**Machine learning** -> Aprender tareas sin ser programadas. Utilizan estadísticas para predecir y reconocer patrones.

**Red neuronal** -> Programa de machine learning que toma decisiones para identificar, sopesar opciones y llegar a conclusiones. Tiene 3 capas:

* **Entradas**: Representa una característica o atributo
* **Oculta** (una y varias): Realizan funciones de suma ponderada y de activación
* **Salida**: Produce predicciones o resultados finales

Cada nodo procesa su información, sino no actúa. Cuantos más datos les des, más aprenderán.

Las **redes neuronales** se basan en datos de entrenamiento para aprender y mejorar su precisión con el tiempo. Ejemplo algoritmo de búsqueda de Google.

A veces se denominan **ANN** (Red neuronal artificial) o **SNN** (Red neuronal simuladas)

**Red neuronal multicapa/profunda**

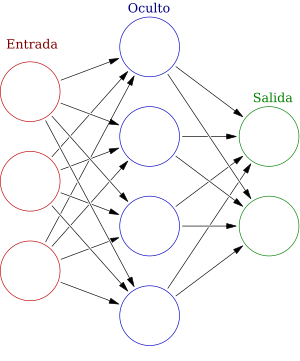
El término **profundo** en las **redes neuronales profundas** se refiere a redes con muchas capas ocultas. Pueden aprender automáticamente a partir de datos sin procesar.

**Retropropagación**: Entrenamiento de la red neuronal, que hace uso de algoritmos de optimización

El proceso es el mismo que la red neuronal normal, lo único que añade son varias capas ocultas más. La descripción básica de la multicapa:

* **Entrada**: Representa una caracteristica o atributo
* **Oculta**: Reapliza funciones de suma ponderada y activacion
* **Pesos y sesgos**: Cada conexión entre neuronas (**sinapsis**) tiene un peso y cada neurona tiene un determinado sesgo. Son parámetros que se pueden aprender y ajustar
* **Suma ponderada y activación**: Primero cada neurona calcula la suma y luego se activa, y se repite el proceso en las capas ocultas
* **Salida**: Produce predicciones o resultados finales
* **Funciones de activación**: Permite moldear relaciones complejas entre los datos. Algunas funciones de activación: Relu, Prelu, tanh

**Neurona**: Grupo de nodos interconectado. Ejemplo de neurona: Donde cada flecha es la conexión



**Deep learning**: Subconjunto del machine learning. Utiliza redes neuronales profundas (multicapa) para simular la toma de decisiones.

**Tipos de IA**

* **IA Débil (o estrecha)** : Que nos facilitan la vida (Asistente virtual, Chatbots, Filtros de spam). Solo cogen información y te devuelven información. No imitan al ser humano.

No busca emular sin la necesidad de comprender o interpretar el mundo en su totalidad. inteligencia humana, se centra en ser práctica y funcional

* **IA Fuerte (o generativa):** Busca replicar las capacidades humanas. Ej: robots que te haga la comida, asistentes de IA avanzados, coches autónomos. Caracteristicas:
  + *Replicar capacidades cognitivas humanas*
  + *Habilidades resolver problemas*

Funciona mediante aprendizaje profundo

La IA débil se le manda algo y lo hace, en cambio la IA fuerte toma sus propias decisiones

Los sistemas expertos se utilizan para simular programas de expertos.

**Sistemas ML**: Son muy eficientes en búsqueda, pero no son capaces de identificar el problema

**Privacidad diferencial se dedica a meter ruido**

**Aprendizaje Automático Clasificación de Sistemas**

* **Supervisado**: Consiste en que, a partir de unos datos, te da unos resultados

*Tipos de etiqueta*:

* + Clasificación: varias etiquetas
  + Regresión: una sola etiqueta

*Algoritmos*:

* + Redes neuronales: Para Deep learning. Cada nodo se compone de entrada, capas y salida
  + Naive bayes: Basado en el teorema de Bayes (una característica no depende de otra).
  + Regresión lineal: Para identificar la relación entre una variable dependiente y una/varias independientes. Simple y versátil para aplicaciones de clasificación binaria. Tipo de etiqueta: Regresion
  + Regresión logística: Se utiliza cuando las variables dependientes son continuas, se selecciona cuando la variable dependiente es categórica (verdadero/falso, 1/0). Se utiliza para clasificación binaria, responde si/no. Tipo de etiqueta: Regresion
  + Máquinas de vectores de soporte (SVM): Se aprovecha de los problemas de clasificación. Desarrollado por Vladimir Vapnik. Tipo de etiqueta Clasificacion
  + K vecino más cercano: Conocido como algoritmo KNN, algoritmo no paramétrico. Si todos menos 1 hacen algo, probablemente acabe haciendo algo. Se usa para problema de clasificación y regresión. Tipo de etiqueta Clasificacion
    - Algoritmo paramétrico: Utiliza un número fijo de parámetros para predecir nuevos puntos de datos.
    - Algoritmo no paramétrico: No hace suposiciones sobre los datos lo que permite una mayor flexibilidad y adaptabilidad a la estructura de los datos.
  + Random forest: Se utiliza con fines de clasificación y regresión. Hace referencia a una colección de árboles de decisión no correlacionados. Tipo de etiqueta Clasificacion.

