

## ANÁLISIS COGNITIVO: INTELIGENCIA ARTIFICIAL

Inspirada por ciertos rasgos que caracterizan a la inteligencia humana, la **computación cognitiva** (*cognitive computing*) aplica, entre otras, técnicas de **inteligencia artificial** (**AI**, *Artificial Intelligence*), **visión por ordenador** (*computer vision*) o **procesamiento de lenguaje natural** (**NLP**, *Natural Language Processing*) con el fin de tratar grandes conjuntos de datos no estructurados y encontrar soluciones a problemas y necesidades complejas.

Estos sistemas suponen un salto cualitativo importante respecto a los sistemas programables, que dominaron la informática desde mediados de los años 50 del siglo pasado. La diferencia radica en que los sistemas cognitivos no tienen que ser programados de forma explícita, sino que pueden aprender a partir de interacciones y experiencias que les son proporcionadas en forma de conjuntos de datos. De alguna manera, la idea detrás de ellos es emular el proceso que llevamos a cabo las personas cuando tomamos decisiones: **observar, interpretar, evaluar y decidir**. Las personas, además, cerramos el ciclo al calibrar el éxito o el fracaso de la decisión mediante un proceso global denominado **aprendizaje**, que nos permite una afinación y mejora continua. Los modelos cognitivos, al igual que los predictivos que veíamos en capítulos anteriores, necesitan todavía en gran medida que un agente externo les guíe en esa calibración, pero la estrategia y el proceso a seguir es básicamente el mismo.

## 9.1 MOTIVACIÓN Y OBJETIVOS

Los siguientes son algunos ejemplos de los ámbitos de aplicación de los sistemas cognitivos:

- ▶ La transcripción automática del audio de una conversación entre una agente y un cliente, detectando el motivo de la consulta, el tono empleado por el agente, y si el cliente solucionó finalmente la incidencia.
- ▶ La síntesis de voz para la generación bajo demanda de audiolibros para personas con dificultades visuales.
- ▶ El reconocimiento automático de tumores en imágenes médicas.
- ▶ La interpretación de una noticia, extrayendo palabras clave, conceptos, entidades y relaciones, así como su traducción a diversos idiomas.
- ▶ La generación de contenido textual (informes, código de programación, correos electrónicos, etc.) y multimedia (imágenes, discursos, música, etc.) a partir de una serie de indicaciones.
- ▶ La elaboración de un perfil psicosociológico de un candidato a un puesto de trabajo a partir de sus publicaciones en redes sociales.
- ▶ La creación de agentes conversacionales, también denominados **asistentes virtuales** (*chatbots*), para la gestión de llamadas en centros de atención telefónica, sistemas de reserva o consulta.
- ▶ La implementación de sistemas de búsqueda integral, combinando fuentes heterogéneas y dispares en cuanto a formato, para la detección de relaciones complejas en investigación medioambiental.
- ▶ La recomendación del mejor tratamiento para un paciente, basado en su historial clínico, la experiencia en casos similares, y la literatura médica y farmacológica existente al respecto.

La relación podría ser más larga, pero estos ejemplos nos dan ya algunas pistas del propósito y la función de estos sistemas. Nos indican también que hablamos de aplicaciones analíticas, pero también operacionales, como los asistentes virtuales o los sistemas de conducción automática, basados estos últimos en tecnologías de visión artificial. Es por ello por lo que estamos hablando de forma genérica de sistemas cognitivos, con independencia de si su funcionalidad es analítica u operacional.

Una de las necesidades comunes de los ejemplos que acabamos de dar es la de **procesar fuentes no estructuradas**: texto, audio e imágenes. Este rasgo es un elemento característico del *Big Data*, pero también están presentes los otros, como el volumen, siendo fácil intuir los tamaños de los conjuntos de datos involucrados, la velocidad a la que estos se generan y, no menos importante, la veracidad alrededor de la generación

de contenidos. En este sentido, el rápido desarrollo del procesamiento cognitivo y la AI es una de las principales consecuencias del auge de las tecnologías de *Big Data* (y viceversa).

La Figura 9-1 muestra como la sinergia de distintos factores, a modo de capas concéntricas, ha posibilitado el desarrollo tan rápido de la AI en los últimos diez años. En primer lugar, el desarrollo tecnológico ha permitido el acceso a importantes recursos de computación en la nube de una manera eficiente, rápida y económica. La disponibilidad de soluciones de *Big Data* para el almacenamiento y procesado de grandes volúmenes de datos y su acceso mediante API, ha facilitado el desarrollo de aplicaciones descentralizadas en las que se combinan componentes de distintos proveedores. En un segundo nivel nos encontramos con la explosión del dato en cuanto a variedad y volumen. Las redes sociales han creado una cantidad tal de información que han hecho posible, en una tercera capa, el desarrollo y entrenamiento de una nueva generación de algoritmos sobre infraestructuras y plataformas asequibles en la nube.

Muchos de estos algoritmos, como las **redes neuronales artificiales** o los **algoritmos genéticos**, habían sido desarrollados años atrás, pero la falta de datos y recursos de cómputo hicieron disminuir su interés en aquel momento.

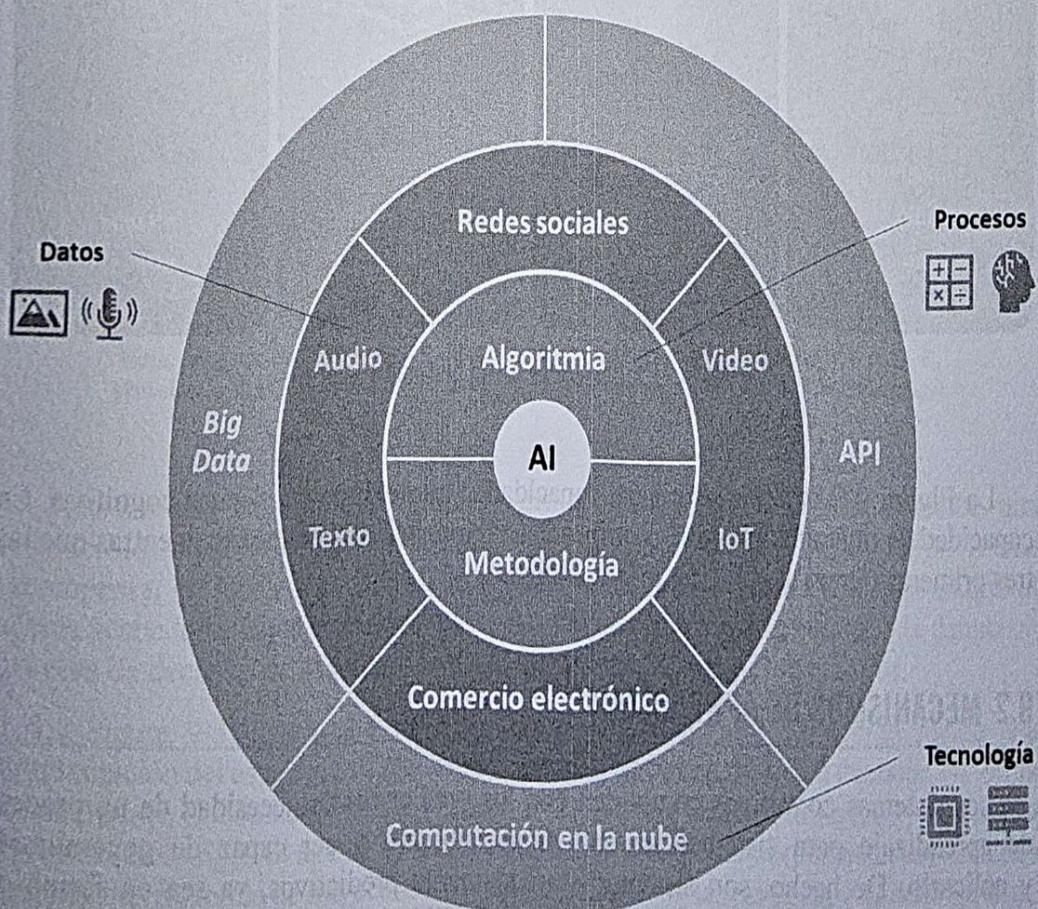


Figura 9-1. Posibilitadores en el desarrollo de la inteligencia artificial.

Aunque comparten muchos enfoques, estrategias de aprendizaje y una base algorítmica común, los sistemas de análisis cognitivo van más allá que los de análisis predictivo, con otros condicionantes marcados precisamente por la variedad y el volumen de los datos a manejar. A la vista de los ejemplos que hemos dado, los objetivos se ven también diferentes, siendo mucho más ambiciosos.

Por dichos objetivos, y también por el tipo de información que manejan, en muchos casos de alta sensibilidad, el empleo de sistemas cognitivos está siempre sujeto a una cierta controversia. No hay que perder de vista las importantes implicaciones que los sistemas de inteligencia artificial pueden tener en temas como la **privacidad**, la toma de decisiones de forma sesgada y poco explicada, o el posible riesgo vital alrededor de algunas de estas decisiones. Si a esto unimos el impacto que puede tener en las relaciones sociales y laborales, el debate ético y moral está servido, dando lugar a distintas **iniciativas regulatorias** y posicionamientos a favor o en contra de estas.



Figura 9-2. Capacidades de un sistema cognitivo.

La Figura 9-2 resume las cuatro capacidades que exhibe un sistema cognitivo. La capacidad de interacción está más enfocada a la aplicación operacional, mientras que las tres primeras tienen un carácter más analítico.

## 9.2 MECANISMOS DE APRENDIZAJE

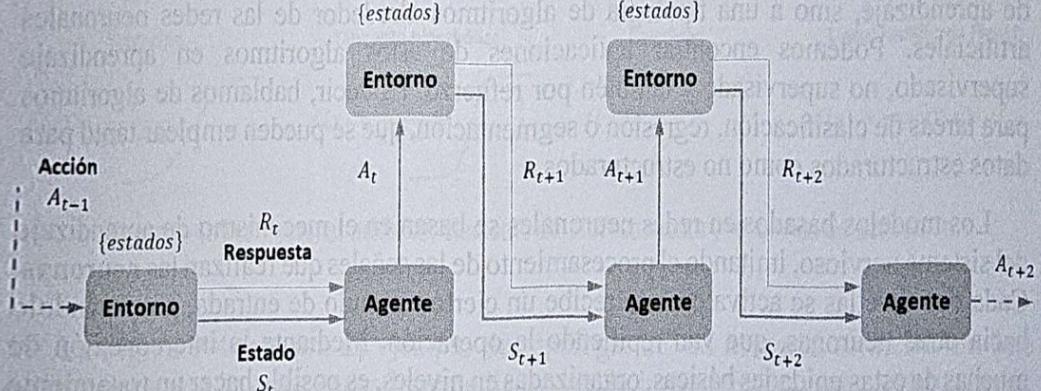
Los sistemas cognitivos comparten con los predictivos la necesidad de un proceso de aprendizaje para sistematizar un comportamiento y ser capaz de generalizarlo y aplicarlo. De hecho, son sistemas marcadamente predictivos, ya sea en forma de modelos de clasificación o regresión, aunque también los podemos encontrar en tareas de reconocimiento de patrones y segmentación. Por ejemplo, un sistema que identifica el tipo de suelo (urbano, industrial o agrícola) a partir de imágenes aéreas del territorio no

deja de implementar un modelo de clasificación. La diferencia radica fundamentalmente en el tipo de dato que hay que tratar y el algoritmo empleado para ello.

Vamos a ver a continuación los mecanismos principales de aprendizaje que podemos encontrar en los sistemas cognitivos. En realidad estos pueden seguir calificándose como **supervisados** o **no supervisados**, tal y como veímos en el Capítulo 7. Sin embargo presentan ciertas características que los hacen hasta cierto punto especiales.

### 9.2.1 Aprendizaje por refuerzo

El **aprendizaje por refuerzo** (*reinforcement learning*) es un tipo de aprendizaje no supervisado con mucha tracción en inteligencia artificial. La idea detrás de este paradigma es que un sistema, denominado **agente**, sea capaz de tomar la decisión adecuada mediante un aprendizaje basado en prueba y error.



**Figura 9-3.** Interacción entre un agente y su entorno en un proceso de Markov.

**Nota.** Adaptado de *Key Concepts of Modern Reinforcement Learning* [Figura], por Bisong, E., [ekababisong.org \(https://ekababisong.org/key-concepts-of-modern-reinforcement-learning/\)](https://ekababisong.org/key-concepts-of-modern-reinforcement-learning/).

