5. INTELIGENCIA ARTIFICIAL, APRENDIZAJE DE MÁQUINA Y TEMAS ASOCIADOS

Autor: Dr. Claudio Delrieux

5.1 INTELIGENCIA ARTIFICIAL

En este capítulo presentamos brevemente qué es la Inteligencia Artificial, su relación con otras tecnologías como el aprendizaje de máquina, el aprendizaje profundo, las metaheurísticas y la minería de datos. Se mencionarán las aplicaciones principales, y cómo se relaciona actualmente con un sinnúmero de tareas y procesos industriales en forma provechosa.

5.1.1 Inteligencia Artificial: Breve introducción

Una definición comúnmente aceptada define a la Inteligencia Artificial (IA) como el análisis y el diseño de sistemas artificiales autónomos capaces de exhibir un comportamiento inteligente. Se asume que para que un agente actúe inteligentemente debe poder percibir su entorno, elegir y planificar sus objetivos, actuar hacia el logro de estos objetivos aplicando algún principio de racionalidad, e interactuar con el medio y con otros agentes inteligentes, sean estos artificiales o humanos. En los últimos años, y con la revolución de las comunicaciones, esta sinergia se ha extendido hacia una ecuación más completa: “Información + Inteligencia + Ubicuidad”. Por poner unos ejemplos, los sistemas de inteligencia ambiental, sensibles al contexto en el que se mueven, el trabajo hacia una web semántica o los dispositivos para la telemonitorización inteligente de parámetros biológicos, muestran de forma patente hacia donde está evolucionado el campo de la IA.

Si bien existe una perspectiva científica de la IA, la cual busca una teoría computable del conocimiento humano (es decir, una teoría en la que sus modelos formales puedan ejecutarse en un sistema de cálculo y tener el mismo carácter predictivo que tiene la física, por ejemplo), en general tiene mayor primacía la IA como ingeniería, con objetivos pragmáticos y de menor alcance. Como se ve en la Fig. 5.1, la IA tiene relación directa con otras ramas aplicadas (minería de datos, aprendizaje de máquina, metaheurísticas y aprendizaje profundo), por lo que cabe preguntarse también cuáles serían los temas “centrales” que quedan circunscriptos dentro de la IA. Entre ellos podemos contar los siguientes:

- Sistemas basados en conocimiento
- Resolución de problemas
- Planificación de tareas
- Lenguaje natural
- Visión artificial

Posiblemente una manera de circunscribir estos temas y demarcarlos respecto de los demás, es que su enfoque es “simbólico” (lo definiremos más abajo) en contraposición con ML, DL, MH y MD, que se basan en enfoques numéricos.

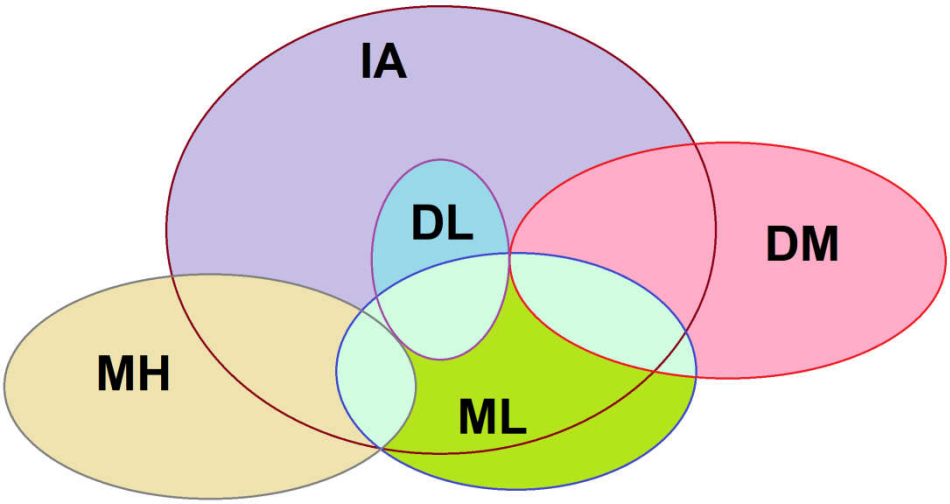

Figura 5.1: Relación entre Inteligencia Artificial (IA), Aprendizaje de Máquina (ML), Aprendizaje Profundo (DL), Minería de Datos (DM) y Metaheurísticas (MH). Fuente: elaboración propia.

5.1.2 Sistemas basados en conocimiento

Posiblemente ésta sea la meta inicial y la más ambiciosa, propuesta desde la década de 1960 pero con raíces en el pensamiento filosófico analítico y Cartesiano. La idea central es definir un sistema de razonamiento (un lenguaje formal para expresar datos, reglas, contextos, situaciones, etc.) tal que dar respuesta a problemas originados en situaciones reales o hipotéticas derive de un “cálculo” (sistema de inferencia), con las propiedades fundamentales de ser objetivo, replicable, transparente y reutilizable.

El marco formal casi universalmente adoptado lo brindan las lógicas (el cálculo de predicados y algunas de sus variantes), dentro del cual el conocimiento se puede expresar en forma de hechos, reglas, mecanismos de inferencia, conclusiones, etc. Como se muestra en la Fig.5. 2, el punto de partida es el diseño de esta representación formal a partir de descripciones realizadas por observadores humanos. Este proceso a veces es iterativo, y se conoce también como “elicitación del conocimiento”.

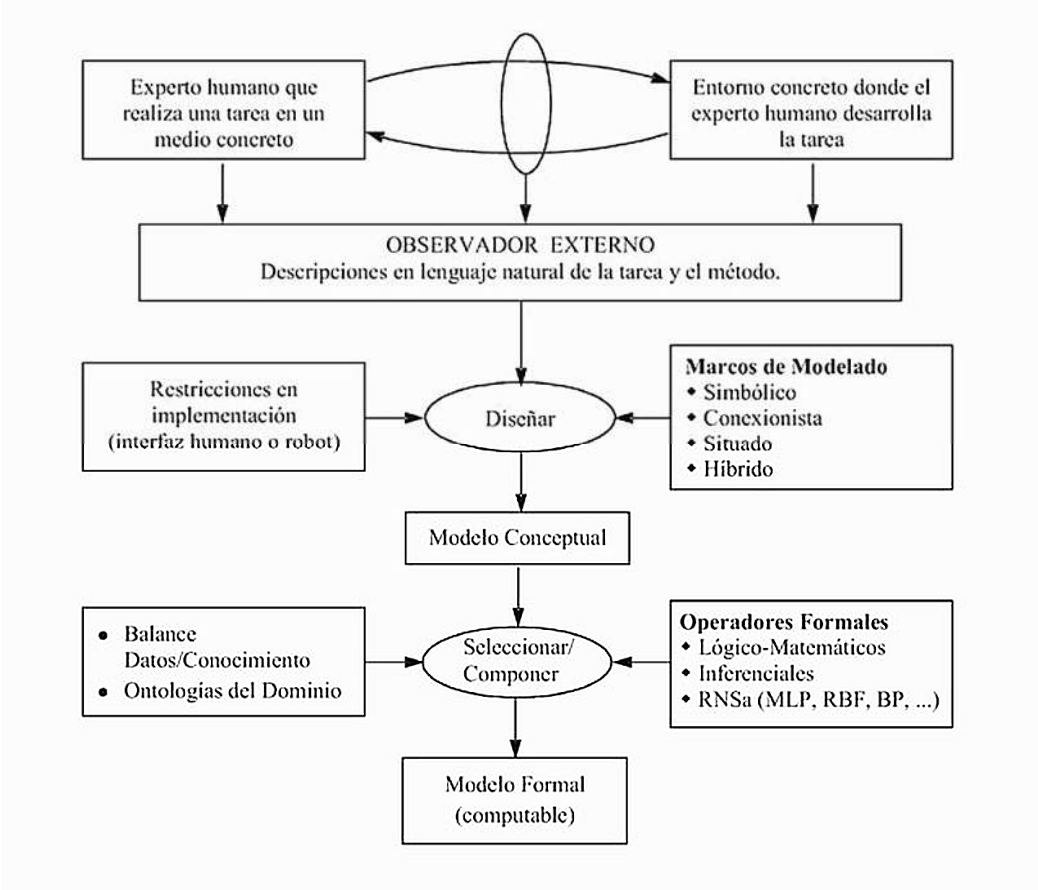

Figura 5.2: Marco conceptual de la IA simbólica (fuente: elaboración propia).

