

# MÁQUINAS QUE APRENDEN





## FELIPE TOBAR

Investigador en el Centro de Modelamiento Matemático, Universidad de Chile, y anteriormente investigador postdoctoral en el Grupo de Aprendizaje de Máquinas, Universidad de Cambridge. Doctor en Procesamiento de Señales, Imperial College London (2014; PhD) e Ingeniero Eléctrico, Universidad de Chile (BSc y MSc; 2010). Sus intereses están en la interfaz entre procesamiento de señales y aprendizaje de máquinas e incluyen modelos bayesianos no paramétricos, algoritmos basados en kernels, inferencia aproximada y aprendizaje adaptativo.  
ftobar@dim.uchile.cl

El aprendizaje de máquinas (AM) es una disciplina que se enfoca en la construcción de algoritmos que aprenden de la misma forma en que los animales aprendemos: a través de la experiencia. En las últimas décadas, la evolución del AM ha provocado un cambio de paradigma en la forma en que hacemos ciencia, gracias a desarrollos teóricos, avances computacionales y disponibilidad de datos masivos; esto ha posicionado al AM como una exitosa herramienta en variadas aplicaciones. En este sentido, el estudio del AM es fundamental no solo por su protagonismo en el estado actual del desarrollo tecnológico, sino que también por su innegable naturaleza interdisciplinaria y sus controversiales motivaciones filosóficas. En este artículo, revisaremos la historia y estado actual del AM, presentando las distintas aplicaciones que ha tenido en diferentes áreas del conocimiento, para luego discutir los desafíos que este campo de estudio enfrenta hoy.

## INTRODUCCIÓN

El AM reúne elementos de optimización, estadística, probabilidades y ciencias cognitivas para construir el motor de aprendizaje dentro de la inteligencia artificial (IA). Definido por Arthur Samuel en 1950, el AM es la disciplina que da a las máquinas la habilidad de aprender sin ser explícitamente programadas. Si bien existen enfoques al AM inspirados en sistemas biológicos, es necesario clarificar que ésta no es la única forma de construir algoritmos o métodos de aprendizaje. Una analogía se puede identificar en los primeros intentos por construir máquinas

voladoras, en donde inicialmente se intentó, infructuosamente, replicar el movimiento de las alas de un ave. Sin embargo, no fue sino hasta la invención del primer avión, el cual no mueve sus alas, que se logró construir la primera máquina voladora.

El AM es una disciplina joven que ha experimentado un vertiginoso crecimiento en las últimas décadas. Esto ha sido posible, en gran parte, gracias a los recientes avances computacionales y a la existencia de bases de datos de un volumen suficiente para el entrenamiento de algoritmos complejos. El uso masivo de técnicas del AM se ha visto reflejado en distintas áreas, como por ejemplo visión computacional, clasificación de secuencias de ADN, marketing, detección de fraude, diagnósticos médicos, análisis financiero y traducción de texto, entre otras. Adicionalmente, si bien el objetivo del AM es desarrollar algoritmos de aprendizaje prescindiendo en gran medida de la intervención humana, otra razón del éxito del AM es su facilidad para acoplarse con otras disciplinas, lo cual permite incorporar conocimiento de aplicaciones específicas, dando cabida a lo que conocemos últimamente como *data science*. Es entonces necesario entender el AM, su motivación y génesis, sus áreas de aplicación y sus desafíos actuales.

La siguiente sección de este artículo abordará los orígenes del AM en relación a la IA. Luego, revisaremos una breve historia del AM, donde veremos cómo ésta se convirtió en una disciplina en sí misma, con una activa comunidad e interacciones con otras áreas del conocimiento. Posteriormente, mencionaremos los tipos de



AM en distintos escenarios para luego clarificar su relación con otras disciplinas como estadística, computación clásica y *data science*, entre otras. Finalmente, presentaremos algunos de los desafíos actuales del área.

