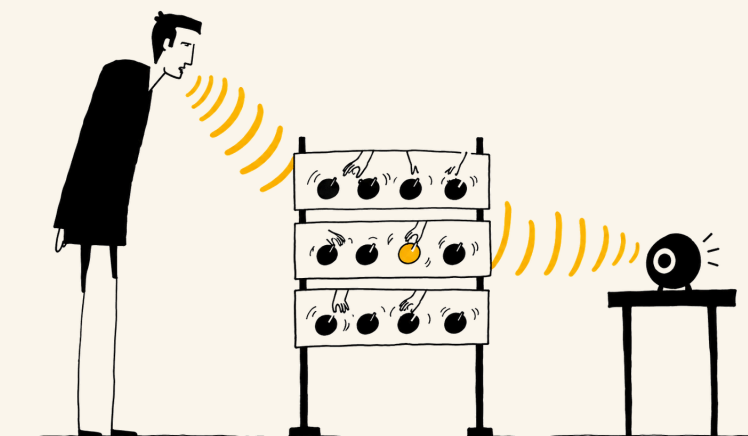


Sumérgete en el aprendizaje profundo



ton Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola · 20

Audiolecciones ilustradas

Lecturized

Sumérgete en el aprendizaje profundo

Audiolecciones ilustradas

Aston Zhang, Zachary C. Lipton, Mu Li, Alexander J. Smola · 2023

Contado como lecciones ilustradas por Lecturized
lecturized.com

Del original: Dive into Deep Learning (Cambridge University Press, 2023), d2l.ai

Índice

Introducción

Un ejemplo motivador

Componentes clave

Modelos y funciones objetivo

Algoritmos de optimización y aprendizaje supervisado

Regresión

Clasificación

Etiquetado, búsqueda y sistemas de recomendación

Aprendizaje de secuencias

Aprendizaje no supervisado y autosupervisado

Interacción con un entorno y aprendizaje por refuerzo

Raíces

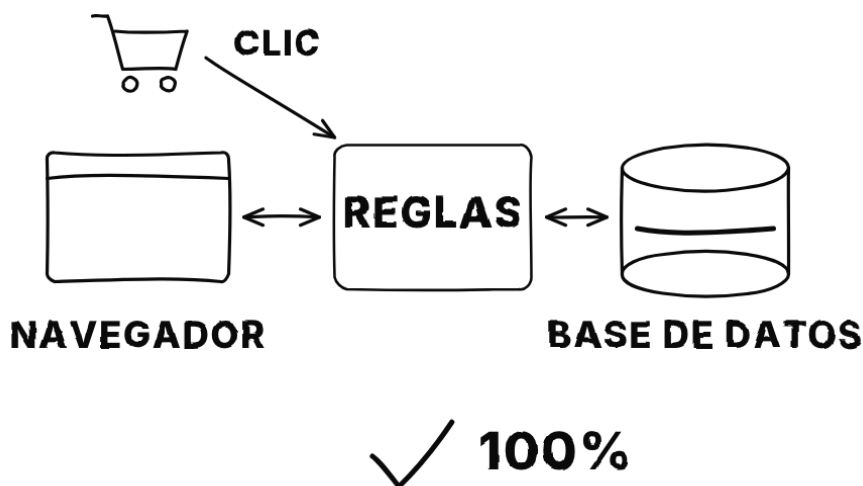
El camino hacia el aprendizaje profundo

Historias de éxito

La esencia del aprendizaje profundo

Introducción

Hasta hace poco, casi cualquier programa informático con el que usted pudiera interactuar en un día corriente estaba escrito como un conjunto rígido de reglas que especificaba con precisión cómo debía comportarse. Supongamos que quisiéramos escribir una aplicación para gestionar una plataforma de comercio electrónico. Tras reunirnos unas horas frente a una pizarra para pensar el problema, podríamos acordar las líneas generales de una solución que funcione, por ejemplo: primero, los usuarios interactúan con la aplicación a través de una interfaz que se ejecuta en un navegador web o en una aplicación móvil; segundo, nuestra aplicación se comunica con un motor de base de datos de nivel comercial para seguir el estado de cada usuario y guardar el historial de transacciones; y tercero, en el corazón de nuestra aplicación, la lógica de negocio (podría decirse, el cerebro) de la aplicación establece un conjunto de reglas que asignan a cada circunstancia imaginable la acción que nuestro programa debe realizar.



