Desafíos epistémicos de la técnica

¿Aprenden las máquinas con su *machine learning*?

Andrés A. Ilcic

La vida contemporánea está rodeada de artefactos que recurren a algoritmos de aprendizaje automatizado, ya sea porque activamente los utilizan durante sus operaciones y —si las hay— con las interacciones con sus usuarios o con otros artefactos, ya sea porque han sido diseñados haciendo uso de dichos algoritmos. En el presente trabajo exploro lo que denomino “desafíos epistémicos” que presentan los artefactos técnicos que hacen uso de los métodos de aprendizaje automatizado, que suelen recibir el nombre colectivo de “*machine learning*”. Como se podrá notar, el uso de “artefacto técnico” aquí es muy amplio, ya que no sólo incluye a dispositivos técnicos propiamente dichos sino también a modelos científicos. Desde el comienzo en necesario aclarar que los desafíos que dichos artefactos presentan no son meramente epistémicos. Dentro de los otros desafíos los más notables son los éticos, que ya han sido discutidos en ámbitos filosóficos, especialmente dados los casos recientes de público conocimiento en los que vehículos automatizados se vieron involucrados en accidentes con víctimas fatales. Otro de los desafíos clásicos gira en torno a cuestiones ligadas con la antropología filosófica, en tanto han surgido debates acerca de la posible condición "post-humana" que podría ser hecha posible por las nuevas tecnologías, incluso al punto de sugerir escenarios en los que las máquinas dominan al hombre: la llamada rebelión de las máquinas, un escenario posible siendo el descrito por Bostrom (2014). Sin embargo, desde un punto de vista epistemológico, podemos encontrar una serie de desafíos que sugieren un ámbito fértil para la interacción entre la filosofía de las ciencias y la filosofía de la tecnología.

Los desafíos principales que trato en este trabajo (que sólo deben considerarse como ejemplares de una lista incompleta del conjunto de desafíos) son los siguientes:

1. Los artefactos técnicos que utilizan durante sus operaciones normales o que han sido diseñados mediante algoritmos de aprendizaje automatizado pueden lograr superar su programación original, en el sentido en que, si bien fueron programados para aprender, el resultado del aprendizaje (i.e. el *output* del proceso) y, muchas veces, el proceso mismo por el que el software llega al resultado adecuado, son desconocidos (y quizás incognoscibles) por los desarrolladores. Así el primer desafío es el de la posibilidad de la reducción de la "opacidad epistémica" que genera el uso de dichos algoritmos.[1]

2. El segundo desafío se desprende del primero y es el de las prácticas dentro del mismo campo de la ciencia de la computación que trabaja sobre aprendizaje automatizado. Una preocupación reciente expresada por importantes miembros de la comunidad es que el campo del aprendizaje automatizado pareciera estar cada vez más cerca de "una alquimia" que del de una "verdadera ciencia" en tanto el

[1] El término “opacidad epistémica” fue introducido por Paul Humphreys (2004) para describir la “distancia epistémica” que un proceso computacional puede generar con respecto a su usuarios dado que la velocidad de dicho proceso es demasiado elevada para seguir seguido por una mente humana o bien porque no hay algoritmos que conecten de manera explícita a las entradas con las salidas.

desconocimiento del funcionamiento interno de dichos programas y la falta de herramientas de testeo sobre la robustez de los mismos pueden llevar a casos en los que los resultados de los mismos programas sean "correctos" bajo cierto standard aunque, pese a su corrección, haber llegado a dichos resultados por procesos equivocados o desconocidos. Además, esto aumenta el riesgo de sobreestimación o sobreajuste [*overfitting*], especialmente cuando la cantidad de datos sobre los que se aplica el algoritmo es demasiado pequeña o no lo suficientemente representativa (Hutson, 2018).

3. El último desafío que analizo se sigue de los dos anteriores y concierne a la cantidad de estos algoritmos que se utilizan hoy en día en la práctica científica, especialmente para procesar y analizar la cantidad de datos que surgen de las técnicas experimentales recientes. Estas grandes bases de datos ya no se limitan a casos de la llamada *Big Science* (como el LHC en el CERN) sino en casos mucho más "pequeños" como en el análisis de imágenes obtenidas por microscopía electrónica de apenas milímetros de corteza cerebral (Helmstaedter et al., 2013). El uso de estos algoritmos se está volviendo prácticamente obligatorio en las ciencias que estudian sistemas complejos, dada la alta dificultad de predecir dichos sistemas mediante modelos clásicos. Un caso similar ocurre en las ciencias sociales, en los que la cantidad de datos obtenidos mediante redes sociales, sistemas de vigilancia comunitaria y sensores en los teléfonos celulares ha creado una fuente de datos inmensa para el testeo de hipótesis de las ciencias sociales, para los cuales también deben usarse algoritmos de aprendizaje.

Hecho este diagnóstico, mi propuesta en este trabajo es la de presentar una posible forma de atender a esta serie de desafíos epistémicos, sugiriendo que una tesis de articulación de modelos puede ser un primer paso para generar confianza en los resultados de los productos de algoritmos de aprendizaje automatizado. Dicha articulación no sólo debe limitarse a observar cómo el avance epistémico se da en la generación de nuevos modelos y la integración de los mismos en distintas disciplinas sino también en la interacción entre "comunidades epistémicas" que tienden a tener métodos de trabajo distintos para el abordaje de los mismos problemas. Como bien lo hace notar Domingos (2015) el campo de la inteligencia artificial en general –y el del aprendizaje automatizado en particular– no es ninguna excepción. A modo de cierre, doy una posible respuesta a la pregunta del subtítulo.