Partiendo de un estado inicial, el agente emprende una acción y recibe una retroalimentación de su entorno. Esta puede consistir en una penalización o una recompensa, en función de la cual evolucionará hacia otro estado. La idea es que el agente acabe transitando a estados que maximicen la recompensa acumulada dentro un **proceso de decisión de Markov**, como el que vimos en el capítulo anterior.

La Figura 9-3 ilustra este proceso. En una interacción  $t$ , el agente recibe una representación del estado de su entorno ( $S_t$ ) y toma, en base a esta, una decisión en forma de acción ( $A_t$ ). Consecuentemente, el entorno le devuelve una respuesta ( $R_{t+1}$ ) y su nueva representación ( $S_{t+1}$ ). El agente va evolucionando por una sucesión de estados, acciones y respuestas en forma de trayectoria, donde los valores de la respuesta y el nuevo estado sólo dependen del estado anterior y la acción tomada sobre este. La función de respuesta, también denominada **función de recompensa**, vendrá definida por el problema a

resolver. En cualquier caso, la recompensa es un número real<sup>208</sup>, siendo el objetivo del agente la acumulación del mayor premio posible a lo largo de toda la trayectoria, no en el corto plazo de una iteración.

El aprendizaje por refuerzo se viene empleando con éxito en robótica y control industrial, con aplicaciones en sistemas de conducción autónoma, visión artificial o manipulación de mercancías. También en el procesamiento del lenguaje natural, tanto en la síntesis de texto como en la traducción. Los juegos de azar son otra área de gran aplicación y exhibición de estos sistemas. Hay que destacar también sus aplicaciones en *marketing*, tanto en la parte de sistemas de recomendación de productos como en la de personalización de la publicidad.

## 9.2.2 Aprendizaje profundo

El **aprendizaje profundo** (*deep learning*) no hace referencia a un nuevo paradigma de aprendizaje, sino a una tipología de algoritmos alrededor de las redes neuronales artificiales. Podemos encontrar aplicaciones de estos algoritmos en aprendizaje supervisado, no supervisado y también por refuerzo. Es decir, hablamos de algoritmos para tareas de clasificación, regresión o segmentación, que se pueden emplear tanto para datos estructurados como no estructurados.

Los modelos basados en redes neuronales se basan en el mecanismo de aprendizaje del sistema nervioso, imitando el procesamiento de las señales que realizan las **neuronas**. Cada una de ellas se activa cuando recibe un cierto estímulo de entrada, propagándolo hacia otras neuronas, que van repitiendo la operación. Mediante la interconexión de muchas de estas unidades básicas, organizadas en niveles, es posible hacer un tratamiento muy elaborado de la información.

Los algoritmos que emulan estas estructuras organizan **elementos de procesamiento**, el equivalente a las neuronas, en distintas **capas**, de forma que la información fluye a lo largo del arreglo. Además de la capa de entrada y la de salida, encargadas de recibir las observaciones y de dar el resultado, existen capas intermedias que permiten un cómputo más complejo. La topología de la red, incluyendo el número de capas, los elementos por capa y la forma de interconectar estos, marcará el tipo de red y sus posibles aplicaciones.

Es importante aclarar que el empleo de redes neuronales no se restringe al ámbito cognitivo y el tratamiento de datos no estructurados. Por el contrario, se han venido aplicando desde hace años en problemas, podríamos decir, más tradicionales de clasificación, predicción, segmentación, reducción de la dimensionalidad u optimización.

208 La recompensa a modo de premio será un número positivo, mientras que una penalización tendrá un valor negativo.

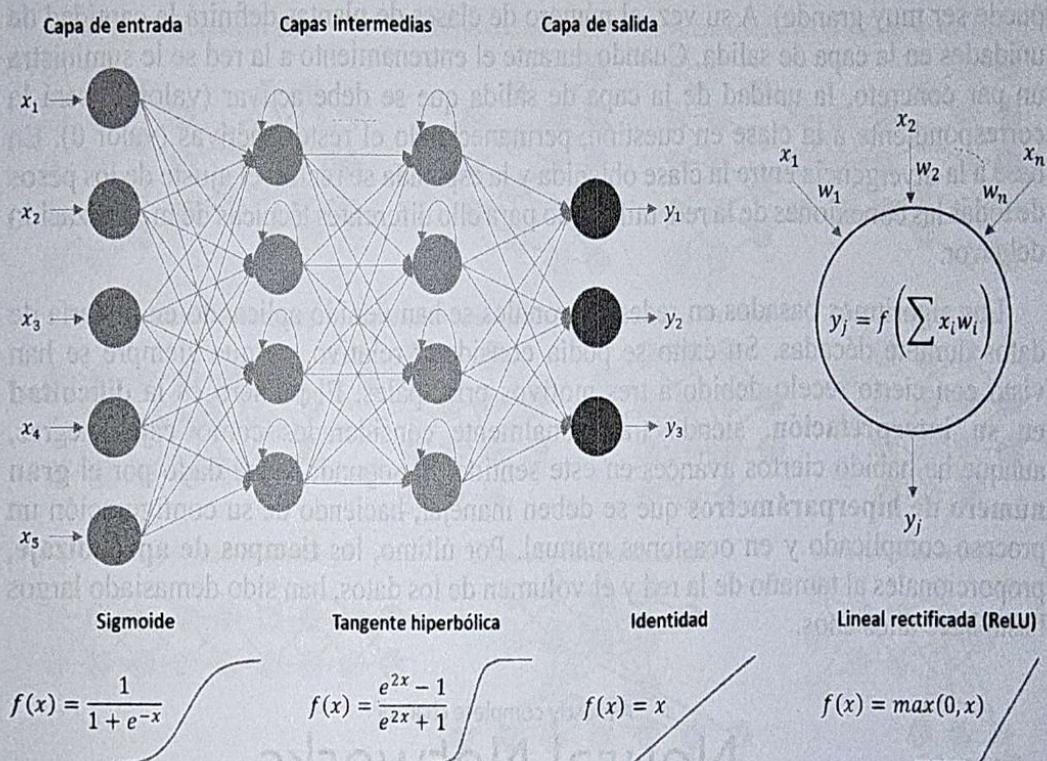


Figura 9-4. Red neuronal artificial, elemento de procesamiento y funciones de activación.

La Figura 9-4 muestra un ejemplo de topología de red neuronal artificial. En este caso vemos que hay dos capas intermedias y que la interconexión entre capas es total; es decir, todos los elementos de una capa reciben una señal de todos los elementos de la capa inmediatamente anterior. Además, los elementos de las capas intermedias tienen **conexiones recurrentes**, recibiendo como señal de entrada su propia salida. A la derecha de la figura tenemos el funcionamiento de cada elemento de procesamiento. Básicamente este consiste en una suma ponderada de las entradas, a la que se aplica una **función de activación**. El aprendizaje en estos modelos consiste en encontrar el valor de esos pesos de ponderación que minimizan una **función de coste** determinada. Lo normal es que todos los elementos que comparten la misma capa tengan la misma función de activación. Su elección dependerá del tipo de problema. En tareas de clasificación la capa de salida acostumbra a equiparse con funciones tipo escalón, como la sigmoide o la tangente hiperbólica, mientras que para regresión se utilizan funciones lineales. La función lineal rectificada, que también muestra la figura, se utiliza mucho en las denominadas **redes neuronales de circunvolución** (*CNN, Convolutional Neural Networks*), empleadas en el procesamiento de imágenes y del lenguaje natural.

Por ejemplo, en un problema de clasificación de imágenes de plantas, una red neuronal será entrenada dentro de un proceso de aprendizaje supervisado. Para ello se le suministrarán ejemplos representativos del problema, consistentes en pares de **imagen-clase**. La imagen de la planta vendrá representada por un conjunto de pixeles, cuyo número definirá el número de unidades en la capa de entrada a la red (el cual

puede ser muy grande). A su vez, el número de clases de plantas definirá la cantidad de unidades en la capa de salida. Cuando durante el entrenamiento a la red se le suministra un par concreto, la unidad de la capa de salida que se debe activar (valor 1) será la correspondiente a la clase en cuestión, permaneciendo el resto inactivas (valor 0). En base a la divergencia entre la clase obtenida y la esperada se realiza el ajuste de los pesos de todas las conexiones de la red, utilizando para ello diferentes técnicas de minimización del error.

Los algoritmos basados en redes neuronales se han venido aplicando en minería de datos durante décadas. Su éxito se podía considerar relativo, ya que siempre se han visto con cierto recelo debido a tres motivos principales. El primero es la **dificultad en su interpretación**, siendo tradicionalmente considerados como cajas negras, aunque ha habido ciertos avances en este sentido. El segundo viene dado por el **gran número de hiperparámetros** que se deben manejar, haciendo de su configuración un proceso complicado y en ocasiones manual. Por último, los **tiempos de aprendizaje**, proporcionales al tamaño de la red y el volumen de los datos, han sido demasiado largos hasta hace unos años.

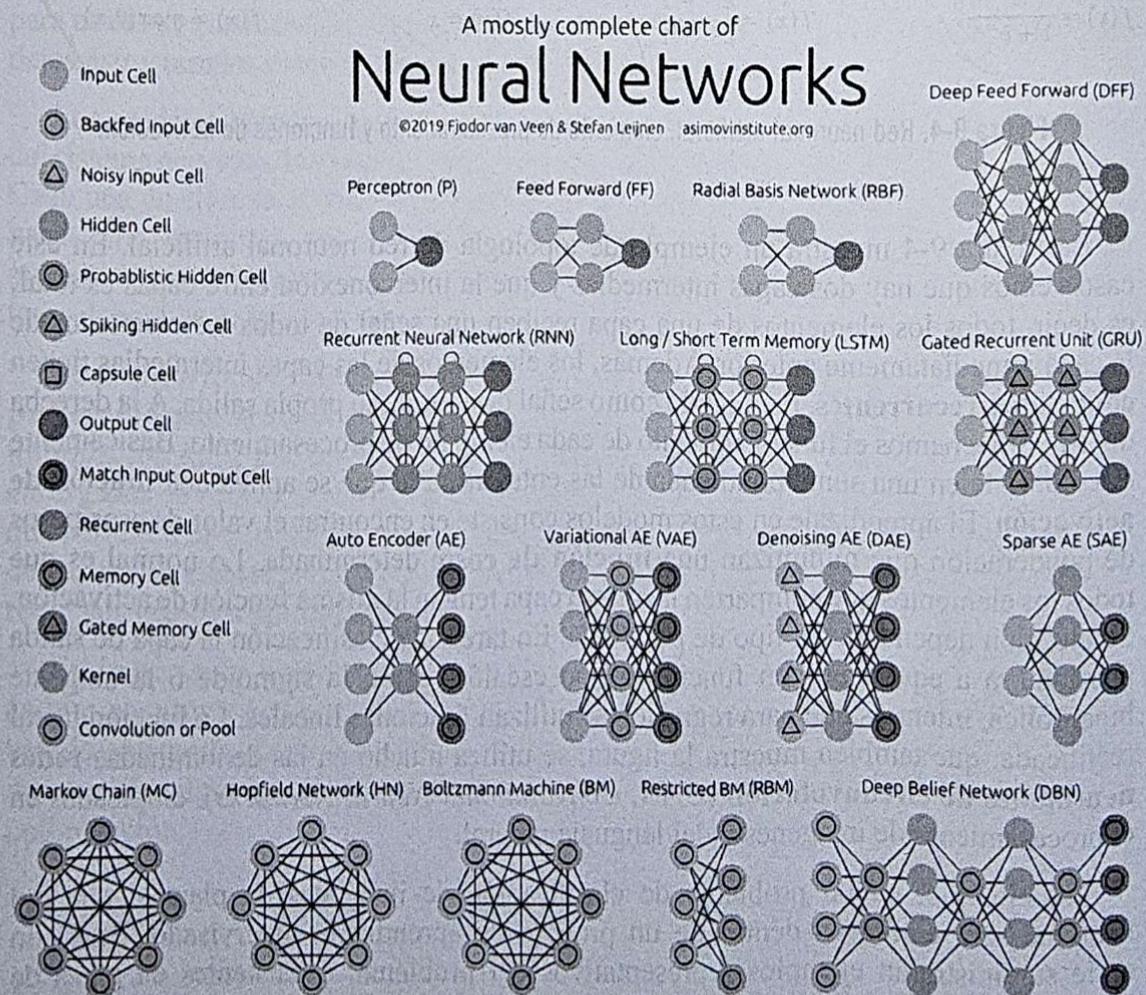


Figura 9-5. El zoo de las redes neuronales.