En la práctica, el uso exclusivo del lenguaje de la lógica es demasiado árido, por lo que se suelen adoptar prácticas híbridas (incluyendo probabilidades, lógica difusa, mecanismos pragmáticos, etc.) a la hora de determinar el modelo conceptual resultante. Este último es el que permite finalmente en la iteración siguiente elaborar el modelo computable, expresándolo en un sistema de programación que puede ser un lenguaje específico (por ejemplo, PROLOG), en un lenguaje genérico con bibliotecas (por ejemplo, Python Logic), o directamente en suites de software propietarias.

Un caso particular que históricamente tuvo mucho éxito académico e industrial durante la década de 1990 fueron los “expert system shells”, que permitían el RAD de sistemas basados en conocimiento a partir de una factorización intuitiva de sus componentes (base de conocimiento, motor de inferencia, interacción con los usuarios). Este modelo permite el rápido deploy de modelos de aplicación bastante genérica, y tienen la ventaja de ser modulares, aunque la utilización conjunta de dos o más bases de conocimiento sigue siendo un problema formal aún no completamente resuelto.

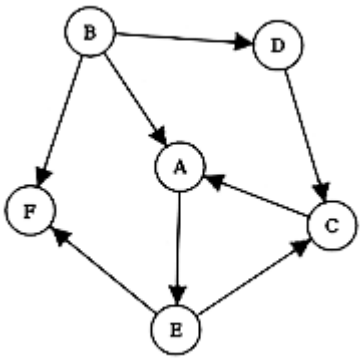

Figura 5.3: grafo de estados (fuente: elaboración propia).

5.1.3 Resolución de problemas

La resolución de problemas es un conjunto de modelos y técnicas específicamente diseñadas para encontrar secuencias de pasos que obtengan soluciones en dominios bien circunscriptos. Un ejemplo típico es solución de problemas de ingenio o en juegos (ajedrez, go), dado que las técnicas empleadas se adaptan exitosamente en problemas del mundo real. El modelo formal en este contexto lo constituye el concepto de grafo de estados *estático*, donde cada nodo es un estado posible del mundo, y los arcos que los interconectan son las posibles acciones a llevar a cabo (ver Fig. 3). De esa manera, las posibles situaciones iniciales quedan representadas como nodos iniciales, algunos nodos son los posibles estados finales esperados, y puede haber nodos indeseados que se deben evitar. A su vez, tanto los nodos como los arcos pueden tener cuantificaciones numéricas relacionadas con costos, beneficios, problemas asociados, etc. La estrategia básica para la resolución de problemas es la búsqueda, en este caso caminos o rutas que vayan de los estados iniciales a los estados deseados sin pasar por los estados indeseados. Estos caminos se pueden buscar con diferentes criterios (más corto, más económico, más rápido, o con funciones ad-hoc a minimizar).

Cuando el problema a solucionar tiene oponentes, pero se cumplen ciertas restricciones (información perfecta, determinismo, suma cero), el problema se conoce como búsqueda adversaria, y se puede modelar utilizando el mismo modelo formal. Si bien el problema en general es inabordable por fuerza bruta (búsqueda exhaustiva), existen algoritmos sofisticados que permiten encontrar soluciones en tiempo aceptable, aunque el espacio de búsqueda sea enorme. Para ello suelen diseñarse *heurísiticas*, es decir, reglas aproximadas que definen criterios generales para restringir la búsqueda a una fracción representativa de todas las posibilidades. Esta técnica es muy eficiente, aunque no infalible. Un ejemplo de ello son los programas que juegan al ajedrez o al go, en los cuales los espacios de búsqueda son astronómicos.

Cuando además de oponentes intervienen otros factores (por ejemplo, incertidumbre, azar, información incompleta, etc.) surge la temática asociada a la búsqueda en juegos, que no solamente permite modelar juegos como el poker o el backgammon, sino que también se utiliza como modelo formal para ciertas ramas de la economía de mercado, donde estos modelos fundamentan la llamada *teoría de la decisión racional*, que ha dado resultados merecedores de varios premios Nobel en Economía.

5.1.4 Planificación de tareas

Esta rama de la IA tiene como objetivo construir algoritmos de control que permitan a un agente sintetizar una secuencia de acciones que le lleve a alcanzar sus objetivos. Un problema de planificación requiere encontrar una secuencia eficiente de acciones para conducir a un sistema desde un estado inicial hasta un estado objetivo maximizando o minimizando una función objetivo. Se diferencia del tema anterior en el sentido de que las acciones que se pueden ejecutar modifican el espacio de estados. Otra diferencia relevante es que en planificación de tareas algunas acciones son *irreversibles* (a diferencia de lo que ocurre en otros contextos). Esto hace que las técnicas de búsqueda sean ineficientes o intratables, y se requiera una estrategia diferente para resolver, típicamente un enfoque jerárquico de descomposición de los objetivos generales en sub-objetivos parciales (ver Fig. 5.4).

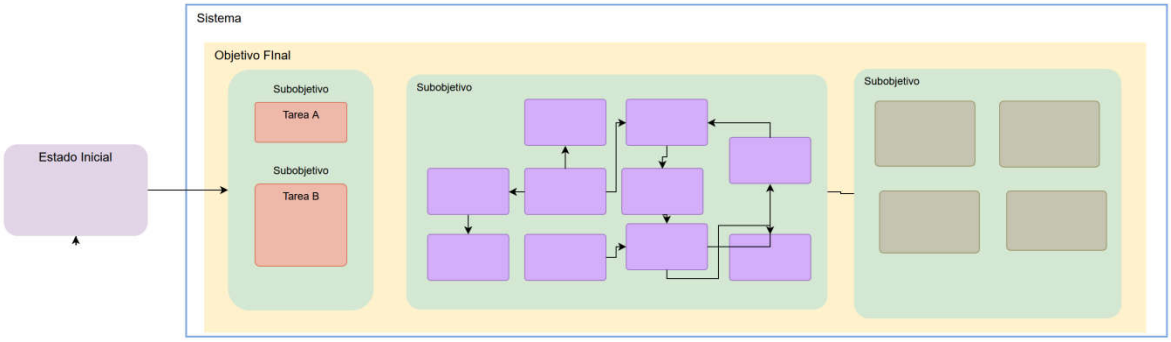

Figura 5.4: Descomposición de tareas por orden parcial y jerárquico (fuente: elaboración propia).

En problemas complejos se suceden múltiples interacciones entre objetivos parciales del problema porque las acciones que se utilizan para resolver un objetivo pueden interferir en la solución de otro objetivo parcial. Por ello se trabaja simultáneamente en una planificación con múltiples objetivos del problema. De este modo, el planificador tendrá que tomar decisiones sobre cómo las acciones que resuelven una parte del problema afectan a la resolución del resto del problema. A

este tipo de aproximación se le denomina planificación por orden parcial, que tiene la ventaja de ser flexible a la hora de establecer un orden entre las acciones que componen un plan, ya que se pueden primero tomar decisiones sobre las partes importantes del plan (en lugar de forzar un orden cronológico) y luego subdividir entre las acciones de cada una de estas partes.