1. Primer desafío: la naturaleza del aprendizaje automatizado

La idea de crear una máquina que sea capaz de simular las capacidades de un ser humano es mucho más antigua de lo que uno podría imaginarse. Cabe recordar, sólo a manera de ejemplo, aquel jugador automático de ajedrez denominado “El Turco”, que hizo su debut en 1769 de la mano de Wolfgang von Kempelen y que simulaba un complicado sistema de relojería que poco tenían que ver con la verdadera capacidad del aparato de jugar al ajedrez: un enano conocedor del juego oculto dentro de la supuesta máquina que no era más que un intrincado disfraz.

Al surgir la computadora moderna, surgieron las dificultades para lograr que una computadora hiciera lo que el usuario quería que la computadora haga. Incluso bajo dominios específicos, programar una computadora era una tarea bastante desafiante. En lo que se podría considerar como la primera referencia al *machine learning* en la literatura, el padre intelectual de la computadora, Alan Turing, sugirió la idea de que si entendemos al cerebro como una computadora,

tranquilamente podríamos imaginarnos una computadora con la complejidad suficiente para ser "programado" de la misma manera en la que se "programa" un cerebro humano común y corriente: tratándolo como un niño y enseñándole cómo funciona el mundo de la misma manera en la que solemos hacer con los niños. Por esa razón sugería que una máquina desorganizada podía ser reorganizada y convertida en una computadora universal si se la sometía a un entrenamiento suficientemente análogo a la forma en la que aprende un niño:

> Esto se puede lograr permitiéndole a la máquina pasar al azar por una secuencia de situaciones y aplicar un estímulo de dolor cuando realice una mala elección y un estímulo de placer cuando haga una buena elección. Es también mejor aplicar un estímulo de dolor cuando hace elecciones irrelevantes. Esto es para evitar que se quede aislado en un anillo de situaciones irrelevantes. La máquina está ahora "lista para su uso" (Turing, 1948/Cooper & Leeuven, 2013, p. 514).

El campo de la inteligencia artificial queda oficialmente inaugurado y bautizado recién unos diez años después con la conferencia de Dartmouth, un workshop de unas 8 semanas de duración propuesto por Claude Shannon y organizado por John McCarthy, que tuvo lugar en New Hampshire entre el 18 de junio y el 17 de agosto de 1956. Quizás lo más concreto que surgió de las discusiones fue el nombre del campo y cierto consenso en concentrarse en métodos simbólicos, acotados a dominios específicos y en sistemas deductivos en lugar de inductivos.

Si bien no se trató estrictamente del primer algoritmo de aprendizaje automatizado, la variante más cercana a lo que hoy conocemos por tal cosa fue presentada por primera vez en enero de 1957 por Frank Rosenblatt.[2] Curiosamente, dicho algoritmo va precisamente en contra de los lineamientos

[2] Por rigurosidad histórica es necesario agregar algunos detalles. La idea de pensar un cálculo lógico para representar a las operaciones presentes en el sistema nervioso llevaron al neurofisiólogo Warren McCulloch y al matemático Walter Pitts a presentar el primer modelo de neuronas artificiales (McCulloch & Pitts, 1943). Sin embargo, en dicho trabajo no presentaron ningún algoritmo de aprendizaje. El objetivo principal era plantear la posibilidad de estudiar al cerebro como, básicamente, una máquina de Turing sofisticada. Las primeras simulaciones computacionales sobre el modelo de McCulloch-Pitts fueron llevadas a cabo por Farley y Clark, quienes utilizaban un algoritmo que modificaba los pesos de las conexiones entre las 128 unidades computacionales o neuronas de su implementación que buscaba reconocimiento de patrones simples. Es curiosa la forma en la que definen "aprendizaje" y cómo se refieren a su implementación como un "organismo": "En términos de nuestro modelo, podemos describir [el aprendizaje] en los siguientes términos. Se elige un número de patrones de entrada y se lo presenta al organismo en un orden y en un tiempo preestablecidos para proveer la "experiencia" requerida. Una o más de estas entradas son designadas como pruebas o tareas de rendimiento, y se construye una medida adecuada de su habilidad, como puede ser una prueba con puntaje. Este puntaje corresponde a la medida que le hemos adjuntado a la transformación general. Si la medida aumenta como resultado de la presentación de entradas "de experiencia", y no incrementa de otra forma, se dice que el organismo aprende" (Farley & Clark, 1954, p. 77). Una implementación un tanto "más rústica" de una red neuronal inspirada por el modelo de McCulloch-Pitts fue el SNARC diseñado por Marvin Minsky en 1951 como parte de su trabajo doctoral que estuvo orientado a una exploración de ideas acerca de cómo el sistema nervioso podría aprender. Las siglas en inglés de la "máquina de Minsky" corresponden a un "Calculador Estocástico Neuronal Analógico por Refuerzo" y consistía en una red de unas 40 neuronas conectadas al azar. La intensidad o peso de las conexiones se ajustaba de acuerdo al éxito que la máquina a la hora de realizar una tarea específica. El peso de las conexiones hacía uso de la memoria de la máquina que consistía, básicamente, en la posición de cada una de las perillas de las sinapsis, que fijaban la probabilidad de propagación de las señales.

