



# DEEP LEARNING

# **ELECTIVA**

# **PROFESIONALIZANTE**

JULIO CHERANTHONY PEÑA PÉREZ-2014-0471

HECTOR JUNIOR MORALES 2016-1208



<b>1. Introducción.....</b>	<b>4</b>
<b>2. Justificación.....</b>	<b>5</b>
<b>3. Planteamiento del Problema.....</b>	<b>6</b>
<b>4. Objetivo de Estudio.....</b>	<b>7</b>
<b>5. Objetivos.....</b>	<b>8</b>
5.1. <b>Objetivo General.....</b>	<b>8</b>
5.2. <b>Objetivos Específicos.....</b>	<b>8</b>
<b>6. Hipótesis.....</b>	<b>9</b>
<b>7. Fundamentacion Teorica.....</b>	<b>10</b>
<b>8. Que es el Deep Learning.....</b>	<b>11</b>
8.1.Redes Neuronales.....	12
8.1.1. Modelo.....	15
8.1.2. Forward Pass.....	17
8.1.3. Backward Pass.....	18
8.2.Deep Learning: Aproximación a la percepción humana.....	19
8.2.1. ¿Cómo funciona el Deep Learning?.....	20
8.2.2. Arquitecturas de Deep Learning.....	23
8.2.2.1. Redes Neuronales Pre-alimentadas.....	24
8.2.2.2. Redes Neuronales Convulcionales.....	24
8.2.2.3. Redes Neuronales Recurrentes.....	26
8.3.Aprendizaje No Supervisado.....	28
8.4.Innovaciones del Deep Learning.....	31
8.5.Logros del Deep Learnig.....	33
8.6.¿Por qué resultados tantardios?.....	34
8.6.1. Disponibilidad de gran cantidad de datos.....	34
8.6.2. Computación paralela masiva con GPUs.....	35
8.6.3. Funciones para Backpropagation.....	35
8.6.4. Nuevas arquitecturas de regularización.....	35
8.6.5. Optimizadores más robustos.....	36
8.6.6. Plataformas de Software.....	36
<b>9. Conclusiones.....</b>	<b>37</b>
<b>10. Bibliografía.....</b>	<b>38</b>

## **INTRODUCCION**

El deep learning es una disciplina que se engloba en lo que se conoce como "*representation learning*" y esta vez está contenida en el archí-conocido "*machine learning*".

Pero, ¿de qué se trata?, ahora todo el mundo habla de deep learning, de la cantidad de cosas que se pueden hacer con este método y de lo maravilloso que es.

En el mundo de la tecnología añadir la etiqueta deep learning a un producto se está convirtiendo en lo mismo que hace unos años añadir "IoT (internet of things)", web semántica, cloud o mucho más lejos en el tiempo AJAX.

La motivación para escribir este artículo es la escasa información que existe en lengua cervantina sobre el tema, esto es algo normal, puesto que es algo novedoso y está en pleno crecimiento.

## **JUSTIFICACION**

Una de las claves de la inteligencia artificial avanzada está en el aprendizaje. Es cada vez más habitual que le pidamos a las maquinas que aprendan por si solas. No podemos permitirnos el lujo de pre-programar reglas para lidiar con las infinitas combinaciones de datos de entrada y situaciones que aparecen en el mundo real.

En vez de hacer eso, necesitamos que las maquinas sean capaces de auto-programarse, en otras palabras, queremos maquinas que aprendan de su propia experiencia. La disciplina del Aprendizaje Automático (Machine Learning) se ocupa de este reto y gracias a la tormenta perfecta en la que nos acabamos de adentrar todos los gigantes de Internet ha entrado de lleno en el mundo del aprendizaje automático, ofreciendo servicios en la nube para construir aplicaciones que aprendan a partir de los datos que ingieren.

## **Planteamiento del problema**

La gran importancia sobre entrar en este tipo de tecnología, radica en que existan algoritmos capaces de aprender por cuenta propia sin necesidad de una supervisión o intervención humana, siendo las maquinas capaces de ir aprendiendo a medida que pase el tiempo, así de manera autónoma recopilando y navegando entre grandes cantidades de datos siendo capaces de llegar a conclusiones por su propia cuenta.

Los primeros modelos solo poseían una capa de procesamiento, la de salida, por lo que se desarrollaron algoritmos que tuvieran capas “ocultas” entre la capa de entrada y la de salida que también se encargaran del procesamiento y clasificación de los patrones de entrada, desarrollando el perceptrón multicapa o MLP.

Pero, ¿Por qué? Es una pregunta con una respuesta inmediata. “Tener ordenadores que sean auto-programables, sin la necesidad de reprogramar un computador, que vaya aprendiendo conforme navegue en las redes y aprenda sobre conductas, costumbres, de los objetos percibidos.

El Deep Learning representa un acercamiento más íntimo al modo de funcionamiento del sistema nervioso humano. Nuestro encéfalo tiene una micro arquitectura de gran complejidad, en la que se han descubierto núcleos y áreas diferenciados cuyas redes de neuronas están especializadas para realizar tareas específicas. Hasta ahora hemos visto que la computación cognitiva se basa en la integración de procesos psicológicos típicamente humanos como el aprendizaje o el lenguaje.

## **Objetivo de Estudio**

El Deep Learning representa un acercamiento más íntimo al modo de funcionamiento del sistema nervioso humano. Aunque existen varias maneras de implementar Deep Learning, una de las más comunes es utilizar redes de neuronas y los autocodificadores.

La visión artificial es una de las áreas donde el Deep Learning proporciona una mejora considerable en comparación con algoritmos más tradicionales. Existen varios entornos y bibliotecas de código de Deep Learning que se ejecutan en las potentes GPUs modernas.

Los modelos computacionales de Deep Learning imitan estas características arquitecturales del sistema nervioso, permitiendo que dentro del sistema global haya redes de unidades de proceso que se especialicen en la detección de determinadas características cultas en los datos. Este enfoque ha permitido mejores resultados en tareas de percepción computacional, si las comparamos con las redes monolíticas de neuronas artificiales.

# **Objetivos**

## **Objetivo General**

Ilustrar la aplicabilidad del aprendizaje profundo en procesamiento digital de señales unidimensionales e imágenes por medio de redes neuronales, capaces de generar algoritmos para llegar a conclusiones sin ser supervisadas por humanos.

## **Objetivos Específicos**

- Descubrir por completo la arquitectura de una red neuronal.
- Implementar una red neuronal capaz de auto-programarse.
- Implementar Deep Learning mediante auto-codificaciones para producir salidas.
- Estudiar el funcionamiento de la técnica de Deep Learning contrastándolo con el funcionamiento de las redes neuronales por medio de sus respectivas aplicaciones.
- Seleccionar aplicaciones del Deep Learning en procesamiento digital de imágenes.