Se trabaja entonces sobre los objetivos parciales del problema simultáneamente y se mantiene un *orden parcial* entre las acciones finales sin tener que comprometer un orden rígido entre las mismas hasta que ésto no sea indispensable. Esta descomposición reduce significativamente el espacio de búsqueda. Otra forma de optimizar la formulación de planes consiste en *proteger* subobjetivos parciales (aquellos que, una vez obtenidos no deben deshacerse).

5.1.5 Procesamiento del lenguaje natural

El objetivo de esta rama de la IA es poder comunicar las máquinas con las personas mediante el uso de lenguaje natural (español, inglés, etc.) y que las máquinas puedan comprender adecuadamente y eventualmente responder de acuerdo a lo requerido. Esto puede hacerse a través de diversos formatos (lenguaje escrito, oral, etc). Dependiendo del formato dado, se requiere un paso previo para extraer los elementos primarios (fonemas, tokens) hasta extraer una estructura lingüística básica, la que luego se representa por medio de gramáticas. En general las etapas se pueden dividir en los siguientes niveles de análisis:

5.1.5.1 Análisis morfológico/léxico. Análisis de las palabras que forman oraciones para extraer lemas, rasgos, unidades compuestas, etc.

5.1.5.2 Análisis sintáctico. Consiste en el análisis de la estructura de las oraciones de acuerdo con el modelo gramatical empleado.

5.1.5.3 Análisis semántico. Proporciona la interpretación de las oraciones (si se trata de enunciados, preguntas u otro tipo de actos de habla).

5.1.5.4 Incorpora el análisis del contexto de uso a la interpretación final. Aquí se incluye el tratamiento del lenguaje figurado (metáfora e ironía) como el conocimiento del mundo específico necesario para entender un texto especializado.

Estos diferentes niveles de análisis se aplicarán dependiendo del objetivo de la aplicación. Por ejemplo, un conversor de texto a voz no necesita el análisis semántico o pragmático. Pero un sistema conversacional o un chatbot requieren información muy detallada del contexto y del dominio temático.

5.1.6 Visión computacional

Es el área de la IA que desarrolla teorías y métodos para la extracción automática o semiautomática de información útil contenida en imágenes y videos. El objetivo es obtener y representar esa información a las máquinas de forma comprensible y útil en otros contextos. Los resultados tienen un sinnúmero de aplicaciones industriales, comerciales, científicas e inclusive artísticas. Los fundamentos de la visión computacional se encuentran en el procesamiento de imágenes, que utiliza técnicas como la manipulación del histograma de luminancias, la conversión a espacios cromáticos, el filtrado de ruido, el procesamiento por convolución, la morfología matemática y el procesamiento espectral, entre otras.

Este enfoque requiere una actividad previa de diseño (*feature engineering*, ver Fig. 5). Actualmente el aprendizaje profundo está teniendo un impacto muy grande en la visión computacional, dado que permite resolver problemas puntuales aprovechando el pre-entrenamiento de redes neuronales muy complejas y con bases de imágenes de entrenamiento de gran tamaño. Estos modelos son más eficaces que los tradicionales, al costo de ser opacos dado que no proveen capacidades explicativas que transparenten su funcionamiento.

Las tareas asociadas a la visión computacional pueden agruparse aproximadamente en el siguiente conjunto de propósitos:

5.1.6.1 Reconocimiento de objetos: Se basa en detectar en la imagen instancias de una clase semántica particular, abstrayéndose de detalles irrelevantes. Es uno de los procesos centrales para sistemas o modelos cognitivos de mayor complejidad.

5.1.6.2 Identificación de patrones: Puede pensarse como una articulación del reconocimiento de varios objetos, cada uno además con otros atributos (tamaño, posición, movimiento, etc.) lo cual permite identificar situaciones semánticamente más complejas.

5.1.6.3 Visión 3D: Se combinan dos o más imágenes o secuencia de video, y aplicando técnicas fotogramétricas se reconstruye la forma y tamaño 3D de objetos.

5.1.6.4 Reconocimiento de acciones: Detectar en una secuencia de video una instancia de una actividad semántica, abstrayéndose de detalles irrelevantes. Típicamente se aplica a movimientos humanos (por ejemplo, deportes) pero también para navegación autónoma o procesos industriales.

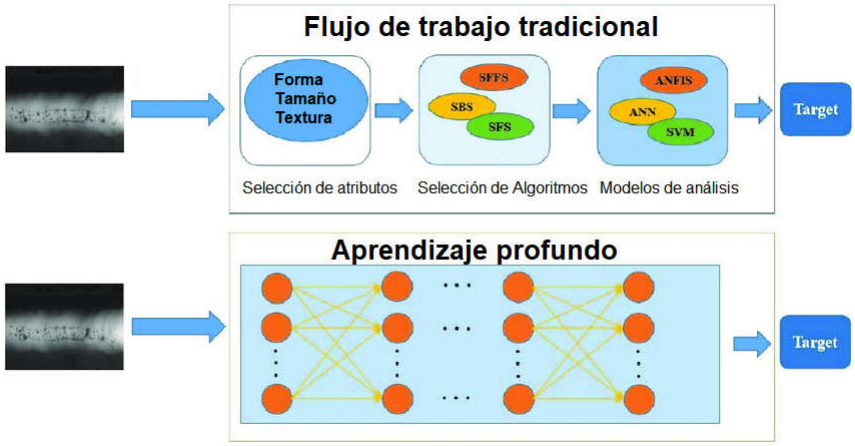

Figura 5:5 Modelo tradicional ("feature engineering") y modelo basado en aprendizaje de máquina para la visión computacional.

5.2 APRENDIZAJE DE MÁQUINA

5.2.1 Aprendizaje de Máquina: Breve introducción

El aprendizaje de máquina (ML) se refiere al campo de estudio en el que las computadoras son capaces de aprender por sí mismas sin ser programadas explícitamente. Esto permite a las computadoras modificar implícitamente su programación a partir de datos, así como de la realimentación que producen sus resultados. La diferencia fundamental entre ML e IA proviene de que los métodos de ML no intentan ser "semánticos" o simbólicos, sino que se basan en representaciones de bajo nivel conceptual. Esto permite por otro lado el diseño de algoritmos y flujos de trabajo de muy alta eficiencia que hacen posible el procesamiento masivo de grandes volúmenes de datos en tiempo real, y obtener de ellos información de alto valor.

El ML está revolucionando actualmente varios contextos científicos y empresariales. Entre estos últimos podemos mencionar específicamente el rol que el ML cumple como apoyo a la ciencia de datos en lo que se conoce como Big Data (grandes datos), que junto con otras tecnologías está modificando el panorama del datawarehousing empresarial, así como la gestión de los procesos industriales en la I4.0 (Industria 4.0). En ambos contextos (entre otros) el rol del ML es clave al poder automatizar el análisis y la extracción de información de conjuntos de datos que en general son heterogéneos, ruidosos y de muy gran tamaño. Las aplicaciones de ML cubren todo el rango de tareas para las cuales una programación estática sería poca práctica:

- No hay expertos humanos que sepan resolver la tarea.

- Hay expertos humanos, pero éstos no sabrían cómo describir su accionar.
- El contexto es abierto o se modifica.
- Hay muchas instancias (usuarios) de una misma tarea, cada una con restricciones locales diferentes.
- La información necesaria/disponible es copiosa/cambiante.

