

Introducción a la Inteligencia Artificial y al Aprendizaje Automático

M. Sc. Saúl Calderón Ramírez
Instituto Tecnológico de Costa Rica,
Pattern Recognition and Machine Learning Group (PARMA-Group)
Escuela de Computación.

18 de junio de 2019

El presente documento introduce conceptos básicos de la inteligencia artificial. Basado en el libro *Nilsson, Nils J., et al. Inteligencia artificial: una nueva síntesis. 2001.*

1. ¿Qué es la Inteligencia Artificial?

La inteligencia es un concepto complejo de definir. En términos generales, la podemos circunscribir como la capacidad de un sistema de percibir, razones, aprender, comunicarse y actuar en entornos complejos, a mayor sofisticación de tales procesos, mayor la *inteligencia* de tal sistema. La naturaleza del sistema a estudiar define los objetivos de estudio de la inteligencia:

1. **Sistemas biológicos:** El estudio del fenómeno de la inteligencia en sistemas biológicos como animales y su interacción social se estudia con frecuencia en campos como la biología. La máquina cognitiva biológica con mayor complejidad e inteligencia hasta ahora conocida, el cerebro humano, se estudia desde distintas perspectivas: la filosofía, la psicología, la psiquiatría, la sociología, la antropología y las ciencias cognitivas, las cuales integran distintas perspectivas con el objetivo de entender mejor los procesos cognitivos en el ser humano. Estudiar el comportamiento y los procesos de sistemas biológicos simples, como un virus, nos arroja pistas sobre los procesos de percepción, razonamiento y reacción, contribuyendo en la construcción de ese concepto que denominamos inteligencia.
2. **Sistemas artificiales:** La inteligencia artificial tiene por objetivo implementar máquinas *pensantes* o *inteligentes*. Esta sentencia en sí misma puede rodearse de mucha controversia, pues es necesario primero definir ¿Qué es pensar?, por lo que su relación con las disciplinas mencionadas en el apartado anterior es inevitable. Entender los procesos cognitivos en sistemas que denominamos *inteligentes* facilita en sí mismo la definición de inteligencia, pero además, inspira el diseño e implementación de máquinas inteligentes, o *sistemas bioinspirados*. La inteligencia artificial como campo de estudio comprende entonces elementos teóricos, relacionados con la definición de inteligencia y la medición *de que tan inteligente es un sistema*, para la cual de paso, uno de los precursores de las ciencias computacionales, Alan Turing, propuso la conocida prueba de Turing, ilustrada en la Figura 1.

La inteligencia artificial en su aspecto ingenieril, trata sobre el diseño, implementación y prueba de sistemas *inteligentes*, capaces entonces de percibir, razones, aprender, comunicarse y actuar en entornos complejos. La separación en la perspectiva ingenieril de la inteligencia artificial respecto, no debe aún así desligarse del problema teórico e incluso filosófico de la inteligencia artificial, pues el diseño e implementación de sistemas inteligentes, como cualquier actividad humana, tiene atrás un asidero paradigmático o conceptual. Las siguientes definiciones de qué es un sistema inteligente, inspiran el desarrollo de sistemas artificialmente inteligentes:

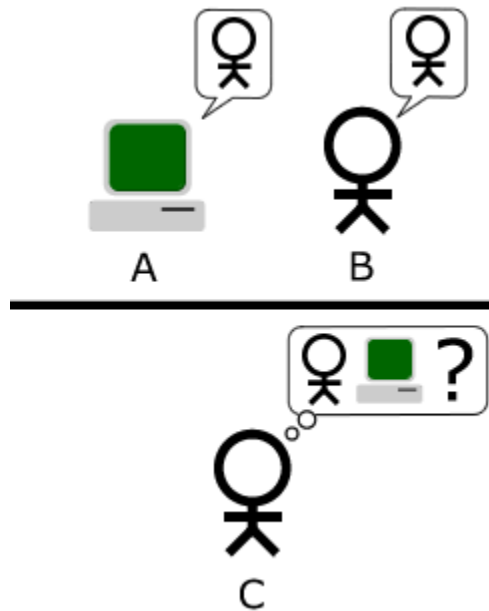


Figura 1: Prueba de Turing consiste en preguntar a una serie de sujetos, si las respuestas a una serie de preguntas fueron producidas por una máquina o una persona: si la mayoría de sujetos no logra distinguir correctamente quién respondió la pregunta, el sistema puede definirse como *inteligente*.

