

1.2. Aprendizaje automático y aprendizaje profundo

El aprendizaje automático o *machine learning* es una de las ramas de las ciencias de la computación y, en concreto, una de las ramas de la inteligencia artificial. Este término fue acuñado en el año 1959 por el informático estadounidense Arthur Samuel y hace referencia a las técnicas centradas en el desarrollo de algoritmos que buscan que las máquinas realicen un aprendizaje.

Se considera que una máquina ha realizado un aprendizaje cuando, en base a su experiencia, está haciendo un mejor desempeño de la tarea que le había sido asignada. De este modo, se pueden resolver problemas sin la necesidad de programarlos de forma explícita, ya que la máquina puede desarrollar una lógica propia utilizando los datos que se le proporcionen.

La limitación de los algoritmos de aprendizaje automático era por un lado, la cantidad de datos que podían utilizar para desarrollarse y, por otro, el tiempo de compilación que requerían. Así, en la década de los 2010, gracias al avance tecnológico tanto en el desarrollo de algoritmos como en el hardware de los ordenadores se pudieron plantear problemas más complejos y nace el concepto de aprendizaje profundo.

El aprendizaje profundo o *deep learning* es una rama del aprendizaje automático enfocada a la creación de sistemas más complejos tales como las redes neuronales artificiales. Estas redes cuentan con un gran número de neuronas y buscan imitar el comportamiento del cerebro humano mediante el análisis de grandes cantidades de datos.

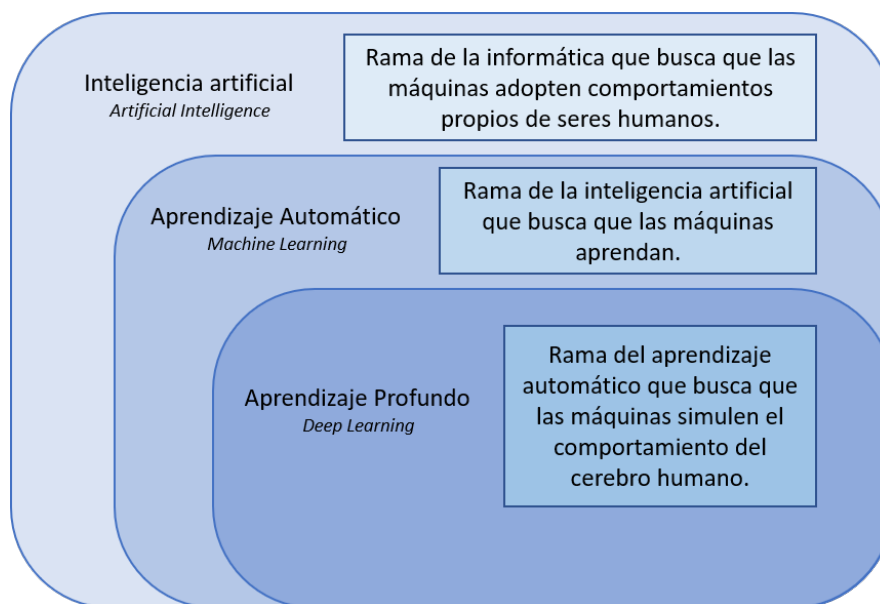


Figura 1.1: Diagrama de la inteligencia artificial.

Algunas de las aplicaciones del aprendizaje profundo son:

- Traducción automática
- Automatización industrial
- Añadir color a imágenes y vídeos
- Mejorar experiencias de clientes mediante *chatbots*
- Detección de objetos en satélites

1.3. Tipos de aprendizaje automático

1.3.1. Aprendizaje supervisado

El aprendizaje supervisado se caracteriza por utilizar un conjunto de datos al que se denomina conjunto de entrenamiento para que los algoritmos aprendan. Así, utilizamos un conjunto de tuplas $\{(X_1, Y_1), \dots, (X_n, Y_n)\}$ donde el conjunto $\{X_i\}_{i=1, \dots, n}$ son los datos de entrada y el conjunto $\{Y_i\}_{i=1, \dots, n}$ son los datos de salida esperada. Introduciendo estos datos, el algoritmo aprende qué salida podemos esperar para cada uno de los datos del conjunto de entrada, por lo que, posteriormente, podremos utilizar el algoritmo para predecir salidas que no conocíamos a priori.