Una tienda hecha de reglas

Para construir el cerebro de nuestra aplicación, podríamos enumerar todos los sucesos habituales que el programa debe atender. Por ejemplo, cada vez que un cliente hace clic para añadir un artículo a su carrito, nuestro programa debe añadir una entrada a la tabla de carritos de la base de datos, asociando el ID de ese usuario con el ID del producto solicitado. Después podríamos intentar recorrer todos los casos límite posibles, comprobando si nuestras reglas son adecuadas y haciendo las modificaciones necesarias. ¿Qué ocurre si un usuario inicia una compra con el carrito vacío? Aunque pocos desarrolladores lo hacen del todo bien a la primera (pueden hacer falta varias pruebas para pulir los fallos), en general podemos escribir programas así y lanzarlos con confianza antes de ver a un solo cliente real. Nuestra capacidad de diseñar a mano sistemas automáticos que

hacen funcionar productos y sistemas, a menudo en situaciones nuevas, es una hazaña cognitiva notable. Y cuando usted es capaz de idear soluciones que funcionan el 100% de las veces, normalmente no debería preocuparse por el aprendizaje automático.

Por suerte para la creciente comunidad de científicos del aprendizaje automático, muchas tareas que querríamos automatizar no se doblegan tan fácilmente al ingenio humano. Imagine que se reúne frente a la pizarra con las mentes más brillantes que conoce, pero esta vez se enfrenta a uno de los problemas siguientes.



Tareas que las reglas no resuelven

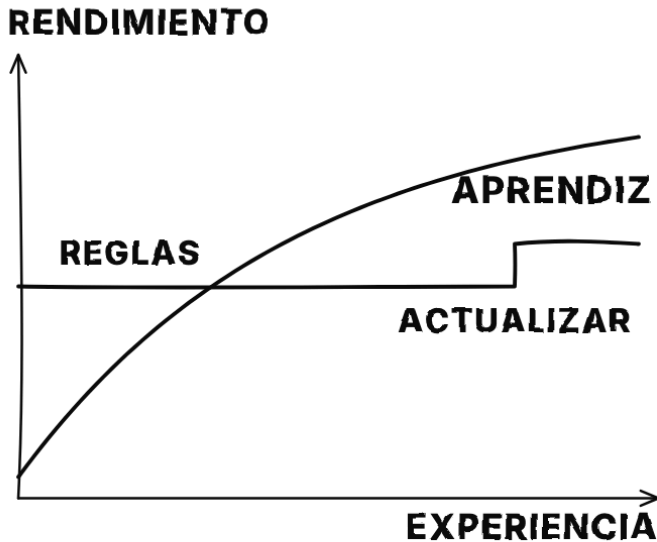
Escribir un programa que prediga el tiempo de mañana a partir de información geográfica, imágenes de satélite y el tiempo de los

últimos días. Escribir un programa que reciba una pregunta sobre un dato, expresada en texto libre, y la responda correctamente. Escribir un programa que, dada una imagen, identifique a cada persona que aparece en ella y dibuje un contorno alrededor de cada una. Escribir un programa que muestre a los usuarios productos que probablemente les gusten pero que difícilmente encontrarían al navegar por su cuenta.