1.1. Sistemas *inteligentes* de procesamiento simbólico

Los filósofos Newell y Simon REF establecen que un sistema físico cualquiera es inteligente, siempre que pueda representar internamente y leer símbolos, tales como números, valores booleanos, etc., sin importar la implementación física de sus componentes (relés mecánicos, transistores, proteínas). La inteligencia artificial clásica se basa en las operaciones lógicas o inferencias sobre el conocimiento *declarativo*, el cual consiste en una serie de sentencias o predicados, los cuales componen el nivel del **conocimiento**. Tales sentencias se representan en estructuras simbólicas como listas en lenguajes como LISP, o predicados en lenguajes de quinta generación como PROLOG, usando álgebra simbólica como la booleana.

1.2. Sistemas *inteligentes* de procesamiento subsimbólico

Otros pensadores [Wilson, 1991 y Brooks, 1990] definen la inteligencia como la capacidad de procesar señales o información subsimbólica, dada la complejidad del entorno y sus estímulos, reemplazando el concepto de *símbolo* por el de *señal*. Este enfoque propone que actividades cognitivas cotidianas en el ser humano, por ejemplo la comunicación oral o el reconocimiento de rostros, no puede descomponerse en un alfabeto finito de símbolos o *lexicons*, por lo que la unidad básica de información es la señal. Este enfoque cambia los fundamentos simbólicos del paradigma en la subsección anterior por la definición de fundamentos físicos o de transmisión y tratamiento de señales. La implementación de lógicas no binarias o borrosas, y el procesamiento paralelo reciente han popularizado este paradigma. [Brooks, 1990], define el concepto de agente, como un procesador de señales a distintos niveles de abstracción, el cual logra un comportamiento complejo al agente interactuar por un período de tiempo con el entorno, tal comportamiento le denomina *comportamiento emergente*.

Este paradigma ha inspirado la implementación de modelos y sistemas como las redes neuronales, las cuales procesan señales, representadas como series de valores numéricos en el caso de su implementación con computadoras discretas, o los algoritmos evolutivos con operadores de cruce, mutación y reproducción implementados para igualar a una señal de aptitud.

1.3. El modelo de agente

Para ilustrar y entender mejor la diferencia entre ambos enfoques, definiremos el modelo de agente como sigue: Un agente es una entidad o sistema el cuál **percibe** estímulos del entorno, a través de uno o más sensores, para según tales estímulos y de forma opcional un **estado** del agente, **planificar** una o más **acciones** a futuro, las cuales son realizadas por uno o más **actuadores**.

Tómese el caso de un robot sencillo el cual en este caso, tomará una acción tomando en cuenta solamente los estímulos externos, por lo cual no implementa un estado interno, para moverse en un espacio con objetos inmóviles. El objetivo del robot será no chocar contra ningún objeto, y cuando encuentre uno en su vecindario próximo, seguirá el perímetro de tal objeto indefinidamente. El robot está dotado de 8 cámaras a escala de grises de resolución 800×600 como sensores para percibir su ambiente, y 2 ruedas, una la cual controla la tracción del robot, y la otra la dirección del mismo. A continuación se modelará el comportamiento del agente usando los paradigmas de aprendizaje simbólico y subsimbólico.

