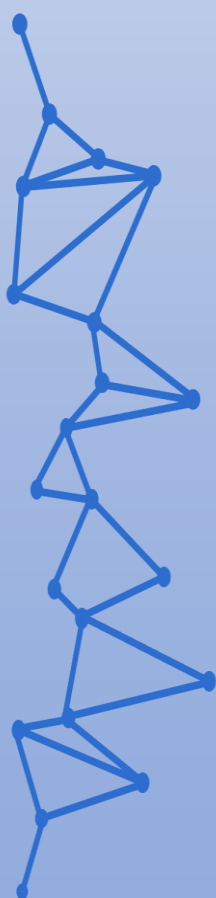




Curso de Especialización de Inteligencia Artificial y Big Data (IABD)



Sistemas de aprendizaje Automático

UD03. Redes neuronales y Deep Learning.
Resumen.

JUAN ANTONIO GARCIA MUELAS

El nombre de las redes neuronales en Inteligencia Artificial procede de las red de neuronas biológicas de los organismos vivos que pretendían imitarse en los inicios de estas técnicas.

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.

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.

A semeja 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.

El **perceptrón** es una **neurona artificial** que realiza ciertos cálculos para detectar capacidades de datos de entrada.

Inicialmente el perceptrón fue planteado por Frank Rosenblatt, en 1957, como un programa y no tanto como una máquina. Fue más adelante cuando su concepto se transformó al ámbito del software.

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.

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**, de **objetivo de sonda**, de **voz** o de **caracteres** manuscritos.
- ✓ **Visión** por computador.

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.

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.

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 y Detección de series temporales.
- ✓ Reconocimiento y Síntesis de voz.
- ✓ Aprendizaje del ritmo.
- ✓ Composición musical.

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.

Red de Unidades Periódicas Cerradas - *Gated Recurrent Unit (GRU)*: 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 y señales de voz.
- ✓ Procesamiento natural del lenguaje.

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 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:

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

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 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.

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 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 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.
- ✓ 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 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 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.

¿En qué ámbitos están dando mejores resultados las redes neuronales convolucionales? En reconocimiento de voz y procesamiento de imágenes.

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 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 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.

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.

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 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.

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).

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.

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:

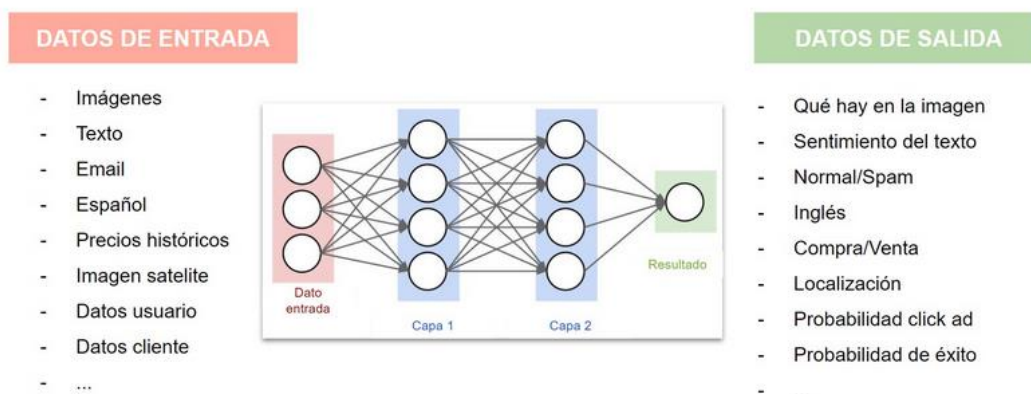
- ✓ Detección de rostro.
- ✓ Categorización de texto.
- ✓ Clasificación.
- ✓ Bioinformática.
- ✓ Reconocimiento de escritura a mano.

Máquina 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.

Funcionamiento práctico de una Red Neuronal

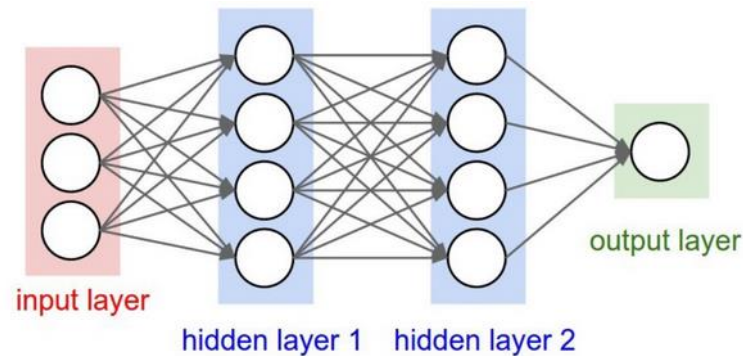


El aprendizaje profundo (Deep Learning) es una de las principales técnicas del aprendizaje automático (Machine Learning).

La principal característica del Aprendizaje Profundo es que el modelo sigue aprendiendo y mejorando su precisión cada vez que se utiliza.