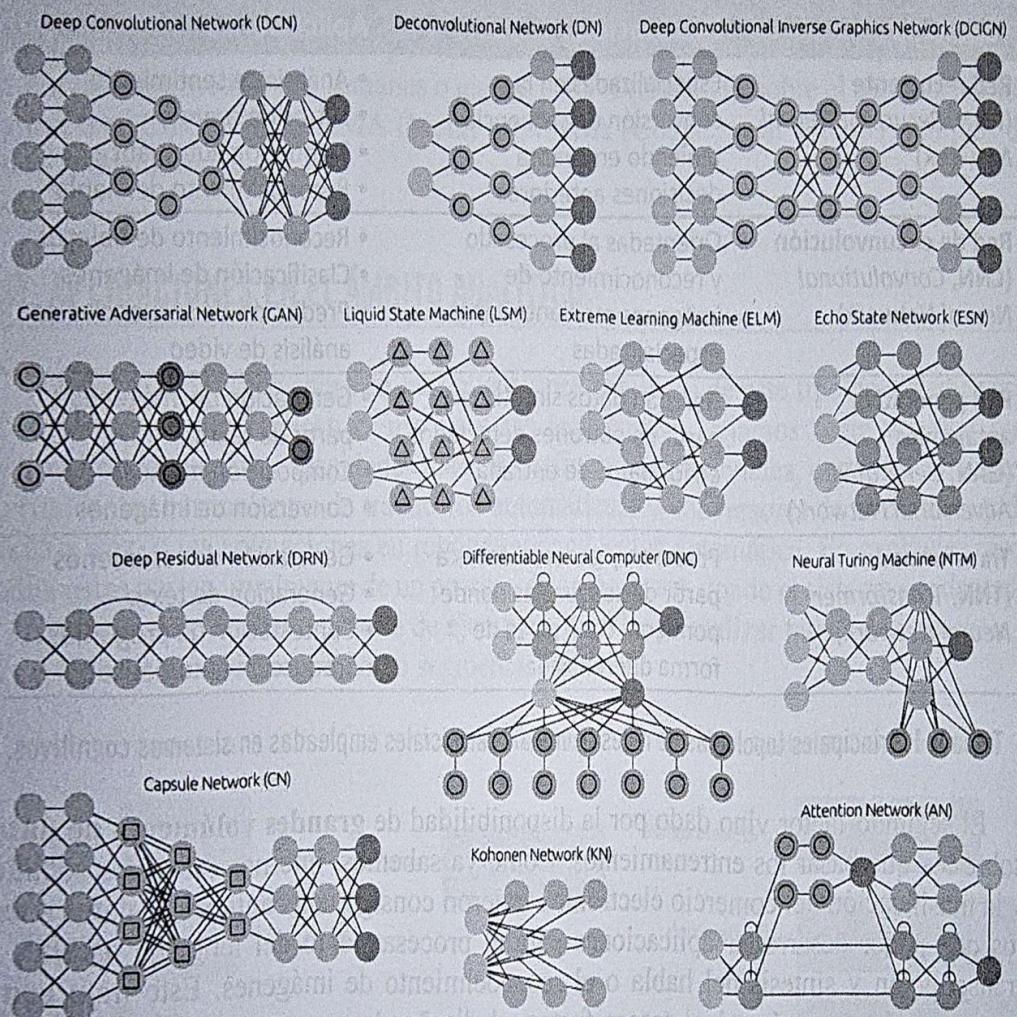


Figura 9-5 (continuación). El zoo de las redes neuronales.

**Nota.** Extraído de *The Neural Network Zoo* [Figura], por Van Veen, F., The Asimov Institute, 2016, (<https://www.asimovinstitute.org/neural-network-zoo/>).

A principios de la década pasada se produce un resurgir en el interés en este tipo de modelos, generándose una explosión de nuevas aplicaciones, principalmente en inteligencia artificial y en el procesamiento de datos no estructurados, como audio, texto o imágenes. Detrás de este impulso hay una serie de factores necesariamente confluentes, que ya hemos venido comentando. El primero tiene que ver con los avances en *hardware* y capacidad de cómputo. El desarrollo y uso de **unidades de procesamiento gráfico (GPU, Graphical Processor Unit)** y otros chips especializados ha permitido pasar de semanas a días en cuanto a tiempos de entrenamiento. Simultáneamente, el despliegue generalizado de **infraestructura y plataformas especializadas** en la nube han facilitado el acceso rápido a potentes entornos de cálculo, sin necesidad de grandes inversiones.

Topología	Descripción	Aplicaciones
<b>Red recurrente (RNN, Recurrent Neural Network)</b>	Especializadas en la conversión de secuencias, teniendo en cuenta decisiones anteriores	<ul style="list-style-type: none"> <li>• Análisis de sentimiento</li> <li>• Filtros de <i>spam</i></li> <li>• Traducción automática</li> <li>• Reconocimiento del habla</li> </ul>
<b>Red de circunvolución (CNN, Convolutional Neural Network)</b>	Orientadas al procesado y reconocimiento de imágenes mediante capas especializadas	<ul style="list-style-type: none"> <li>• Reconocimiento de objetos</li> <li>• Clasificación de imágenes</li> <li>• Predicción de secuencias en análisis de video</li> </ul>
<b>Red generativa antagónica (GAN, Generative Adversarial Network)</b>	Generan datos sintéticos a partir de patrones detectados en los datos de entrada	<ul style="list-style-type: none"> <li>• Generación de imágenes a partir de texto</li> <li>• Composición musical</li> <li>• Conversión de imágenes</li> </ul>
<b>Transformador (TNN, Transformer Neural Network)</b>	Procesan y generan datos a partir de secuencias donde ponderan cada parte de forma diferenciada	<ul style="list-style-type: none"> <li>• Generación de resúmenes</li> <li>• Generación de texto</li> <li>• Contestación de preguntas</li> <li>• Traducción automática</li> </ul>

**Tabla 9-1.** Principales topologías de redes neuronales artificiales empleadas en sistemas cognitivos.

El segundo factor vino dado por la disponibilidad de **grandes volúmenes de datos** sobre los que basar los entrenamientos. Como ya sabemos, internet, las redes sociales y la masificación del comercio electrónico trajeron consigo una cantidad de datos sobre los que poder desarrollar aplicaciones para el procesamiento del lenguaje natural, la transcripción y síntesis del habla o el reconocimiento de imágenes. Esto trajo como consecuencia, y aquí está el tercer factor, el diseño de **nuevas topologías** de redes neuronales y mecanismos de aprendizaje y optimización de hiperparámetros. Estas arquitecturas se caracterizaban por incrementar de forma considerable tanto el número de capas intermedias como la distribución de elementos de procesado, apareciendo arreglos en dos y tres dimensiones. Este crecimiento en profundidad es lo que le ha dado nombre a este tipo de modelos<sup>209</sup>. Sirva la Figura 9-5, un clásico ya en la literatura sobre redes neuronales, como ejemplo de la variedad de elementos de procesamiento y topologías disponibles, orientada cada una a resolver una tipología de problema.

La Tabla 9-1 resume cuatro de las principales arquitecturas de redes neuronales empleadas para procesar datos no estructurados. Aunque todas ellas tienen relevancia en su campo, cabe destacar los **transformadores** (*transformers*), que constituyen la base actual de los **grandes modelos de lenguaje** (LLM, *Large Language Model*). Estos modelos son capaces de generar contenido discursivo de gran naturalidad a partir de un entrenamiento sobre enormes conjuntos de datos textuales, donde se extraen

209 A partir de dos capas intermedias se puede considerar a la red, y por lo tanto al aprendizaje, como profunda.

patrones, estructuras y relaciones presentes en el lenguaje<sup>210</sup>. Las redes neuronales que los sustentan están constituidas por miles de millones de hiperparámetros, con tiempos de aprendizaje medidos en semanas o meses. GPT-4 (OpenAI), sobre el que está basado en popular ChatGPT, LLaMA (Meta) o LaMDA (Google), son algunos ejemplos de estos modelos del lenguaje.

### 9.3 APLICACIONES EN EL ÁMBITO ANALÍTICO

Vamos a centrarnos ahora en las aplicaciones analíticas de este tipo de sistemas con dos ejemplos que implican el uso de estas tecnologías. Dejamos fuera aquellas que giran alrededor de las capacidades más interactivas y operacionales, como los asistentes virtuales, las herramientas de traducción automática, los conversores de voz a texto y de texto a voz, o las aplicaciones en robótica y conducción automática. En cualquier caso, todas estas parten igualmente de un proceso de aprendizaje, donde el sistema es adaptado a un contexto en base a una serie de ejemplos, para luego realizar tareas de inferencia a nivel de clasificación, regresión o segmentación.

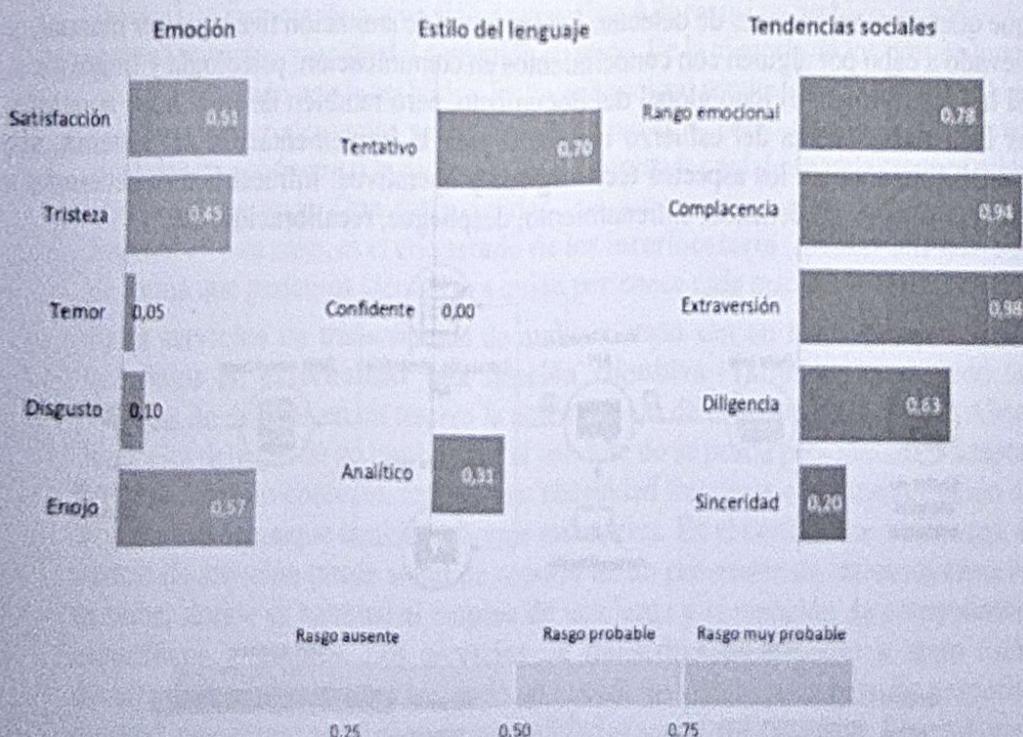


Figura 9-6. Análisis de tono en un correo electrónico.

<sup>210</sup> Un aspecto importante de estos modelos es la forma como se codifica el texto para alimentar a la red neuronal durante el proceso de entrenamiento e inferencia. Para ello se utilizan técnicas que hacen corresponder las entradas del texto a vectores continuos que las representan (*embeddings*).

### 9.3.1 Análisis de conversaciones

El **análisis de tono** es un buen ejemplo de lo sencillo, provechoso y económico que puede resultar montar un sistema cognitivo. El objetivo no es otro que detectar y cuantificar rasgos que caractericen una comunicación en términos del estilo, el carácter y el estado anímico que expresa. Esta comunicación puede tener la forma de un correo electrónico, una noticia en prensa, un discurso o una conversación entre varias personas. La Figura 9-6 muestra un ejemplo de este tipo de análisis realizado sobre un correo electrónico, donde se identifican una serie de rasgos en el documento, agrupados en tres familias. Este puede servir para estudiar cómo está evolucionando el tono de las comunicaciones entre un vendedor y un cliente alrededor de una oportunidad de negocio, deduciendo a partir de ahí si esta progresiona adecuadamente o, por el contrario, si hay que cambiar algo en la estrategia. Con un sentido más práctico y operativo, otra aplicación es la revisión del tono de un correo antes de enviarlo, con el fin de validar la forma en la que se está haciendo la notificación y si se adapta o no al estilo de comunicación corporativo.