Los conjuntos de datos (datasets) pueden ser muy variados y heterogéneos, y dentro del ML se desarrollaron técnicas para poder limpiar, estandarizar y fusionar varios datasets y de esa forma extraer información latente que aisladamente no podía ser detectada. Este análisis, además de las fuentes de datos tradicionales, incluye datasets de muy difícil manejo como puede ser el texto libre (extraído de redes sociales, por ejemplo), audios, imágenes, video, etc.

5.2.2 Tipos de Aprendizaje de Máquina

Los métodos de ML se pueden catalogar como “supervisados” y “no supervisados” dependiendo de que se cuente o no con datasets “etiquetados” que permitan diseñar un entrenamiento por prueba y error. También es importante determinar si el objeto de análisis es una predicción numérica o se busca predecir una categoría. Eso permite determinar los tipos más frecuentes de modelos de ML (si bien hay otros más):

5.2.2.1 Clasificación: Es un modelo de aprendizaje supervisado en el que se busca predecir una categoría, por ejemplo, inferir el género de una persona a partir de una fotografía.

5.2.2.2 Regresión: Es un modelo de aprendizaje supervisado en el que se busca predecir una cantidad numérica, por ejemplo, el volumen de venta de un producto en un determinado período.

5.2.2.3 Clustering (agrupamiento): Es un modelo de aprendizaje no supervisado en el que se busca inferir categorías o grupos sin contar con pautas a-priori. Por ejemplo, segmentar la base de clientes de un comercio a partir de los comportamientos de cada cliente.

5.2.2.4 Reducción de dimensionalidad: Es un tipo de aprendizaje no supervisado en el que se desea simplificar un modelo con un gran conjunto de variables, obteniendo uno equivalente más simple y transparente. Por ejemplo,

encontrar un pequeño conjunto de factores relevantes que influencien los patrones de compra de los clientes.

5.2.2.5 Detección de atípicos: Es un modelo de análisis que puede ser supervisado o no supervisado, en el que se busca determinar si un dato en particular se comporta dentro de un determinado rango de variabilidad. Por ejemplo, para determinar si una transacción electrónica es legal o fraudulenta.

El resultado del análisis así efectuado normalmente se automatiza, dado que los modelos se basan en el uso de algoritmos de alta eficiencia, y pueden ser presentados a través de tableros de comando u otro tipo de métodos de comunicación. Estos resultados se denominan *analíticos* (analytics) y se pueden a su vez clasificar en cuatro categorías, de complejidad y valor crecientes para la empresa o industria:

Analíticos *descriptivos*, para responder preguntas sobre eventos que ya han sucedido. Las preguntas de ejemplo pueden incluir:

- ¿Cuáles son los datos de venta de los últimos 12 meses?
- ¿Cuál es el número de llamadas de soporte según la clasificación de la severidad y la ubicación geográfica?
- ¿Cuál es la comisión mensual ganada por cada agente de ventas?

Alrededor del 80% de los analíticos son de carácter descriptivo.

Los **analíticos de *diagnóstico*** tienen por objeto determinar la causa de un fenómeno que ocurrió en el pasado, utilizando preguntas enfocadas en descubrir la razón subyacente al evento. Algunas preguntas pueden ser:

- ¿Por qué las ventas del segundo semestre fueron menores que las del primero?
- ¿Por qué hubo más llamadas solicitando soporte técnico provenientes de la región este que de la región oeste?
- ¿Por qué hubo un incremento en las tasas de readmisión de pacientes durante los últimos tres meses?

Los analíticos de diagnóstico proporcionan más valor que los descriptivos, requiriendo modelos más avanzados. Por lo general requieren recoger datos de múltiples fuentes y almacenarlos en una estructura que se preste para realizar drill-downs y roll-ups. Los resultados suelen presentarse con herramientas de visualización interactivas que permiten a los usuarios identificar tendencias y patrones.

Los **analíticos *predictivos*** intentan determinar el resultado de un evento que podría producirse en el futuro. Las preguntas generalmente son formuladas utilizando un «qué pasaría si» lógico, como las siguientes:

- ¿Cuáles son las chances que un cliente se declare moroso al pago de un préstamo si ya ha se ha retrasado en un pago mensual?
- ¿Cuál será la tasa de venta del producto B si hay un aumento de precio en el producto A?
- Si un cliente compró productos A y B, ¿cuáles son las chances de que el cliente también compre el producto C?

Los analíticos predictivos intentan predecir eventos futuros, y las predicciones están basadas en patrones, tendencias y excepciones encontradas en datos históricos y actuales. Esto puede llevar a la identificación de riesgos y oportunidades, implicando el uso de grandes datasets (compuestos tanto por datos internos como externos), y modelos mucho más elaborados que los necesarios para los analíticos anteriores.

Los **Analíticos *prescriptivos***, finalmente, típicamente se construyen sobre los resultados predictivos, prescribiendo acciones que deben tomarse en función de objetivos empresariales (maximizar una oportunidad, minimizar un riesgo, etc.). El foco está en la opción prescripta a seguir, y por qué y cuándo debe ser seguida, para obtener ventajas o mitigar riesgos. Las preguntas de ejemplo pueden ser:

- ¿Cuándo es el mejor momento para comercializar un stock en particular?
- ¿Qué acción tomar si los competidores anuncian un nuevo producto en el mercado?
- ¿Qué campaña de marketing sería adecuada frente a un cambio de contexto (por ejemplo, un nuevo ASPO)?

En este tipo de análisis se analizan diversos resultados posibles, y se sugiere el mejor curso de acción para cada uno. El enfoque cambia de explicativo a consultivo y puede incluir la simulación de varios escenarios. Se incorporan (además de datos internos como ventas históricos y actuales, información del cliente, datos del producto, reglas de negocio) datos externos de varios tipos, (datos de redes sociales, datos del clima, datos demográficos), y también las reglas de negocio. Los prescriptivos proporcionan más valor que cualquier otro tipo de analíticos, y correspondientemente requieren desarrollos y habilidades más avanzadas, así como una alta especialización en herramientas y software.

5.2.3 Clasificadores

Este tipo de modelos infiere una función que establece una correspondencia entre las entradas y las salidas deseadas, las cuales son categorías o valores nominales. La base de aprendizaje del sistema está formada por ejemplos etiquetados (por eso es aprendizaje supervisado). Este tipo de aprendizaje puede llegar a ser muy útil en un sinnúmero de contextos. Por ejemplo, predecir un tipo de

falla a partir de las condiciones contextuales. Existe un buen arsenal de métodos de clasificación, entre los cuales se cuentan K vecinos más cercanos, regresión logística, clasificador Bayesiano, máquinas de soporte vectorial, perceptrones multicapa, árboles de decisión, etc.

En la clasificación tenemos tres etapas, entrenamiento y validación (con datos conocidos y en el contexto de aprendizaje) y testeo (con datos desconocidos, y en el contexto de aplicación). En la primera etapa, el algoritmo específicamente elegido es sometido a los casos de entrenamiento provistos, y con ello modifica implícitamente su programación para adaptarse a ellos. Esto permite validar los resultados utilizando parámetros específicos de evaluación de calidad (precisión, exactitud, sensibilidad, especificidad, etc.).

Existe una serie de estrategias para detectar y mitigar algunos problemas que pueden surgir durante este proceso, como ser el sobreajuste o el subajuste. Si la performance no fuese adecuada, se pueden adoptar estrategias extra de aprendizaje por ensamble (boosting y bagging). Finalmente, el modelo es testeado con datos hasta el momento no utilizados, para garantizar que el clasificador no fue influenciado por ellos. Existen además técnicas de *regularización*, que controlan la complejidad del modelo resultante sin que éste pierda necesariamente poder predictivo.