1.3.1. Ejemplo de agente con procesamiento simbólico

Para el caso del agente con procesamiento simbólico, modelaremos la percepción utilizando el álgebra booleana o discreta. El módulo de **percepción** del agente se detalla a continuación. El espacio se modela como una cuadrícula de celdas, para las cuales el robot dispondrá de ocho entradas sensoriales booleanas a las 8 celdas del vecindario inmediato al agente $s_1(t), s_2(t), \dots, s_8(t)$, las cuales tienen un valor de $s_i(t) = 1$ si la celda i está ocupada por un obstáculo y un valor $s_i(t) = 0$ de lo contrario, en el tiempo t . Cada entrada sensorial está conectada a una cámara, la cual toma su valor binario a partir de la intensidad acumulada en la matriz de 800×600 píxeles, usando un umbral τ para ello. Para simplificar más el problema de navegación en tal entorno discreto, se decide evitar que el agente pase por pasillos estrechos, con menos de una celda de ancho, por lo que entonces el módulo de percepción incorpora lo que denominaremos una simple etapa de **extracción de características**, la cual consiste en este caso en un arreglo de 4 elementos $x_1(t), x_2(t), \dots, x_4(t)$, definidos como se ilustra en la Figura 2. Por ejemplo $x_1(t) = s_2(t) \wedge s_3(t)$. Los valores booleanos de las características **representan el conocimiento** del agente acerca de su entorno para un tiempo t .

Dado que el agente no implementa un estado interno, y además no necesita de una planificación compleja para lograr su objetivo, el cuál en este caso es *permanente*, es decir, el agente no llega a un objetivo el cual le permita detenerse luego de lograrlo, definiremos el módulo de **acción** integrando la simple planificación a implementar por el agente. El módulo de acción se implementará usando un **sistema basado en reglas** o también conocido como **sistema de producción**. Una regla se define como una condición $c_i(t)$ dada por una función booleana, la cual si se cumple, lleva a la ejecución de una acción $a_i(t)$ y se escribe como sigue:

$$c_i(t) \rightarrow a_i(t)$$

La acción en el tiempo t modificará probablemente los valores de las características $x_1(t+1), x_2(t+1), \dots, x_4(t+1)$ en el tiempo $t+1$. Usualmente se implementa una regla *por defecto*, la cual se realiza si ninguna de las reglas anteriores se cumple. Para el caso del agente en desarrollo, se definen las siguientes reglas:

$$\begin{aligned} \overline{x_4(t)} \wedge \overline{x_3(t)} \wedge \overline{x_2(t)} \wedge \overline{x_1(t)} &\rightarrow n(t) \\ x_4 \wedge \overline{x_1(t)} &\rightarrow n(t) \\ x_3(t) \wedge \overline{x_4(t)} &\rightarrow o(t) \\ x_2(t) \wedge \overline{x_3(t)} &\rightarrow s(t) \\ x_1(t) \wedge \overline{x_2(t)} &\rightarrow e(t) \\ 1 &\rightarrow nil \end{aligned}$$

Con n, o, s, e correspondientes a acciones de movimiento al norte, oeste, sur y este respectivamente. El caso de existir un objetivo que amerite que el agente se detenga, se implementa en la primera regla. Si ninguna de las reglas se cumple, se ejecuta la última regla, que hace que el robot se detenga.