generales que surgieron del workshop de Dartmouth y más cerca de las especulaciones de Turing, aunque las mismas no se conocían entonces ya que su trabajo de 1947 fue recién publicado en 1970. En su reporte técnico, Rosenblatt comienza por describir el interés creciente en crear una máquina que sea "capaz de conceptualizar entradas [*inputs*] de luz, sonido, temperatura, etc., que surgen directamente del ambiente –el mundo fenoménico con el que todos estamos familiarizados– en lugar de requerir la intervención de un agente humano para digerir y codificar la información necesaria" (Rosenblatt, 1957, p. 1). Entrenado en la psicología fisiológica de la época, Rosenblatt reconoce en la tarea de reconocer patrones complejos de información algo análogo al proceso de asociación o generalización de estímulos que parece hacer el cerebro humano. Impone dos requisitos fundamentales para llevar a cabo la tarea. Por un lado, las identidades entre los patrones deben ser "aprendidas o adquiridas por la experiencia". Esta es la primera referencia directa a que dicha máquina efectivamente puede "aprender". Por otro lado, impone un requisito económico, que es el de que "el número de unidades funcionales en el sistema de almacenamiento o memoria debe ser mucho menor que el número de formas o memorias a ser retenidas" (Rosenblatt, 1957, p. 1).

Para poder cumplir con dichos requisitos, Rosenblatt se basa en sus resultados teóricos anteriores que le sugieren que debería ser factible implementar dicho sistema que "aprenderá a reconocer similaridades o identidades entre patrones [...] de una manera que puede ser análoga a los procesos de percepción de un cerebro biológico" (1957, p. 2). El punto clave que le permitirá hablar de aprendizaje es lo que señala inmediatamente a continuación, que es que el "sistema propuesto depende de principios probabilísticos para su operación, en lugar de deterministas, y gana su confiabilidad de las propiedades de las mediciones estadística obtenidas de una gran población de elementos" (Rosenblatt, 1957, p. 2). El nombre que recibirá este sistema es el de "Perceptrón" y es la arquitectura básica del aprendizaje automatizado supervisado. Curiosamente es ya en este mismo trabajo que aparece la primera referencia a lo que he denominado la "opacidad epistémica" del aprendizaje automatizado: su característica de "caja negra". Pensando en su implementación como un sistema electrónico específico, aunque el nombre se generalizó para esta arquitectura de aprendizaje, Rosenblatt comenta que

> podemos considerar al perceptrón como una caja negra, con una cámara de TV para la entrada y una impresora alfabética o un conjunto de señales de luz como salida. Su rendimiento [*performance*] puede entonces ser descrito como un proceso de aprender a dar la misma señal de salida [...] para todos los estímulos ópticos que pertenecen a alguna clase constituida arbitrariamente (Rosenblatt, 1957, p. 1).

Más allá de cuál sea la arquitectura de aprendizaje automatizado que se esté implementado, esta característica de "caja negra" es común a todas.[3] En el caso del Perceptrón simple, la "caja negra" es

---

El refuerzo era introducido por el operador a manera de recompensa cuando la máquina lograba un avance en la tarea y las últimas probabilidades se ajustaban mecánicamente para guardar dicho "aprendizaje".

El primer uso en la literatura del término "machine learning" fue por Arthur Samuel –uno de los asistentes al Workshop de Darthmouth– en 1959, en su trabajo sobre máquinas que aprenden a jugar al juego de las damas mejor que sus programadores (Samuel, 1959).

[3] Curiosamente, Turing ya había hecho notar la posible ignorancia del maestro con respecto a la "máquina pupilo" en su discusión un tanto más conocida de las máquinas niño en Turing (1950): "una característica importante de una máquina que aprende es que su maestro a menudo será bastante ignorante de lo que ocurre precisamente adentro, pese a que hasta algún punto todavía pueda predecir el comportamiento de su pupilo. Esto

el Perceptrón en sí pero al tratarse de una sola capa en la estructura de la red, la opacidad epistémica que presenta es limitada. Además, para esta clase de arquitectura, existen pruebas de carácter matemático que aseguran que para un conjunto de datos que es linealmente separable el algoritmo converge a una solución. Aquí, sin embargo, ya se puede notar otra fuente de opacidad epistémica que tiene que ver con nuestra capacidad de conocer las propiedades de un conjunto de datos *antes* de ser usado para entrenar una red. El problema de los datos y el problema de la caja negra se multiplican en los casos más interesantes de arquitecturas de aprendizaje actuales, en los que la naturaleza de los datos es extremadamente heterogénea y difícil de explicar, mientras que los algoritmos cuentan con una infinidad de cajas negras conectadas entre sí. Para una buena lectura epistemológica es necesario aquí pasar a otro plano de análisis, en el que entran en juego los usuarios y los ingenieros de los métodos de aprendizaje y la robustez de sus prácticas.

Ahora bien, antes continuar es necesario aclarar que en el presente trabajo estoy poniendo el énfasis en una clase general de estructuras y algoritmos de aprendizaje automatizado: el paradigma que suele denominarse como "conexionista" y en el que las redes neuronales artificiales son la figura estrella. La relación de este paradigma con los otros dentro del campo es tan intrincada como interesante y comentaré más adelante algo al respecto. Por ahora es suficiente justificar el énfasis de este trabajo comentando que dentro de los paradigmas de aprendizaje automatizado las redes neuronales artificiales han recuperado el rol de paradigma reinante, en particular gracias a que con ellas se han logrado implementar por primera vez sistemas que sean capaces de resolver tareas específicas –seguimos en el nivel de una inteligencia artificial especializada, no una general– con una eficacia al menos equivalente a la de un ser humano, e incluso mucha veces superior que la de su creador. Algunos ejemplos de estas tareas son el reconocimiento de patrones complejos, de objetos en imágenes, el reconocimiento de voz y la traducción de textos entre varios idiomas naturales.