## ORÍGENES: INTELIGENCIA ARTIFICIAL

Tradicionalmente, los seres humanos nos hemos comprendido a nosotros mismos en oposición al resto de los animales. En este ejercicio, nuestras habilidades cognitivas han sido el principal foco de atención, destacando particularmente nuestra capacidad de adaptarnos a diferentes situaciones ambientales de forma rápida y sin la necesidad de un cambio evolutivo. Por ejemplo, para migrar desde África al norte de Europa y a Oceanía el *Homo sapiens* no se adaptó biológicamente a climas distintos, sino que manipuló herramientas y materiales para producir embarcaciones y vestimenta adecuada (Harari, 2015). Como sociedad, siempre hemos estado interesados en entender la inteligencia. En efecto, desde sus inicios la filosofía, psicología y ciencias cognitivas se han dedicado al estudio de la forma en que entendemos, recordamos, razonamos y aprendemos. La IA es una disciplina mucho más reciente que las anteriores y su objetivo no es solo la comprensión teórica de la inteligencia como habilidad humana; por el contrario, su principal objetivo es replicar la inteligencia y, por ende, su fin último es la construcción de seres inteligentes no-humanos (Russell & Norvig, 2003).

En estricto rigor, la definición más ampliamente aceptada de la IA tiene que ver con la capacidad de una máquina de realizar acciones inteligentes u objetivamente racionales. Ésta es la posición de Alan Turing, quien propone que una máquina es inteligente si es capaz de desarrollar tareas cognitivas a un nivel humano suficiente para engañar a un interrogador (también humano)

(Turing, 1950). En este contexto, para que una máquina sea inteligente, o equivalentemente, apruebe el test de Turing, es necesario que posea (Russell & Norvig, 2003):

- *procesamiento de lenguaje natural para comunicarse con seres humanos, en particular, el interrogador,*
- *representación del conocimiento para guardar información recibida antes y durante la interrogación,*
- *razonamiento automático para usar la información almacenada y formular respuestas y conclusiones, y*
- *aprendizaje de máquinas para adaptarse a nuevas circunstancias y descubrir patrones.*

El test de Turing sigue siendo un tópico de investigación en filosofía hasta el día de hoy, sin embargo, los avances actuales de la IA no están necesariamente enfocados en diseñar máquinas para aprobar dicho test. Actualmente, la IA apunta hacia el desarrollo de metodologías para aprender de grandes volúmenes de datos sin necesariamente actuar de forma humana (recordemos el ejemplo de las aves y las máquinas voladoras en la introducción), en particular, el componente de aprendizaje de máquinas en la lista anterior ha jugado un papel fundamental en esta nueva etapa, en donde su aporte en distintas áreas de aplicación es cada vez más evidente.

## BREVE HISTORIA DEL APRENDIZAJE DE MÁQUINAS

El conexionismo es un movimiento que pretende explicar las habilidades intelectuales mediante redes neuronales artificiales, es decir, modelos compuestos por capas de unidades básicas de procesamiento (i.e., neuronas) conectadas entre

sí. Con base en las definiciones de la inteligencia de máquinas asentadas por Turing, las primeras redes neuronales artificiales comenzaron a emerger en los trabajos seminales de (Minsky, 1952), quien programó el primer simulador de una red neuronal, (Farley & Clark, 1954), que implementaron un algoritmo de prueba y error para el aprendizaje, y (Rosenblatt, 1958), quien propuso el Perceptrón, un clasificador lineal que ajusta sus parámetros en línea. En los años siguientes, la investigación en redes neuronales (RN) se vio afectada por las limitaciones de dichas estructuras, expuestas en (Minsky & Papert, 1969), dando origen así al denominado primer invierno de la IA. Es en este período cuando el Profesor Sir James Lighthill expuso frente al parlamento inglés que la IA se fijaba objetivos no realistas y solo servía para escenarios básicos (Lighthill, 1973). Esta desconfianza en los alcances de la IA ralentizó su desarrollo, no obstante, los conexionistas seguirían investigando sobre formas de diseñar y entrenar redes neuronales. Finalmente, la propuesta de (Werbos, 1974) -que culminaría en el algoritmo de *backpropagation* propuesto por (Rumelhart, Hinton, & Williams, 1986)- y los avances de (Hopfield, 1982) en RN recurrentes permitirían terminar con este primer receso de la teoría conexionista.