Para desarrollar un sistema de análisis de tono hay que disponer de una colección importante de documentos para el entrenamiento de un modelo<sup>211</sup>. Cada uno de estos documentos debe estar previamente anotado con los diferentes rasgos que contiene y que queremos ser capaces de detectar. Este proceso de anotación tiene que ser manual, y llevado a cabo por alguien con conocimientos en comunicación, psicología y lingüística. Si la idea es medir el tono global del documento, pero también de cada frase concreta, es fácil darse cuenta del esfuerzo necesario para la implementación del sistema, sin contabilizar todavía los aspectos tecnológicos y operativos: infraestructura necesaria y especialistas en modelizado, entrenamiento, despliegue, recalibración, etc.

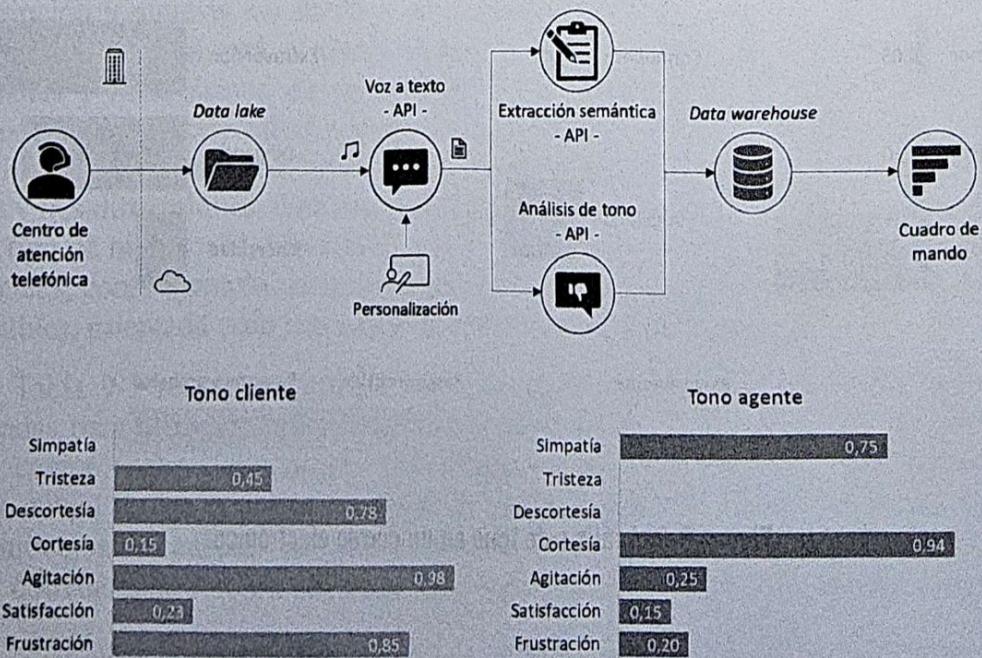


Figura 9-7. Análisis de tono y semántico en las conversaciones con los clientes

211 Una red neuronal recurrente sería un modelo apropiado para este caso.

Sin embargo, existen proveedores que nos facilitan todo este trabajo, proporcionando un conjunto de **servicios cognitivos** en la nube en forma de **API**. La Figura 9-7 enseña un ejemplo de cómo sería una implementación con ellos. En este caso, se trata de analizar las conversaciones en un centro de atención telefónica con el objetivo, entre otros, de medir el tono de los agentes y relacionarlo con la eficacia a la hora de resolver las incidencias. Esta medición puede hacerse a nivel de conversación completa o de cada manifestación concreta, distinguiendo entre las realizadas por el cliente y el agente. En este sentido, un análisis muy interesante sería estudiar cómo va variando el tono del cliente desde que la conversación se inicia hasta que termina, viendo la habilidad del agente en todo el proceso.

El sistema para implementar este análisis consistiría en una aplicación que se encarga de coordinar el flujo de los datos, la llamada a las diferentes API, y la persistencia y presentación de resultados. Los pasos serían los siguientes:

1. En primer lugar, las grabaciones de las conversaciones son almacenadas en un contendor dentro de un almacén de objetos en la nube. Esto puede hacerse por lotes (una vez finalizado el día) o tan pronto como la conversación concluye.
2. El audio de las conversaciones debe ser transcrita a texto, ya que los servicios de análisis posteriores requieren este formato de entrada<sup>212</sup>. Para ello es necesario invocar un servicio de conversión de voz. Este API recibe un archivo de audio<sup>213</sup> y genera una transcripción del contenido en texto. En la mayoría de los proveedores, esta transcripción puede ser básica, incluyendo solo el resultado literal, pero también puede contener elementos adicionales bajo petición, como término alternativos, medidas de confianza o la detección de determinadas palabras en la transcripción (*keyword spotting*). Otra funcionalidad que proporcionan estos API, y que nos interesa en este caso, es el **etiquetado de los interlocutores** (*speaker diarization*), de forma que podemos identificar a quien pertenece cada manifestación transcrita.

Estos servicios de transcripción de audio a texto son un buen ejemplo de los beneficios de **externalizar una función cognitiva**. Ya hemos comentado las ventajas de delegar en un tercero la construcción de este tipo de modelos. Ahora bien, esta delegación no implica que el servicio no se pueda personalizar y adaptar a nuestro propio contexto. Esto es una necesidad frecuente en la transcripción de audio a texto, aunque también en otros escenarios. En el caso en que nos ocupa, el centro de atención puede ser el de soporte de un proveedor de infraestructura en la nube, donde es habitual el empleo de una jerga y la mención de componentes específicos y propios. Los servicios de transcripción de audio a texto están desarrollados y entrenados teniendo en cuenta un vocabulario base de propósito general, por lo que no ofrecerían una calidad aceptable en este caso. Sin embargo,

<sup>212</sup> Esto no tendría que ser necesariamente así. El servicio de análisis de tono podría trabajar directamente sobre audio si hubiese sido entrenado con este formato. Además, de esta manera el tono podría extraerse no solo en base al contenido de la conversación, sino también teniendo en cuenta la locución. De esta manera, el análisis sería más completo, aunque el modelo cognitivo sería más complejo de desarrollar.

<sup>213</sup> Se acostumbran a soportar otros interfaces de entrada (HTTP, WebSockets).

la mayoría de ellos ofrecen la posibilidad de realizar una **adaptación lingüística**, mediante la cual se puede entrenar un modelo propio que expande el modelo base incluyendo la terminología específica. Algo parecido ocurre con las características del audio. Si nuestro centro de atención está empezando y sólo atiende en inglés, será frecuente la llamada de usuarios no nativos con acentos variados. En estos casos también suele ser posible la realización de una **adaptación acústica** con el fin de mejorar la resolución de la transcripción a casuísticas complicadas<sup>214</sup>. Para cualquiera de estas dos adaptaciones será necesario proporcionar datos de ejemplo, pero el servicio nos proporciona métodos dentro del API para la realización del entrenamiento de uno o varios modelos personalizados, su seguimiento y su posterior utilización<sup>215</sup>. En definitiva, no solo tenemos la capacidad de usar un sistema cognitivo gestionado por un tercero, sino que podemos adaptarlo a nuestras propias necesidades.

3. Una vez que tenemos la transcripción del audio podemos pasar a analizarla. Para ello invocamos a dos servicios. El primero de ellos nos proporciona una **extracción semántica** del contenido. Esta puede incluir la identificación de palabras clave, conceptos que subyacen en el contenido pero que no son nombrados de forma explícita, entidades mencionadas (personas, empresas, lugares, etc.), relaciones entre estas entidades, etc. Al igual que en el caso de la transcripción de audio a texto, esta extracción se puede personalizar mediante el desarrollo de **anotadores** que son capaces de identificar entidades y relaciones concretas después de un proceso de entrenamiento. El objetivo de esta extracción semántica es pasar de un contenido no estructurado a otro que sí lo está, cuantificando además la relevancia de cada elemento extraído. Es decir, automatizar la identificación del objetivo y el contenido de la conversación entre el usuario y el agente. Esto nos permitirá el análisis posterior de las consultas más habituales, sobre qué componentes son y acerca de qué incidencias.
4. Paralelamente, la transcripción pasa por el servicio de **análisis de tono**, donde obtendremos una serie de rasgos (la Figura 9-7 muestra siete) asociados a cada comentario, tanto del cliente como del agente. Cada rasgo estará cuantificado por un nivel de relevancia.
5. Finalmente, tanto el resultado de la extracción semántica como del análisis de tono son almacenados en una base de datos relacional, bajo un modelo que permite su representación y consulta en un **cuadro de mando**.

En definitiva, este proceso nos permitirá añadir nuevos indicadores (KPI) a nuestro sistema de medición del centro de atención telefónica a partir de un contenido no estructurado, difícil de abordar de otra manera.

214 Otro caso complicado se da cuando la calidad del audio es mala y no puede ser mejorada. Por ejemplo, en las comunicaciones entre un helicóptero de rescate y su base.

215 Otra adaptación típica es la basada en gramáticas. Esta es útil en sistemas de reconocimiento de voz que esperan respuestas concretas del usuario ante determinadas preguntas, limitando el vocabulario reconocible, pero proporcionando mejor resolución y rapidez en la transcripción.

### 9.3.2 Análisis de imágenes

El **análisis de imágenes** es otra área donde se han desarrollado múltiples y variados servicios cognitivos, existiendo un gran número de aplicaciones.

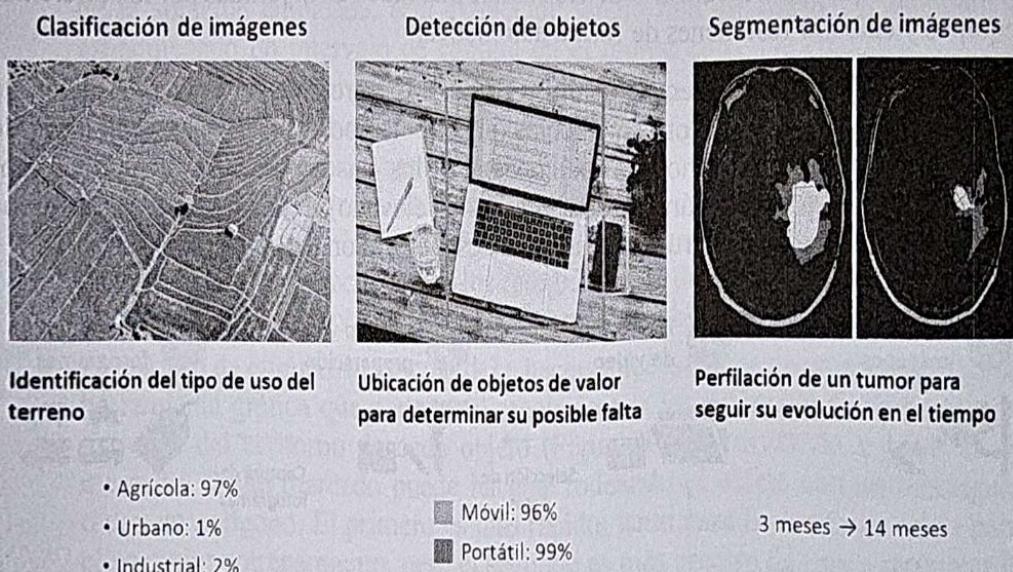


Figura 9-8. Casos de uso del análisis de imágenes.

La Figura 9-8 muestra los tres casos de uso principales dentro del análisis de imágenes, mientras que la Tabla 9-2 los detalla y da algunos ejemplos.

Caso de uso	Objetivo	Aplicaciones
<b>Clasificación de imágenes</b>	Determinar en qué medida una imagen completa pertenece a una o más categorías predefinidas	<ul style="list-style-type: none"> <li>• Identificación de piezas defectuosas o estropeadas</li> <li>• Reconocimiento de personas</li> <li>• Clasificación de imágenes aéreas</li> </ul>
<b>Detección de objetos y acciones</b>	Marcar el contenido de una imagen o video empleando una serie de etiquetas predefinidas que indican objetos o acciones	<ul style="list-style-type: none"> <li>• Conteo de personas, vehículos, etc.</li> <li>• Seguimiento de objetos en cadenas de producción</li> <li>• Conducción automática</li> </ul>
<b>Segmentación de imágenes</b>	Perfilar y etiquetar la posición precisa de una serie de objetos en una imagen en base a una serie de etiquetas predefinidas y formas	<ul style="list-style-type: none"> <li>• Ubicación de objetos en imágenes de satélite</li> <li>• Localización y seguimiento de tumores</li> <li>• Reconocimiento dactilar</li> </ul>

Tabla 9-2. Casos de uso en análisis visual.