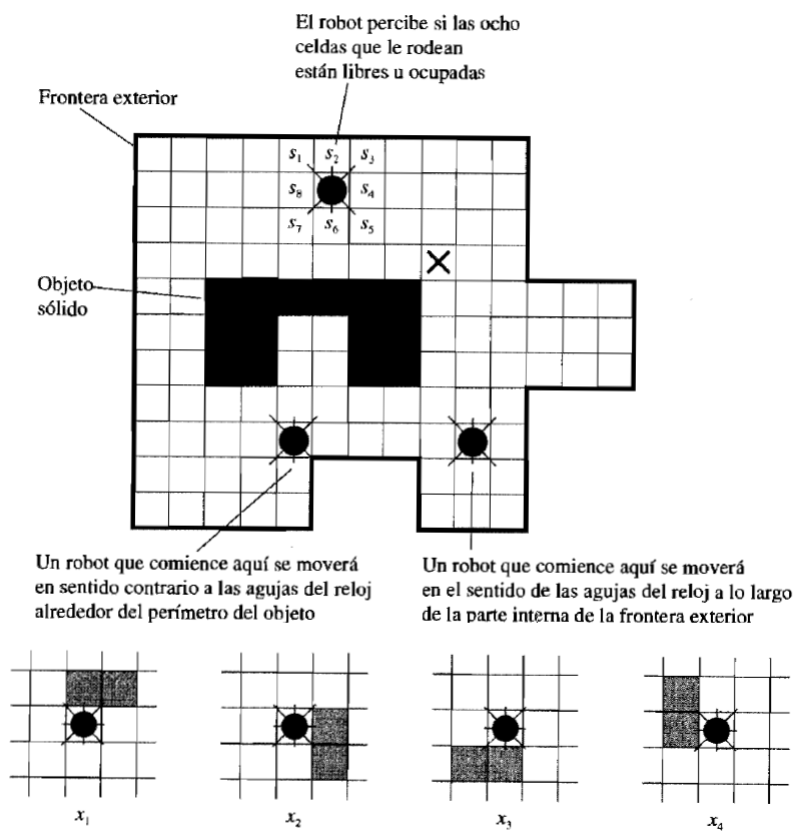


Figura 2: Percepción del agente usando procesamiento simbólico, tomado de Nilsson, Nils J., et al. *Inteligencia artificial: una nueva síntesis*. 2001..

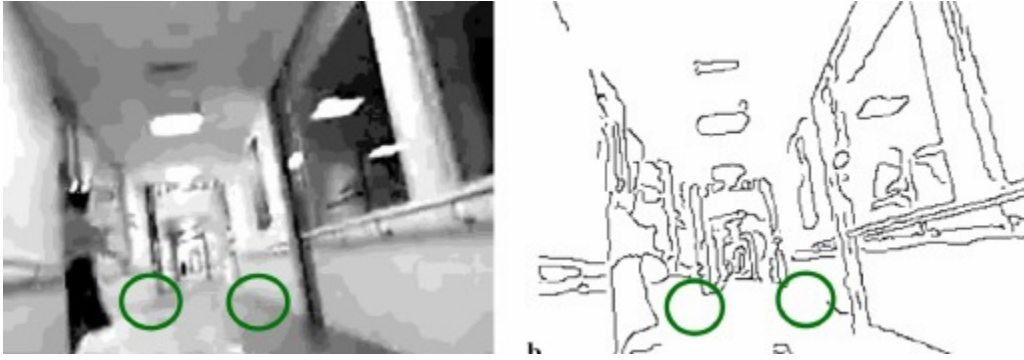


Figura 3: Extracción de características para el sistema de percepción.

1.3.2. Ejemplo de agente con procesamiento subsimbólico

Un agente implementado bajo el paradigma del procesamiento subsimbólico, utiliza señales como unidades fundamentales en vez de símbolos. El módulo de **percepción** se puede implementar bajo este paradigma, como un sistema el cual manipula directamente las entradas de los 8 sensores $s_1(t), s_2(t), \dots, s_8(t)$, con dimensiones de 800×600 pixeles, por lo que entonces $s_i(t) \in \mathbb{Z}^{800 \times 600}$, con $0 \leq s_i(t, x, y) \leq 255$, puesto que cada pixel x, y tiene un valor en la escala de grises. La etapa de extracción de características puede implementarse como un estimador *continuo* de bordes, usando el operador de la convolución $*$ el cual se estudiará más adelante, junto con una función estimadora de bordes, como el operador laplaciano L de forma que: $x_i(t) = s_i(t) * L$, lo cual resulta en algo similar a lo ilustrado en la Figura 3.