Durante la pausa del conexionismo, tuvo lugar el desarrollo de los sistemas basados en reglas, en particular, de los sistemas compuestos por reglas condicionales de la forma “si-entonces” (*if-then*)<sup>1</sup>. Estos sistemas pretenden replicar el comportamiento de un humano experto; consecuentemente, son conocidos como “sistemas experto”. Estos métodos se convirtieron en la primera herramienta exitosa de la IA en aplicaciones reales, sin embargo, los sistemas experto no aprenden por sí solos, es decir, las reglas si-entonces deben ser explícitamente programadas por un humano. En problemas complejos, el considerable costo asociado al diseño humano de reglas hizo colapsar a este tipo de sistemas, pues la cantidad de información disponible aumentaba y dicho enfoque carecía de escalabilidad (i.e., la habilidad de un algoritmo para continuar funcionando bien cuando el volumen de los

1. Por ejemplo, una regla “si-entonces” simple para un termostato podría ser: “si” la temperatura está por debajo de la deseada, “entonces” el calefactor debe continuar encendido.

datos aumenta). Un sistema basado en reglas que aún se utiliza son los llamados árboles de decisión (Breiman, Friedman, Olshen, & Stone, 1984), los cuales difieren de los sistemas experto en que las reglas no son definidas por un humano sino que descubiertas sobre la base de la elección de variables que mejor segmentan los datos de forma supervisada (ver Sección Taxonomía del aprendizaje de máquinas).

Oportunamente, antes del colapso de los sistemas experto, todavía en la década de los ochenta, las RN vieron un resurgimiento con el método de *backpropagation* (Rumelhart, Hinton, & Williams, 1986). Este permitía entrenar RN de más de dos capas usando gradiente descendente y la regla de la cadena (i.e., ajustando los parámetros de la red mediante la minimización de una función de desempeño en base a derivadas calculadas capa a capa). Esto, a su vez, permitió finalmente validar la teoría conexionista en tareas complejas, específicamente en reconocimiento de caracteres usando redes convolucionales como el Neocognitron (Fukushima, 1980) y LeNet-5 (LeCun, Boser, Denker, Henderson, Howard & Jackel, 1989), y reconocimiento de voz usando RN recurrentes (Hopfield, 1982). Tras este breve resurgimiento del paradigma conexionista, hacia fines de los ochenta tuvo lugar una segunda caída del conexionismo, debido a la inexistencia de una clara teoría que explicara el desempeño de las redes neuronales, su elevado costo computacional y su tendencia al sobreajuste<sup>2</sup>.

Posteriormente, a principios de los noventa, y basado en la teoría del aprendizaje estadístico (Vapnik & Chervonenkis, 1971), surgieron los métodos basados en kernels, resaltando específicamente las máquinas de soporte vectorial (MSV) (Boser, Guyon, & Vapnik, 1992). Esta nueva clase de algoritmos estaba fundamentada en una base teórica que combinaba elementos de estadística y análisis funcional, para caracterizar los conceptos de sobreajuste, optimalidad de soluciones y funciones de costo en el contexto del aprendizaje de máquinas. Además de sus fundamentos teóricos, las MSV mostraron ser una alternativa competitiva a las RN en cuanto a su desempeño y costo computacional en distintas aplicaciones.

También en la década de los noventa, surgieron nuevos enfoques de aprendizaje de máquinas en donde el manejo de incertidumbre era abordado usando teoría de probabilidades, siendo éste el punto de mayor similitud entre el AM y la estadística (Ghahramani, 2015). Este enfoque procede definiendo una clase de modelos generativos probabilísticos, en otras palabras, se asume que los datos observados son generados por un modelo incierto y aleatorio. Luego, el aprendizaje consiste en convertir la “probabilidad de los datos dado el modelo” en la “probabilidad del modelo dado los datos”. Para ello, se utiliza el teorema de Bayes: una fórmula que relaciona, justamente, la probabilidad del “efecto de una causa” con la probabilidad de dicha causa, sobre la base de la observación del efecto. El enfoque bayesiano permite definir todos los elementos del problema de aprendizaje (i.e., modelos, parámetros y predicciones) mediante distribuciones de probabilidad. Esto, con la finalidad de caracterizar la incertidumbre en el modelo y definir intervalos de confianza en las predicciones, permitiendo incluso hacer inferencia sobre modelos con infinitos parámetros (Hjort, Holmes, Müller, & Walker, 2010). En estos casos, el espacio de parámetros pueden ser todas las posibles soluciones de un problema de aprendizaje, por ejemplo, el conjunto de las funciones continuas en regresión (Rasmussen & Williams, 2006) o el conjunto de las distribuciones de probabilidad en el caso de estimación de densidades (Ferguson, 1973). Éste es un enfoque elegante y formalmente sustentado en la teoría de probabilidades el cual permitió reinterpretar métodos anteriores, a pesar de ello, muchas veces esta formulación resulta en que las estimaciones del modelo y las predicciones no tienen forma explícita. Por ello, para el éxito del AM bayesiano fue necesario recurrir a técnicas de inferencia aproximada basadas en Monte Carlo (Neal, 1993) o métodos variacionales (Jordan, Ghahramani, Jaakkola, & Saul, 1999).