La **detección de objetos** es uno de los tres casos de uso con más aplicaciones. Estas pueden consistir en la identificación de una serie de tipos de objetos predefinidos en una imagen o video, incluyendo su posición, pero también la de acciones o movimientos, como puede ser la consecución de un gol en un partido de fútbol. Esto último puede ser útil, por ejemplo, para localizar todos los tantos marcados en la jornada por los diferentes equipos y elaborar resúmenes de forma automática.

Otro ejemplo es el conteo de distintos objetos. Concretamente, la identificación de coches, motos, camiones u otros vehículos que circulan por una autopista, con el fin de determinar en qué franjas horarias se concentran los atascos y cuál es el motivo (por ejemplo, debido a que el número de camiones es elevado). La Figura 9-9 contiene los pasos necesarios para construir un sistema de este tipo. Son los siguientes:

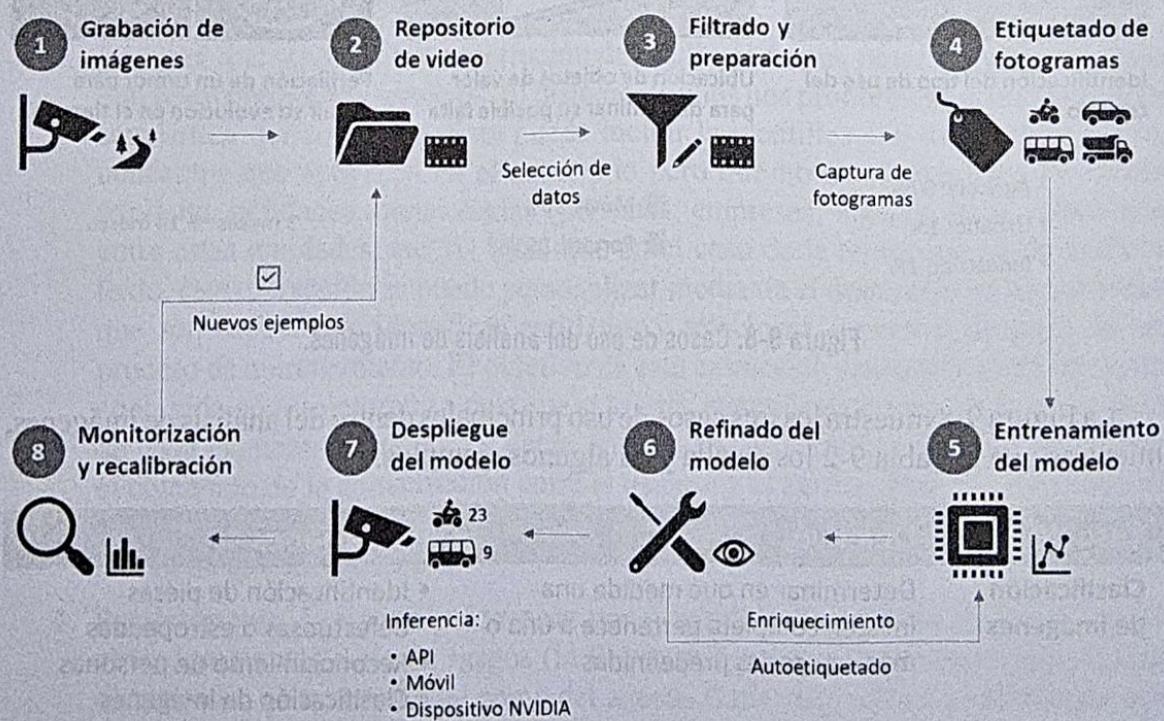


Figura 9-9. Pasos en la construcción de un modelo de detección de vehículos.

1. En primer lugar necesitaremos un conjunto de grabaciones sobre los que basar el entrenamiento del modelo. Estas grabaciones deben ser reales y representativas de las situaciones que se quieren modelizar: no solo se trata de que aparezcan los distintos objetos que se quieren identificar (automóviles, motos, distintos tipos de camiones, autobuses, etc.), sino que deben hacerlo sobre un escenario real, no montado artificialmente.
2. Estas grabaciones se documentan y almacenan en un repositorio, formando el conjunto de datos de entrenamiento.
3. Antes de proceder al etiquetado, los videos se pueden tratar con el fin de disminuir el posible ruido, eliminando las de menor calidad y representatividad. En este

punto es necesario también tener definidos los objetos que se quieren identificar en las imágenes.

4. El paso de **etiquetado** es el más importante. En primer lugar se deben extraer los fotogramas de cada grabación que se emplee para el entrenamiento. Dependiendo de la solución empleada, esta extracción tendrá que ser manual o automática, especificando un intervalo de captura en el último caso. De cualquier manera, es importante asegurar que los fotogramas obtenidos son representativos de los diferentes objetos que se quieren identificar. Esto significa que en el total de las imágenes de entrenamiento así generadas, cada objeto debe aparecer un mínimo de veces<sup>216</sup>. Por lo tanto, es habitual utilizar una extracción automática de fotogramas y luego complementarla con fotogramas adicionales con el fin de alcanzar estos mínimos. Una vez seleccionados los fotogramas, se debe proceder a su etiquetado manual, marcando el contorno de cada uno de los objetos que en ellos aparecen.

El proceso de etiquetado manual debe hacerse necesariamente empleando alguna herramienta gráfica que permita el recorrido por los diferentes fotogramas y el marcado del contorno de cada objeto (Figura 9-10), existiendo muchas en el mercado. Dicho marcado puede hacerse rodeando el objeto con un rectángulo o con un polígono. El primero es más rápido, tanto para la anotación como para el posterior entrenamiento, pero el segundo es más preciso de cara a recuperar la posición del objeto en la inferencia del modelo.

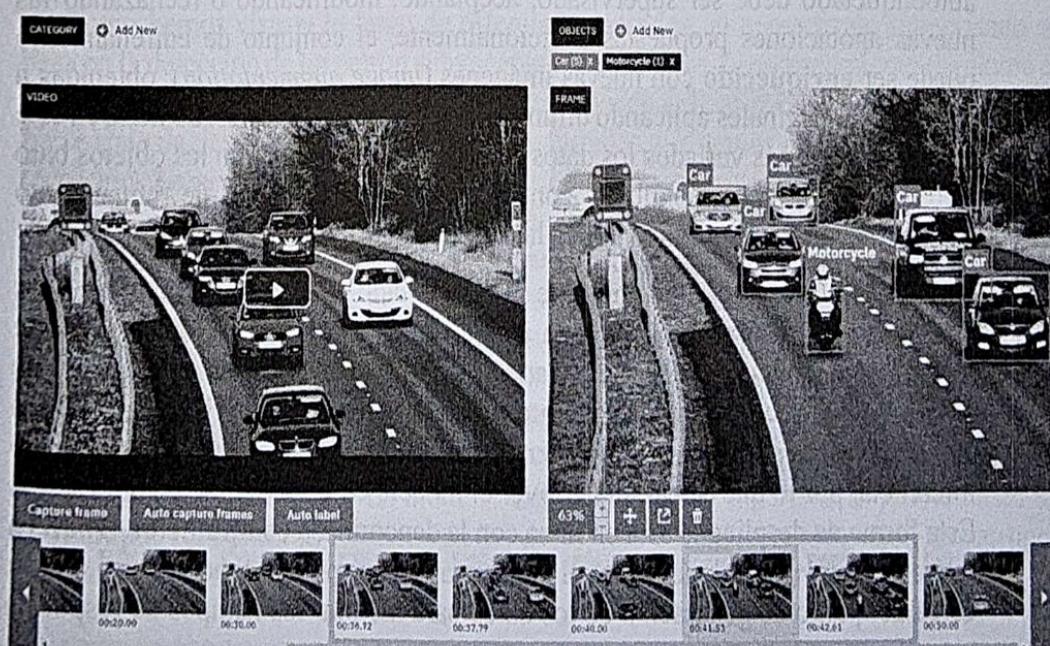


Figura 9-10. Etiquetado de objetos en IBM Maximo Visual Inspection.

216 Es difícil establecer un número mínimo general de fotogramas por objeto, ya que este dependerá tanto de la complejidad y densidad de las imágenes, como del número total de objetos que se deben identificar. Como primera aproximación, no debería bajar de 5–10 apariciones.

5. Como en todo aprendizaje supervisado, una vez que disponemos del conjunto de entrenamiento etiquetado podemos pasar a la **generación del modelo**. Dependiendo del caso de uso y de la forma en la que desplegaremos posteriormente el modelo, habrá que elegir entre distintos tipos de algoritmos y optimizaciones. Por ejemplo, para la detección de objetos hay que tener en cuenta el dispositivo donde se realizará la inferencia, si dispone de CPU o GPU, si hay que trabajar con imágenes de alta resolución, o si el marcado está hecho con rectángulos o polígonos. **YOLO** (*You Only Look Once*) o **Detectron**, basados en redes neuronales de circunvolución, son dos ejemplos de estos algoritmos<sup>217</sup>.

En el caso de la clasificación de imágenes, y para agilizar el entrenamiento, se puede partir de **modelos preentrenados**, especializados en determinadas áreas, como comida, plantas o escenas. La mayoría de los proveedores de servicios de reconocimiento visual en la nube disponen de estos modelos como punto de partida. A su vez, es una práctica habitual desarrollar un modelo base preentrenado sobre el que ir construyendo después versiones cada vez más especializadas.

6. Una vez entrenada una primera versión del modelo, es posible hacer un **refinamiento** para mejorar su resolución. En el caso de los modelos de detección de objetos, este suele consistir en un **autoetiquetado**: se utiliza el propio modelo para etiquetar nuevos fotogramas, de forma que el conjunto de entrenamiento aumenta y se puede entrenar una nueva versión del modelo más precisa. Este autoetiquetado debe ser supervisado, aceptando, modificando o rechazando las nuevas anotaciones propuestas. Adicionalmente, el conjunto de entrenamiento puede ser **enriquecido** con nuevas imágenes (*image augmentation*), obtenidas a partir de las originales aplicando difuminados, filtros y rotaciones. Se trata, en este caso, de hacer más variados los datos de aprendizaje, mostrando los objetos bajo otros puntos de vista, ángulos y contornos. Estos mecanismos de refinamiento estarán disponibles en función de la herramienta utilizada.
7. Una vez validado<sup>218</sup>, el modelo puede ser desplegado ya en producción para tareas de inferencia. La forma más directa es mediante la **publicación de un API**, de forma que el modelo puede ser invocado pasando la señal de la imagen (*streaming*) y obteniendo los conteos de los distintos tipos de vehículos cada cierto tiempo. Sin embargo, esta forma de invocación centralizada introduce latencia en el proceso de inferencia, por lo que es habitual mover el modelo cerca del punto de aplicación. Esta forma de despliegue está en línea con la denominada **computación frontera** (*edge computing*)<sup>219</sup>, donde tanto el almacenamiento como el procesado se mueve cerca del punto de generación de los datos para mejorar los tiempos de respuesta

217 Cada uno de estos algoritmos ofrece distintos niveles de exactitud, requiriendo también más o menos tiempo de entrenamiento.

218 Como en cualquier otro modelo de clasificación, aquí aplican distintas métricas de precisión, exactitud o sensibilidad a la hora de verificar y dar por bueno el entrenamiento.

219 A veces traducida también como **computación en el borde**, aunque el término inglés sin traducir es el empleado habitualmente.

y disminuir el consumo de ancho de banda. Para ello, el modelo es exportado e importado en un dispositivo con capacidad para hacer inferencia (conectado a las cámaras de control de la autopista, en nuestro caso) y almacenar y/o enviar los resultados. Esto puede hacerse empaquetando el modelo en un contenedor, o bien utilizando algún marco de despliegue. **TensorRT**, que se ejecuta en dispositivos NVIDIA equipados con GPU, es uno ellos; **Core ML**, para sistemas equipados con Apple iOS, es otro.

8. El último paso es la **monitorización del modelo** en producción, con el fin de detectar posibles pérdidas de precisión en la identificación de los vehículos y la necesidad de recalibración. La aplicación del modelo a nuevas imágenes y su verificación permitirá ampliar la base de datos de entrenamiento.

En el caso de querer detectar acciones, el procedimiento sería muy similar al que acabamos de plantear. La diferencia radica en que la anotación consistiría en marcar con una etiqueta el principio y el fin cada tipo de acción, con lo que esta estaría compuesta por un número determinado de fotogramas.