Posteriormente, las 8 señales de características $x_1(t), x_2(t), \dots, x_8(t)$ son utilizadas por *unidad con umbral o perceptrón* para tomar la decisión de a qué velocidad **accionar** la rueda 1 r_1 , la cual tiene como máxima velocidad 50 revoluciones por minuto, y la rueda 2 r_2 , la cual utiliza un motor tipo servo, y puede girar de 0 a 180 grados. Para decidir cual valor continuo puede tener tanto la rueda 1 $0 \leq r_1 \leq 50$ y la rueda 2 $0 \leq r_2 \leq 180$.

El perceptrón implementa dos funciones, la función de peso neto y la función de activación. La función de peso neto calcula la combinación lineal de las entradas del perceptrón, definidas en el arreglo $\vec{x} \in \mathbb{R}^n$ con los pesos o parámetros $\vec{w} \in \mathbb{R}^n$ haciendo entonces:

$$p = \sum_i^n x_i w_i.$$

En este caso, las entradas vendrían a ser los valores de todos los pixeles en todos los sensores, por lo que entonces $n = 8 \times 800 \times 600$. Posteriormente el perceptrón implementa una función de activación, la cual cumple la función de *acotar* la salida del perceptrón, por ejemplo, puede estar dada por la función sigmoideal:

$$y(p) = \frac{1}{1 + e^{-p}}$$

aconchado la salida entre 0 y 1, en valores continuos, la salida del perceptrón, por lo que su salida se puede multiplicar por 50 para el caso de la primer rueda, y por 180 por el caso de la segunda. Los pesos en el arreglo \vec{w} son calibrados o definidos a través de un proceso llamado *entrenamiento*, en el cual se les presentan muestras a los perceptrones muestras o tuplas de entradas y salidas deseadas, las cuales el modelo trata de igualar con todas las muestras mostradas.

Observe como en esta simple descripción de dos implementaciones de un sistema con el mismo objetivo, se utilizan operadores y principios teóricos distintos. Como ventajas del paradigma simbólico, la construcción basada en reglas facilita la explicabilidad del agente para una decisión tomada. Sin embargo, el enfoque subsimbólico tiene una mayor capacidad de generalización en entornos complejos, a un costo computacional usualmente mayor y con necesidad de mayor cantidad de muestras para entrenar o verificar el modelo. Ambos contratiempos en los últimos años han sido gradualmente solventados con la irrupción de la internet a alta velocidad y de la compu-

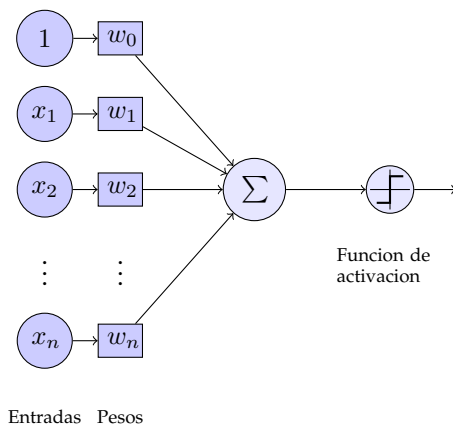


Figura 4: Diagrama conceptual de un perceptrón, abstracción del concepto biológico de neurona.

tación paralela a gran escala, por lo que el paradigma subsimbólico se ha popularizado en ámbitos académicos e industriales.

Sistemas multi-agente: Hasta ahora nos hemos enfocado en el modelado de un agente, sin embargo existen sistemas que enfatizan la interacción entre múltiples agentes y en la construcción de conocimiento colectivo entre agentes de baja o mediana complejidad.