Un nuevo resurgimiento de las RN se vio en los primeros años de la década del 2000, donde el área se renombró como *deep learning* (Bengio, 2009), nombre inspirado por las nuevas intuiciones que guían las arquitecturas de las RN. Pro-

gresivamente, el foco de la comunidad migró desde temáticas probabilistas o basadas en kernels para volver a RN, pero ahora con un mayor número de capas. Es así como el enfoque conexionista finalmente logra las metas propuestas más de dos décadas atrás. Este éxito se debe principalmente a dos factores: la gran cantidad de datos disponibles, e.g., la base de datos ImageNet, y la gran capacidad computacional y paralelización del entrenamiento mediante el uso de tarjetas gráficas. Estas condiciones permitieron implementar RN con billones de parámetros, las cuales eran complementadas con técnicas para evitar el sobreajuste.

El hecho de que (i) hasta el momento no hay otro enfoque que supere a *deep learning* en variadas aplicaciones, lo cual ha sido confirmado por los avances de Google, DeepMind y Facebook, y que (ii) la gran cantidad de parámetros de las RN “profundas” no tenga claro significado para entender relaciones entre datos, hace que el AM se aisle de la estadística en pos de un objetivo es netamente predictivo y no inferencial. De acuerdo a Max Welling, si bien la irrupción de *deep learning* distancia al AM de la estadística, aún hay temáticas que se nutren de ambas áreas, como programación probabilista y computación bayesiana aproximada (Welling, 2015). Adicionalmente, Yoshua Bengio sostiene que aún hay muchos aspectos inciertos de *deep learning* en los cuales los estadísticos podrían contribuir, tal como los especialistas de las ciencias de la computación se han dedicado a los aspectos estadísticos del aprendizaje de máquinas en el pasado (Bengio, 2016).

---

## TAXONOMÍA DEL APRENDIZAJE DE MÁQUINAS

---

En la sección anterior se mencionaron distintos métodos de AM que son transversales a variados tipos de aprendizaje, en esta sección veremos los tres principales tipos de aprendizaje de máquinas que se implementan hoy en día.

2. El sobreajuste es un efecto que suele ocurrir al entrenar modelos complejos (con muchos parámetros) y tiene relación con el aprendizaje de características específicas de los datos de entrenamiento que no son representativos de la estructura principal de los datos. Un algoritmo sobreajustado aprende relaciones únicas del conjunto de entrenamiento y es incapaz de generalizar el aprendizaje a nuevos datos. →



En primer lugar, está el aprendizaje supervisado, en el que los datos consisten en pares de la forma (dato, etiqueta) y el objetivo es estimar una función  $f(\cdot)$  tal que  $\text{etiqueta} = f(\text{dato})$ , de acuerdo a cierta medida de rendimiento. El término supervisado viene del hecho de que los datos de entrenamiento cuentan con etiquetas correctamente asignadas por un “supervisor”, muchas veces humano. Ejemplos de aprendizaje supervisado son la identificación de *spam* en correos electrónicos (clasificación) y la estimación del precio de una propiedad en función de características como su tamaño y ubicación (regresión).