## 9.4 PROBLEMAS DE SESGO Y FALTA DE EQUIDAD EN LOS MODELOS

Como hemos visto a lo largo de diferentes capítulos, el aprendizaje supervisado se basa en la detección de un patrón subyacente en unos datos de entrenamiento mediante el desarrollo de un modelo que posteriormente se puede aplicar para realizar predicciones. Ya se trate de datos estructurados o no estructurados, o estemos empleando una sencilla regresión lineal o una red neuronal recurrente con millones de hiperparámetros, el mecanismo es siempre el mismo, y viene marcado por la inferencia estadística más clásica: utilizar una muestra para caracterizar a toda una población estadística, de manera que asumimos que el comportamiento de la primera es representativo de la segunda<sup>220</sup>. Pero, ¿qué ocurre cuando la muestra no es representativa de la población que se está intentando modelizar? En este caso, el modelo obtenido adolecerá de falta de **exactitud** (*accuracy*), de forma que producirá resultados que se apartan de los reales<sup>221</sup>. Ahora bien, una falta de representatividad de la muestra no es el único motivo para obtener resultados inexactos cuando aplicamos un modelo. Otra explicación, inherente al problema en sí, es la propia **variabilidad** existente en los datos, que hace que la extracción de patrones y tendencias no sea tarea fácil.

220 La inferencia estadística nos da métodos para calcular los límites dentro de los cuales esta asunción es válida.

221 No pretendemos aquí hacer una exposición rigurosa alrededor de las fuentes de error y su medición, sino introducir una serie de conceptos habituales cuyo significado es necesario conocer a la hora de diseñar y construir modelos.

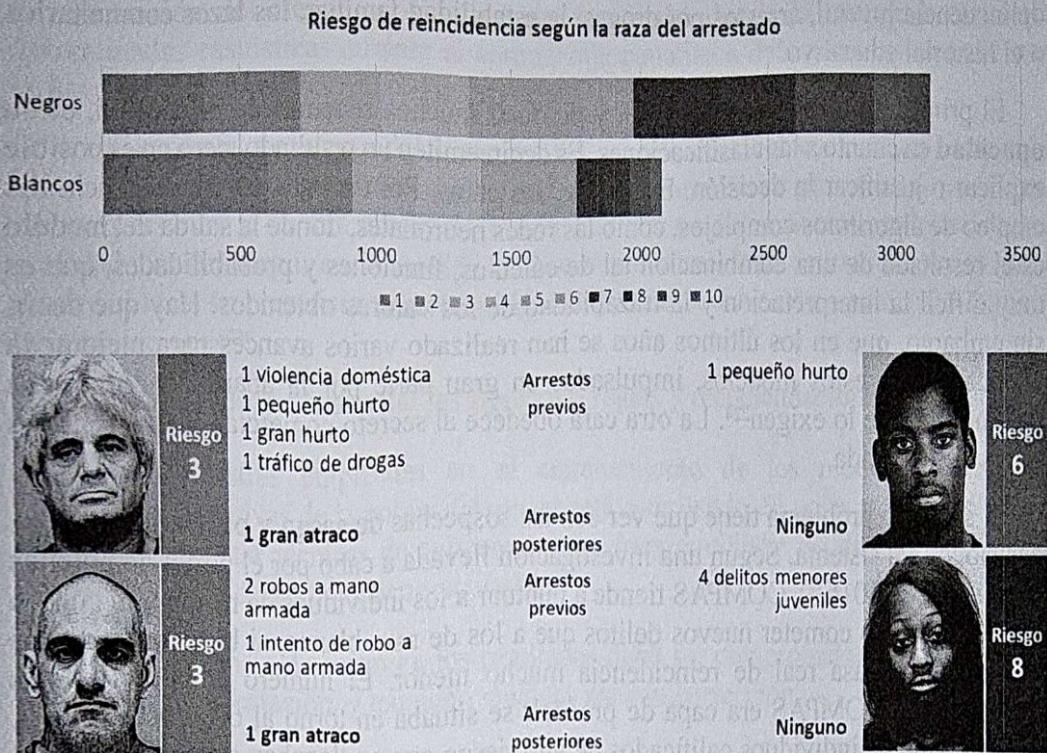
Por ejemplo, si queremos construir un sistema de reconocimiento facial que clasifique a una persona según su ciudad de nacimiento a partir de una foto, la variabilidad es tal, y debida a tan variados motivos, que la exactitud del sistema, medida como el número de clasificaciones correctas<sup>222</sup>, dejará mucho que desear. La disponibilidad de una amplia y variada muestra de fotos de personas, etiquetadas con su correspondiente lugar de nacimiento, no sería suficiente para modelizar adecuadamente un problema tan complejo. En un caso como este, diremos que el sistema de reconocimiento facial producirá **errores aleatorios**, fruto de esa variabilidad, pero también de cualquier otra imprecisión cometida en el proceso de toma o recolección de las fotos y su etiquetado.

Otro problema distinto es cuando nos enfrentamos a modelos que producen **errores sistemáticos**. En el ejemplo anterior del reconocimiento facial, y con un conjunto de entrenamiento ideal, las clasificaciones incorrectas pueden darse en cualquier sentido, etiquetando como coruñés a una persona nacida en Torrelodones con la misma probabilidad que a un barcelonés se le califica como porteño. Sin embargo, el error sistemático es aquel que se produce siempre, de manera consistente y repetitiva, en la misma dirección. Por ejemplo, si el sistema tiene tendencia a confundir a los naturales de Vilanova i la Geltrú con aquellos nacidos en Hortolândia, estaremos ante un error que se da por sistema. En este caso diremos que existe un **sesgo** (*bias*) en la respuesta del modelo, o también que el modelo está sesgado, en la medida en que tiende a proporcionar unas respuestas frente a otras.

El sesgo es un concepto estadístico. Este puede ser debido a diferentes causas. Una de ellas es la realización de mediciones de forma defectuosa, por ejemplo empleando instrumentos mal calibrados. Si cuando se tomaron las fotos a los nacidos en Vilanova i la Geltrú la cámara tenía una tara en el objetivo que distorsionaba la imagen y les hacía adquirir rasgos de hortolandenses, es de esperar que el modelo ubique a los naturales de la capital de la barcelonesa comarca del Garraf en el municipio brasileño del estado de São Paulo.

Otra de las causas del sesgo es el empleo de muestras no representativas en el entrenamiento, tal y como planteábamos anteriormente. Concretamente, de **muestras desequilibradas**, donde unas clases dominan frente a otras, estando estas últimas subrepresentadas. Si en la muestra los nacidos en la ciudad de Barcelona son muchos menos que los de Buenos Aires, entonces la probabilidad de identificar correctamente a los primeros es, por sistema, muy inferior a la de los segundos.

222 Se puede matizar mucho más el rendimiento del modelo con métricas como la **precisión** (*precision*) y la **sensibilidad** (*recall*) que miden en qué sentido se está equivocando el modelo al producir una clasificación incorrecta.



**Figura 9-11.** Sesgo en el sistema COMPAS.

**Nota.** Extraído de *Machine Bias [Figuras]*, por Angwin, J., Larson, J., Matu, S. y Kirchner, L. F., ProPublica, 2016, (<https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>).

En este último sentido es donde este sesgo estadístico comienza a tomar **connotaciones psicosociológicas**, dejando de ser únicamente un problema de exactitud o precisión matemática. Si en determinados contextos un modelo produce decisiones sesgadas, perjudicando a ciertas personas o colectivos, entonces estamos ante un problema de **falta de equidad (fairness)**, normalmente con importantes implicaciones morales. Se han dado, quizás cada vez de forma más notoria, diferentes casos al respecto.

Uno de los más llamativos fue el del **sistema COMPAS** (*Correctional Offender Management Profiling for Alternative Sanctions*), un software para la evaluación de la probabilidad de reincidencia de una persona que ha cometido un delito. Esta desarrollado por la compañía Equivant, formando parte de la familia de productos **Northpointe Suite** para la gestión de casos legales<sup>223</sup>, empleado en los tribunales de ciertos estados de los Estados Unidos de América. Mediante la definición de unas escalas de riesgo de reincidencia en distintos tipos de infracciones, COMPAS evalúa, entre otras cosas, la probabilidad de que un arrestado vuelva a cometer un delito mientras está pendiente de juicio, o bien una vez quede en libertad. Para ello utiliza distintos algoritmos de clasificación que tienen en cuenta los antecedentes de la persona, incluyendo actos de

223 <https://www.equivant.com/northpointe-suite-case-manager/>

delincuencia juvenil, arrestos por drogas, la estabilidad familiar, los lazos comunitarios o el historial educativo.

El primer problema de COMPAS, y de otros muchos sistemas de predicción, es su **opacidad** en cuanto a las clasificaciones. Es decir, emiten un resultado, pero no es posible explicar o justificar la decisión. Esto tiene dos caras. Por un lado, es consecuencia del empleo de algoritmos complejos, como las redes neuronales, donde la salida del modelo es el resultado de una combinación tal de cálculos, funciones y probabilidades, que es muy difícil la interpretación y la trazabilidad de los valores obtenidos. Hay que decir, sin embargo, que en los últimos años se han realizado varios avances para mejorar la explicación de estos modelos, impulsados en gran parte por la aparición de marcos regulatorios que lo exigen<sup>224</sup>. La otra cara obedece al secreto comercial, y en ocasiones es más complicada.

El segundo problema tiene que ver con las sospechas de **sesgo y parcialidad** en las decisiones del sistema. Según una investigación llevada a cabo por el portal de noticias ProPublica en 2016<sup>225</sup>, COMPAS tiende a puntuar a los individuos de raza negra con un mayor riesgo de cometer nuevos delitos que a los de raza blanca, si bien los primeros muestran una tasa real de reincidencia mucho menor. El número de reincidencias correctas que COMPAS era capaz de predecir se situaba en torno al 60%. Ahora bien, el porcentaje de individuos calificados de alto riesgo que no llegaban a reincidir era del 23% en el caso de personas blancas, pero subía al 45% en el caso de personas negras. Inversamente, el 48% de las personas blancas etiquetadas de bajo riesgo acababan siendo arrestadas de nuevo, mientras que el porcentaje bajaba al 28% en las personas negras. La Figura 9-11 da más detalles de este estudio, mostrando la diferencia en las puntuaciones de riesgo entre una raza y otra, así como algunos de los casos que se mencionan en el estudio.

#### 9.4.1 Mitigación del sesgo

La investigación de ProPublica ha sido discutida en diferentes frentes, el de Equivant el primero. En cualquier caso, nos puede servir como ejemplo de los riesgos que conlleva la utilización de conjuntos de entrenamiento desequilibrados, si es este, y no otro, el motivo que puede estar detrás de la supuesta falta de imparcialidad del sistema. Si la distribución de los casos empleados en el aprendizaje del sistema obedece a la mostrada en la Figura 9-11, entonces es de esperar que la respuesta del mismo sea tendenciosa. Este comportamiento lo podemos encontrar en distintos sistemas, como los de reconocimiento facial, donde los ejemplos de entrenamiento no están equilibrados en cuanto a raza, sexo o rango de edad; o en los modelos

224 Una de las líneas en este sentido consiste en alterar los atributos de entrada al modelo en diferentes sentidos y magnitudes, midiendo como varía la respuesta; el modelo sigue funcionando como una caja negra, pero al menos se puede cuantificar y documentar lo sensible que es a distintos valores y si la respuesta es siempre consistente.

225 <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>

de evaluación del riesgo crediticio o de accidente, donde la preponderancia de determinadas casuísticas durante el aprendizaje penaliza a determinados colectivos a la hora de obtener un seguro o una hipoteca.

La solución al problema del sesgo en los modelos predictivos no está exenta de discusión. Algunos argumentan que el sesgo está presente de por sí en la sociedad, en las personas y en sus actos, de forma que los modelos acaban reflejando una realidad. Sin embargo, en la medida en que supone una falta de equidad en la toma de decisiones, su presencia debería mitigarse, y en esta dirección avanzan las leyes y las regulaciones<sup>226</sup>.