2. Segundo desafío: entre la alquimia y la ciencia

Parafraseando a la clásica línea de Eugene Wigner sobre la irrazonable efectividad de la matemática en las ciencias naturales –que sigue siendo uno de los problemas centrales en la intersección entre la filosofía de la matemática y la filosofía de la ciencia–, Pedro Domingos comenta que "el aprendizaje automatizado es lo que se obtiene cuando la irrazonable efectividad de las matemáticas se encuentra con la irrazonable efectividad de los datos" (Domingos, 2015, p. 31). Si antes de introducir a los datos ya teníamos un problema filosófico difícil de resolver, ahora debemos enfrentarnos a uno que es varios órdenes de magnitud más complejo. Esto se vuelve todavía más difícil de elucidar cuando la estructura de aprendizaje es de muchas capas de unidades que aprenden y que son extremadamente sensibles a las propiedades de los datos, que son la característica principal de las llamadas arquitecturas de aprendizaje profundo [*deep learning*]:

> Una arquitectura de aprendizaje profundo es un apilamiento multicapa [*multilayer stack*] de módulos, de los cuales todos (o la mayoría) están sujetos a aprendizaje y mucho de los

debería aplicarse con más fuerza a la educación posterior de una máquina que surja de una máquina niño con un diseño bien probado (o programa). Esto está en claro contraste con el procedimiento normal al usar una máquina para hacer computaciones: el objeto de uno es tener una clara imagen mental del estado de la máquina en cada momento de la computación" (Turing, 1950/Cooper & Leeuven, 2013, p. 567).

> cuales pueden computar mapeos de entrada-salida no lineales. Cada módulo en la pila transforma su entrada para incrementar tanto la selectividad como la invarianza de la representación. Con múltiples capas no lineales [...] un sistema puede implementar funciones extremadamente intrincadas de sus entradas que son simultáneamente sensibles a detalles pequeños [...] e insensibles a grandes variaciones irrelevantes [...] (LeCun, Bengio, & Hinton, 2015, p. 438)

Estas arquitecturas pueden ser extremadamente precisas en tareas que siempre fueron muy difíciles de lograr mediante otros métodos de aprendizaje o de programación directa como el reconocimiento de caracteres manuscritos, de clasificación de objetos y de reconocimiento de patrones de voces, tareas en los que los humanos siempre fueron superiores dado lo intuitivas que son dichas tareas. Ahora bien, para un humano en su condición de agente epistémico sobre un sistema que puede realizar tal proeza, la tarea no es tan sencilla. Esto es, conocer las razones por la que una red neuronal artificial hace tanto lo que puede llegar a hacer como lo que efectivamente logra al ser aplicada a un problema específico, puede fácilmente tornarse en una tarea tan difícil como la de saber por qué el cerebro humano es tan capaz de realizar dichas proezas tan eficientemente. Paradójicamente, la utilidad de los métodos de aprendizaje profundo *radica* en esta "distancia epistémica" que toman de quienes la desarrollan.

Cualquier implementación de aprendizaje automatizado puede describirse en cuatro pasos fundamentales una vez que se ha definido el problema de aplicación: (1) la obtención de datos, (2) el diseño de una función de pérdida o coste adecuada según el problema, con la cual se podrá medir el rendimiento de la implementación, (3) la selección de una arquitectura de red adecuada con sus parámetros correspondientes, y (4) el reajuste de los parámetros mediante un algoritmo de optimización que minimice la función elegida. Cuando la arquitectura empleada es tan compleja como la que describen LeCun et al. (2015), la cantidad de niveles de abstracción que la red termina implementando para "representar" los datos con los que está trabajando se escapa de cualquier intento humano de comprender dichas abstracciones. De hecho, uno hasta podría pensar otra similitud con lo que logra el cerebro humano, en tanto no somos conscientes de la cantidad de abstracciones que éste realiza para "representar" un estímulo, por ejemplo.

En última instancia, uno puede observar que tanto el ensamblado como el testeo de dichas arquitecturas complejas se vuelve un trabajo con mucho más "ensayo y error" que el que uno esperaría de una disciplina formal, especialmente en lo que corresponde al paso (3) de la implementación. Esto dejaría al campo del aprendizaje profundo como una disciplina más bien empírica –si es que no netamente ingenieril– en el que el método de estudio es más inductivo y artesanal, en el que se estudian los sistemas con métodos de corte empírico, como si se tratara de un fenómeno natural a explicar.

Algunos miembros de la comunidad han incluso sugerido que muchas veces se ven sorprendidos por los resultados de una implementación de aprendizaje profundo y han demostrado su preocupación dada la cantidad de resultados que se publican por año en los que no se aclara si dicha implementación fue lo suficientemente testeada, tanto en términos de los datos usados para entrenarla y validarla, como en términos de poner a prueba los distintos módulos para asegurarse de que estén realizando lo que los desarrolladores pretendían implementar y, por lo tanto, que los resultados que obtienen se obtienen efectivamente por las razones esperadas. Estos múltiples factores de "ensayo y error" (término que uno hasta puede leer en los libros de texto del campo), que de momento no pueden ser guiados por una teoría matemática robusta, han incluso llevado a algunos a sugerir que el

aprendizaje profundo se encuentra más cerca de una alquimia que de una ciencia, ya que muchas veces se publican resultados o se implementan en sistemas de consumo masivo o de toma de decisiones sin haber pasado por suficientes pruebas (cf. Hutson, 2018).