Ante estos problemas, incluso los programadores de élite tendrían dificultades para escribir una solución desde cero. Las razones pueden variar. A veces el programa que buscamos sigue un patrón que cambia con el tiempo, ¡así que no hay una respuesta correcta fija! En esos casos, cualquier solución que funcione debe adaptarse con soltura a un mundo cambiante. Otras veces, la relación (por ejemplo, entre los píxeles y las categorías abstractas) puede ser demasiado complicada: exige miles o millones de cálculos y sigue principios desconocidos. En el reconocimiento de imágenes, los pasos precisos que requiere la tarea quedan fuera de nuestra comprensión consciente, aunque nuestros procesos cognitivos inconscientes la realizan sin esfuerzo.

El aprendizaje automático es el estudio de los algoritmos que pueden aprender de la experiencia. A medida que un algoritmo de aprendizaje automático acumula más experiencia, normalmente en forma de datos observados o de interacciones con un entorno, su rendimiento mejora. Compárelo con nuestra plataforma de comercio electrónico determinista, que sigue la misma lógica de negocio por mucha experiencia que se acumule, hasta que los propios desarrolladores aprenden algo y deciden que es hora de actualizar el software. En este libro le enseñaremos los fundamentos del aprendizaje automático, centrándonos en particular en el aprendizaje profundo, un potente conjunto de técnicas que impulsa innovaciones en campos tan

diversos como la visión por computador, el procesamiento del lenguaje natural, la sanidad y la genómica.



Aprender mejora con la experiencia

Preguntas

1. Según el texto, ¿cuándo no debería usted preocuparse normalmente por el aprendizaje automático?
 - a. Cuando el programa vaya a funcionar en un navegador web
 - b. Cuando pueda idear reglas que funcionen el 100% de las veces

- c. Cuando el programa deba guardar registros de las transacciones
 - d. Cuando los desarrolladores aún no hayan visto a un cliente real
2. Según la primera razón del texto, ¿por qué un conjunto fijo de reglas fallaría al recomendar productos que gusten a los compradores?
- a. Porque la base de datos no puede guardar suficientes productos
 - b. Porque el programa tiene que funcionar tanto en teléfonos como en ordenadores
 - c. Porque el patrón puede cambiar con el tiempo, así que no hay una respuesta correcta fija
 - d. Porque los compradores nunca hacen clic en los productos recomendados
3. ¿Por qué es difícil programar a mano el reconocimiento de imágenes, aunque las personas lo hagan sin esfuerzo?
- a. Los pasos precisos quedan fuera de nuestra comprensión consciente
 - b. Las imágenes ocupan demasiada memoria
 - c. Solo unas pocas personas reconocen caras con fiabilidad
 - d. Las imágenes cambian demasiado despacio para aprender de ellas
4. ¿Qué distingue a un algoritmo de aprendizaje automático de la plataforma de comercio electrónico descrita en el texto?
- a. No tiene ninguna regla dentro

- b. Funciona más rápido en el mismo ordenador
- c. Nunca necesita desarrolladores que lo construyan
- d. Su rendimiento mejora a medida que acumula experiencia

Respuestas

1. **b)** Cuando usted es capaz de idear soluciones que funcionan el 100% de las veces, normalmente no debería preocuparse por el aprendizaje automático.
2. **c)** A veces el patrón que buscamos cambia con el tiempo, así que cualquier solución que funcione debe adaptarse con soltura a un mundo cambiante.
3. **a)** Los pasos precisos que requiere la tarea quedan fuera de nuestra comprensión consciente, aunque nuestros procesos inconscientes la realizan sin esfuerzo.
4. **d)** Un algoritmo de aprendizaje automático mejora a medida que acumula experiencia, mientras que la plataforma sigue la misma lógica de negocio hasta que los desarrolladores la actualizan.

Ejercicios

1. Tome el problema de predecir el tiempo de mañana a partir de información geográfica, imágenes de satélite y el tiempo pasado. Con las dos razones que da el texto, explique por qué a los programadores les costaría escribirlo como un conjunto fijo de reglas.