La segunda categoría es el aprendizaje no supervisado, el cual consiste en extraer información de datos que no están etiquetados para encontrar su estructura subyacente. La búsqueda de dicha estructura tiene usualmente fines de visualización, almacenamiento o búsqueda de características (i.e., una forma de preprocesar datos antes de aplicar otros métodos de AM). Dos técnicas principales del aprendizaje no supervisado son (i) el *clustering*, es decir, la agrupación de datos en subconjuntos (*clusters*) cuyos miembros tienen algún grado de relación o propiedades en común; y (ii) los métodos de reducción de dimensionalidad que pretenden expresar los datos usando “menos variables” que las originales. Un ejemplo de aprendizaje no supervisado usando redes neuronales es (Hinton & Salakhutdinov, 2006) donde un gran número de artículos (los datos en este caso son texto) fueron segmentados en *clusters* sobre la base de la frecuencia de aparición de las palabras contenidas en el texto. Los *clusters* encontrados correspondieron precisamente a las categorías de los artículos: economía, accidentes, asuntos gubernamentales, etc.

La tercera categoría del AM es el aprendizaje reforzado, en donde el objetivo es aprender a tomar decisiones mediante la maximización de un funcional de recompensa. Éste es probablemente el tipo de aprendizaje más cercano a la forma en que los animales aprendemos, i.e., mediante prueba y error. Un ejemplo ilustrativo

de este tipo de aprendizaje es el adiestramiento de perros mediante recompensas como comida y afecto. El reciente resultado de DeepMind, donde una máquina aprendió a jugar Go usando una búsqueda de árbol y una RN profunda, es uno de los ejemplos más exitosos de aprendizaje reforzado (Silver, et al., 2016).

## RELACIÓN CON OTRAS DISCIPLINAS

AM y Estadística. En el análisis de datos es posible identificar dos extremos, a saber, el aprendizaje inductivo y el aprendizaje deductivo. El primero tiene por objetivo realizar predicciones con algoritmos complejos que no necesariamente expliquen los datos coherentemente, los supuestos *a priori* sobre la naturaleza de los datos son heurísticos y es indispensable trabajar con una gran cantidad de datos. El aprendizaje deductivo, por otra parte, tiene por objetivo que los agentes aprendan relaciones significativas entre los datos usando modelos simples; en este caso, sí se adoptan supuestos sobre la naturaleza de los datos y no se trabaja con una cantidad masiva de ellos. Podemos entender a la estadística por su cercanía al segundo extremo (i.e., el aprendizaje deductivo), mientras que el AM tiene componentes que son de enfoque deductivo y muy relacionado con estadística (inferencia bayesiana), y componentes de enfoque inductivo (e.g., redes neuronales). Una consecuencia de esto es la importancia que tienen los parámetros en distintos métodos de AM: en los métodos deductivos (estadísticos) los parámetros tienen un rol explicativo de la naturaleza del fenómeno en cuestión que es revelado por los datos, mientras que los métodos inductivos se caracterizan por tener una infinidad de parámetros sin una explicación clara, pues el objetivo muchas veces es directamente hacer predicciones. La relación entre estadística y algunos enfoques al AM es tan cercana que Robert Tibshirani se ha referido

al AM como “estadística pretenciosa” (*glorified statistics*).

AM y Programación Clásica. La programación clásica construye algoritmos basados en reglas, esto es, listas de instrucciones que son ejecutadas en una máquina. Esto requiere que el programador conozca de antemano el algoritmo a programar, e.g., calcular la transformada de Fourier rápida (FFT) de una grabación de audio. Sin embargo, hay determinadas tareas en las cuales el algoritmo apropiado no es conocido. En estos casos, el enfoque que adopta el AM es programar directamente la máquina para que aprenda la lista de instrucciones. Tomemos el caso del ajedrez: de acuerdo a (Shannon, 1950) el número de combinaciones posibles para el juego de ajedrez es del orden de  $10^{43}$ , esto significa que, usando programación clásica, un programa básico para jugar ajedrez tendría que tener al menos esa cantidad de instrucciones de la forma “para la combinación C, ejecutar la acción A”. Si todos los humanos sobre la faz de la tierra ( $7.4 \times 10^9$  estimado a abril 2017) se uniesen para programar esta rutina y cada uno pudiese escribir 10 de estas instrucciones por segundo, nos tomaría  $4 \times 10^{25}$  años, esto es casi mil billones ( $10^{15}$ ) de veces la edad de la tierra ( $4.54 \times 10^9$  años), lo cual hace impracticable adoptar un enfoque clásico de programación para jugar ajedrez. Una alternativa basada en AM es un programa simple en el cual la máquina explora distintos posibles escenarios del tablero e inicialmente toma decisiones aleatorias de qué acción ejecutar para luego registrar si dicha movida llevó a ganar o perder el juego. Este enfoque de programación no pretende programar la máquina para “jugar ajedrez” sino para “aprender a jugar ajedrez”. Una exitosa implementación de este concepto usando aprendizaje reforzado y redes neuronales profundas para el juego de Go puede verse en (Silver, et al., 2016), como comentábamos en el apartado [Taxonomía del aprendizaje de máquinas](#).