3. Tercer desafío: entre técnicas y modelos

Un tercer desafío epistémico –y que incorpora a los dos desafíos ya planteados– gira en torno a la aplicación contemporánea de dichos sistemas de aprendizaje automatizado tanto para el modelado de fenómenos naturales como para los casos en los que un experimento, una simulación o una combinación de ellos, produce un volumen de datos extremadamente grande que no puede ser procesado por los métodos tradicionales de análisis de datos. Un ejemplo del primero puede ser el uso de una red neuronal con aprendizaje automatizado para simular una red biológica de un cerebro animal. En dicho caso, hay cierta presunción de que ambos sistemas comparten cierta estructura por lo que se suele apelar a la similaridad como fuente de garantía epistémica de los resultados de la simulación. Sin embargo, este aspecto "representacional" se pierde en casos en los que modelos de redes neuronales se utilizan, por ejemplo, para generar predicciones sobre el comportamiento de un huracán sin hacer uso alguno de la teoría física que rige dicha clase de fenómenos (la mecánica de fluidos en este caso).

La otra cara del tercer desafío incluye los casos de la práctica científica en el que los sistemas de aprendizaje deben ser utilizados no como un modelo de un fenómeno particular sino como una técnica auxiliar para el procesamiento de los datos, sin la cual dicho análisis no podría ser llevado a cabo dado el volumen de datos con el que se trabaja. Uno de los ejemplos más vívidos de esta cara del desafío es el del Sloan Digital Sky Survey (SDSS), un proyecto de investigación astronómica que pretende cartografiar un cuarto del cielo visible y cuya última publicación de datos ronda los 156 Terabytes. Este es un ejemplo paradigmático de la transformación de algunas disciplinas hacia una ciencia dirigida por datos [*data driven*] en lugar de ser dirigida y guiada por teorías fundamentales, en el que la interacción de distintos agentes epistémicos como astrónomos, técnicos y analistas de datos tienen un rol similar en la producción de conocimiento. Sin embargo, allí en donde yace su principal riqueza, en las múltiples posibilidades para el análisis de los datos, yace también su gran desafío epistémico, que es el de distinguir entre características reales del fenómeno y de los artefactos o ruido introducidos por los sistemas de detección o por los algoritmos de análisis de datos.

4. Desafiando los desafíos

Dada la forma en la que estos desafíos están conectados entre sí, enfrentar a uno de ellos significa estar enfrentando a los otros simultáneamente. Más importante aún, dada la naturaleza de los mismos, dichos desafíos no pueden ser estrictamente resueltos, sino que deben ser entendidos como parte integral de la relación de los agentes epistémicos con los sistemas de aprendizaje automatizado. Es

decir, deben ser atendidos en todas sus facetas por los distintos usuarios de dichos modelos y sus resultados. Conocer la naturaleza del método empleado para generar conocimiento sobre un fenómeno y las incertidumbres que dicho método incorpora, no sólo es clave para la misma empresa científica, sino que es un aspecto crucial a considerar cuando se tienen que tomar decisiones, en contextos tanto técnicos como políticos, en base a los resultados.

De esta manera se deben implementar una serie de prácticas para disminuir los peligros asociados con su uso. Muchas de estas prácticas son similares a las llevadas a cabo por las diferentes comunidades científicas que estudian un fenómeno natural por vía experimental o recurriendo a simulaciones computacionales. Si bien en ningún campo científico ni en la ingeniería existe un conjunto de prácticas que siempre asegure la obtención de un buen resultado, sí se suelen encontrar una serie de recomendaciones de "mejores prácticas" que tienen por objetivo disminuir tanto como sea posible la incertidumbre de los resultados. Poner a prueba la robustez del sistema de aprendizaje para asegurarse qué es lo que contribuye cada componente y replicar el experimento con una estructura de aprendizaje diferente son ejemplos de la metodología científica que pueden ser aplicados al caso del aprendizaje automatizado.

Por otro lado, dada la dependencia de los sistemas de aprendizaje en los datos con los que son entrenados y luego aplicados, gran parte de la disminución de nuestra ignorancia sobre el sistema dependerá de nuestra capacidad de poder manejar, replicar y entender dichos datos. Por esta razón, los estudios sobre la naturaleza de los datos y de los procesos de adquisición de los mismos deberían también apelar a estrategias diferentes para asegurar la convergencia de los resultados. Dada la misma naturaleza de un dato, y de lo compleja que puede ser su historia causal, esto no es fácil de lograr. Aquí, de nuevo, el rol de los modelos es clave, ya que ellos pueden ser empleados tanto para describir los posibles mecanismos de creación de datos como para crear nuevos datos basados en dichos modelos. Estos datos pueden ser utilizados para poner a prueba el sistema de aprendizaje que se pretende entender o aplicar. Es importante notar que dichos modelos alternativos no deberían incorporar métodos de aprendizaje automatizado a efectos de controlar posibles sesgos introducidos en el modelo por recurrir a un mecanismo similar que el que se quiere poner a prueba. Mientras mayor sea la independencia de los modelos mucho más confiable serán los resultados de la comparación.