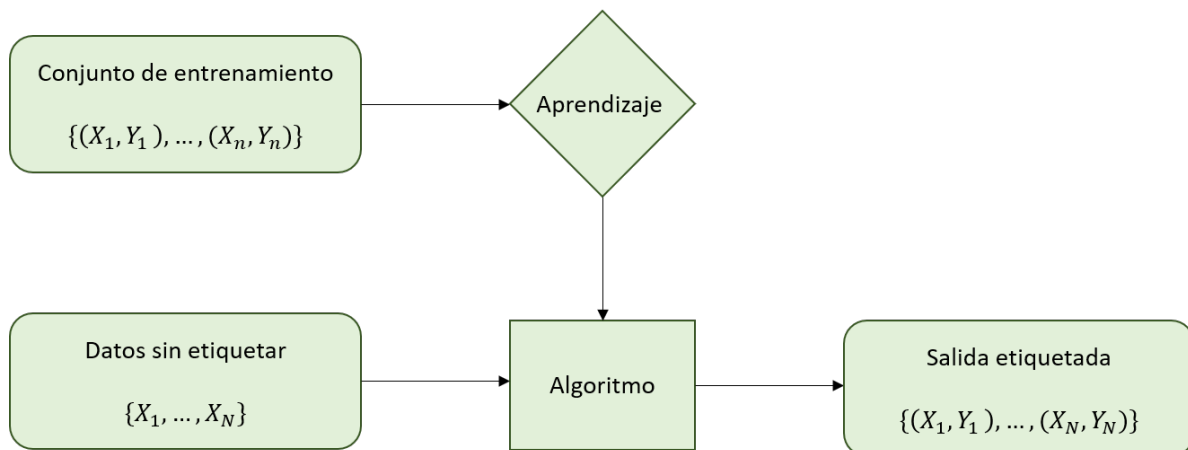


Figura 1.2: Diagrama de funcionamiento de un algoritmo de aprendizaje supervisado.

En función del tipo de datos de salida, diferenciamos dos tipos de algoritmos de aprendizaje supervisado:

- **Modelos de regresión:** en este tipo de algoritmos, los datos de la variable de salida toman valores numéricos, es decir, son modelos cuantitativos.

Los problemas que se pueden resolver utilizando este tipo de algoritmos son, por ejemplo, estimaciones del tiempo que se tarda en hacer una tarea determinada, predicciones del precio que va a tener un producto, etc.

Los algoritmos más utilizados en problemas de regresión son: regresión lineal (aplicado a variables respuesta continuas), regresión logística (aplicado a variables respuesta discretas) y árboles de decisión.

- **Modelos de Clasificación:** en este caso, los datos de la variable de salida son etiquetas, es decir, son modelos cualitativos. El tipo de problemas que se pueden resolver en este caso son de dos tipos.

Si las etiquetas que puede tener la variable de salida son dos, tenemos un problema binario, por ejemplo, para filtrar correos electrónicos utilizaríamos las etiquetas “spam” y “no spam”. En cambio si las etiquetas son más de dos, tenemos un problema multiclase. Un ejemplo de este tipo de problemas sería el reconocimiento de imágenes de distintos animales.

Los algoritmos más utilizados en este tipo de modelos son: clasificación de Naïve-Bayes, bosque aleatorio o máquinas de vectores de soporte.

Cabe destacar que hay algunos algoritmos que sirven para resolver tanto problemas de regresión como de clasificación. Es el caso de la regresión logística, los árboles de decisión o las redes neuronales.

1.3.2. Aprendizaje no supervisado

El aprendizaje no supervisado es un conjunto de técnicas de aprendizaje automático con las que los algoritmos aprenden utilizando únicamente un conjunto de datos de entrada. Así como en el aprendizaje supervisado teníamos un conjunto de entrenamiento formado por las tuplas $\{(X_i, Y_i)\}_{i=1, \dots, n}$, en este caso no sabemos cual es la salida esperada, por lo que nuestro único conjunto de datos es el $\{X_i\}_{i=1, \dots, n}$.