1.4. Ramas de la Inteligencia Artificial

Los paradigmas simbólico y subsimbólico detallados en las secciones anteriores, alimentan las tres principales ramas de la inteligencia artificial:

Búsqueda heurística y optimización: La búsqueda de un estado meta en un árbol de estados para problemas en distintas aplicaciones es un área de estudio frecuente en la inteligencia artificial. Como se estudiará posteriormente, su relación con el problema de la optimización de una función $f(x)$ es muy cercana, por lo que áreas de la matemática aplicada como la optimización discreta y continua, las probabilidades (pues muchos algoritmos implementan enfoques probabilísticos, de máxima verosimilitud), además de las ciencias computacionales como la computabilidad; pues es de interés comparar los costos computacionales de distintos enfoques, fundamentan esta área de estudio. La búsqueda heurística específicamente se refiere a técnicas que implementan **heurísticas** (el griego εὕρισκειν) que significa «hallar, inventar» (etimología que comparte con eureka) aparece en más de una categoría gramatical) o estimaciones de estados más cercanos al estado meta en un problema de búsqueda. Algunos ejemplos de problemas de búsqueda son las soluciones de juegos (4 reinas, rompecabezas, etc.) y navegación en grafos.

Representación del conocimiento: Esta área de la inteligencia artificial estudia distintos métodos, paradigmas y enfoques para representar el conocimiento de un dominio específico en un computador, con las estructuras de datos, procedimientos de interpretación e incluso lenguajes necesarios para facilitar la implementación de una *conducta inteligente* en un programa. Un enfoque específico en la representación del conocimiento construye entonces modelos computacionales con convenciones de descripción de objetos, relaciones y proceso en el mundo. Los paradigmas simbólicos y subsimbólicos han motivado la generación de lenguajes como PROLOG y Octave, respectivamente.

Aprendizaje automático: El aprendizaje automático se refiere al estudio, diseño e implementación de algoritmos que *aprenden* a partir de conjuntos de datos, referidos como *datos de entrenamiento*, construyendo un modelo que permita realizar estimaciones a partir de nuevos casos, como lo muestra la Figura 5.

El aprendizaje automático ha recibido mucha atención en los últimos años, con la irrupción de los computadores con paralelismo de datos a gran escala, la internet y los sensores de bajo costo. Existen distintos enfoques, los principales son:

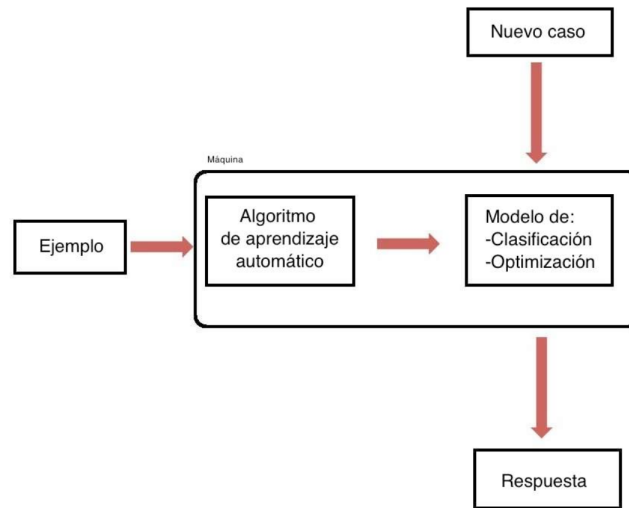


Figura 5: Diagrama de un sistema basado en el aprendizaje automático.



Figura 6: Árbol de decisión de ejemplo, tomado de <http://www.cs.us.es/~fsancho/?e=104>

1. El **enfoque lógico**, inspirado en el paradigma simbólico, con técnicas como los árboles de decisión. La Figura 6 muestra un árbol de decisión de ejemplo, construido a partir de un conjunto de datos.
2. El **enfoque probabilístico**, basado en la estadística bayesiana, ejemplos de técnicas son las cadenas de Markov y los modelos basados en la maximización de la verosimilitud. La Figura 7 muestra una pequeña red Bayesiana para la estimación de enfermedades.
3. El **enfoque geométrico**: Los modelos geométricos tienen por objetivo construir superficies de decisión en espacios de n dimensiones, usualmente utilizando técnicas de optimización para definir los parámetros de tales superficies.