Esto no quiere decir que no se puedan ni deban usar otros métodos de aprendizaje automatizado para testear o mejorar los resultados en un contexto particular. De hecho, el trabajo en simultáneo con muchos modelos de la misma clase es una práctica común —e indispensable en algunos casos—, tanto a la hora de intentar predecir el estado futuro de sistemas complejos con simulaciones computacionales como el tiempo atmosférico a corto plazo o el clima a largo plazo, como en casos en donde lo que se pretende hacer es mejorar el rendimiento de un modelo individual de aprendizaje. A esta clase de combinación de modelos se les llama "ensambles" porque intentan combinar distintos factores (como diferentes condiciones iniciales) o, directamente, distintos modelos, esperando que el promedio de sus resultados sea mucho más preciso que el de los modelos individuales. Si los modelos empleados son de la misma clase, no se ataca el problema de la caja negra que describimos antes pero sí se pueden usar para atacar el problema del sobreaprendizaje, que estrictamente también es una característica de estos métodos, en especial del *deep learning* que puede verse como una forma explícita de tomar ventaja de esta supuesta deficiencia de los modelos. Se habla de sobreaprendizaje cuando un sistema aprende con tantos detalles la naturaleza del conjunto de datos con el que se la entrena que luego falla cuando se procede a implementar dicho modelo con un conjunto de datos diferente. Como en cualquier técnica de generalización, siempre existirá la necesidad de realizar un *trade-off* o un equilibrio entre dos desiderata que son incompatibles: tasa de error nula y máxima generalizabilidad. Por este motivo, es necesario encontrar un balance entre una

tasa de error lo suficientemente baja y una generalizabilidad acotada pero suficiente para el rango de problemas que se estén tratando.

La construcción de un ensamble de modelos puede verse también como un caso de aprendizaje automatizado supervisado (aunque quizás sea necesario introducir el término "metaaprendizaje" por tratarse de aprender de lo aprendido) ya que se emplean los resultados de los distintos modelos como factor de corrección para el "modelo ensamblado" que se obtiene también mediante una técnica de aprendizaje automatizado. Si bien el hecho de generar un modelo que sobrepase a los anteriores, en el que por definición el espacio de estados posibles es mayor y por lo tanto el riesgo de sobreaprendizaje también lo sería, podría verse más como un problema que una solución, si se usan adecuadamente y se priorizan modelos cuyos resultados originales tienden a divergir, el modelo resultante sea probablemente mucho más robusto que cualquiera de los originales.[4]

Estrictamente todavía queda un desafío que atender, que es aspecto de caja negra de los algoritmos. Digo "estrictamente" porque el uso de múltiples modelos ya atiende a parte del desafío, en tanto pueden ser usados para entender por qué un algoritmo podría haber tomado una decisión, si bien probablemente nunca se sepa por qué efectivamente lo hizo. Agregar "capas semánticas" a las distintas instancias del algoritmo, que permitan interrogar o saber el estado del mismo, puede ser una estrategia a implementar en algunos modelos simples, pero parece algo bastante difícil de lograr en la mayoría de los modelos de aprendizaje profundo, al menos por el momento, pese a que en principio podría ser una forma de comenzar a develar el entramado entre el paradigma conexionista y el paradigma simbólico.

En realidad, todas estas formas de atacar a los desafíos epistémicos de los sistemas de aprendizaje automatizado nos revelan un problema epistemológico de segundo orden, que es el de describir el complejo entramado que puede existir entre modelos, ya que muchas veces los modelos exitosos incluyen, como una parte propia o asociados a el trabajo con ellos, modelos de aprendizaje automatizado. Esta mirada de segundo orden indica también que los modelos no deben juzgarse por su capacidad de resolver problemas particulares ni por su posible realismo con respecto al fenómeno que modelan, sino más bien por la robustez de las prácticas asociadas a su implementación, la

[4] Una de las técnicas más comunes y robustas disponibles es la denominada agregación de *bootstrap*, también conocida como empaquetado o *bagging*. Esta técnica fue originalmente descrita por Breiman (1996) y consiste en generar nuevos conjuntos de entrenamiento mayor mediante un muestreo uniforme y con reemplazo. Esta es la parte de "*bootstrap*" (que literalmente sugiere la idea de tirar de las propias botas aunque también hace referencia a cualquier técnica que pueda crear algo más complejo y efectivo sin introducir recursos externos). Todos los modelos se aproximan con las muestras resultantes del *bootstrap* y, si se trata de un problema de regresión, se promedia el resultado o, si se trata de un problema de clasificación, se toma al resultado de cada modelo como un voto a favor de cierta decisión. Una idea similar está detrás de otra de idea del mismo Leo Breiman: los denominados bosques aleatorios o *random forests*. En este trabajo me he concentrado en el paradigma conexionista pero no es la única forma de ver al aprendizaje automatizado. Otra técnica consiste en la formación de árboles de decisión, nombre que reciben porque pueden visualizarse como una estructura de un diagrama de flujos en los que un nodo representa una prueba sobre un atributo y cada rama el resultado de dicha prueba hasta llegar a un "nodo hoja" que representa la etiqueta de una clase o la decisión después de considerar una prueba en todos los atributos. El camino entre el primer nodo y la hoja representa una regla de clasificación. Utilizar un conjunto aleatorio de dichos árboles permite obtener resultados que corrigen algunos de los problemas propios de los árboles de decisión, especialmente su tendencia a sobreaprender su conjunto de entrenamiento y fallar en la generalización (Breiman, 2001).

fertilidad de los modelos (entendida en términos de la familia de modelos que pueden generar dentro de su clase) y la manera en la que distintos modelos de un mismo proceso, natural o artificial, pueden llegar a mostrar resultados que convergen. Si los modelos provienen de distintos paradigmas, tanto mejor para los modelos (o la naturaleza, según corresponda).