**Que es un proceso de regresión:** Proceso estadístico que permite analizar la relación entre 2 o mas variables

* **No supervisado**: Se retroalimenta a través de sus propios resultados. Aparecen 2 tipos:
  + **Association**: Relación entre los datos.
  + **Clustering (o agrupación):** A partir de unos resultados vas haciendo agrupaciones. A partir de esas agrupaciones da resultados (posiblemente más específicos). Tipos de algoritmos basados en:
    - Densidad: Encuentra los lugares más densos en puntos de datos y los llama grupos. Pueden tener cualquier forma, no toma en cuenta valores atípicos
    - Distribución: A partir de un punto central, según se aleja del centro la probabilidad disminuye.
    - Centroides: Es el centro de algo, siendo equidistante de los vértices. Media de todos los puntos de datos dentro de un clúster. El cuadrado del valor del centroide es la distancia al cuadrado (Si es 3, 3^2 = 9)

Cada cluster va a tener un centroide.

Ventajas:

* + - * Escalabilidad
      * Interpretabilidad
      * Flexibilidad

Limitaciones:

* + - * Sensibilidad a las condiciones iniciales
      * Asunción de grupos esféricos
    - Jerarquías: Se utiliza en datos jerárquicos. Es perfecto para tipos específicos de conjunto de datos (Es el más restrictivo). Es no supervisado que se utiliza para agrupar puntos de datos no etiquetados. Construye un árbol de grupos para que todo esté organizado de arriba hacia abajo.\*

Existen 2 tipos

* + - * **Divisivo**: todos los puntos de datos se tratan como un gran cluster y el proceso de agrupación implica la división del gran cluster en varios clusters pequeños
      * **Aglomeración**: puntos de datos se agrupan utilizando un enfoque ascendente

**\*Dendrograma**: resultado es una estructura en forma de árbol invertido

**Taxonomía**: clasificación de cosas(animales, insectos) micotómicas (2 versiones), como un árbol de decisión. Necesita muchos datos.

* **Semisupervisado**: Estudia y trabaja con la mitad de los datos (los más significativos)
* **Por refuerzo**: Basada en un sistema de recompensas. Se retroalimentan.

**Tipos de datos**

* **Categóricos**: algo que no tiene continuidad, es decir, si no, perro gato. Puntos al azar en una gráfica. los números enteros son categóricos
* **Numéricos**: Números, que tienen una continuidad (una línea dentro de una gráfica) (ejemplo, números reales)

Si puedes calcular la media, se consideran datos numéricos. Si no puedes averiguar la media, entonces se consideran datos categóricos.

**MALENTENDIDOS**

Correlación implica causalidad.

* Realidad: Causalidad es más que solo establecer correlaciones.

Al desarrollar sistemas de aprendizaje automático, cuanto más datos y mayor sea la variedad, mejor.

* Realidad: Los conjuntos de datos de entrenamiento de ML deben seleccionarse para cumplir umbrales de precisión y representatividad.

ML necesita datos de entrenamiento completamente libres de errores.

* Realidad: Para conseguir un buen rendimiento los modelos de ML se necesitan datos de entrenamiento con tan solo una calidad superior a un cierto umbral.

El desarrollo de sistemas de ML requiere grandes repositorios de datos o el intercambio de conjuntos de datos de diferentes fuentes.

* Realidad: El aprendizaje federado permite el desarrollo de sistemas de aprendizaje automático sin compartir datos de entrenamiento.

Los modelos de ML mejoran automáticamente con el tiempo.

* Realidad: Una vez implementados, el rendimiento de los modelos de ML puede deteriorarse y no mejorará a menos que reciba entrenamiento adicional.

Las decisiones automáticas tomadas por los algoritmos de ML no pueden ser explicadas.

* Realidad: Un modelo de ML bien diseñado puede producir decisiones comprensibles para todas las partes interesadas relevantes.

La transparencia en ML viola la propiedad intelectual y no es entendida por el usuario.

* Realidad: Es posible proporcionar una transparencia significativa a los usuarios de IA sin dañar la propiedad intelectual.

Los sistemas de ML están sujetos a menos sesgos que los propios humanos.

* Realidad: Los sistemas de ML están sujetos a diferentes tipos de sesgos y algunos de estos provienen de sesgos humanos.

ML puede predecir con precisión el futuro.

* Realidad: Las predicciones de un sistema ML solo son precisas cuando los eventos futuros reproducen tendencias pasadas.

Los interesados son capaces de anticipar las posibles salidas que los sistemas de ML pueden dar con sus datos.

* Realidad: La capacidad de ML para encontrar correlaciones no evidentes en los datos puede terminar por revelar información personal adicional, sin que el interesado sea consciente de ello.