5.2.4 Regresores

En la regresión, el resultado esperado es una predicción numérica o cuantitativa. Por ejemplo, predecir el tiempo esperado de ocurrencia de una falla a partir de datos contextuales. Esto modifica algunos aspectos de los algoritmos subyacentes y de su entrenamiento, pero en general se siguen las etapas que vimos en clasificación (entrenamiento, validación, testeo). El modelo básico es la regresión lineal (uni o multivariada), que se generaliza fácilmente a regresiones polinomiales o con otro tipo de funciones. Asimismo, muchos de los algoritmos para clasificación se adaptan muy eficazmente para la regresión, incluyendo K vecinos más cercanos, árboles de regresión, máquinas de soporte vectorial y perceptrones. Finalmente, también son aplicables los métodos de ensamble de modelos.

A diferencia de lo que ocurre con clasificación, los parámetros de calidad del modelo evalúan diversos aspectos del error numérico de la predicción, por ejemplo, el error cuadrático medio o el error medio de la estimación. También se tiene en cuenta la calidad del ajuste (lineal o polinomial) al conjunto de entrenamiento, por lo que se utilizan parámetros estadísticos como por ejemplo el coeficiente de regresión. También se aplican técnicas de regularización de modelos, aunque éstas difieren sustancialmente de la regularización de clasificadores.

5.2.5 Clustering

Se conoce como “aprendizaje no supervisado”, porque las categorías o clases subyacentes son desconocidas (o inexistentes). El objetivo consiste en agrupar el conjunto de datos o registros (tradicionalmente denominados *patrones*) en agrupamientos (o clases), por algún criterio particular (similitud, proximidad, etc.). La similitud o proximidad se determinan por algún criterio de distancia dentro del espacio de atributos. Al igual que en los otros tipos de ML, los atributos nominales requieren tratamiento especial para ser representados en el espacio de atributos. El espacio de búsqueda de este problema es enorme: $m^k/m!$ para k datos en m clases. Por lo tanto, se requieren algoritmos heurísiticos de buen desempeño, tanto en eficiencia computacional como en calidad de resultados.

El algoritmo más difundido se conoce como k-medias (k-means). Se basa en iterar asignaciones de centroides a las clases, seguidas de una etapa de ajuste. Es un algoritmo aglomerativo, con k conocido o asignado arbitrariamente. Se repiten dos pasos: (i) asignar cada dato al grupo cuyo centroide sea más cercano, y (ii) recalcular los centroides de cada grupo. Existen teoremas que muestran que este algoritmo converge siempre, aunque esta convergencia depende de las condiciones iniciales.

Las aplicaciones del clustering en ciencia de datos son innumerables. Es la metodología básica para inferir patrones o regularidades en datos “opacos”. Por ejemplo, inferir grupos de usuarios a partir de su comportamiento, agrupar tipos de fallas en función de los valores de funcionamiento en cada falla, determinar grupos de conductas a partir del registro de los movimientos, etc. Uno de los problemas fundamentales es la determinación de la calidad de un resultado. A diferencia de lo que ocurre con regresión o clasificación (donde hay casos de entrenamiento supervisados), el clustering genera información nueva para la cual no hay un método de contraste adecuado. Las medidas de calidad entonces son indirectas, referidas a la integridad, robustez y coherencia de los clusters obtenidos.

5.2.6 Reducción de dimensionalidad

Trabajar con datasets muy *anchos* (espacios de atributos de muchas dimensiones) genera una gran cantidad de problemas, más allá del costo computacional. Esto se conoce como la “maldición de la dimensionalidad”. Al aumentar la dimensionalidad del espacio, la densidad de los datos baja exponencialmente, por lo que la estabilidad y robustez de las técnicas de análisis vistas más arriba se compromete, y la significatividad estadística de los resultados se debilita. Se requiere recolectar una enorme cantidad de datos de entrenamiento para garantizar un cubrimiento razonable de los posibles casos.

Además, cuando la cantidad de dimensiones es muy alta las métricas de distancia pierden las propiedades intuitivas, y los métodos basados en distancias (k-NN, por ejemplo) se sesgan muy rápidamente. Por otro lado, durante las tareas de curado del dataset y análisis exploratorio, muchas veces es necesario visualizar los datos de alguna manera razonablemente accionable. Finalmente, datos con muchos atributos tienen mucha probabilidad de tener atributos irrelevantes, valores faltantes o contaminados, etc., por lo que se generan modelos con mucha variancia y el sobreajuste es difícil de controlar.

La reducción de la dimensionalidad consiste en transformar el dataset original en otro de menor cantidad de atributos (tablas “más angostas”), pero que retenga las propiedades significativas del original con respecto al propósito de análisis. Muy crudamente, estos métodos pueden catalogarse en lineales y no lineales, y varias de las técnicas que vimos más arriba pueden utilizarse directa o indirectamente para reducir la dimensionalidad. Si bien estas técnicas no son específicamente un modelo de análisis, muchas veces una reducción de dimensionalidad adecuada produce resultados útiles y valiosos (por ejemplo, identificar los atributos más relevantes, o agrupar los datos respecto de una baja cantidad de atributos).

Entre los métodos “lineales” podemos mencionar como más útiles la selección de atributos, que consiste buscar exhaustivamente todas las combinaciones de subconjuntos de atributos y testearlas respecto de alguna métrica de performance, lo cual es intratable. Por ello suelen utilizarse árboles de decisión (ya vistos) si los datos están etiquetados. Caso contrario, es más común y factible emplear extracción de atributos (por ejemplo, a través de análisis de componentes principales) para “comprimir” el dataset dentro de un nuevo espacio de atributos, que conserve la mayor cantidad posible de factores y características del dataset original, y así poder aplicar clustering sin los problemas mencionados arriba.

Respecto de los métodos “no lineales”, muchas veces permiten una reducción agresiva de la cantidad de dimensiones (por ejemplo, pasar de cientos a solo dos o tres) y de esa manera el resultado ya constituye implícitamente un agrupamiento que muchas veces es útil o adecuado. Entre las técnicas más usuales en este contexto podemos mencionar los mapas auto-organizados, métodos basados en divergencias estadísticas (t-SNE) o en análisis topológico (UMAP).

5.2.7 Detección de atípicos

La detección de valores atípicos es el proceso de encontrar datos que son significativamente diferentes o inconsistentes con el resto de los datos en un dataset dado en base a algún criterio. Esta técnica de machine learning es usada para identificar anomalías, anormalidades y desviaciones que pueden ser ventajosas (tales como oportunidades) o desventajosas (tales como riesgos). La detección de valores atípicos está relacionada con la clasificación y el clustering (sea, respectivamente,

que tengamos datos supervisados o no), aunque sus algoritmos se centran en encontrar casos anormales. Las aplicaciones de la detección de valores atípicos son innumerables, incluyendo detección de fraudes, diagnóstico médico, análisis de datos de redes, análisis de datos de sensores, etc.

La detección de atípicos puede basarse en cualquiera de los modelos que ya vimos. Por ejemplo, en regresión, un atípico podría ser un valor que se encuentra muy lejos del modelo de ajuste previamente entrenado para los datos supervisados. En un clasificador, podría ser un dato que no cuadra adecuadamente con ninguna de las clases establecidas. Finalmente, en clustering podría ser un dato que se encuentra muy lejos de todos los centroides de los clusters que se formaron al entrenar el modelo.