5. ¿Aprenden las máquinas? ¿Aprendemos nosotros?

En esta última sección, retomo la pregunta del subtítulo, tomando en cuenta los desafíos epistémicos y los desafíos a los desafíos. Dada lo especulativa que es la pregunta en sí, me permito ser mucho más especulativo en esta sección que en las anteriores.

Tal y como puede esperarse, la respuesta a la pregunta depende de cómo definimos "aprender". En un obvio sentido, los algoritmos de aprendizaje automatizado aprenden en tanto son capaces de ser literalmente *entrenados* para realizar una tarea que antes de dicho entrenamiento no podían realizar. Lo que hace ruido en dicha mirada es que si pensamos en la clase de aprendizaje que realizan los seres humanos, cuando ellos aprenden suelen poder hacer más cosas sobre lo aprendido que lo que puede un algoritmo de *machine learning*. Por ejemplo, un humano puede traspasar dicho conocimiento a otro dominio y, por sobre todo, puede ser interrogado para determinar qué tanto aprendió. De alguna manera, las máquinas entrenadas también son "interrogadas" cuando se pretende que resuelvan un problema que implica un conjunto de datos diferente con el que fue entrenada. Estrictamente no nos dicen cómo lo hacen, pero tienen un *know-how* de dominio específico que les permite hacer la tarea y "pasar la prueba".

Remarco la idea de prueba ya que en esto el lector quizás piense en una estrategia similar a la usada por Turing (1950), quien en lugar de dar una definición de inteligencia para explorar la idea de si una computadora podía pensar inteligentemente o no, considerando el aspecto polisémico del término "inteligencia", optó por dar una definición más bien operativa de la inteligencia como la capacidad de superar cierta prueba, asociada a su capacidad lingüística (de nuevo un dominio acotado de habilidades). Si bien los seres humanos también somos capaces de tener conocimiento práctico de dominio que nos es imposible pasar a proposiciones, solemos al menos poder intentar traspasar dicho conocimiento de dominio a una explicación de por qué decimos que aprendimos algo o, en condiciones óptimas, a decir qué es lo que aprendimos.

Esta es una distinción binaria clásica en epistemología y en filosofía de la mente entre conocimiento proposicional (o conocimiento de algo [*knowing that* o *knowing what*] y el conocimiento implícito o procedimental a la hora de realizar una tarea [*knowing how*]. Quizás sea una buena distinción analítica pero no creo que haga a una buena definición de conocimiento, al menos en el dominio que compete, por el momento, al aprendizaje automatizado. Ya llegará el tiempo en que las computadoras puedan hacer filosofía ellas mismas y, si les sirve, discutirán dicha clasificación. Por el momento, sugiero considerar como una mejor clasificación del conocimiento humano y del maquínico la distinción propuesta por Minsky de "clasificar al conocimiento en términos de las clases de pensamiento que podemos aplicarle" (2006, p. 174), del que salen las siguientes habilidades:

Experiencia positiva: saber en qué situaciones aplicar qué fragmento particular de conocimiento.

Experiencia negativa: saber qué acciones no tomar, ya que pueden hacer que una situación empeore.

Habilidades de depuración: conocer maneras alternativas para proceder cuando el método usual falla.

Habilidades adaptativas: conocer cómo adaptar el conocimiento viejo a nuevas situaciones.

Bajo este nuevo esquema, probablemente los algoritmos de aprendizaje automatizado queden todavía más lejos de un "aprendizaje verdadero" ya que recibirán un puntaje menor en los cuatro puntos. De hecho, dada la discusión anterior acerca de cómo se trabaja con el aprendizaje automatizado podemos notar que todas estas "habilidades" no son realizadas por el algoritmo, sino que son realizadas por los diseñadores de dichos algoritmos. Parte de la razón para ello es que la clasificación sugerida por Minsky tiene un elemento de sentido común de muy alto nivel que todavía no está al alcance de los algoritmos de aprendizaje disponibles (pero que son necesarios considerar si se pretende hablar de la posibilidad de una inteligencia artificial general).[5]

Siguiendo esta última línea, propongo revisar para finalizar al aprendizaje profundo, paradójicamente de manera un tanto superficial. Si analizamos la clasificación de conocimiento de Minsky, podemos notar que en todas hay una estrategia básica que es la que permite, a fin de cuentas, que puedan ser implementadas. Todas ellas requieren poder realizar una abstracción sobre un conocimiento previo o sobre una situación, de manera tal de poder determinar qué características son esenciales a dicho conocimiento o situación y que permite aplicarlo a otro dominio o tomar una decisión sobre distintas alternativas.[6] El punto a señalar es que algún proceso básico de abstracción, entendida de manera general como la aislación de un conjunto de elementos para aplicarlo en otro lugar para intentar resolver un problema o hasta para hacer el proceso inverso y substituir lo abstraído por otro elemento, es precisamente lo que ocurre en una red neuronal profunda y, como si eso fuera poco, ocurre automáticamente. Dada la forma en la que están estructuradas con múltiples capas en la que sólo una está "expuesta" directamente a los valores de entrada, cada capa siguiente sólo tiene acceso a la abstracción realizada por la capa anterior. Esto hace que tanto su arquitectura como su dinámica pueda entenderse en términos modulares cuya función es precisamente la de construir características a partir de otras características, a medida que la red neuronal actúa sobre los valores de entrada. La clave está en el aspecto modular. Distribuir a los elementos de una red en múltiples capas con una organización jerárquica, en donde los elementos pueden tener distintas intensidades de