## **Hipótesis**

Para bien o para mal, hoy en día lo realmente importante a nivel tecnológico y económico se juega en el terreno de la inteligencia artificial aplicada. No obstante, la expresión más frecuente en el ámbito de Data Science es Machine Learning y, desde hace unos años, la que más impresiona es Deep Learning.

Tomando en cuenta que este tipo de tecnología se manifiesta de manera automática y es llamado así mismo, aprendizaje automático que es un tipo de inteligencia artificial que proporciona a las computadoras la capacidad de aprender, sin ser programadas explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden cambiar cuando se expone a nuevos datos.

La idea de Deep Learning mediante auto-codificadores aplicados es precisamente esa, usar varios codificadores, y entrenarlos uno a uno, usando cada codificador entrenado para entrenar al siguiente. Podríamos llamarlo un algoritmo voraz, y este es realmente el gran avance del Deep Learning que permite hacer todas esas cosas tan fantásticas que leemos últimamente. ¿Qué pasa si realmente necesitamos realizar aprendizaje supervisado? Por ejemplo, quizás no nos vale con que a la red vea videos de YouTube y descubra por si sola el concepto de gato, sino que queremos que aprenda a etiquetar objetos concretos que nos interesan.

Deep Learning se basa fundamentalmente en un montón de técnicas que ya existían con anterioridad. Algunas de estas técnicas son algo complejas, como el entrenamiento en redes de neuronas, pero existen un montón de bibliotecas para múltiples lenguajes de programación que las implementan por vosotros. El gran avance ha sido la manera de combinarlas, especialmente el uso de auto-codificadores aplicados que se entrenan capa a capa para lograr aprendizaje no supervisado eficiente.

## **Fundamentación teórica**

Básicamente es un conjunto de algoritmos de aprendizaje que usan técnicas de Redes neuronales y que en cada capa extra y procesan información útil para una siguiente capa de forma que es capaz de obtener como resultado un aprendizaje minucioso de la información de entrada y poder extraer conclusiones relevantes.

Existen dos grandes subtipos de aprendizaje supervisado: regresión, en el cual el valor predicho cae dentro de un rango (estilo ¿cuánto tiempo? ¿cuántas unidades?) o clasificación (estilo ¿es el tumor canceroso?).

A su vez, el aprendizaje no supervisado puede dividirse en dos subtipos: el clustering, que es esencialmente organizar los datos en grupos, y los algoritmos que persiguen proyectar los datos en un espacio de dimensión inferior y en este último caso las dimensiones deben interpretarse como factores semánticos.

Es posible que se necesiten aprender varias tareas simultáneamente. En el más famoso concurso de ML hasta la fecha, el premio del millón de dólares de Netflix al mejor recomendador de películas, lo óptimo es poder contar con las opiniones que cada usuario ha realizado sobre las películas que ha visto, y comparar dichas opiniones con las de otros usuarios para averiguar qué le gustará ver. Pero, por otra parte, existen usuarios recién llegados quienes no han publicado ningún rating. A estos usuarios por supuesto también se desea ofrecerles alguna recomendación por otro medio que no sea cotejar opiniones. Por consiguiente, se necesita alguna forma estadística de encontrar su similitud con otros usuarios. Evidentemente, ambas tareas están relacionadas. El truco para conseguir mejores modelos con ambas es no compartir demasiado ni demasiado poco entre ellas, lo cual depende de cuántos datos se disponga y cuánto conocimiento previo para el ajuste fino de cada tarea. Esto se conoce como multi-task learning.

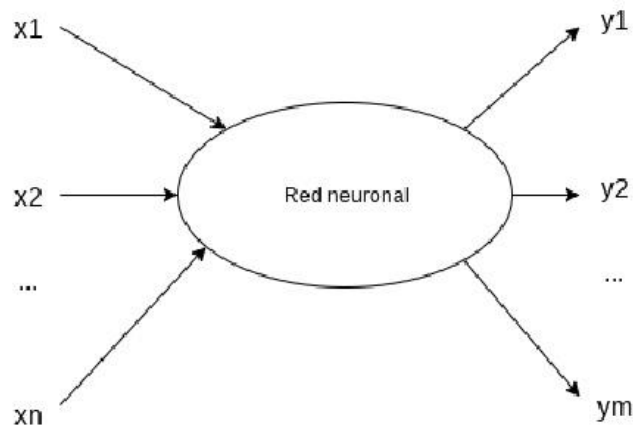
# **Deep learning**

¿Qué es el Deep learning?

El deep learning es un método que nos va a permitir generalmente clasificar cosas, es cierto que nos puede ayudar en otras áreas como, por ejemplo: predicciones, sistemas de compresión, generación automática de contenidos, traducciones, etc. Pero estas áreas no van a dejar de ser un subconjunto del problema de clasificación.

El deep learning se sustenta en el uso de redes neuronales, aunque pueden leer muchas cosas por ahí, estas redes no se basan en el funcionamiento del cerebro, básicamente por que desconocemos el funcionamiento exacto del cerebro. Este hecho es muy importante en cuanto a terminología, una red neuronal no está formada de neuronas, está formada de unidades.

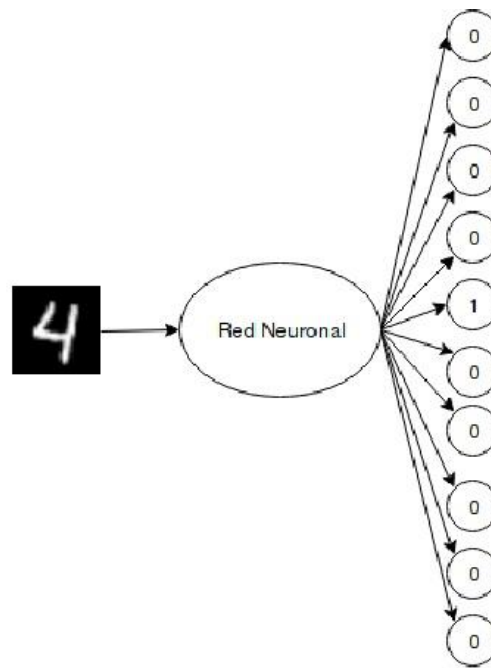
De momento veamos lo que es una red neuronal:



Una red neuronal puede ser representada básicamente por la ilustración superior. Es una función que acepta una serie de parámetros ( $x_1, x_2, \dots, x_n$ ) y devuelve uno o más valores ( $y_1, y_2, \dots, y_m$ ).

La particularidad que tiene esta función es que normalmente es suficientemente compleja como para adaptarse a una clasificación que utilizaríamos en la vida real, como por ejemplo etiquetar una imagen con el valor numérico del número que contiene.

Veamos un ejemplo práctico utilizando un conjunto de datos publico llamado "*MNIST*".