Estas técnicas sencillas funcionan adecuadamente únicamente con datos atípicos individuales (lo que constituye un caso relativamente raro). Para situaciones más complejas, que involucran atípicos contextualizados, o grupos de casos que individualmente son normales, pero colectivamente son atípicos, se requieren técnicas de análisis más elaboradas.

5.2.8 Filtrado

El filtrado es un proceso automatizado de encontrar datos pertinentes dentro de un dataset (típicamente de gran tamaño) en función de criterios de búsqueda o de análisis. Por ejemplo, en un sistema de recomendación para venta on-line, los artículos a ofrecer pueden ser filtrados o bien basados en un comportamiento propio del cliente o buscando coincidir con el comportamiento de múltiples clientes similares. El filtrado generalmente es aplicado a través de los siguientes dos enfoques: el filtrado colaborativo y el filtrado basado en el contenido.

El filtrado colaborativo es una técnica basada en la colaboración (fusión) de casos pasados con el caso actual. De esa manera, cada registro del dato actual es utilizado implícitamente como un parámetro de selección dentro del dataset para seleccionar otros registros que tengan valores similares (en base a ciertos criterios). Finalmente, entre los registros seleccionados se puede realizar una clasificación, regresión o agrupamiento y con el resultado obtenido se obtiene la predicción que se desea.

Por ejemplo, se cuenta con un dataset del comportamiento de diversos sensores y máquinas durante un determinado proceso industrial. Se desea determinar el mejor curso de acción dentro de una determinada situación. Para ello se utilizan los datos históricos como dataset de consulta, y la o las condiciones actuales como criterio de filtrado colaborativo, y se seleccionan casos históricos que podrían aportar valor a determinar el objetivo deseado. El filtrado colaborativo se basa únicamente en

la similitud entre comportamientos o situaciones pasadas y la actual, y requiere una gran cantidad de datos históricos para obtener predicciones valiosas.

El filtrado basado en el contenido, en cambio, utiliza un modelo más semántico o estructurado. El dataset histórico no requiere ser muy grande, pero sí requiere estar organizado en esquemas de mayor estructura, utilizando ontologías o marcos conceptuales. El filtrado en este caso busca una analogía entre la situación actual y algunas de las históricamente registradas, y realiza predicciones en función de algún criterio de análisis.

5.2.9 Analíticos visuales

La información que se obtiene del análisis de datos es crucial, pero encontrar la manera de hacer análisis y descubrimientos significativos basados en analíticas no es tan fácil o elocuente como parece. Si bien la ciencia de datos y el ML pueden verse como un avance respecto de los antiguos reportes ad-hoc del data warehousing empresarial, en ciertos aspectos seguimos teniendo problemas similares, fundamentalmente la necesidad de tener especialistas entre los usuarios finales y los datos, para traducir los requerimientos de información a las tareas de análisis correspondientes, y para interpretar los resultados. Y siempre se tiene dificultad para traducir estos resultados en decisiones que aprovechen el análisis de datos.

El viejo adagio de que “una imagen vale mil palabras” podría aplicar a la ciencia de datos diciendo que los datos por sí mismos no tienen sentido si no están representados de manera elocuente. El propósito de la representación visual de la información es facilitar el entendimiento, y si la representación es accionable (interactivamente modificable para formular análisis exploratorio y confirmatorio) entonces estamos en el territorio de los llamados analíticos visuales. Las visualizaciones aprovechan las capacidades superlativas del cerebro humano para captar y digerir grandes cantidades de datos de una manera integral y comprensible. Al representar visualmente la información, si esto se hace de manera adecuada, la convertimos en un mapa que se puede explorar y recorrer visualmente. Y si la visualización es además interactiva, entonces la representación de ese mapa se puede adaptar, personalizar y modificar arbitrariamente. Cuando estamos perdidos en la información, un mapa de estas características es sobresalientemente útil.

En la figura 5.6 mostramos esquemáticamente algunos de los elementos fundamentales de las analíticas visuales, y cómo éstos modifican el territorio tradicional del análisis de datos. El recorrido “inferior”, pasar de los datos al conocimiento a través de modelos, representa las actividades tradicionales (refinamiento de los datos por medio de preparación y limpieza, minería de datos para proponer modelos, refinamiento de los modelos para optimizar sus parámetros, y finalmente la generación de conocimiento a través de procesos cognitivos conceptuales o semánticos). El recorrido “superior” genera visualizaciones a partir de

los datos utilizando un “mapeo visual” (por ejemplo, un scatter-plot, o un mapa de calor, por mencionar algunos de los mapeos más populares). La visualización es interactiva, permitiendo tareas exploratorias como el “drill-down”, el filtrado, la agregación, el “roll-up” y otras tareas.

Finalmente, a través de procesos cognitivos visuales (que son más inmediatos y holísticos) se genera conocimiento y comprensión por parte de los usuarios. En ambos caminos, siempre hay una realimentación entre el conocimiento y los datos, que básicamente implica que frente a los resultados (visuales o conceptuales) si la tarea no fue completa o satisfactoria, se buscan nuevos datos. Es interesante también la interacción que plantean las analíticas visuales con el análisis tradicional a través de la interacción entre los pasos intermedios (modelos y representación visual), dando origen a tareas relevantes como pueden ser la visualización de los modelos (y no de los datos) y la elaboración de modelos de visualización (y no de los datos).

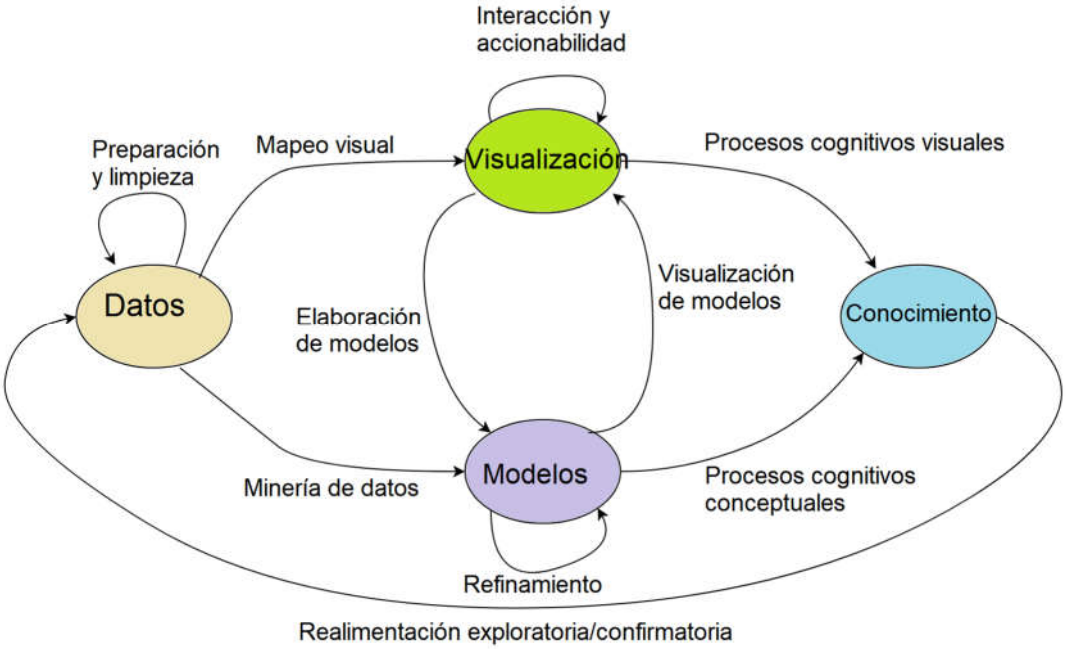

Figura 5.6: Una posible conceptualización del análisis visual de datos y su relación con el análisis de datos tradicional (fuente: elaboración propia).