Pista: Pregúntese si el patrón se mantiene igual con el tiempo y lo complicada que es la relación entre las imágenes y un pronóstico.

Respuesta: La relación entre las imágenes de satélite, el tiempo pasado y el de mañana es demasiado complicada para detallarla: exige una enorme cantidad de cálculos que siguen principios que nadie puede escribir como reglas. Además, los patrones pueden cambiar con el tiempo,

así que una respuesta fija escrita una vez no se mantendría al día; una buena solución debe adaptarse a un mundo cambiante.

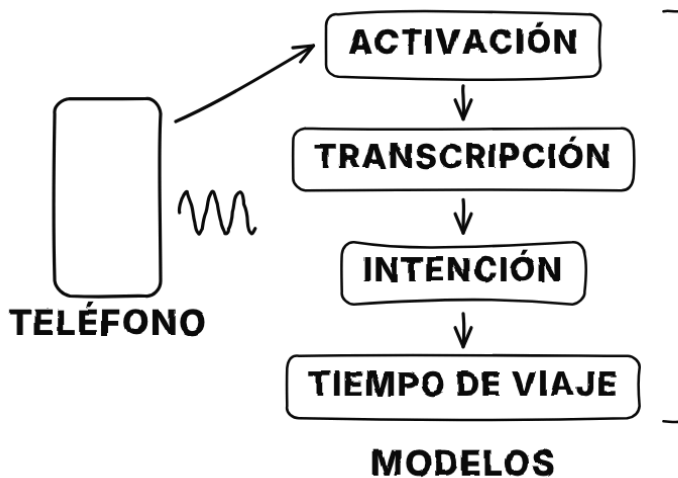
2. Explique con sus propias palabras la diferencia entre cómo mejora la plataforma de comercio electrónico y cómo mejora un algoritmo de aprendizaje automático.

Pista: Pregúntese quién aprende en cada caso.

Respuesta: La plataforma sigue la misma lógica de negocio por mucha experiencia que se acumule; solo cambia cuando los propios desarrolladores aprenden algo y actualizan el software. Un algoritmo de aprendizaje automático mejora por sí mismo a medida que reúne más experiencia, como datos observados o interacciones con un entorno.

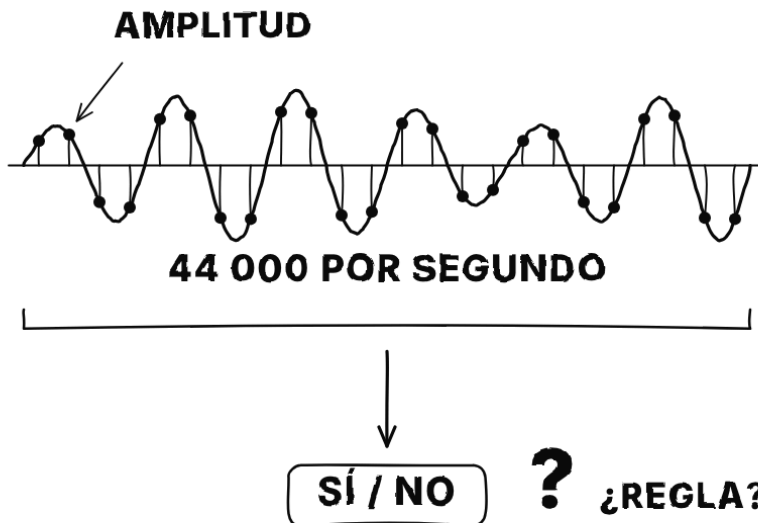
Un ejemplo motivador

Antes de empezar a escribir, los autores de este libro, como buena parte de la población trabajadora, necesitaban un café. Subimos al coche y nos pusimos en marcha. Con un iPhone, Alex dijo «Hey Siri», despertando el sistema de reconocimiento de voz del teléfono. Luego Mu ordenó: «indicaciones para ir a la cafetería Blue Bottle». El teléfono mostró rápidamente la transcripción de su orden. También reconoció que estábamos pidiendo indicaciones y abrió la aplicación de mapas para atender la petición. Una vez abierta, la aplicación de mapas encontró varias rutas. Junto a cada ruta, el teléfono mostró un tiempo de viaje previsto. Aunque esta historia se inventó por comodidad pedagógica, muestra que en apenas unos segundos nuestras interacciones cotidianas con un teléfono inteligente pueden poner en marcha varios modelos de aprendizaje automático.