1.5. Tipos de aprendizaje

Los enfoques geométrico, probabilístico y lógico inspiran distintos tipos de aprendizaje:

1. **Aprendizaje supervisado**: El aprendizaje supervisado se basa en la construcción de un modelo $\tilde{T}^{(v)} = y(X^{(v)})$, utilizando un conjunto de muestras de entrenamiento $X^{(v)} = \{\vec{x}_1^{(v)}, \vec{x}_2^{(v)}, \dots, \vec{x}_n^{(v)}\}$. El conjunto de muestras $\tilde{T}^{(v)}$ se asocia a las estimaciones del modelo para las muestras de validación. La construcción de un modelo supervisado requiere de un conjunto de etiquetas manualmente generadas, con intervención

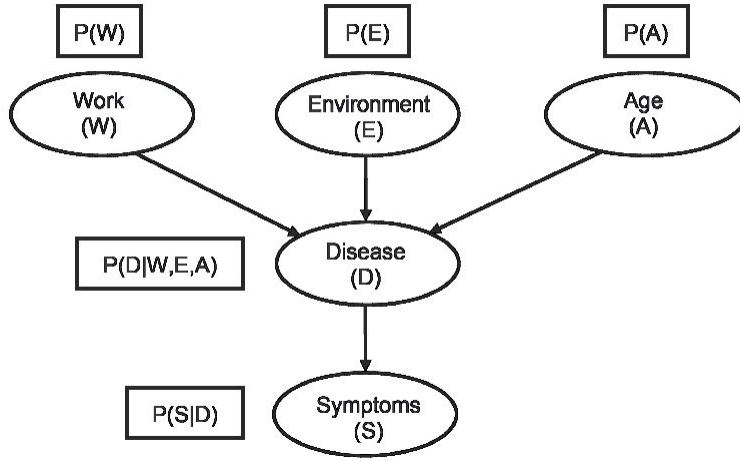


Figura 7: Red Bayesiana de ejemplo para la estimación de enfermedades, tomado de <https://www.intechopen.com/books/current-topics-in-public-health/from-creativity-to-artificial-neural-networks-problem-solving-methodologies-in-hospitals>

humana usualmente $T = \{\vec{t}_1^{(v)}, \vec{t}_2^{(v)}, \dots, \vec{t}_n^{(v)}\}$. A a través de lo que se conoce como un proceso de **entrenamiento**, se estima el modelo y :

$$y = \text{train} \left(X^{(t)}, T^{(t)} \right)$$

donde en este caso se utilizan los conjuntos de muestras de entrenamiento $X^{(t)}$ y $T^{(t)}$.

- a) Si $t_i = y(\vec{x}_i) \in \mathbb{R}$, el modelo es de **regresión**, y si $t_i = y(\vec{x}_i) \in \mathbb{N}$, el modelo es de **clasificación**.
- b) Si el modelo está compuesto por un vector de pesos $\vec{w} \in \mathbb{R}^m$, de forma que $t_i = y(\vec{x}_i, \vec{w})$, el modelo es **paramétrico**.

2. **Aprendizaje conjunto:** Conocido también como *ensemble learning*, entrena un conjunto de r modelos:

$$\begin{aligned} y_1 &= \text{train} \left(X^{(t_1)}, T^{(t_1)} \right) \\ y_2 &= \text{train} \left(X^{(t_2)}, T^{(t_2)} \right) \\ &\vdots \\ y_r &= \text{train} \left(X^{(t_r)}, T^{(t_r)} \right) \end{aligned}$$