5.3. MINERÍA DE DATOS Y METAHEURÍSTICAS

5.3.1 Introducción

Además de los métodos más formales de la IA clásica, y más numéricos del aprendizaje de máquina, podemos mencionar como dos grupos de temas relacionados, pero no propiamente incluidos a la minería de datos y a las

metaheurísiticas. Ambos grupos de temas se caracterizan por tener ámbitos de aplicación específicos.

5.3.2 Minería de datos

La minería de datos (MD) o *exploración de datos* (también *Knowledge Discovery in Databases* o KDD) es un grupo de procesos que intenta descubrir patrones, tendencias, correlaciones y estructuras ocultas o subyacentes en grandes volúmenes de conjuntos de datos. Utiliza métodos de estadística, IA y ML, pero con un objetivo específico, que es extraer información de un conjunto de datos y transformarla en una estructura comprensible para su aprovechamiento posterior. Está muy influenciada y moldeada por la gestión de los motores de bases de datos en el Data Warehousing SQL de los últimos 30 años en los aspectos de gestión de datos y del procesamiento en bases de datos, así como las métricas de aprovechamiento en la inteligencia de negocios tradicional. Como tal, en los últimos 10 años experimentó grandes cambios relacionados con la irrupción de las tecnologías de Big Data y sus "múltiples V (Volumen, Velocidad, Variedad, Verosimilitud, Valor), el Data Warehousing parcialmente vinculado a SQL (NoSQL y sistemas distribuidos), y a los analíticos en tiempo real.

La MD fue un "buzzword" las dos últimas décadas del siglo pasado, y como tal fue utilizado "laxamente" para referirse a cualquier forma de gestión de datos a mediana escala, el procesamiento de la información (recolección, extracción, almacenamiento, análisis y estadísticas), y hasta a cualquier tipo de sistema informático de apoyo a la toma de decisiones. De ahí el solapamiento (y la confusión) con inteligencia artificial, aprendizaje automático y las disciplinas asociadas. La clave en la MD es el *descubrimiento*, que comúnmente se acepta como la *detección de algo nuevo*, siempre que este descubrimiento luego pueda explotarse dentro de la inteligencia de negocios de la empresa o institución.

La tarea de la MD el análisis asistido o automático o de grandes datos (típicamente relacionales, aunque contemporáneamente se incluyen también los datos no estructurados y los datos semiestructurados). Se busca extraer patrones, tendencias, relaciones o estructuras desconocidas hasta el momento, y que tengan algún valor o utilidad. Entre las técnicas más utilizadas podemos mencionar la correlación y técnicas estadísticas similares para encontrar relaciones entre grupos de variables, el clustering para encontrar patrones o estructuras similares entre grupos de registros, la detección de atípicos, anomalías, e imputación de valores faltantes, y el análisis de dependencias por reglas de asociación. Estas técnicas son en general parte del material ya visto en IA y ML, pero lo que las distingue como MD es su contexto de aplicación.

Otra característica distintiva de la MD es que los resultados de una etapa de análisis típicamente son utilizados como insumo de etapas ulteriores. Por ejemplo, un

paso previo para identificar grupos en los registros (por clustering) luego puede ser utilizado para obtener otros resultados. Las etiquetas de los cluster obtenidos se agregan a los registros como un atributo más, y eso permite otros tipos de procesos más precisos de predicción (por ejemplo, perfilar clientes a través de procesos de clasificación). Finalmente, estos perfiles se pueden utilizar con un mecanismo de regresión para generar predictores en un sistema de recomendación.

Procesos asociados a la MD son la colección y recolección de datos (a veces por *harvesting* y *scraping*), el *munging* y *wrangling* de datos (cuyo objetivo es obtener tablas relacionales o por lo menos bien curadas), la imputación (para corregir valores faltantes o corruptos), la normalización, codificación, validación y estructuración de datos. Estos procesos no son de suyo MD, pero están definidos y relacionados fuertemente con ella.

5.3.3 Metaheurísticas

Las metaheurísticas (MH) son métodos computacionales propuestos para resolver problemas que no tienen algoritmos adecuados o satisfactorios, o bien cuando estos algoritmos existen, pero son computacionalmente intratables. La mayoría de las metaheurísticas tienen como objetivo resolver problemas de optimización en espacios de búsqueda de complejidad combinatoria, encontrando soluciones satisfactorias en tiempos bajos. Como afortunadamente muchos problemas del mundo real pueden conceptualizarse dentro de este tipo de formulaciones, eso hace que las MH tengan un amplio espectro de aplicaciones en el mundo real, como por ejemplo el problema del viajante (camino óptimo) u otros similares. Esto hace que las MH se relacionen con otros aspectos de la IA y el ML, aunque su campo de aplicación es casi específicamente el de la optimización.

El objetivo de una optimización es encontrar una representación (por ejemplo, un *feature vector* como en ML, pero en este contexto se lo denomina *vector de estado*) que minimice una función de costo u objetivo. El posible conjunto de vectores de estado determina un espacio de búsqueda, y es posible modificar un estado a través de una o más transiciones. Las MH exploran este espacio de búsqueda en forma serial o paralela manteniendo uno o más estados en diferentes vectores y aplicando una o más transiciones a cada uno. Si bien la estrategia directa sería aplicar la transición que lleve al estado de menor costo posible, esto tiene problemas a veces sutiles, por lo que se aplican diferentes técnicas probabilísticas. La variedad de maneras de aplicar estas técnicas es la que da origen a las MH más conocidas, como por ejemplo la programación evolutiva, los métodos de enjambre, la búsqueda tabú, etc.

Como ejemplo notorio de MH, la programación evolutiva propone una solución inspirada en la evolución natural. Ésta se basa en mantener una población de estados, de los cuales “sobreviven los más aptos” (es decir, una proporción de los

estados que tengan el menor costo sobrevive, y el resto es descartado). Los estados sobrevivientes, a su vez, tienen la posibilidad de generar “descendencia” a través de un proceso de copia del vector de estados a la que se aplican cambios aleatorios similares a las mutaciones genéticas, el crossover (reproducción) y demás procesos. Estas dos etapas, supervivencia y reproducción, se iteran hasta encontrar un vector de estados satisfactorio. Dado que el número de operaciones puede ser muy grande, normalmente este tipo de MH incluyen parámetros de terminación (por tiempo máximo, cantidad máxima de iteraciones, falta de convergencia hacia una solución, etc.)

5.4. APRENDIZAJE PROFUNDO

5.4.1 Introducción

El aprendizaje profundo (DL por “deep learning”) es una clase de algoritmos ideados para el aprendizaje automático basado en datos. Se basa en redes neuronales de una gran cantidad de capas y diferentes arquitecturas de conectividad. Por ello los modelos tienen una enorme cantidad de grados de libertad y para evitar el overfitting se hace necesario contar con un alto volumen de datos de entrenamiento. Típicamente estas arquitecturas contienen una cascada de capas neuronales con unidades tipo perceptron o convolucional, seguidas de un procesamiento no lineal (RELU o Tanh).

Cada capa utiliza la salida de la capa anterior como entrada. A su vez, entre las capas perceptrón o convolucionales suelen agregarse capas de reducción o downsampling utilizando max-pooling. La definición de una arquitectura dada, por lo tanto, es muy libre y variada, aunque existen en la actualidad arquitecturas típicas y con algunas capas iniciales pre-entrenadas para propósitos específicos. Los algoritmos pueden utilizar aprendizaje supervisado o aprendizaje no supervisado, y las aplicaciones actuales son virtualmente ilimitadas.