Para transformar la imagen en algo entendible por la red tenemos que preguntarnos que es una imagen del MNIST. Una imagen del MNIST es una matriz cuadrada que contiene valores de 0 a 255, siendo 0 los pixeles de color negro más oscuro y 255 el color más blanco más intenso, como ejemplo la siguiente imagen:



En este caso los valores están normalizados entre 0 y 1 una imagen de MNIST únicamente tenemos que dividir la matriz entre el escalar 255. La normalización es un tema de gran importancia en el *deep learning* pero lo trataremos más adelante. Los colores también están invertidos, pero para este caso nos resulta indiferente.

Ahora tenemos una matriz cuadrada y normalizada de 28x28, pero necesitamos convertirla en un vector, el proceso para conseguirlo es muy sencillo, simplemente concatenaremos todas las filas una detrás de otra para acabar teniendo un tensor de rango 1 que tiene 784 componentes.

Por otro lado, tenemos la salida, otro tensor de orden 1 con 10 componentes, cada componente representa una categoría, así por ejemplo  $y_1$  representa la categoría del número 0  $y_2$  representa la categoría del número 1,  $y_3$  la del 2, etc.

Lo que vamos a tratar de buscar, es que cuando metamos una imagen que represente un número, la componente de la salida que representa la categoría de la entrada valga un valor muy cercano a 1 y el resto de componentes su valor sea cercano al 0.

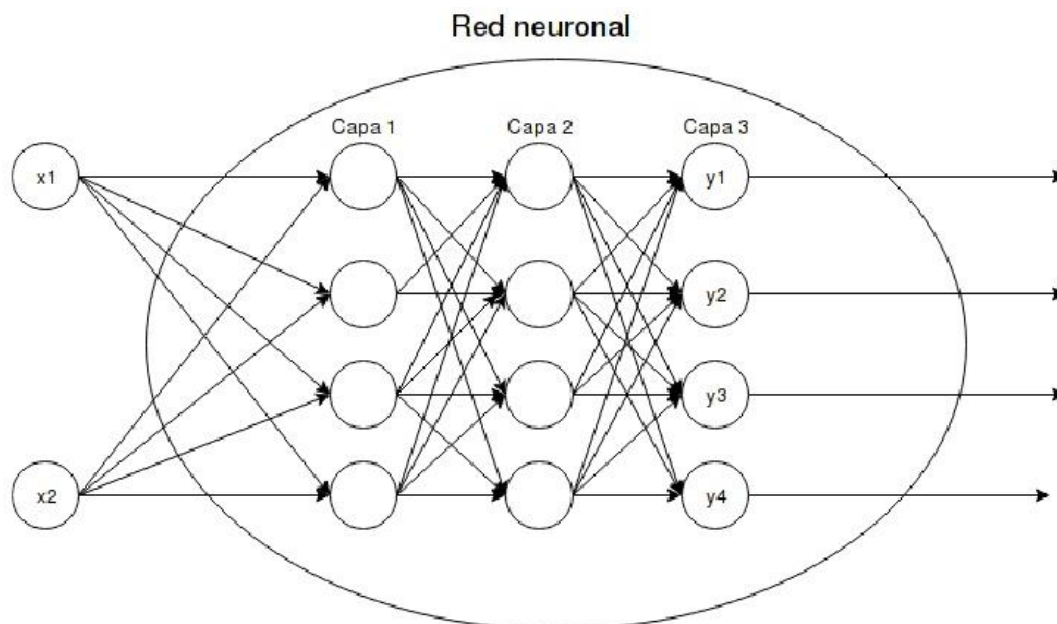
En el caso de la imagen de antes, donde se introduce un 4 a la red neuronal, vemos que la salida es perfecta: todos los componentes valen 0 excepto  $y_5$  que representa la categoría 4 y vale exactamente 1. Lamentablemente esta exactitud es muy difícil que suceda en la vida real.

## Modelo

Bueno, ya tenemos un modelo que podría llegar a predecir algo, pero por desgracia, si le pasáramos una imagen convertida en un tensor únicamente acertaríamos un 10% de las veces, como si la clasificación fuera aleatoria.

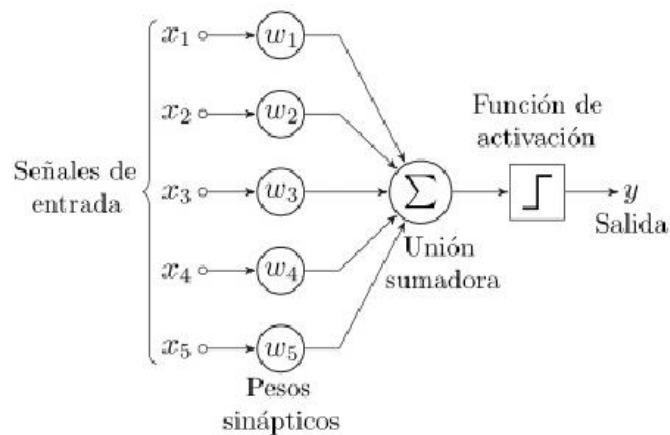
¿Por qué sucede esto?, bueno, esto sucede por cómo funciona internamente una red neuronal. Hemos visto cómo funciona a nivel general, pero una red neuronal está formada por pequeños componentes llamados unidades, estas unidades son funciones más pequeñas que aportan cierto valor a la salida de la red neuronal.

Observemos la siguiente imagen:



Cada círculo pequeño representa una unidad de la red neuronal, como pueden ver en la imagen, están agrupados en capas, y cada capa está completamente conectada a la siguiente. ¿Qué significa estar completamente conectadas?, significa únicamente que la salida de cada unidad es la entrada de cada unidad de la siguiente capa.

Vamos a ir un poco más allá para entender cómo se conectan las unidades entre sí, para ello veremos a detalle el funcionamiento de una unidad:



Como podemos ver, la imagen de atrás representa la función:  

$$y = f(x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + x_4 * w_4 + x_5 * w_5)$$

Siendo "f" la función de activación,  $x_n$  los valores de entrada de la unidad y "w" los pesos. Normalmente a la función se le añade una variable más llamada peso de sesgo que permitirá activar o desactivar la unidad, por lo que lo que la función quedaría:

$$y = f(x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + x_4 * w_4 + x_5 * w_5 + b)$$



Siendo es este caso  $b$  el peso de sesgo y que no está representado en la ilustración superior.

La función de activación ( $f$ ) que suele aplicar varía dependiendo de lo que queremos conseguir, pueden ser funcionales lineales o no lineales.

## Forward pass

El forward pass es el proceso por el cual, dada una entrada a la red neuronal, por ejemplo, la imagen con el número 4, obtenemos una salida  $(0,0,0,0,1,0,0,0,0,0)$ , esto básicamente se consigue multiplicando, sumando y aplicando funciones de activación de manera encadenada hasta que llegamos a la salida de la red neuronal. Es decir, es aplicar la función red neuronal a una entrada.

Para que la salida sea la que esperamos, antes debemos ajustar los pesos ( $w$ ) y sesgos ( $b$ ) para cuando se hagan todos los cálculos el resultado que obtengamos sea el que esperamos.