Varios modelos en un teléfono

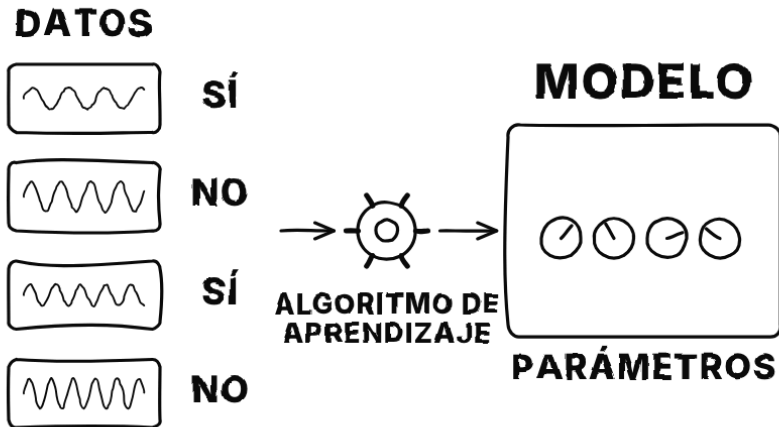
Imagine que solo tiene que escribir un programa que responda a una palabra de activación como «Alexa», «OK Google» o «Hey Siri». Intente programarlo usted solo en una habitación, sin nada más que un ordenador y un editor de código, como se ilustra en la figura. ¿Cómo escribiría un programa así a partir de primeros principios? Piénselo... el problema es difícil. Cada segundo, el micrófono recoge aproximadamente 44 000 muestras. Cada muestra es una medida de la amplitud de la onda sonora. ¿Qué regla podría pasar de forma fiable de un fragmento de audio en bruto a una predicción segura, sí o no, sobre si el fragmento contiene la palabra de activación? Si se ha atascado, no se preocupe. Nosotros tampoco sabemos escribir un programa así desde cero. Por eso usamos el aprendizaje automático.



Del audio en bruto a sí o no

Este es el truco. A menudo, aunque no sepamos decirle explícitamente a un ordenador cómo pasar de las entradas a las salidas, nosotros mismos sí somos capaces de realizar esa hazaña cognitiva. Dicho de otro modo, aunque usted no sepa programar un ordenador para que reconozca la palabra «Alexa», usted mismo sí es capaz de reconocerla. Armados con esta capacidad, podemos reunir un enorme conjunto de datos con ejemplos de fragmentos de audio y sus etiquetas, que indican qué fragmentos contienen la palabra de activación. En el enfoque del aprendizaje automático que domina hoy, no intentamos diseñar explícitamente un sistema que reconozca palabras de activación. En su lugar, definimos un programa flexible cuyo comportamiento está determinado por una serie de parámetros. Después usamos el conjunto de datos para encontrar los mejores

valores posibles de los parámetros, es decir, los que mejoran el rendimiento del programa según una medida de rendimiento elegida.