5.4.2 Diferencias del Aprendizaje Profundo con las demás técnicas

El Dl se caracteriza por poseer un conjunto de propiedades distintivas (algunas ventajosas y otra no tanto) que lo diferencian del aprendizaje de máquina en general. Entre ellas podemos mencionar las siguientes:

- Como el modelo tiene muchas variables (cientos de miles de pesos entre las neuronas) se necesitan grandes cantidades de datos de entrenamiento para un correcto entrenamiento.
- Depende de máquinas rápidas dado que realiza un gran número de operaciones matemáticas sencillas, sobre todo durante el entrenamiento. Se utilizan

actualmente GPUs o virtualización en la nube para optimizar eficazmente estas operaciones.

- No requiere *feature engineering* dado que puede aprender automáticamente las características de bajo y alto nivel de los datos, y además crea nuevas características (opacas) automáticamente.
- Divide el proceso de aprendizaje en pasos más pequeños. Luego, combina los resultados de cada paso en una salida. Este proceso es normalmente "caja negra" (no es posible modelarlo ni interpretarlo fácilmente).
- Requiere muy alto costo computacional para entrenarse cuando los modelos tienen muchas capas. El modelo entrenado, en cambio, puede ser ejecutado con mayor rapidez
- La salida no es necesariamente un valor numérico o una clasificación como en el ML, sino que puede tener varios formatos, como por ejemplo texto, imágenes o sonido.

5.4.3 Aplicaciones del aprendizaje profundo

Son virtualmente ilimitadas, siempre que existan datos de entrenamiento suficientes. A causa de la estructura de la red neuronal artificial, el aprendizaje profundo es excelente para identificar patrones, clasificar, o hasta generar elementos nuevos en datos no estructurados, como imágenes, sonido, vídeo y texto. Por esta razón, el aprendizaje profundo está transformando rápidamente muchos sectores, como la atención sanitaria, la energía, las finanzas y el transporte. Gracias a ello, estos sectores se están replanteando los procesos empresariales tradicionales. Mencionamos brevemente algunos ejemplos:

5.4.3.1 Reconocimiento de entidades con nombre. Es un método de aprendizaje profundo que toma un fragmento de texto como entrada y lo transforma en una clase especificada previamente. Esta nueva información podría ser un tópico literario, un código postal, una fecha o un identificador de producto. Asimismo, esa información se puede almacenar en un esquema estructurado para compilar una lista de direcciones, o puede servir como banco de pruebas de un motor de validación de identidades.

5.4.3.2 Detección/reconocimiento de objetos. Esto incluye entre otras, tareas como la clasificación de imágenes y la localización de imágenes. La clasificación de imágenes identifica los objetos de la imagen, como automóviles o personas. La localización de imágenes, por su parte, proporciona la ubicación específica de estos objetos. La detección de objetos ya se está usando en sectores como los videojuegos, los comercios minoristas, el turismo y los vehículos autónomos.

5.4.3.3 Etiquetado de imágenes. De forma similar a la tarea de reconocimiento de imágenes, la generación etiquetas (tags) para imágenes debe generar una descripción textual del contenido de la imagen. Una vez que puede detectar y etiquetar objetos en fotografías, el siguiente paso es convertir esas etiquetas en oraciones descriptivas.

5.4.3.4 Traducción automática. La traducción automática toma palabras u oraciones de un idioma y las traduce automáticamente a otro. Además de esta tarea, se está aplicando aprendizaje profundo en otras dos áreas específicas: la traducción automática de texto (y de voz a texto) y la traducción automática de imágenes. Con la transformación apropiada de los datos, una red neuronal es capaz de comprender texto, audio y señales visuales. La traducción automática se puede usar para identificar fragmentos de sonido en archivos de audio mayores y transcribir la palabra hablada o la imagen como texto.

5.4.3.5 Análisis de texto. Esta tarea implica el análisis de grandes cantidades de datos de texto (por ejemplo, documentos médicos o comentarios de usuarios), el reconocimiento de patrones y la creación de información organizada y concisa como resultado de dicho análisis. Las empresas usan el análisis de texto para un sinnúmero de objetivos (detectar negociaciones, minar “sentimientos” de los usuarios o lectores, detectar fraudes en seguros, etc.).

5.4.4 Tipos de redes neuronales artificiales

Como mencionamos más arriba, las redes neuronal artificiales se forman con capas de nodos conectados en un conjunto potencialmente ilimitado de arquitecturas. Los modelos de aprendizaje profundo usan redes neuronales que tienen un gran número de capas. Históricamente surgieron algunas arquitecturas más o menos genéricas que se volvieron populares y fueron estableciéndose como puntos de partida para exploraciones más complejas.

5.4.4.1 Red neuronal del tipo feedforward. Es el tipo más simple de red neuronal artificial. En una red de tipo feedforward, la información se desplaza solo en una dirección: desde la capa de entrada a la de salida. Las redes neuronales de tipo feedforward transforman una entrada pasándola por una serie de capas ocultas. Cada capa consta de un conjunto de neuronas, donde cada capa está totalmente conectada a todas las neuronas de la capa anterior. Por último, hay una última capa totalmente conectada (la capa de salida) que representa las predicciones generadas.

5.4.4.2 Red recurrente. Estas redes guardan la salida de una capa y la reenvían a la capa de entrada para poder predecir el resultado de esa capa. Las redes neuronales recurrentes tienen grandes capacidades de aprendizaje. Suelen utilizarse en tareas complejas, como la predicción de series temporales, el aprendizaje de escritura a mano y el reconocimiento de idiomas.

5.4.4.3 Red convolucional. Es un tipo especialmente eficaz para el procesamiento de imágenes y video. Las capas se organizan en tres dimensiones: ancho, alto y profundidad. Además, las neuronas de una capa no se conectan con todas las neuronas de la capa siguiente, sino que solo se conectan a una pequeña región de la misma a través de operaciones de convolución digital discreta (cuyos parámetros o kernels son entrenables). Las capas convolucionales además se intercalan con otras capas de reducción no lineal de dimensionalidad (max pooling). La salida, además, puede reducirse a un solo vector de probabilidades (softmax). Existen varias arquitecturas predefinidas (por ejemplo, VGG-16) con las capas iniciales pre-entrenadas para propósitos de visión computacional.

5.4.4.4 Red generativa antagónica (GAN). Son modelos generativos entrenados para crear contenido realista, como por ejemplo imágenes o sonido. Se compone de dos redes, conocidas como el generador y el discriminador. Ambas redes se entrenan simultáneamente. Durante el entrenamiento, el generador usa valores aleatorios para crear nuevos datos sintéticos que se parecen mucho a los datos reales. El discriminador toma la salida del generador como entrada y usa datos reales para determinar si el contenido generado es real o sintético. Las redes compiten entre sí. El generador intenta generar contenido sintético que no se pueda distinguir del contenido real y el discriminador intenta clasificar correctamente las entradas como reales o sintéticas. A continuación, la salida se usa para actualizar los pesos de ambas redes para ayudarles a alcanzar mejor sus respectivos objetivos. Las redes generativas antagónicas se usan para resolver problemas muy complejos, como por ejemplo la transformación de imagen a imagen (pix2pix), la transferencia de estilos, la super-resolución, etc.

5.4.4.5 Redes transformadoras. Son una arquitectura para resolver problemas que contienen secuencias, como texto o datos de serie temporal. Constan de capas de codificador y descodificador. El codificador toma una entrada y la asigna a una representación numérica que contiene información. El descodificador usa la información del codificador para generar una salida, como texto traducido. Se han usado exitosamente para resolver problemas de procesamiento de lenguaje natural, como la traducción, la generación de texto, la respuesta a preguntas y el resumen de texto.