En este sentido, podemos dar las siguientes reglas y recomendaciones:

- ▶ Las muestras empleadas en el entrenamiento de los modelos deben ser representativas de la población que se está estudiando. Si no es posible, se deben implementar técnicas de **sobremuestreo** para equilibrar las distribuciones y la presencia de grupos poco frecuentes.
- ▶ La existencia de sesgo en los resultados debe ser comprobada, identificando su causa y afectación, y tomando medidas para su corrección.
- ▶ Emplear solo atributos que sean relevantes para la predicción, sin introducir aquellos de los que se conoce de antemano su efecto en este sentido.
- ▶ Como norma general, el empleo en los modelos de atributos que puedan suponer una **discriminación**, como la raza, el sexo, la religión, la edad o cualquier otro factor sensible debería evitarse. Si su empleo es necesario, entonces no deben ser la única fuente de información para que el algoritmo tome las decisiones.
- ▶ Evitar la introducción de **ideas preconcebidas** o prejuicios por parte de los desarrolladores y analistas a cargo del modelo. Utilizar equipos variados en cuanto a perspectivas.

En cualquier caso, los modelos deben ser monitorizados, evaluados y auditados de forma regular, efectuando los ajustes y recalibraciones necesarias en el caso de identificar sesgos en los resultados. Como veremos en el siguiente apartado, existen herramientas y soluciones que implementan y facilitan estas tareas.

226 GDPR (*General Data Protection Regulation*) en Europa o ECOA (*Equal Credit Opportunity Act*) en los Estados Unidos de América.

## 9.5 HERRAMIENTAS Y SOLUCIONES PARA ANÁLISIS COGNITIVO

Al igual que en el caso del análisis predictivo, existen distintos paquetes y librerías para el desarrollo de modelos cognitivos, especialmente alrededor de las redes neuronales y el aprendizaje profundo<sup>227</sup>. **TensorFlow** (Google), **PyTorch** (Meta) o **MXNet** (Apache) son algunos de los entornos más empleados para el desarrollo de aplicaciones de visión por ordenador y procesamiento del lenguaje natural. Todos ellos ofrecen interfaces para distintos lenguajes de programación, cubriendo funciones que van desde el entrenamiento hasta la inferencia de modelos y puesta en producción.

### 9.5.1 Aceleración de la inferencia de modelos por hardware

**TensorFlow** es sin duda uno de los marcos de desarrollo más empleados. Cuenta, además, con distintas librerías construidas sobre él que facilitan la construcción de modelos al proporcionar una capa de abstracción. **Keras** (Python) es una de las más populares. Con el fin de acelerar y optimizar las cargas de trabajo de TensorFlow, Google desarrolló ya en 2016 un circuito integrado específico denominado **TPU** (*Tensor Processing Unit*), especialmente adecuado para redes neuronales de circunvolución. Actualmente, Google proporciona acceso a máquinas virtuales equipadas con **TPU como un servicio** en Google Cloud Platform. Una vez creada la instancia de la máquina, el modelo de TensorFlow debe ser desplegado en la misma, tanto para tareas de entrenamiento como de inferencia<sup>228</sup>.

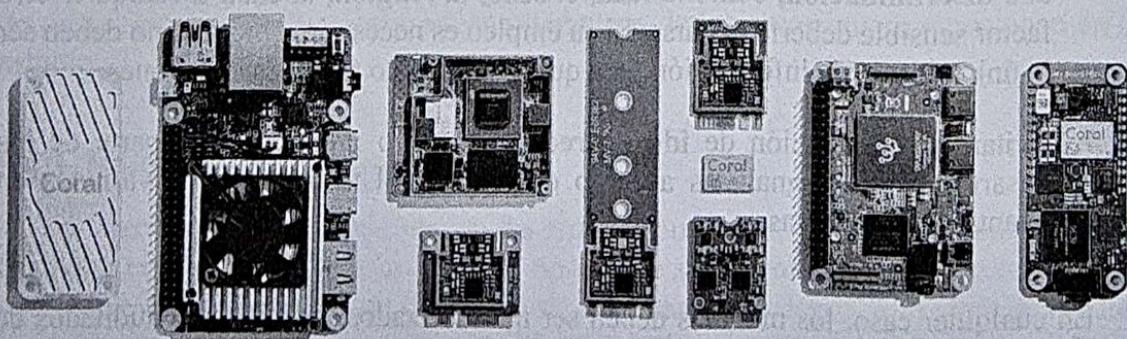


Figura 9-12. Dispositivos equipados con TPU de la familia Google Coral.

Nota. Extraído de *Coral [Imagen]*, Google, 2023, (<https://coral.ai/products/>).

227 [https://en.wikipedia.org/wiki/Comparison\\_of\\_deep\\_learning\\_software](https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software)

228 Una TPU se encarga principalmente de acelerar y paralelizar la multiplicación de matrices, que es una de las operaciones más importantes y costosa en los modelos neuronales. Esto se nota especialmente en los procesos de entrenamiento, que se ven reducidos en tiempo, consumiendo también menos energía que empleando una CPU o GPU. Esta aceleración es tan bien considerable en las tareas de inferencia donde, por ejemplo, se deben identificar múltiples objetos en tiempo real y con muy baja latencia.

Con el fin de desplegar tareas de inferencia en la frontera, especialmente de clasificación y detección de objetos, Google desarrolló también unos circuitos más pequeños, denominados **Edge TPU**, que pueden instalarse en dispositivos pequeños con capacidad de alimentación eléctrica limitada. En el ejemplo del conteo de vehículos que veíamos en apartados anteriores, cada cámara que capta la señal en vivo del estado de la autopista estaría conectada a uno de estos dispositivos, sobre el que se habría desplegado previamente el modelo para la identificación de los vehículos. La propia Google comercializa dispositivos equipados con TPU a través de Coral, una de sus subsidiarias (Figura 9-12). Por ejemplo, el **Coral USB Accelerator** permite añadir un coprocesador Edge TPU a un sistema, como una **Raspberry Pi**, a través del puerto USB. La cámara se conectaría a la Raspberry donde se desplegaría también el modelo de detección de vehículos, utilizando la Edge TPU para acelerar la inferencia<sup>229</sup>.

Para dar alguna idea del rendimiento de estos coprocesadores, una Edge TPU puede realizar 4 TOPS<sup>230</sup>, consumiendo 2 vatios de potencia (2 TOPS por watio). En modelos de visión artificial esto equivale a procesar cerca de 400 fotogramas por segundo para la clasificación imágenes en video mediante redes neuronales de circunvolución<sup>231</sup>.

Desde el momento en que las **GPU** están especializadas en cálculo matricial, son también una alternativa para la aceleración de modelos de aprendizaje profundo, con un rango de aplicaciones más general y amplio que las TPU. **NVIDIA**, uno de los principales fabricantes de GPU, desarrolla las familias **Tesla** y **Titan**, orientadas para tareas de entrenamiento, y **TensorRT** y **Jetson** para inferencia. Todas ellas se apoyan en una plataforma y modelo de programación paralela denominado **CUDA** (*Compute Unified Device Architecture*), soportado por TensorFlow y PyTorch, entre otros.

### 9.5.2 Servicios cognitivos en la nube

Como ya hemos venido comentando, la alternativa al desarrollo propio de modelos cognitivos está en los servicios en la nube, donde podemos encontrar distintas API especializadas según el tipo de dato y el objetivo.

229 La Edge TPU utiliza un formato especial de TensorFlow, más compacto, denominado **TensorFlow Lite**, al que hay que convertir los modelos. Es posible también desplegar modelos entrenados con otras librerías, como PyTorch o Keras, mediante un proceso de conversión análogo.

230 1 TOPS (*Trillion Operations Per Second*) equivale a  $10^{12}$  operaciones por segundo; es una métrica empleada, entre otras aplicaciones, para medir y comparar la velocidad de los aceleradores de modelos de aprendizaje profundo.

231 <https://coral.ai/docs/edgetpu/benchmarks/>

La Tabla 9-3 muestra los principales servicios para el procesamiento del lenguaje natural en la nube, ofrecidos por AWS, Google Cloud, Microsoft Azure e IBM Cloud. En mayor o menor medida, todos ellos permiten la personalización y adaptación de algunas de las características. En algunos casos, esto es indispensable, como en la **clasificación de textos**, donde es el usuario el que debe proporcionar las categorías y los ejemplos correspondientes para hacer el entrenamiento. En otras situaciones, como en la **extracción de relaciones y entidades**, el servicio está preentrenado con sistemas de tipos genéricos, por lo que es recomendable realizar una adaptación al dominio mediante un entrenamiento basado en la anotación manual de textos<sup>232</sup>. Otra característica interesante es la disponibilidad en algunos casos de **modelos contextuales específicos**. Por ejemplo, Google proporciona un modelo especializado en salud y ciencias de la vida para el procesamiento de contenido médico.

En el caso de la transcripción de voz a texto, la oferta es también similar entre proveedores, proporcionando todos características en la línea de lo expuesto en el Apartado 9.3.1: identificación de interlocutores, extracción de palabras clave, adaptación lingüística, uso de vocabularios, etc.

AWS, Google y Microsoft tienen también sus correspondientes servicios de reconocimiento visual<sup>233</sup>: **Amazon Rekognition**, **Google Vision AI** y **Azure Cognitive Services for Vision**. Este último proporciona capacidades de clasificación de imágenes, detección de objetos, identificación facial y **reconocimiento de caracteres (OCR, Optical Character Recognition)**, soportando la extracción de texto manuscrito en diferentes idiomas. Amazon Rekognition añade el reconocimiento de gestos y emociones en la detección facial y la identificación de celebridades. Como Google Vision AI, también detecta la presencia de contenido explícito. Todos ellos soportan el entrenamiento de modelos a medida, proporcionando herramientas para el etiquetado de objetos y el enriquecimiento de imágenes.

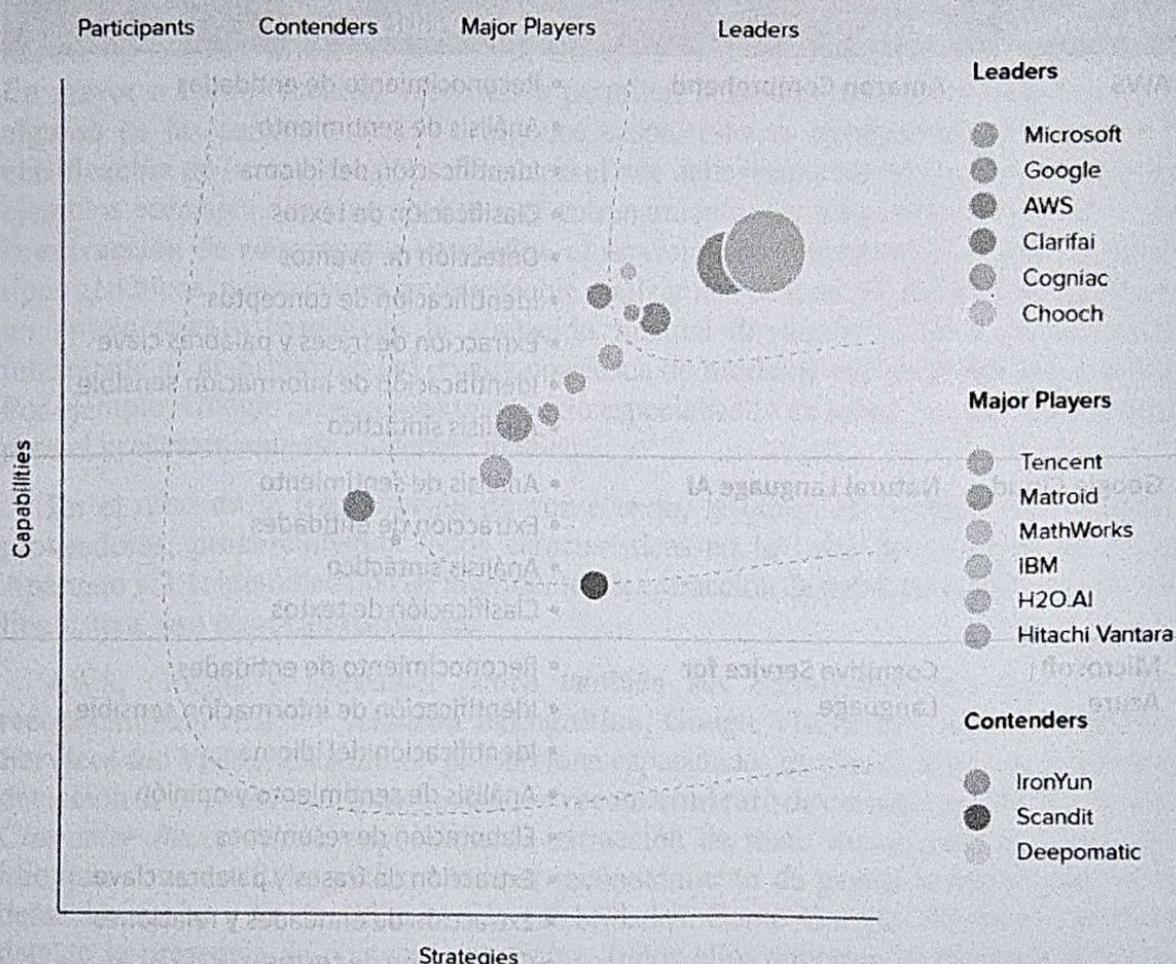
232 Los proveedores también proporcionan aplicaciones para realizar y supervisar esta anotación manual, incluyendo la monitorización del entrenamiento, la validación y el despliegue del modelo. **IBM Watson Knowledge Studio** es una opción en este sentido.