¿Los podemos ajustar a mano?, bueno, todo depende de tu paciencia, pero en general ajustarlos a mano únicamente vale para aproximar funciones sencillas como por ejemplo la función AND, OR, XOR, etc. Para todo lo demás deberemos utilizar otro método.

Hay que tener en cuenta que los modelos pueden tener decenas de capas, con cientos de unidades en cada capa y cuya salida influye en el resto de unidades que están por delante, ajustar todo a ojo sería lo más cercano al infierno que puede existir.

## Backward pass

El backward pass es lo que el común de los mortales suele llamar entrenamiento, el objetivo de esta fase es ir ajustando poco a poco las variables de peso ( $w$ ) y sesgo ( $b$ ) para que ajusten la salida de la red neuronal a lo que nosotros esperamos, para ello necesitaremos bastantes datos de entrada y el valor que esperamos obtener como salida en la red neuronal, en el caso del MNIST tendremos un montón de imágenes con números y el valor del número de la imagen.

Vamos a introducir otra función llamada función de coste o función de pérdida, esta función va a tomar como entrada lo que devuelve la red neuronal y va a devolver cuanto nos hemos confundido, algo tal que así:

$$\text{loss} = f(\text{Red neuronal})$$

Nuestro objetivo es minimizar el error, es decir, loss debe valer el menor valor posible, aunque no quiero entrar en mucho detalle, seguro que recordaran con cariño sus clases de cálculo encontrando mínimos en funciones. Pues esto es lo mismo, pero con una función con miles de variables: nuestros amigos los pesos ( $w$ ) y sesgos ( $b$ ).

No voy a entrar en como calcular el mínimo de una función, pero si destacare un par de cosas, calcular mínimos en funciones sencillas es algo fácil, el problema es que cuando tu función es gigantesca calcular el mínimo implica un coste computacional inasumible, es decir, el tiempo que vas a tardar en calcular los valores de  $w$  y  $b$  se dispara de tal forma que se vuelve inaceptable.

Una vez hemos realizado las funciones backward passes en nuestra red neuronal los valores de los pesos y sesgos deberían estar suficientemente bien ajustados como para que la predicción de nuestro modelo sea lo suficientemente buena como para usarlo en la vida.

## **Deep Learning: la aproximación a la percepción humana**

Posiblemente el futuro del aprendizaje automático pase por un giro hacia el aprendizaje no supervisado. En este paradigma los algoritmos son capaces de aprender sin intervención humana previa, sacando ellos mismos las conclusiones acerca de a la semántica embebida en los datos. Ya existen compañías que se centran completamente en enfoques de aprendizaje automático no supervisado, como “Loop AI Labs”, cuya plataforma cognitiva es capaz de procesar millones de documentos no estructurados y construir de forma autónoma representaciones estructuradas.

La disciplina del aprendizaje automático está en plena ebullición gracias a su aplicación en el mundo del Big Data y el “IoT”. No dejan de aparecer avances y mejoras de los algoritmos más tradicionales, desde los conjuntos de clasificadores hasta el Deep Learning. Que está muy de moda en la actualidad por su capacidad de acercarse cada vez más a la potencia perceptiva humana.

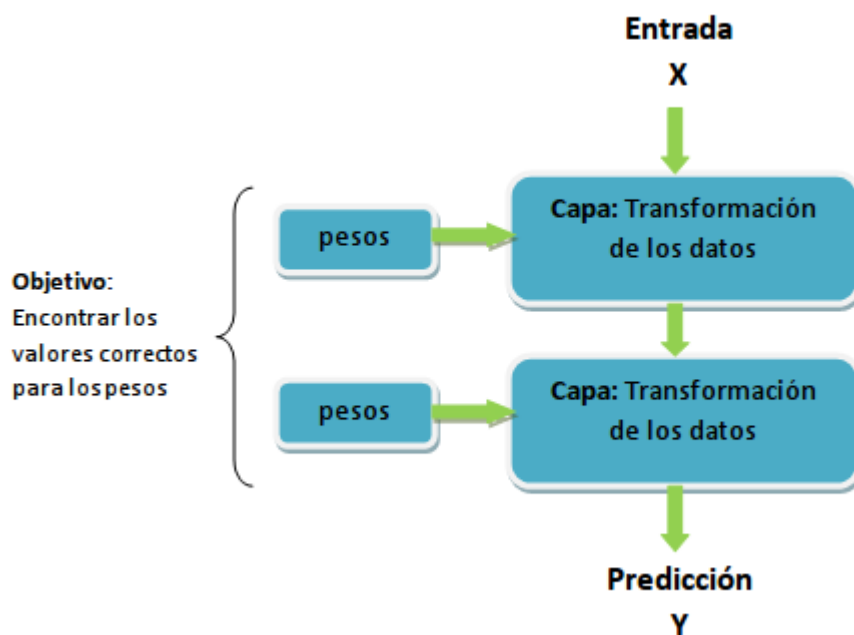
## **¿Cómo funciona el Deep Learning?**

En general, cualquier técnica de Machine Learning trata de realizar la asignación de entradas a salidas objetivo, mediante la observación de un gran número de ejemplos de entradas y salidas. El Deep Learning realiza este mapeo de entrada-a-objetivo por medio de una red neuronal artificial que está compuesta de un numero grande de capas dispuestas en forma de jerarquía. La red aprende algo simple en la capa inicial de la jerarquía y luego envía esta información a la siguiente capa. La siguiente capa toma esta información simple, lo combina en algo que es un poco más complejo, y lo pasa a la tercera capa. Este proceso continuo de forma tal que cada capa de la jerarquía construye algo más complejo de la entrada que recibió de a la capa anterior. De esta forma, la red ira aprendiendo por medio de la exposición a los datos de ejemplo.

La especificación de lo que cada capa hace a la entrada que recibe es almacenada en los pesos de la capa, que, en esencia, no son más que números.

Utilizando terminología más técnica podemos decir que la transformación de datos que se produce en la capa es parametrizada por sus pesos. Para que la red aprenda debemos encontrar los pesos de todas las capas de forma tal que la red realice un mapeo perfecto entre los ejemplos de entrada con sus respectivas salidas objetivo.