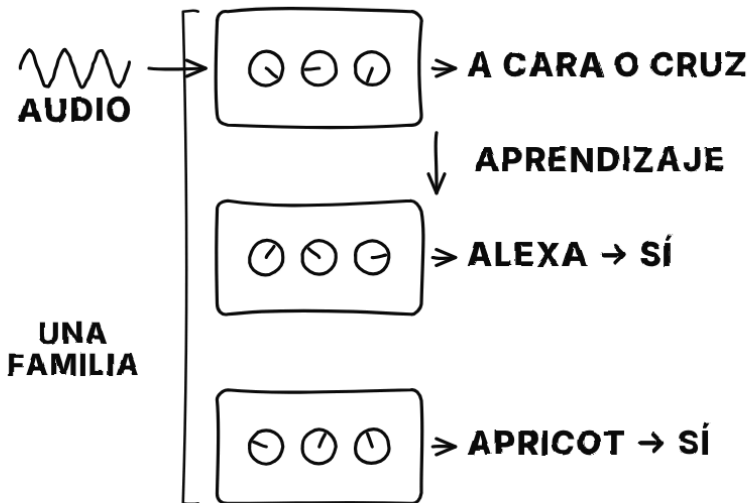


Datos, perillas y algoritmo de aprendizaje

Puede imaginar los parámetros como perillas que podemos girar para modificar el comportamiento del programa. Una vez fijados los parámetros, llamamos al programa modelo. El conjunto de todos los programas distintos (correspondencias entre entradas y salidas) que podemos obtener solo con manipular los parámetros se llama familia de modelos. Y el «metaprograma» que usa nuestro conjunto de datos para elegir los parámetros se llama algoritmo de aprendizaje.

Antes de poner en marcha el algoritmo de aprendizaje, tenemos que definir el problema con precisión, fijando la naturaleza exacta de las

entradas y las salidas, y elegir una familia de modelos adecuada. En este caso, nuestro modelo recibe como entrada un fragmento de audio y genera como salida una elección entre sí y no. Si todo sale según lo previsto, las conjeturas del modelo sobre si el fragmento contiene la palabra de activación serán normalmente correctas.

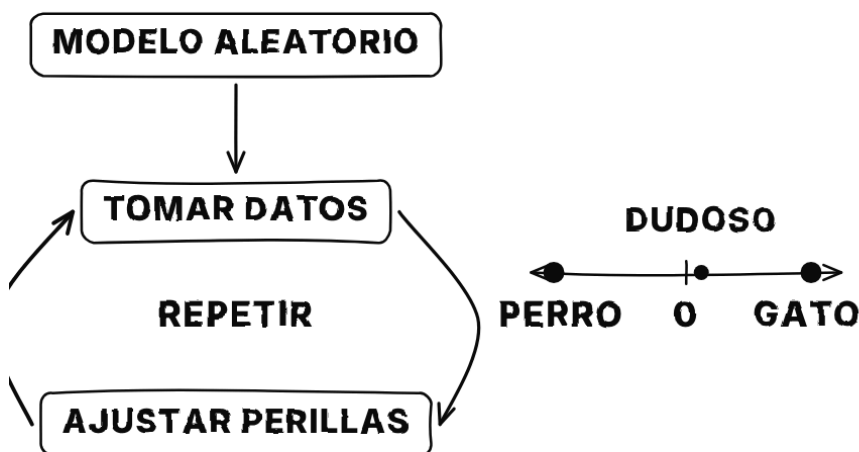


Una familia, muchos ajustes

Si elegimos la familia de modelos adecuada, debería existir una posición de las perillas con la que el modelo responda «sí» cada vez que oiga la palabra «Alexa». Como la elección de la palabra de activación es arbitraria, probablemente necesitaremos una familia de modelos lo bastante rica como para que, con otra posición de las perillas, responda «sí» solo al oír la palabra «Apricot». Esperamos que la misma familia de modelos sirva para reconocer «Alexa» y para

reconocer «Apricot», porque intuitivamente parecen tareas similares. Sin embargo, podríamos necesitar una familia de modelos completamente distinta si quisiéramos tratar entradas o salidas fundamentalmente diferentes, por ejemplo, pasar de imágenes a pies de foto, o de frases en inglés a frases en chino.