AM, *Knowledge Discovery in Databases* (KDD) y Minería de Datos: KDD (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) tiene relación con la

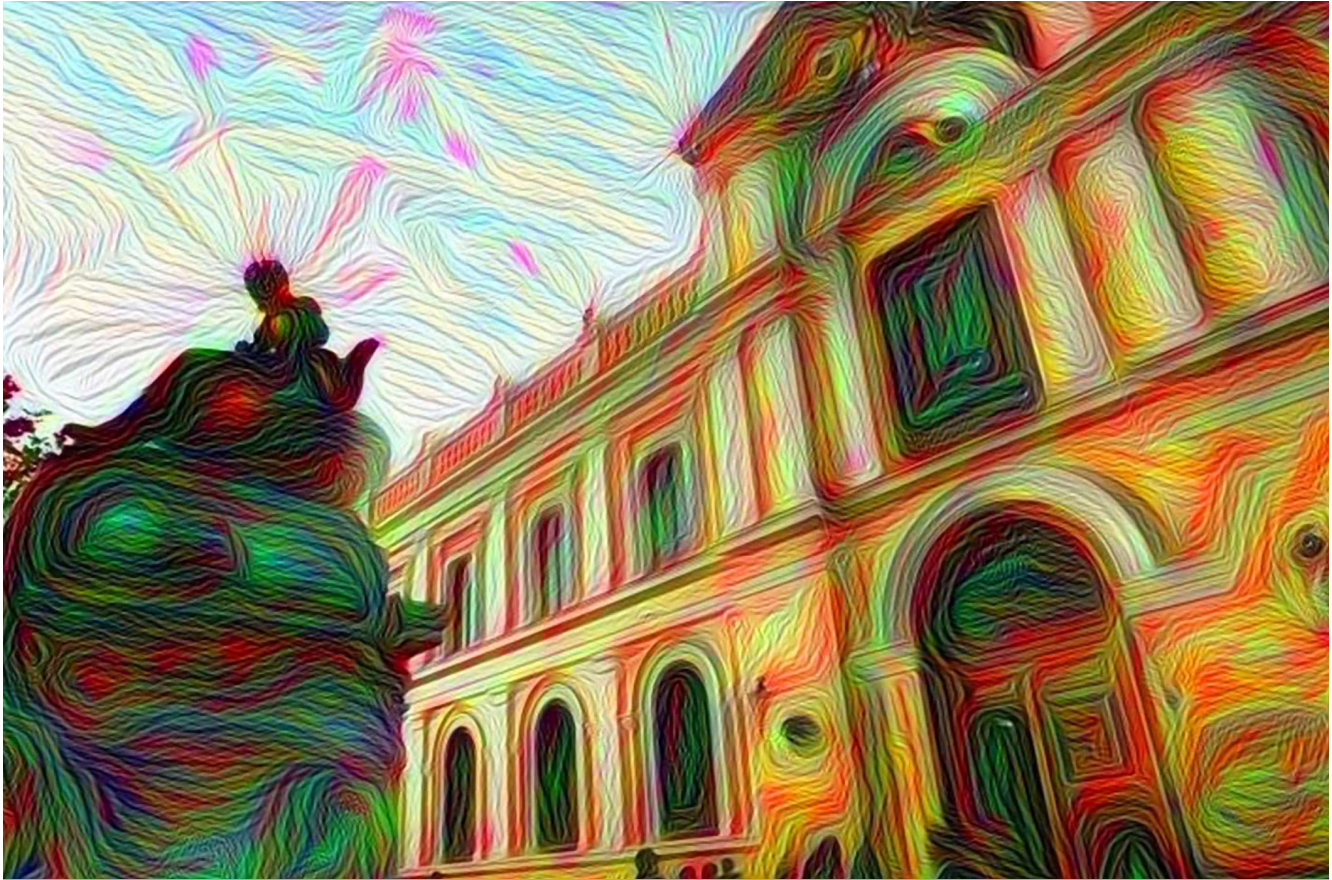


IMAGEN 1.  
FOTOGRAFÍA DEL FRONTIS DE LA CASA CENTRAL DE LA UNIVERSIDAD DE CHILE PROCESADA CON EL ALGORITMO *DEEP DREAM*, EL CUAL REALZA PATRONES USANDO UNA RED NEURONAL CONVOLUCIONAL.