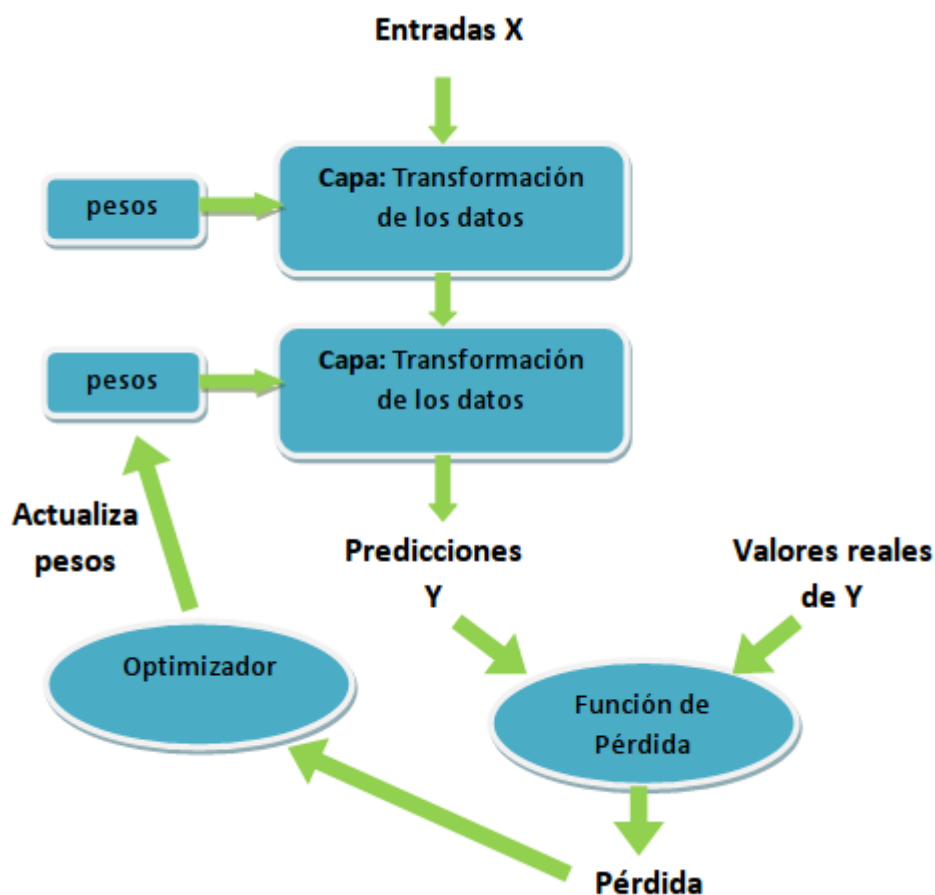
Pero el problema reside en que una red de Deep Learning puede tener millones de parámetros, por lo que encontrar el valor correcto de todos ellos puede ser una tarea realmente difícil, especialmente si la modificación del valor de uno de ellos afecta a todos los demás.



Para poder controlar algo, en primer lugar, debemos poder observarlo. En este sentido, para controlar la salida de la red neuronal, deberíamos poder medir cuán lejos está la salida que obtuvimos de la que se esperaba obtener. Este es el trabajo de la función de pérdida de la red. Esta función toma las predicciones que realiza el modelo y los valores objetivos, y calcula cuán lejos estamos de ese valor, de esta manera,

podemos capturar que tan buen está funcionando el modelo para el ejemplo especificado.

El truco fundamental del Deep Learning es utilizar el valor que nos devuelve esta función perdida para retroalimentar la red y ajustar los pesos en la dirección que vayan reduciendo la pérdida del modelo para cada ejemplo. Este ajuste, es el trabajo del optimizador, el cual implementa la propagación hacia atrás.



Resumiendo, el funcionamiento sería el siguiente: inicialmente, los pesos de cada capa son asignados en forma aleatoria, por lo que la red simplemente implementa una serie de transformaciones aleatorias.

En este primer paso, obviamente la salida del modelo dista bastante del ideal que deseamos obtener, por lo que el valor de la función de pérdida va a ser bastante alto.

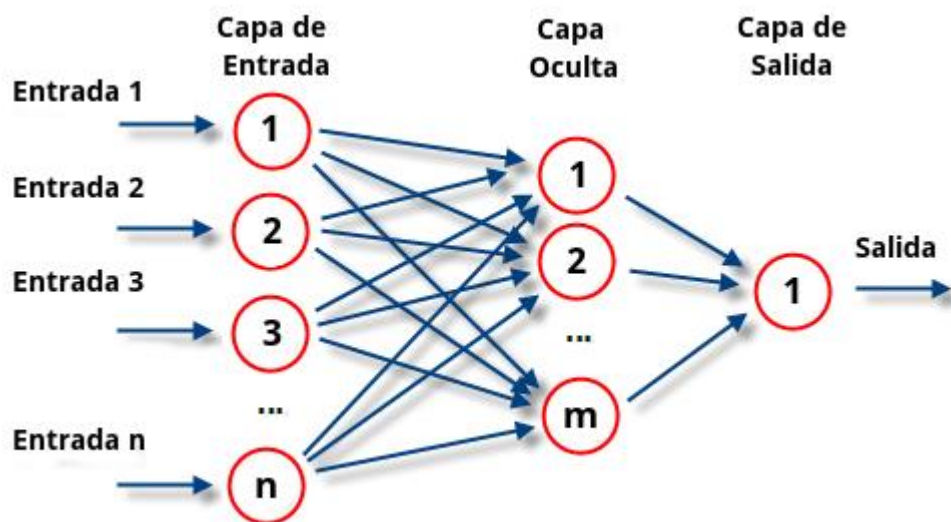
Pero a medida que la red va procesando nuevos casos, los pesos se va ajustando de forma tal de ir reduciendo cada vez más el valor de la función de pérdida. Este proceso es el que conoce como entrenamiento de la red, el cual, repetido una suficiente cantidad de veces, generalmente 10 interacciones de miles de ejemplos, logra que los pesos se ajusten a los que minimizan la función de pérdida. Una red que ha minimizado la pérdida es la que logra los resultados que mejor se ajustan a las salidas objetivo, es decir, que el modelo se encuentra entrenado.

## **Arquitecturas de Deep Learning**

La estructura de datos fundamental de una red neuronal esta vagamente inspirada en el cerebro humano. Cada una de nuestras células cerebrales está conectada a muchas otras neuronas por sinapsis. A medida que experimentamos e interactuamos con el mundo, nuestro cerebro crea nuevas conexiones, refuerza algunas conexiones y debilita a los demás. De esta forma, en nuestro cerebro se desarrollan ciertas regiones que se especializan en el procesamiento de determinadas entradas. Así vamos a tener un área especializada en la visión, otra que se especializa en la audición, otra para el lenguaje, etc. De forma similar, dependiendo del tipo de entradas con las que trabajemos, van a existir distintas arquitecturas de redes neuronales que mejor

se adaptan para procesar esa información. Algunas de las arquitecturas más populares son las siguientes:

- **Redes neuronales pre-alimentadas:** Las redes neuronales pre-alimentadas fueron las primeras que se desarrollaron y son el modelo más sencillo. En estas redes la información se mueve en una sola dirección: hacia adelante. Los principales exponentes de este tipo de arquitectura son el perceptrón y el perceptrón multicapa. Se suelen utilizar en problemas de clasificación simples.



- **Redes neuronales convolucionales:** Las redes neuronales convolucionales son muy similares a las redes neuronales ordinarias como el perceptrón multicapa; se componen de neuronas que tienen pesos y sesgos que pueden aprender. Cada neurona recibe algunas entradas, realiza un producto escalar y luego aplica una función de activación. Al igual que en el perceptrón multicapa también vamos a tener una función de pérdida o costo sobre la última capa, la cual estará totalmente conectada.