[5] Si usamos una mirada de nivel un tanto más bajo quizás podamos reemplazar algunos términos en la clasificación (como sustituir "conocimiento viejo" por "la matriz de pesos de las conexiones entre neuronas en un tiempo *t* determinado") y pretender hablar de aprendizaje en dichos términos, aunque no creo que sea una buena estrategia. Quizás el atento lector también considere que mi estrategia de pasar tan rápido y quizás sin cuidado entre "cocimiento" y "aprendizaje" sea una mala estrategia pero creo que es algo que hacemos todo el tiempo y que puede ser útil para cambiar la mirada sobre ambos conceptos, pese a la crítica de "suciedad analítica" que dicha estrategia recibiría en casi cualquier contexto. De alguna manera es una sugerencia a volver a usar a los métodos de aprendizaje automatizado para entender cómo es que conocemos y aprendemos los seres humanos. Si no hay en eso una tarea de naturalización de la epistemología, habrá que buscarla en otro lado y dudo mucho que aquel otro lado no recurra a algún que otro algoritmo de "aprendizaje".

[6] Usar el término "abstracción" bien podría considerarse una forma de *obscurum per obscurius* ya que se trata de un término mucho más polisémico que el de aprendizaje y sería necesario hacer un trabajo pormenorizado de qué sentidos de abstracción se pueden aplicar a este caso, lo que quedará para otra oportunidad. Sin embargo, quizás sea interesante señalar una forma de pensar la abstracción que va un poco más allá de la ingeniería inversa de características descrita aquí y que no suele ser considerada como una actividad de abstracción y se trata de alguna forma de pensamiento causal. Judea Pearl no entiende al causalidad explícitamente como un proceso de abstracción pero su trabajo es una buena dirección para atacar el problema (Pearl & Mackenzie, 2018).

interacción de acuerdo a si están en el mismo subsistema o no es una de las hipótesis más plausibles sobre el funcionamiento y el diseño de sistemas complejos (Simon, 1996).

Más allá del costo computacional que requieren y que sólo estuvo disponible desde hace alrededor de una década, probablemente una de las razones por la que no se haya trabajado con este tipo de redes antes es el hecho de que una red neuronal multicapa ya es un aproximador universal. Esto es, dado el suficiente número de neuronas en una única capa oculta, dicha red puede aproximar cualquier función que se desee. De ahí que originalmente no se haya pensado en la necesidad de aumentar la complejidad de la red (entendida como el número de capas ocultas) y sólo en su capacidad teórica para resolver un problema con una capa. Pero, si la naturaleza nos enseña algo, es que el tiempo es un factor más que importante y un sistema modular con muchos componentes separados en subsistemas puede evolucionar mucho más rápido que un sistema que no tenga esta arquitectura. Cuando el tiempo es un factor, los argumentos "en principio" deben ser tomados con cuidado ya que quizás haya que poner cosas "en la práctica" a la hora de lograr algo. Curiosamente, esta arquitectura modular que hace que una red de aprendizaje profundo sea extremadamente difícil de estudiar, la vuelve un recurso ideal para estudiar sistemas de complejidad similar como los del mundo natural. Y el aprendizaje está, después de todo, soportado por el cerebro humano, probablemente el sistema más complejo que conocemos o que pretendemos conocer.

Referencias

Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies* (Reprint edition). Oxford, United Kingdom ; New York, NY: Oxford University Press.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Cooper, S. B., & Leeuwen, J. van (Eds.). (2013). *Alan Turing: His Work and Impact* (1 edition). Waltham, MA : Kidlington, Oxford: Elsevier Science.

Domingos, P. (2015). *The master algorithm: how the quest for the ultimate learning machine will remake our world*. New York, NY: Basic Books.

Farley, B., & Clark, W. (1954). Simulation of self-organizing systems by digital computer. *Transactions of the IRE Professional Group on Information Theory*, *4*(4), 76–84.

Helmstaedter, M., Briggman, K. L., Turaga, S. C., Jain, V., Seung, H. S., & Denk, W. (2013). Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, *500*(7461), 168-174. https://doi.org/10.1038/nature12346

Humphreys, P. (2004). *Extending ourselves: computational science, empiricism, and scientific method*. New York: Oxford University Press.

Hutson, M. (2018). Has artificial intelligence become alchemy? *Science*, *360*(6388), 478-478. https://doi.org/10.1126/science.360.6388.478

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Minsky, M. (2006). *The emotion machine: commonsense thinking, artificial intelligence, and the future of the human mind*. New York: Simon & Schuster.

Pearl, J., & Mackenzie, D. (2018). *The Book of Why: The New Science of Cause and Effect* (1 edition). New York: Basic Books.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210–229.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed). Cambridge, Mass: MIT Press.

Turing, A. M. (1948). *Intelligent Machinery* (Report Written by Alan Turing for the National Physical Laboratory).

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *59*(October), 433–60.