233 IBM retiró **Watson Visual Recognition** de IBM Cloud en enero de 2021, meses después de anunciar que dejaría de invertir en el desarrollo de tecnología de reconocimiento facial, debido a la inquietud generada en el momento sobre problemas de privacidad y sesgo racial en su aplicación.

Proveedor	Servicio	Características
AWS	Amazon Comprehend	<ul style="list-style-type: none"> <li>• Reconocimiento de entidades</li> <li>• Análisis de sentimiento</li> <li>• Identificación del idioma</li> <li>• Clasificación de textos</li> <li>• Detección de eventos</li> <li>• Identificación de conceptos</li> <li>• Extracción de frases y palabras clave</li> <li>• Identificación de información sensible</li> <li>• Análisis sintáctico</li> </ul>
Google Cloud	Natural Language AI	<ul style="list-style-type: none"> <li>• Análisis de sentimiento</li> <li>• Extracción de entidades</li> <li>• Análisis sintáctico</li> <li>• Clasificación de textos</li> </ul>
Microsoft Azure	Cognitive Service for Language	<ul style="list-style-type: none"> <li>• Reconocimiento de entidades</li> <li>• Identificación de información sensible</li> <li>• Identificación del idioma</li> <li>• Análisis de sentimiento y opinión</li> <li>• Elaboración de resúmenes</li> <li>• Extracción de frases y palabras clave</li> <li>• Extracción de entidades y relaciones</li> <li>• Clasificación de textos</li> </ul>
IBM Cloud	Watson Natural Language Understanding	<ul style="list-style-type: none"> <li>• Análisis sintáctico</li> <li>• Análisis de sentimiento y emoción</li> <li>• Extracción de palabras clave</li> <li>• Extracción de entidades y relaciones</li> <li>• Identificación de conceptos y categorías</li> <li>• Clasificación de textos</li> <li>• Elaboración de resúmenes</li> <li>• Extracción de roles semánticos</li> <li>• Extracción de metadatos</li> </ul>

Tabla 9-3. Principales servicios en la nube para el procesamiento del lenguaje natural.

Por su parte, IBM centra su oferta de análisis visual en **IBM Maximo Visual Inspection**. Entre otras cosas, lo interesante de esta plataforma es que permite gestionar todo el ciclo de vida de los modelos, incluyendo el inventariado y despliegue a dispositivos en la frontera. En 2022, la consultora IDC calificó a Microsoft como el líder en plataformas de visión por ordenador de propósito general (Figura 9-13).



**Figura 9-13.** Evaluación de vendedores de plataformas de visión por ordenador y ordenador de propósito general – IDC MarketScape 2022.

**Nota.** Extraído de *General-Purpose Computer Vision AI Software Platforms 2022 Vendor Assessment [Figura]*, IDC Custom Solutions, 2022, (<https://microsoft.idc-custom.com/marketscape/us49776422/>).

### 9.5.3 Soluciones para la detección y mitigación de sesgo

Por último, vamos a dar algunas referencias sobre soluciones para la identificación y corrección del sesgo en modelos predictivos. Por un lado, existen distintas herramientas dirigidas al desarrollador para examinar, documentar y corregir la existencia de comportamientos discriminantes en modelos de aprendizaje automático. **AI Fairness 360**<sup>234</sup> es una librería de código abierto desarrollada por IBM que proporciona una API (Python y R) para obtener métricas de medición de sesgo y equidad, conteniendo diversos algoritmos para su mitigación. Estos incluyen el rebalanceo y ponderación de los datos de entrenamiento, la modificación de objetivos y el ajuste de las predicciones con el fin de obtener modelos más equitativos. Entre las métricas de medición destacan

234 <https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/>

aquellas que evalúan la igualdad de oportunidades que otorga el modelo a diferentes grupos sensibles. Esto sirve para evaluar, por ejemplo, si un modelo de aprobación de préstamos exhibe ratios diferentes para hombres que para mujeres.

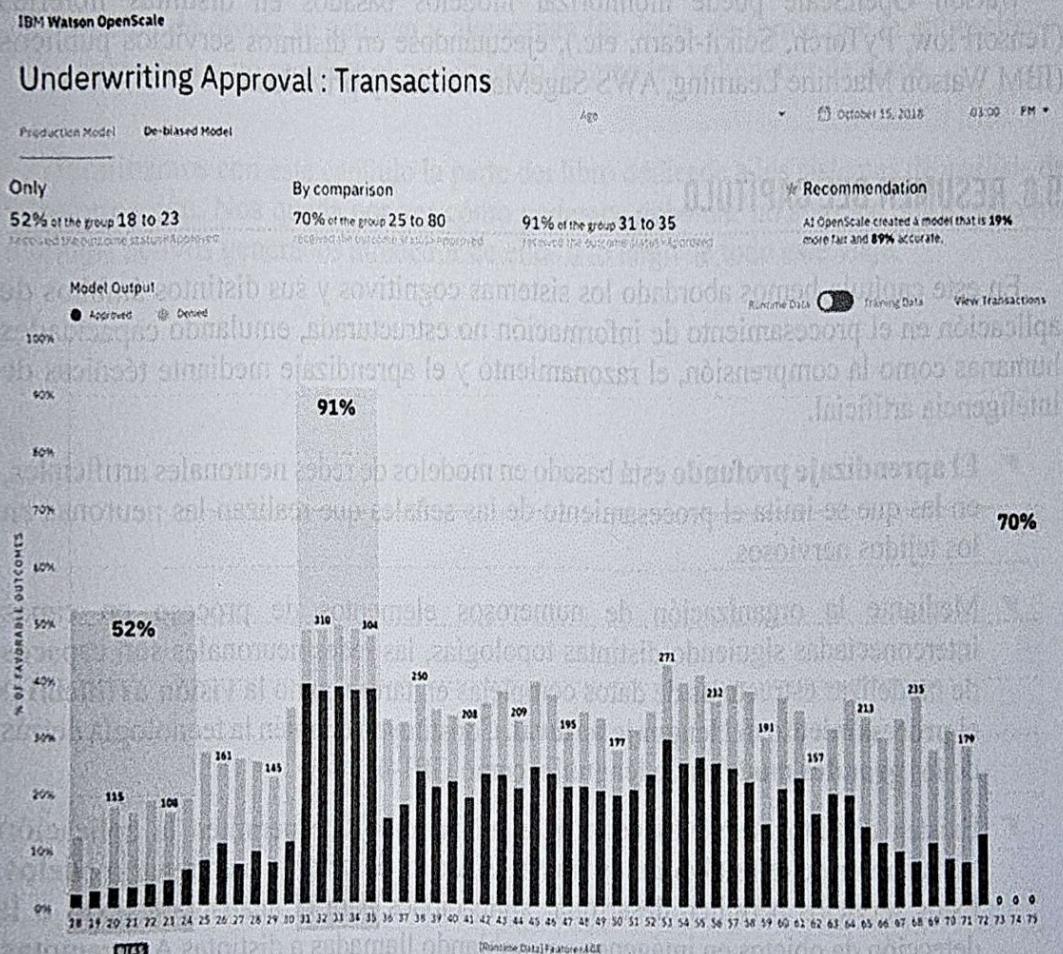


Figura 9-14. Detección de sesgo con IBM Watson OpenScale.

Basado en parte en AI Fairness 360, **IBM Watson OpenScale** es una plataforma para el seguimiento y la monitorización de modelos de aprendizaje automático en producción. Entre otras funcionalidades, incluye herramientas para la explicación de modelos, detección de desviaciones en la calidad de las predicciones, métricas de rendimiento y **monitorización de la equidad** en las inferencias (*fairness monitoring*). La Figura 9-14 muestra un ejemplo de monitorización de un modelo en producción para la aprobación de créditos. En la gráfica se puede apreciar que la concesión de estos a solicitantes de entre 18 y 23 años, el grupo monitorizado, es sustancialmente inferior que en otras franjas de edad: 52% frente al 70% en la franja de 25 a 75 años, que actúa como referencia. Por el contrario, alrededor del 91% de los solicitantes entre 31 y 35 años ven el crédito aprobado. En el primer caso, la ratio de puntuaciones favorables del modelo respecto al grupo de referencia es de  $52/70 = 0,74$ , mientras que en el segundo es de  $91/70 = 1,30$ . Si el umbral de sesgo se sitúa en el 0,80 (algo que es configurable), entonces Watson

OpenScale generaría una alarma calificando el modelo como sesgado<sup>235</sup>. En este caso, además, podría generar automáticamente un modelo alternativo empleando las técnicas y algoritmos que hemos comentado anteriormente.

Watson OpenScale puede monitorizar modelos basados en distintas librerías (TensorFlow, PyTorch, Scikit-learn, etc.), ejecutándose en distintos servicios públicos (IBM Watson Machine Learning, AWS SageMaker, etc.) y privados.

## 9.6 RESUMEN DEL CAPÍTULO

En este capítulo hemos abordado los sistemas cognitivos y sus distintos ámbitos de aplicación en el procesamiento de información no estructurada, emulando capacidades humanas como la comprensión, el razonamiento y el aprendizaje mediante técnicas de inteligencia artificial.

- ▶ **El aprendizaje profundo** está basado en modelos de redes neuronales artificiales, en las que se imita el procesamiento de las señales que realizan las neuronas en los tejidos nerviosos.
- ▶ Mediante la organización de numerosos elementos de proceso en capas interconectadas siguiendo distintas topologías, las redes neuronales son capaces de modelizar estructuras de datos complejas en tareas como la **visión artificial** o el **procesamiento del lenguaje natural**. Constituyen también la tecnología detrás de los **grandes modelos de lenguaje**, como GPT-4.
- ▶ Existen distintos **servicios cognitivos en la nube** que permiten la aplicación transparente de estos algoritmos sin necesidad de desarrollar o entrenar modelos. Estos servicios permiten desarrollar aplicaciones para el análisis de texto o la detección de objetos en imágenes ensamblando llamadas a distintas API remotas.
- ▶ En los modelos cognitivos, pero en los de aprendizaje automático en general, existe el riesgo de proporcionar **resultados sesgados**, entendidos como respuestas que de forma sistemática da el algoritmo favoreciendo unos resultados frente a otros para determinados casos. En la medida en que supone una **falta de equidad**, sobre todo al tratar información sensible (raza, edad, religión, sexo, etc.), el sesgo debe mitigarse empleando conjuntos de entrenamiento equilibrados y otras medidas correctoras a través de herramientas y soluciones especializadas.

235 Para la determinación del sesgo aquí se está empleando el llamado **impacto desigual** (*disparate impact*), una métrica que mide si la ratio del porcentaje de puntuaciones favorables de un grupo monitorizado respecto al mismo porcentaje en un grupo de referencia es inferior a un determinado umbral. En este caso, el grupo a monitorizar es el de jóvenes entre 18 y 23 años. La franja de edad entre 31 y 35 años presenta un comportamiento llamativo en el otro sentido en cuanto a la concesión de créditos; esta circunstancia sería interesante investigarla aunque, de acuerdo a la métrica empleada, no constituiría un comportamiento sesgado.

- El desarrollo de modelos cognitivos requiere una alta capacidad de proceso. Para ello existen distintos **dispositivos de hardware aceleradores** que se pueden emplear tanto en el entrenamiento como en la inferencia. En el caso de modelos de reconocimiento de imágenes, esta última suele desarrollarse en la frontera (*edge*), cerca de donde se generan y almacenan los datos, de forma que la aplicación es más ágil y no requiere el movimiento de grandes volúmenes de datos.

Terminamos con este capítulo la parte del libro dedicada a los sistemas de análisis de la información. Nos queda por ver cómo podemos gobernar no solo los datos, sino los distintos activos generados alrededor de ellos a lo largo de todo este viaje.