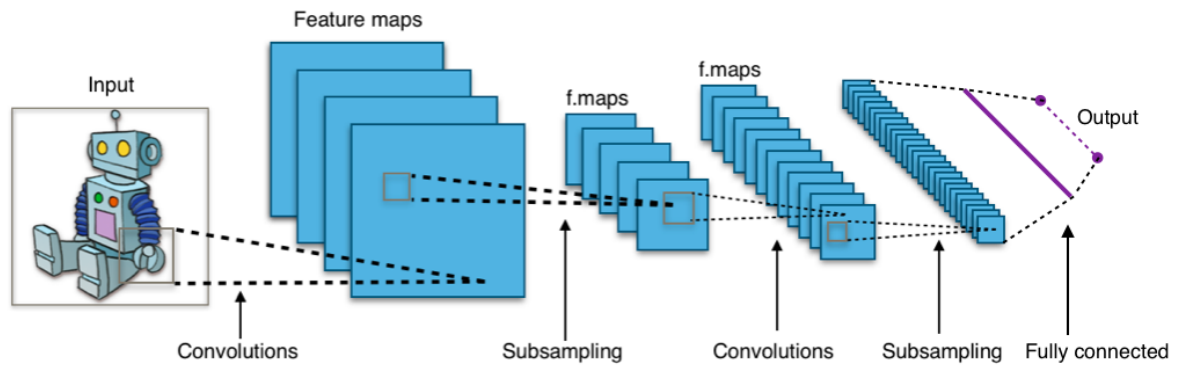
Lo que diferencia a las redes neuronales convolucionales es que suponen explícitamente que las entradas son imágenes, lo que nos permite codificar ciertas propiedades en la arquitectura; permitiendo ganar en eficiencia y reducir la cantidad de parámetros en la red.

En general, las redes neuronales convolucionales van a estar construidas con una estructura que contendrá 3 tipos distintos de capas:

1. Una capa convulcional, que es la que le da nombre a la red.
2. Una capa de reducción o de pooling, la cual va a reducir la cantidad de parámetros al que darse con las características más comunes.
3. Una capa clasificadora totalmente conectada, la cual nos va dar el resultado final de la red.

Algunas implementaciones específicas que podemos encontrar sobre este tipo de redes son: "inception v3, RestNet, VGG16 y xception, etc...".

Todas ellas han logrado excelentes resultados.

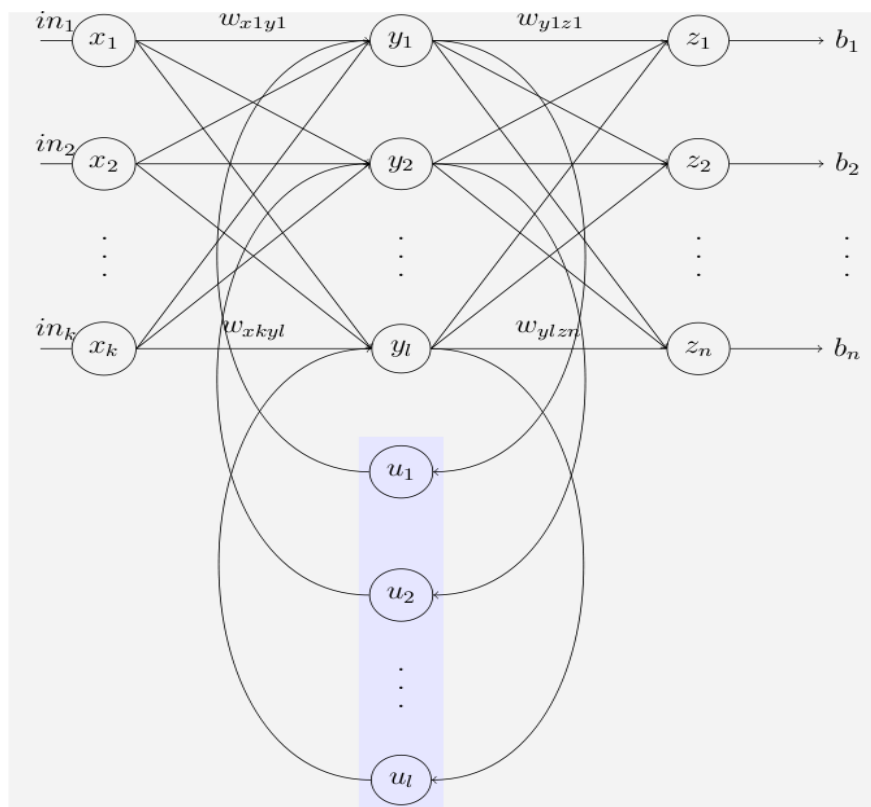


- **Redes neuronales recurrentes:** Los seres humanos no comenzamos nuestro pensamiento desde cero cada segundo, sino que los mismos tienen una persistencia. Las redes neuronales pre-alimentadas tradicionales no cuentan con esta persistencia, y esto parece una deficiencia importante. Las redes neuronales recurrentes abordan este problema. Son redes con bucles de retroalimentación, que permiten que la información persista.

Una red neuronal recurrente puede ser pensada como una red con múltiples copias de ella misma, en las que cada una de ellas pasa un mensaje a su sucesor. Esta naturaleza en forma de cadena revela que las Redes neuronales recurrentes están íntimamente relacionadas con las secuencias y listas; por lo que son ideales para trabajar con este tipo de datos. En los últimos años, ha habido un éxito increíble aplicando Redes neuronales recurrentes a una variedad de problemas como: reconocimiento de voz, modelado de lenguaje, traducción, subtítulos de imágenes y la lista continúa.

Las redes de memoria de largo plazo a corto plazo generalmente llamadas “LSTMs” son un tipo especial de redes neuronales recurrentes, capaces de aprender dependencias a largo plazo.

Elas también tienen una estructura como cadena, pero el módulo de repetición tiene una estructura diferente. En lugar de tener una sola capa de red neuronal, tiene un cuatro, que interactúan de una manera especial permitiendo tener una memoria a más largo plazo.