identificación de patrones útiles y explicativos en datos, y consta de cinco etapas: selección, preparación, transformación, minería e interpretación de datos. En particular, la etapa de minería de datos<sup>3</sup> consiste en extraer información desde datos disponibles, por ejemplo, agruparlos en subconjuntos afines o identificar situaciones anómalas. Para esto se usan herramientas de AM, especialmente de aprendizaje no supervisado debido a la cantidad de los datos, los cuales en general no están etiquetados y existe poco conocimiento *a priori* de su naturaleza.

AM y *Data Science*. En línea con sus orígenes en la IA, el objetivo del AM es encontrar relaciones entre datos y hacer predicciones prescindiendo de la intervención humana, es decir, con escaso conocimiento *a priori*. Sin embargo, las

técnicas de AM pueden ser complementadas con el conocimiento de un problema específico, en donde especialistas en AM colaboran con expertos en (i) el área en cuestión, (ii) minería de datos y (iii) visualización. Esto es *data science*, una disciplina colaborativa donde distintas áreas unen fuerzas para hacer un análisis detallado de los datos, con la finalidad de resolver un problema en particular, usualmente con fines comerciales. El perfil del *data scientist* es muy completo, pues debe tener conocimiento de AM, estadística, programación, minería de datos, interactuar con especialistas y entender el impacto comercial del análisis. De hecho, *data scientist* ha sido considerado el puesto de trabajo más sexy del siglo XXI según Harvard Business Review (Davenport & Patil, 2012).

## ESTADO DEL APRENDIZAJE DE MÁQUINAS Y DESAFÍOS ACTUALES

El aprendizaje de máquinas es una reciente disciplina que provee de herramientas a una gran cantidad de campos de estudio, pero también es un área de investigación en sí misma con una activa comunidad y muchas preguntas abiertas. Desde el punto de vista algorítmico, es posible identificar, en primer lugar, el desafío del entrenamiento en línea, esto es, ajustar un algoritmo cada vez que se dispone de un nuevo dato para

3. Nótese que minería de datos se refiere a extraer información desde datos, no extraer datos como su nombre lo sugiere.

operar continuamente (e.g. análisis de series de tiempo). Este es un concepto fundamental en procesamiento adaptativo de señales y no ha sido tomado en cuenta satisfactoriamente aún en AM. Otro desafío que tiene relación con la implementación de algoritmos es la capacidad de escalar el entrenamiento del AM para *Big Data*, pues en general los métodos utilizados en el AM son costosos, ya que se enfocan en aprender estructuras complejas y no en procesar datos de alta dimensión *per se*. Sin embargo, esta habilidad es cada vez más necesaria en la era de la información, donde los conceptos que actualmente se usan en *Big Data* son básicos comparados con el estado del arte del AM. También es posible identificar desafíos en un plano teórico, como por ejemplo la transferencia de aprendizaje, donde la experiencia adquirida en la realización de una tarea (e.g., reconocimiento de

automóviles) sirve como punto de partida para una tarea relacionada (e.g., reconocimiento de camiones), de la misma forma en que un físico o matemático se familiariza rápidamente con el área de finanzas a pesar de no tener necesariamente conocimientos previos de dicha área.