Debido a esto, el aprendizaje que realizan los algoritmos de aprendizaje no supervisado viene dado por el análisis de los datos de entrada y las relaciones entre ellos, por lo que no se pueden resolver los problemas de regresión y clasificación que planteábamos en el aprendizaje supervisado.

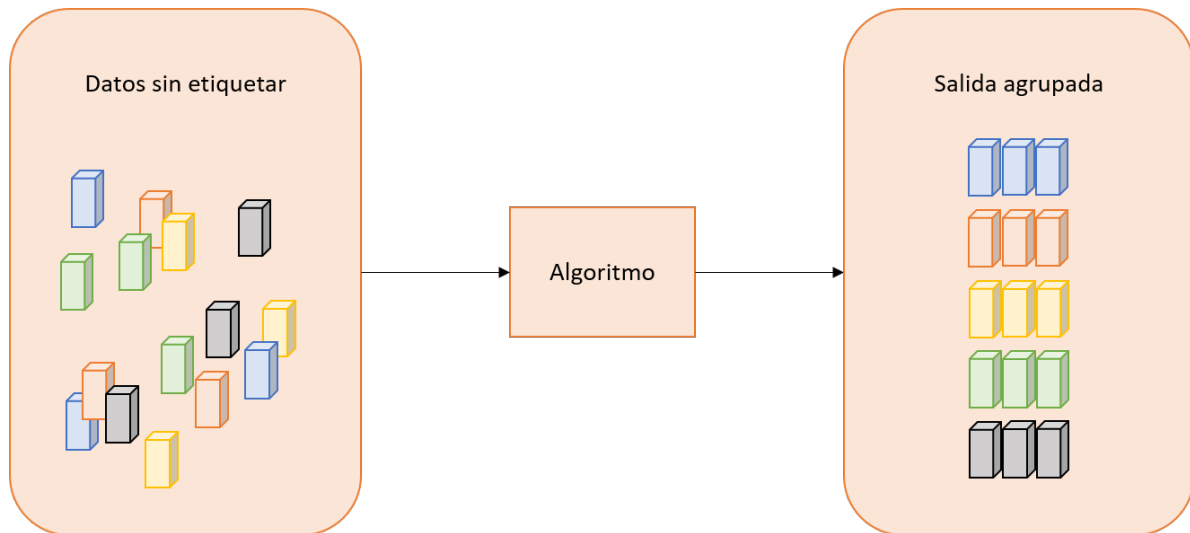


Figura 1.3: Diagrama de funcionamiento de un algoritmo de aprendizaje no supervisado (agrupamiento).

Este tipo de técnicas se pueden utilizar para resolver principalmente tres tipos de problemas:

- Problemas de agrupamiento o *clustering*: consisten en, dado un conjunto de datos de entrada, encontrar patrones o características similares entre ellos para poder agruparlos. Existen distintos tipos de agrupamiento:
 - Exclusivo: cada dato puede pertenecer a un único grupo o *cluster*.
 - Solapamiento: cada dato puede pertenecer a uno o más grupos teniendo en cada uno de ellos un grado de afiliación distinto.
 - Aglomerativo: cada dato constituye un grupo por sí mismo y los grupos se asocian por cercanía formando grupos más grandes.
 - Probabilístico: crea los grupos en base a la distribución de probabilidad.

Los problemas a resolver son, por ejemplo, los de análisis estadístico y algunos de los algoritmos más utilizados son: K-means, Fuzzy C-means, DBSCAN etc.

- Problemas de reducción de dimensiones: este tipo de problemas consiste en reducir el conjunto de datos utilizando la información que proporciona cada uno de ellos. Así, si encontramos que varios datos aportan la misma información o están relacionados se puede expresar el conjunto de una forma más simple para aumentar el rendimiento.

Algunos de los algoritmos que se utilizan para resolver estos problemas son: Análisis de Componentes Principales (PCA), escalado multidimensional o factorización de matriz no negativa.