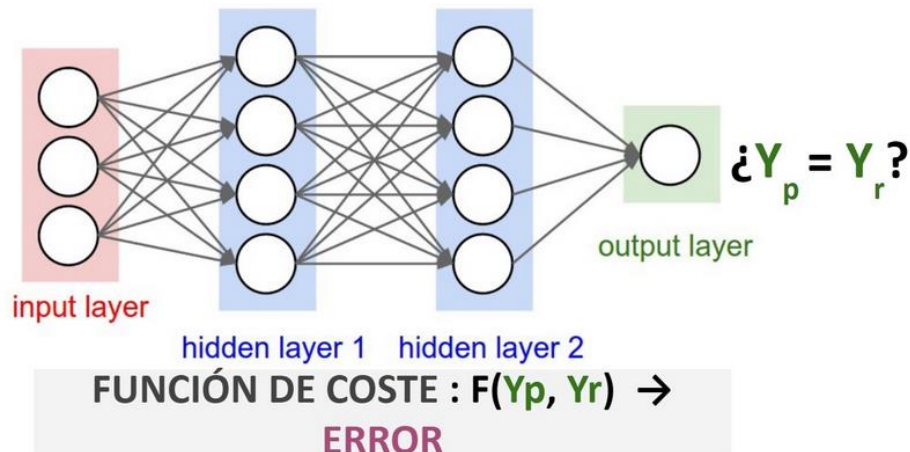
Fases del entrenamiento de una Red Neuronal Profunda.

- ✓ **Fase 1 - Forward Pass:** Se procesa la información de entrada y se genera un resultado.

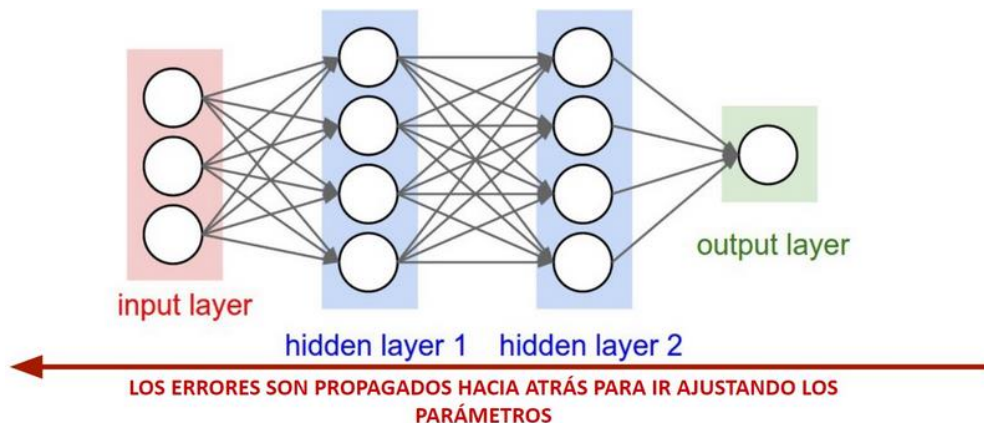


$X \rightarrow (\text{Suma ponderada} \rightarrow \text{Función de activación}) \times N^{\circ} \text{ Capas} \rightarrow Y_p$

- ✓ **Fase 2 - Función de coste:** Se comparan los resultados y se detectan errores.



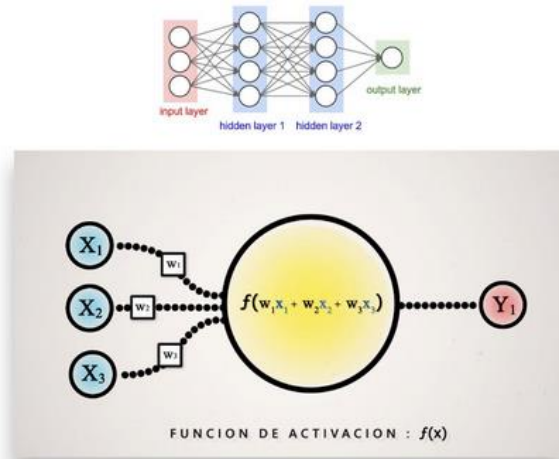
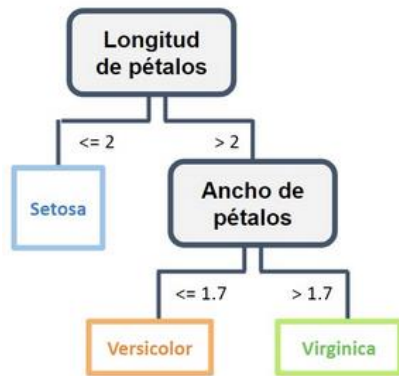
- ✓ **Fase 3 - Backward Pass:** Analiza los resultados clasificados como errores o no suficientemente buenos, retropropaga dichos errores en el resultado en el algoritmo hacia atrás, para encontrar qué parámetros ha utilizado para llegar a esa conclusión.



- ✓ **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.



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.