Por otro lado, existe un desafío ético relacionado a los riesgos del uso del AM: el avance de esta disciplina parece a veces descontrolado y abre interrogantes con respecto de dimensiones como la legislación sobre el actuar de máquinas inteligentes. Por ejemplo, preguntas como ¿quién es responsable en un accidente en el que está involucrado un automóvil autónomo?, o bien, ¿debemos dejar que una máquina evalúe a un humano que solicita un crédito hipotecario o que postula a un trabajo?, sugieren que el avance en el AM no sólo debe ser técnico sino

que también ético y social. El desafío ético del AM es un área que no hemos explorado y sugiere que, a pesar de los avances teóricos y sobre todo aplicados del AM, estamos lejos de entender la inteligencia. Como ha sido expuesto en (Gal, 2015), nuestra relación con el desarrollo de la inteligencia mediante el uso del AM puede ser entendido como el *Homo erectus* hace 400.000 años frotando dos ramas para producir fuego: ellos se beneficiaron del fuego para abrigarse, cocinar y cazar, sin embargo, esto no quiere decir que comprendían por qué al frotar dos ramas generaban fuego o, más aún, qué es el fuego. Estamos en una etapa temprana del estudio del aprendizaje, en la que usamos estas herramientas “inteligentes” para nuestro bienestar. No obstante, estamos lejos de comprender realmente la ciencia que hay detrás. ■



## AGRADECIMIENTOS

EL AUTOR ESTÁ PROFUNDAMENTE AGRADECIDO DE ANITA TOBAR (UNIVERSITY OF EDINBURGH), ANGELO FALCHETTI Y MATÍAS MATTAMALA (AMBOS DE LA UNIVERSIDAD DE CHILE) POR ESTIMULANTES Y ENTRETENIDAS DISCUSIONES SOBRE LOS TEMAS TRATADOS AQUÍ, Y TAMBIÉN POR COMENTARIOS SOBRE VERSIONES PRELIMINARES DE ESTE ARTÍCULO. ESTE TRABAJO FUE REALIZADO BAJO FINANCIAMIENTO DE LOS PROYECTOS CONICYT PAI-82140061 “*MACHINE LEARNING MEETS SIGNAL PROCESSING*” Y BASAL-CMM.

## REFERENCIAS

- Bengio, Y. (19 de enero de 2016). What's Yoshua Bengio's opinion on Max Welling's position paper "Are ML and Statistics Complementary"? Visitado el 11 de diciembre 2016 en Quora: <https://www.quora.com/Whats-Yoshua-Bengios-opinion-on-Max-Wellings-position-paper-Are-ML-and-Statistics-Complementary>
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends® in Machine Learning*, 2 (1), pp. 1-127.
- Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, (p. 144).
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth & Brooks.
- Davenport, T. H., & Patil, D. (octubre de 2012). Data scientist: The sexiest job of the 21st century. Visitado en noviembre de 2016 en *Harvard Business Review*: <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193–202.
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17 (3), 37-54.
- Farley, B., & Clark, W. (1954). Simulation of self-organizing systems by digital computer. *IRE Transactions on Information Theory*, 4 (4), 76-84.
- Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1 (2), 209– 230.
- Gal, Y. (17 de diciembre de 2015). The Science of Deep Learning. Visitado el 12 de diciembre del 2016 en Yarin Gals' website: [http://mlg.eng.cam.ac.uk/yarin/blog\\_5058.html](http://mlg.eng.cam.ac.uk/yarin/blog_5058.html)
- Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521, 452–459.
- Harari, Y. N. (2015). *Sapiens: A Brief History of Humankind*. Harper.
- Hjort, N., Holmes, C., Müller, P., & Walker, S. (2010). *Bayesian Nonparametrics*. Cambridge University Press.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *In Proceedings of the National Academy of Sciences of the USA*, 79, pp. 2554–2558.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T., & Saul, L. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37, 183-233.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., W. H., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1 (4), 541-551.
- Lighthill, J. (1973). Artificial Intelligence: A general survey. *Artificial Intelligence: A Paper Symposium*. Science Research Council.
- Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. University of Toronto, Department of Computer Science.
- Minsky, M. (1952). A neural-analogue calculator based upon a probability model of reinforcement. Harvard University Psychological Laboratories.
- Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT.
- Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313, 504-507.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Driessche, G. v., et al. (2016). Mastering the game of Go with deep neural networks & tree search. *Nature*, 529, 484-488.
- Shannon, C. (1950). Programming a computer for playing chess. *Philosophical Magazine*, 41 (314).
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323 (6088), 533-536.
- Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.
- Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.
- Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 (6), 386–408.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59 (236), 433-460.
- Welling, M. (2015). Are ML and Statistics complementary? *Roundtable discussion at the 6th IMS-ISBA meeting on "Data Science in the next 50 years"*.
- Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University.
- Vapnik, V., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16, 264-280.