Como puede suponer, si ponemos todas las perillas al azar, es poco probable que nuestro modelo reconozca «Alexa», «Apricot» o cualquier otra palabra inglesa. En el aprendizaje automático, el aprendizaje es el proceso por el que descubrimos la posición correcta de las perillas para obtener del modelo el comportamiento deseado. Dicho de otro modo, entrenamos nuestro modelo con datos. Como se muestra en la figura, el proceso de entrenamiento suele ser el siguiente.



Bucle de entrenamiento y puntuación de gato

Primero, se parte de un modelo inicializado al azar que no sabe hacer nada útil. Segundo, se toma una parte de los datos (por ejemplo, fragmentos de audio y sus etiquetas de sí o no). Tercero, se ajustan las perillas para que el modelo funcione mejor en esos ejemplos. Cuarto, se repiten los pasos dos y tres hasta que el modelo sea estupendo.

En resumen, en lugar de programar un reconocedor de palabras de activación, programamos un programa que puede aprender a reconocerlas si se le presenta un gran conjunto de datos etiquetados. Este acto de determinar el comportamiento de un programa presentándole un conjunto de datos puede verse como programar con datos. Es decir, podemos «programar» un detector de gatos proporcionando a nuestro sistema de aprendizaje automático muchos

ejemplos de gatos y perros. Así, el detector aprenderá con el tiempo a emitir un número positivo muy grande si es un gato, un número negativo muy grande si es un perro, y algo más cercano a cero si no está seguro. Esto apenas roza la superficie de lo que el aprendizaje automático puede hacer. El aprendizaje profundo, que explicaremos con más detalle más adelante, es solo uno entre muchos métodos populares para resolver problemas de aprendizaje automático.

Preguntas

1. Según el texto, ¿por qué podemos crear un conjunto de datos para reconocer la palabra de activación aunque no sepamos escribir el programa?
 - a. Porque los micrófonos etiquetan el audio automáticamente
 - b. Porque nosotros mismos reconocemos la palabra de activación y podemos etiquetar los fragmentos
 - c. Porque las palabras de activación tienen una forma fija en la onda sonora
 - d. Porque el teléfono ya contiene una lista de todas las palabras
2. ¿Qué convierte un programa flexible en lo que el texto llama un modelo?
 - a. Reunir un conjunto de datos para él
 - b. Elegir una medida de rendimiento
 - c. Fijar los valores de sus parámetros
 - d. Ejecutarlo en un teléfono inteligente

3. ¿Por qué la misma familia de modelos podría servir para «Alexa» y «Apricot», pero no para convertir imágenes en pies de foto?
 - a. Los pies de foto necesitan muchos más datos que las palabras
 - b. Las imágenes no pueden guardarse como números
 - c. «Apricot» es más fácil de decir que «Alexa»
 - d. Poner pies de foto tiene entradas y salidas fundamentalmente distintas
4. En el proceso de entrenamiento que describe el texto, ¿qué pasa después de ajustar las perillas con algunos ejemplos?
 - a. Se toman más datos y se vuelve a ajustar, hasta que el modelo sea bueno
 - b. Se vuelven a poner todas las perillas al azar
 - c. Se fijan los parámetros definitivamente y se para
 - d. Se sustituye la familia de modelos por una más rica

Respuestas

1. **b)** Aunque usted no sepa programar un ordenador para que reconozca la palabra, usted mismo es capaz de reconocerla, así que puede reunir fragmentos de audio con etiquetas.
2. **c)** Los parámetros son perillas; una vez fijados los parámetros, llamamos al programa modelo.
3. **d)** Las dos palabras de activación son tareas similares, pero para entradas o salidas fundamentalmente distintas, como de imágenes a pies de foto, puede hacer falta otra familia.
4. **a)** Los pasos son: partir de un modelo aleatorio, tomar datos, ajustar las perillas para mejorar en ellos y repetir los pasos dos y tres hasta que el

modelo sea estupendo.