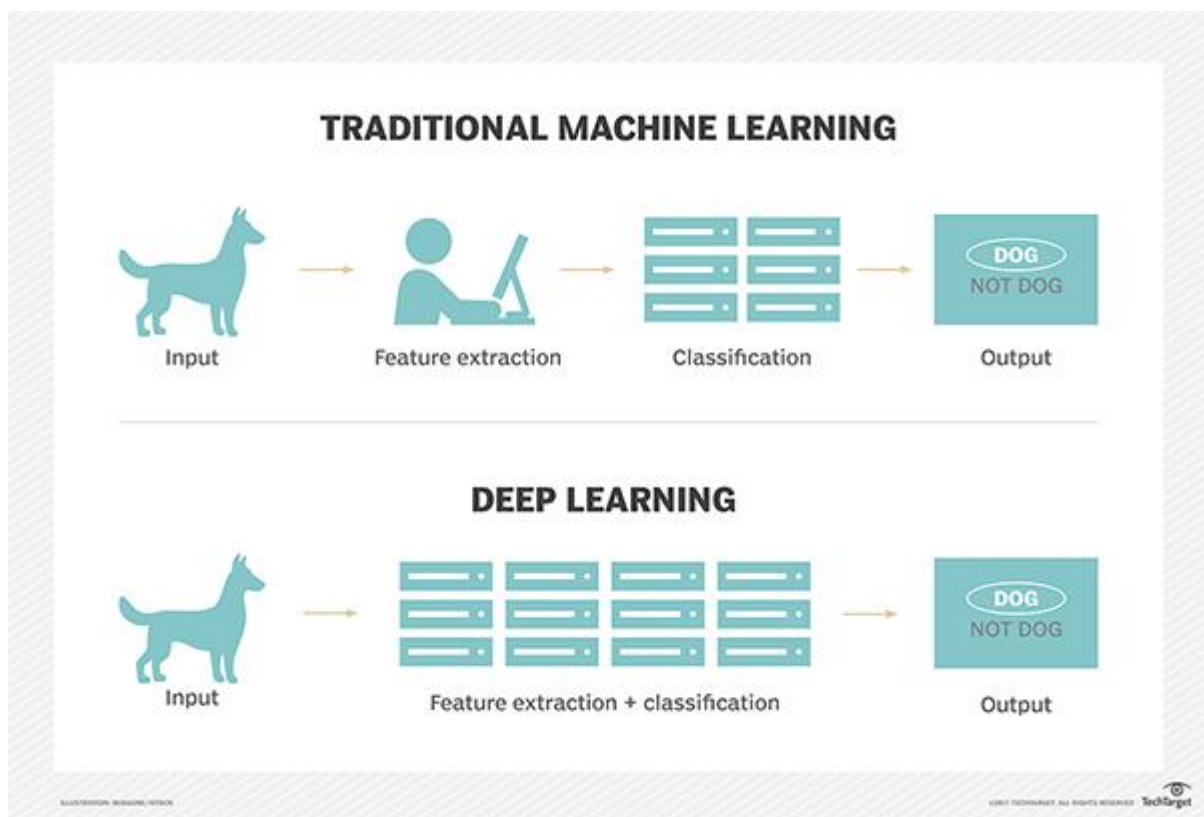
## **Posiblemente el futuro del aprendizaje automático pase por un giro hacia el aprendizaje no supervisado**

En el enfoque Deep Learning se usan estructuras lógicas que se asemejan en mayor medida a la organización del sistema nervioso de los mamíferos, teniendo capas de unidades de proceso (neuronas artificiales) que se especializan en detectar determinadas características existentes en los objetos percibidos. La visión artificial es una de las áreas donde el Deep Learning proporciona una mejora considerable en comparación con algoritmos más tradicionales. Existen varios entornos y bibliotecas de código de Deep Learning que se ejecutan en las potentes GPUs modernas tipo CUDA, como por ejemplo “NVIDIA cuDNN”.

El Deep Learning representa un acercamiento más íntimo al modo de funcionamiento del sistema nervioso humano. Nuestro encéfalo tiene una micro arquitectura de gran complejidad, en la que se han descubierto núcleos y áreas diferenciados cuyas redes de neuronas están especializadas para realizar tareas específicas.

En el aprendizaje tradicional de las máquinas, el proceso de aprendizaje es supervisado y el programador tiene que ser muy, muy específico al decirle a la computadora que tipos de cosas debe buscar para decidir si una imagen contiene un perro o no contiene un perro.

Este es un proceso laborioso llamado extracción de características y la tasa de éxito de la computadora depende totalmente de la capacidad del programador para definir con precisión un conjunto de características para “perro”. La ventaja del aprendizaje profundo es que el programa construye el conjunto de características por sí mismo sin supervisión. Esto no es solo más rápido, si no que por lo general es más preciso.



Inicialmente, el programa de computadora podría ser provisto de datos de entrenamiento, un conjunto de imágenes para las cuales un humano ha etiquetado cada imagen “perro” o “no perro” con meta-etiquetas.

El programa utiliza la información que recibe de los datos de entrenamiento para crear un conjunto de características para el perro y construir un modelo predictivo. En este caso, el modelo que la computadora crea por primera vez podría predecir que cualquier cosa en una imagen que tenga cuatro patas y una cola debería estar etiquetada como “perro”. Por supuesto, el programa no es consciente de las etiquetas “cuatro patas” o “cola”, simplemente buscaría patrones de píxeles en los datos digitales. Con cada iteración, el modelo predictivo que crea el equipo de cómputo se vuelve más complejo y más preciso.

A diferencia del niño pequeño, que tardara semanas o incluso meses en comprender el concepto de “perro”, un programa informático que utiliza algoritmos de aprendizaje profundo puede mostrar un conjunto de entrenamiento y ordenarlo a través de millones de imágenes, identificando con precisión que imágenes tienen perros en tan solo unos minutos.

Con el fin de lograr un nivel aceptable de precisión, los programas de Deep Learning requieren acceso a inmensas cantidades de datos de entrenamiento y poder de procesamiento, ninguno de los cuales estaba fácilmente disponible para los programadores hasta la era de Big Data y la computación en la nube.

Debido a que la programación del Deep Learning es capaz de crear modelos estadísticos precisos a partir de grandes cantidades de datos no etiquetados y no estructurados.

Esto es importante a medida que el Internet de las cosas (IoT) continúa haciéndose más penetrante, porque la mayoría de los datos que los seres humanos y las maquinas crean están desestructurados y no están etiquetados. Los casos de uso de hoy para el Deep Learning incluyen todos los tipos de aplicaciones de análisis de Big Data, especialmente aquellos enfocados en el procesamiento del lenguaje natural (NLP), traducción de idiomas, diagnóstico médico, señales de comercio bursátil, seguridad de redes e identificación de imágenes.

## **¿Qué innovaciones hemos conseguido gracias al Deep Learning?**

Las multinacionales tecnológicas más importantes ya utilizan el Deep Learning para resolver problemas del modo actual y conseguir que los ordenadores actúen y piensen como humanos. Como de costumbre, Google ha sido el pionero en la materia. Procura imitar el funcionamiento cerebral con su proyecto de Google Brain con el que han revolucionado el reconocimiento de voz e imagen.

Mismamente, en 2012, el equipo de Andrew NG necesito 16.000 ordenadores para reconocer un gato en concreto entre los más de 10 millones de videos de YouTube. Gracias al I+D en Deep Learning ahora se necesitan muchísimos ordenadores menos.