Usando comunmente conjuntos de entrenamiento distintos y aleatorios, particiones del conjunto de datos de entrenamiento. Para realizar la estimación para un conjunto de datos de validación $X^{(v)}$ y $T^{(v)}$, en los enfoques más sencillos se promedian o pesan las r estimaciones para el conjunto de datos:

$$\begin{aligned} \tilde{T}^{(v_1)} &= y_1 \left(X^{(v)}, T^{(v)} \right) \\ \tilde{T}^{(v_2)} &= y_2 \left(X^{(v)}, T^{(v)} \right) \\ &\vdots \\ \tilde{T}^{(v_r)} &= y_r \left(X^{(v)}, T^{(v)} \right) \end{aligned}$$

3. **Aprendizaje por transferencia:** Un modelo supervisado y paramétrico:

$$y_1 = \text{train} \left(X^{(t)}, T^{(t)} \right)$$

entrenado con datos $X^{(t)}, T^{(t)}$ puede ser empleado para inicializar los pesos o parámetros $\vec{w} \in \mathbb{R}^m$, el cual tiene por objetivo entrenarse con un conjunto de datos $X^{(d)}, T^{(d)}$, de forma que:

$$y_2 = \text{train} \left(X^{(d)}, T^{(d)}, \vec{w} \right)$$

4. **Aprendizaje no supervisado:** El aprendizaje no supervisado prescinde de las etiquetas de entrenamiento $T^{(t)}$, por lo que la estimación del modelo se realiza de la siguiente forma:

$$y = \text{train} \left(X^{(t)} \right).$$

Existen distintas variantes de aprendizaje no supervisado como el aprendizaje por agrupamiento (clustering), jerárquico, por máxima verosimilitud, etc.

5. **Aprendizaje activo:** El aprendizaje activo toma un modelo previamente entrenado (supervisado o no) $\tilde{T}^{(a)} = y_1(X^{(a)})$, y posibilita al corrección manual o semi-manual de las etiquetas estimadas $\tilde{T}^{(a)}$ con intervención humana, lo cual se modela en la siguiente función:

$$\tilde{T}^{(c)} = \text{correct} \left(X^{(a)}, \tilde{T}^{(a)} \right).$$

Las etiquetas corregidas $\tilde{T}^{(c)}$ se utilizan para reentrenar el modelo:

$$y_2 = \text{train} \left(X^{(a)}, T^{(c)} \right),$$

con lo cual este proceso se puede realizar iterativamente.

6. **Aprendizaje semi-supervisado:** Similar al aprendizaje activo, el aprendizaje semi-supervisado se basa en utilizar un conjunto de datos etiquetados $X^{(t)}$ y sus etiquetas $T^{(t)}$, y un conjunto de muestras no etiquetadas $X^{(s)}$, para iterativamente refinar el modelo. Existen distintos enfoques, el más sencillo es el auto-entrenado supervisado, el cual usa los datos etiquetados para entrenar el modelo:

$$y_1 = \text{train} \left(X^{(t)}, T^{(t)} \right)$$

para luego realizar las estimaciones correspondientes $\tilde{T}^{(s)} = y_1(X^{(s)})$, para las muestras no etiquetadas $X^{(s)}$. Las muestras no etiquetadas clasificadas con la mayor **confianza** la cual se mide con una función $c(\vec{x}_i)$, se toman para reentrenar el modelo:

$$y_2 = \text{train} \left(X^{(s)}, \tilde{T}^{(s_c)} \right).$$

Donde las etiquetas $\tilde{T}^{(s_c)}$ entonces corresponden a las etiquetas de las muestras con mayor confianza. Este proceso se puede realizar iterativamente.

7. **Aprendizaje reforzado:** En el aprendizaje reforzado, para un conjunto de muestras $X^{(t)}$, no se cuenta con el conjunto de etiquetas correspondiente. A través de lo que se conoce como una serie de *experiencias*, en términos muy simples (y con distintas variantes) la función de recompensa estima un conjunto de etiquetas:

$$\tilde{T}^{(t)} = \text{reward} \left(X^{(t)} \right)$$

para posteriormente entrenar el modelo

$$y = \text{train} \left(X^{(t)}, \tilde{T}^{(t)} \right).$$

Este proceso se realiza también de forma iterativa.