- Problemas de asociación: dado el conjunto de datos de entrada, en estos problemas lo que se busca es la incidencia que tienen los unos en los otros, es decir, cuánto afecta la presencia de ciertos datos a la de otros.

Un ejemplo de este tipo es, teniendo los patrones de comportamiento de los usuarios de cierto producto, ver cuanto afecta cada uno de ellos a la compra del mismo. Uno de los algoritmos más utilizados para resolverlos es Apriori.

1.3.3. Aprendizaje semi-supervisado

Este tipo de aprendizaje es un término medio entre el aprendizaje supervisado y el no supervisado descritos anteriormente. Supongamos que tenemos un conjunto grande de datos sin etiquetar, es decir, para los que no conocemos la salida esperada $\{Y_i\}_{i=1,\dots,n}$. El coste de etiquetarlos todos de forma manual utilizando un agente humano puede ser muy grande, además, el rendimiento es mucho más bajo que si lo hiciese un ordenador.

Así, estas técnicas consisten en etiquetar manualmente un conjunto pequeño de datos (o bien tenerlos ya etiquetados) que sirva como grupo de entrenamiento para después aplicarle uno o varios algoritmo de aprendizaje supervisado. Después, utilizamos el resultado de estos algoritmos para etiquetar el resto de los datos. Una vez tenemos todos los datos etiquetados, se puede volver a aplicar un algoritmo de aprendizaje supervisado con el conjunto de entrada completo.

Este tipo de algoritmos se puede utilizar para la filtración de correos “spam”, reconocimiento facial o detección de anomalías.

1.3.4. Aprendizaje por refuerzo

El aprendizaje por refuerzo, también conocido como aprendizaje por ensayo y error, es uno de los tipos más utilizados en áreas como la teoría de juegos, la investigación de operaciones o la teoría de control. La diferencia entre este aprendizaje y los que hemos explicado en apartados anteriores es que en este caso se busca maximizar la recompensa en vez de minimizar el error.

Para que este algoritmo aprenda tiene que hacer una interacción con el ambiente que se le plantee. Este entorno viene definido por reglas y el algoritmo va realizando acciones y observando como estas interacciones se relacionan con el ambiente en base a las limitaciones de este. A raíz de cada acción se obtiene una recompensa o un castigo (ensayo/error) y así el algoritmo sabe si lo hace bien o mal.

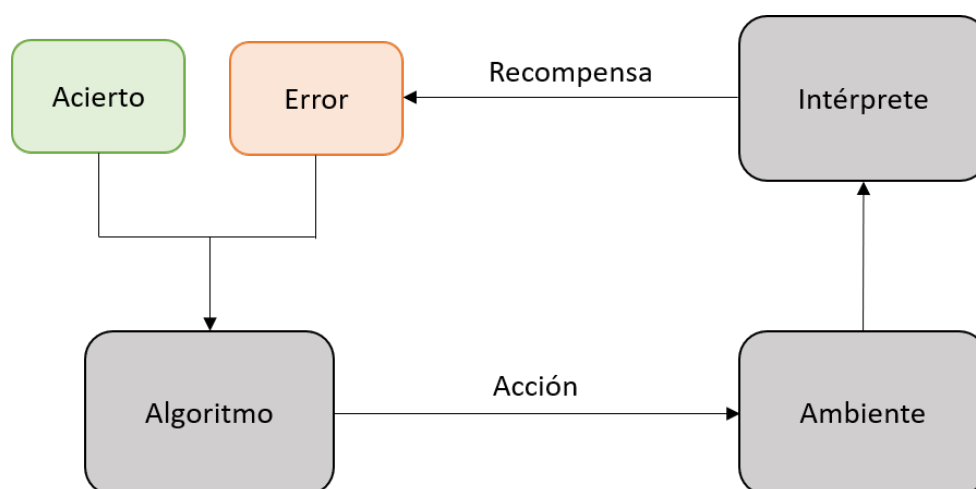


Figura 1.4: Esquema del aprendizaje por refuerzo.