Un ejemplo que conocemos todos es el reconocimiento fotográfico que utiliza Facebook para conseguir etiquetar las distintas caras que aparece en una foto y al mismo tiempo para identificar la localización de una imagen. En un trabajo similar ha trabajado el Brain Team de Google con un proyecto denominado "Show and Tell" que utiliza el sistema de reconocimiento TensorFlow que dispone de un motor que con una precisión del 93,9% muestra en un pequeño texto lo que ocurre en la imagen, es decir no solo a quien pertenece una cara, sino lo que ocurre en una foto, el lugar, el ambiente....

Twitter, que no iba a ser menos, ha preferido potenciar el Deep Learning para mejorar la calidad de imagen de sus videos en streaming. Las redes de neuronas permiten comprimir más el video.

Prometen facilitar la rutina diaria deduciendo con una simple foto cuantas calorías puede tener un plato o incluso que los coches puedan conducir de una forma autónoma reconociendo patrones de conducción en la que quizá sea la idea más ambiciosa relacionada con la IA.

En el deporte ya nos permiten reconocer de forma sencilla si existe en una jugada fuera de juego o quien ha cruzado la meta primera en los 100m lisos.

Pero todo tiene un proceso y en los inicios siempre hay muchos bugs y fallos. Pero para que se puedan utilizar de verdad todas estas ideas necesitan el más profundo perfeccionamiento, no sirve el 99% si ese 1% puede provocar daños en los humanos.



## Logros del Deep Learning

En los últimos años el Deep Learning ha producido toda una revolución en el campo del Machine Learning, con resultados notables en todos los problemas de percepción, como ver y escuchar, problemas que implican habilidades que parecen muy naturales e intuitivas para los seres humanos, pero que desde hace tiempo se han mostrado difíciles para las máquinas.

En particular, el Deep Learning ha logrado los siguientes avances, todos ellos en áreas históricamente difíciles del Machine Learning.

- Un nivel casi humano para la clasificación de imágenes.
- Un nivel casi humano para el reconocimiento del lenguaje hablado.
- Un nivel casi humano en el reconocimiento de escritura.
- Grandes mejoras en traducciones de lenguas.
- Grandes mejoras en conversaciones text-to-speech.
- Asistentes digitales como Google Assistant o Siri.
- Un nivel casi humano en autos autónomos.
- Mejores resultados de búsqueda en la web.
- Grandes mejoras para responder preguntas en lenguaje natural.
- Alcanzado “Nivel Maestro” en varios juegos.

En muchos sentidos, el Deep Learning todavía sigue siendo un campo misterioso para explorar, por lo que seguramente veremos nuevos avances en nuevas áreas utilizando estas técnicas. Tal vez algún día el Deep Learning ayude a los seres humanos a hacer ciencia, desarrollar software y mucho más.

## ¿Por qué estos sorprendentes resultados surgen ahora?

Muchos de los conceptos del Deep Learning se desarrollaron en los años 80s y 90s, algunos incluso mucho antes. Sin embargo, los primeros resultados exitosos del Deep Learning surgieron en los últimos 5 años. ¿Qué fue lo que cambió para lograr la popularidad y éxito de los modelos basados en Deep Learning en estos últimos años?

Si bien existen múltiples factores para explicar esta revolución del Deep Learning, los dos principales componentes parecen ser la disponibilidad de masivos volúmenes de datos, lo que actualmente se conoce bajo el nombre de “Big Data”; y el progreso en el poder de cómputo, especialmente gracias a los GPUs. Entonces, dentro de los factores que explican esta popularidad de los modelos de Deep Learning podemos encontrar:

- **La disponibilidad de conjuntos de datos enormes y de buena calidad**

Gracias a la revolución digital en que nos encontramos, podemos generar conjuntos de datos enormes con los cuales alimentar a los algoritmos de Deep Learning, los cuales necesitan de muchos datos para poder generalizar.

- **Computación paralela masiva con GPUs**

En líneas generales, los modelos de redes neuronales no son más que complicados cálculos numéricos que se realizan en paralelo. Gracias al desarrollo de los GPUs estos cálculos ahora se pueden realizar de forma mucho más rápida, permitiendo que podamos encontrar modelos más profundos y grandes.

- **Funciones de activación amigables para Backpropagation**

La propagación hacia atrás o Backpropagation es el algoritmo fundamental que hace funcionar a las redes neuronales; pero la forma en que trabaja implica cálculos realmente complicados. La transición desde funciones de activación como "tanh o sigmoid" a funciones como "ReLU o SELU" han simplificado estos problemas.

- **Nuevas arquitecturas de regularización**

Técnicas como "dropout, batch normalization y data-augmentation" nos permiten entrenar redes más grandes con menos peligros de sobreajuste.

## ▪ Optimizadores más robustos

La optimización es fundamental para el funcionamiento de las redes neuronales. Mejoras sobre el tradicional procedimiento “SGD”, como “ADAM” han ayudado a mejorar el rendimiento de los modelos.

## ▪ Plataformas de software

Herramientas como “TensorFlow, Theano, Keras, CNTK, PyTorch, Chainer, y mxnet” nos permiten crear prototipos en forma más rápida y trabajar con GPUs sin tantas complicaciones.

Nos permiten enfocarnos en la estructura del modelo sin tener que preocuparnos por los detalles de más bajo nivel.

Otra razón por la que el Deep Learning ha tenido tanta repercusión últimamente además de ofrecer un mejor rendimiento en muchos problemas; es que el Deep Learning está haciendo resolución de problemas mucho más fácil, ya que automatiza completamente lo que solía ser uno de los pasos más difíciles y cruciales en el flujo de trabajo de Machine Learning: la ingeniería de atributos. Antes del Deep Learning, para poder entrenar un modelo, primero debíamos refinar las entradas para adaptarlas al tipo de transformación del modelo; teníamos que cuidadosamente seleccionar los atributos más representativos y desechar los poco informativos.

## **CONCLUSIÓN**

No existe una única definición de Deep Learning. En general se trata de una clase de algoritmos para aprendizaje automático. A partir de este punto común, diferentes publicaciones se centran en distintas características, como, por ejemplo:

- Usar una cascada de capas con unidades de procesamiento no lineal para extraer y transformar variables. Cada capa usa la salida de la capa anterior como entrada. Los algoritmos pueden utilizar aprendizaje supervisado o aprendizaje no supervisado, y las aplicaciones incluyen modelización de datos y reconocimiento de patrones.
- Estar basados en el aprendizaje de múltiples niveles de características o representaciones de datos. Las características de más alto nivel se derivan de las características de nivel inferior para formar una representación jerárquica.
- Aprender múltiples niveles de representación que corresponden con diferentes niveles de abstracción. Estos niveles forman una jerarquía de conceptos.

Todas estas maneras de definir el aprendizaje profundo tienen en común: múltiples capas de procesamiento no lineal.

## **BIBLIOGRAFIA**

- <https://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial>
- <http://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-profundo-deep-learning>
- <https://lapastillaroja.net/2015/02/ml-algols/>
- <https://relopezbriega.github.io/blog/2017/06/13/introduccion-al-deep-learning/>
- <http://www.iebschool.com/blog/machine-learning-deep-learning-big-data/>