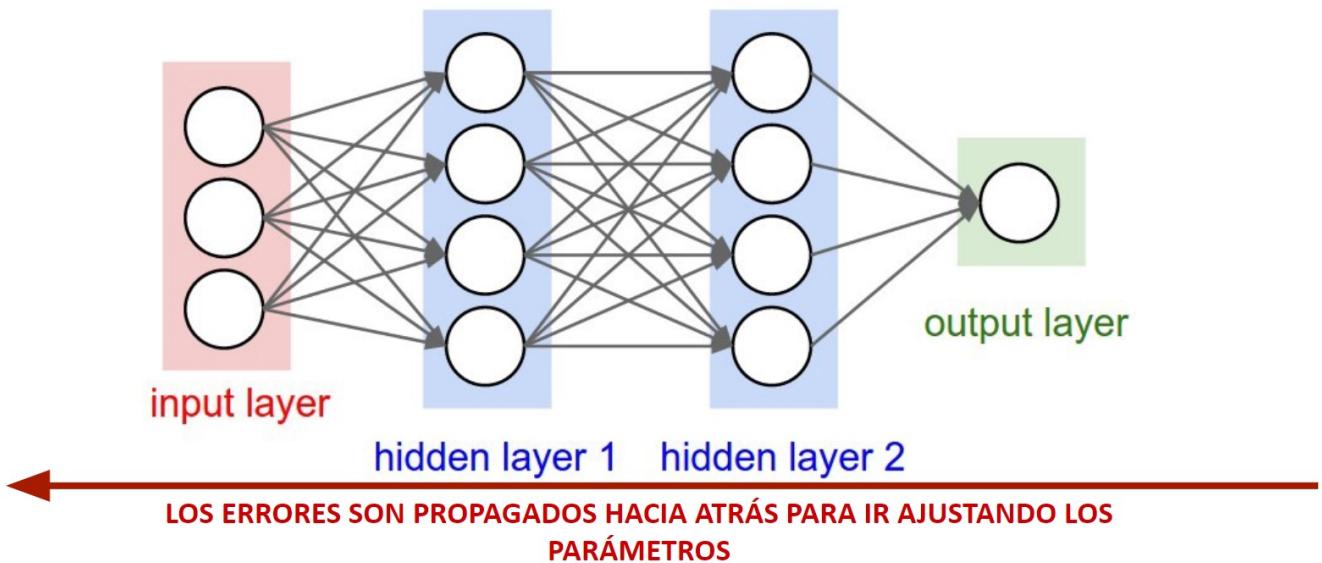
Se comparan los resultados y se detectan errores



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

✓ Fase 3 - Backward Pass

Analiza los resultados clasificados como errores o no suficientemente buenos, retropropaga dichos errores en el resultado en el algoritmo hacia atrás, para encontrar qué parámetros ha utilizado para llegar a esa conclusión.



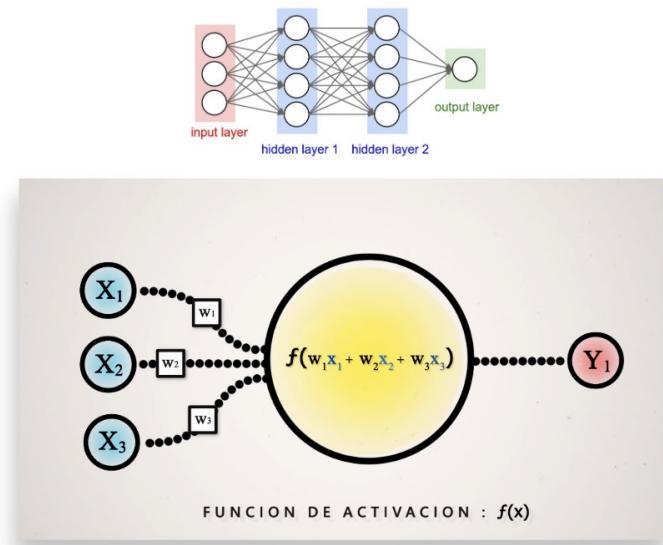
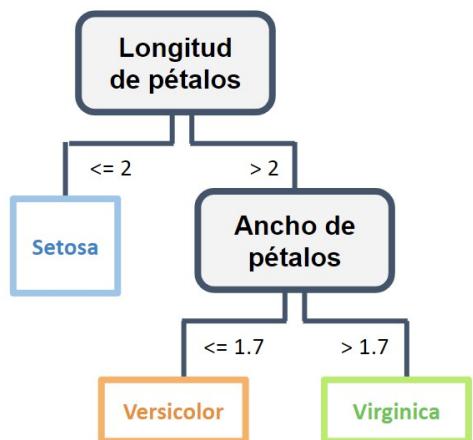
Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

✓ Fase 4 - Descenso del Gradiente

Optimiza la red ajustando los parámetros que han influido en los resultados erróneos para evitar que se vuelvan a producir errores en casos similares.

Los parámetros de la red neuronal profunda.

A medida que el entrenamiento del modelo avanza, se van fijando los parámetros que definen el comportamiento de la red. Por ejemplo, en un modelo entrenado para clasificar unos tipos de plantas, el modelo, después de iterar con los datos de entrada y aplicar redes neuronales profundas, determina que los parámetros determinantes son los referidos a la longitud y ancho de los pétalos de las plantas.



Fran Bartolomé - Elaboración propia ([CC BY-SA](#))

Los modelos entrenados por redes neuronales profundas nos dan resultados de tanto por ciento de confiabilidad para un posible resultado. En el ejemplo de la clasificación de plantas, nos diría que para un caso x, habría, por ejemplo, un 95% de confiabilidad de que se tratara de una planta setosa. Casi nunca nos da un 100% de confiabilidad.



Autoevaluación

Completa la frase con las palabras que faltan.

Una red neuronal es un procesador distribuido en _____ de forma masiva con una propensión natural a almacenar conocimiento experimental y convertirlo en disponible para su uso

Enviar



Autoevaluación

La principal característica del Aprendizaje Profundo es que el modelo sigue
[REDACTED] y mejorando su [REDACTED] cada vez que se
utiliza.

Enviar