Ejemplos resueltos

Cuánto oye el micrófono El micrófono recoge unas 44 000 muestras por segundo. A un modelo de palabra de activación se le da un fragmento de 2 segundos de audio en bruto. ¿Cuántas medidas de amplitud recibe como entrada?

Cada muestra es una medida de la amplitud de la onda sonora, así que contamos muestras.

El micrófono toma unas 44 000 muestras en un segundo.

En 2 segundos toma $2 \times 44\,000 = 88\,000$ muestras.

El modelo debe pasar de todos estos números a una sola respuesta, sí o no, y por eso ninguna regla sencilla sirve.

Respuesta: Unas 88 000 medidas de amplitud

Ejercicios

1. El detector de gatos emite un número positivo muy grande para un gato y un número negativo muy grande para un perro. ¿Qué debería emitir para una foto borrosa sobre la que no está seguro, y por qué?

Pista: Piense en los dos extremos de la escala y en dónde queda la duda entre ellos.

Respuesta: Algo cercano a cero: un número positivo grande significa gato, uno negativo grande significa perro, y un valor cerca de cero es el centro de la escala, donde el detector no está seguro.

2. Explique con sus propias palabras qué quiere decir el texto con «programar con datos».

Pista: Compare escribir usted mismo las reglas con mostrar ejemplos al programa.

Respuesta: En lugar de escribir nosotros las reglas de un reconocedor de palabras de activación o de un detector de gatos, escribimos un programa flexible y le mostramos un gran conjunto de datos etiquetados; después, un algoritmo de aprendizaje fija sus parámetros. El comportamiento del programa lo determinan los ejemplos que le presentamos, así que son los datos los que programan.

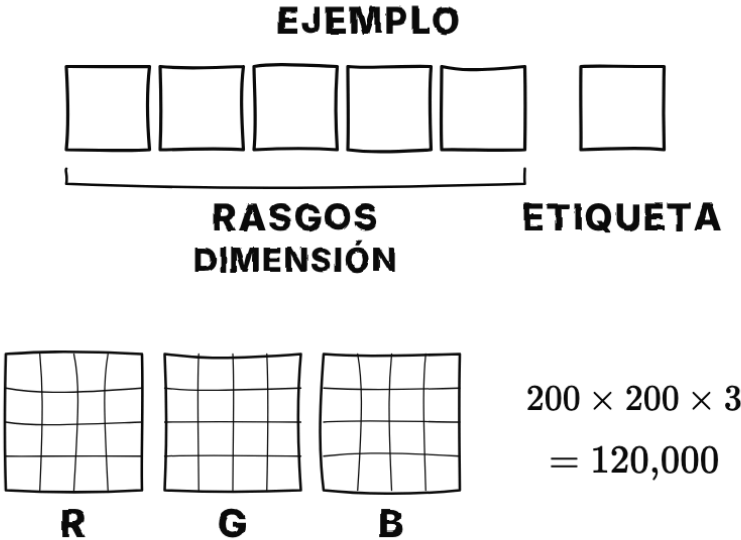
Componentes clave

En nuestro ejemplo de la palabra de activación describimos un conjunto de datos formado por fragmentos de audio y etiquetas binarias, y dimos una idea aproximada de cómo podríamos entrenar un modelo para aproximar una correspondencia entre fragmentos y clasificaciones. Este tipo de problema, en el que intentamos predecir una etiqueta desconocida concreta a partir de entradas conocidas, disponiendo de un conjunto de datos con ejemplos cuyas etiquetas sí se conocen, se llama aprendizaje supervisado. Es solo uno entre muchos tipos de problemas de aprendizaje automático. Antes de explorar otras variedades, queremos aclarar algunos componentes centrales que nos acompañarán sea cual sea el problema de aprendizaje automático que abordemos.

Primero, los datos de los que podemos aprender. Segundo, un modelo de cómo transformar los datos. Tercero, una función objetivo que cuantifica lo bien (o lo mal) que lo está haciendo el modelo. Cuarto, un algoritmo para ajustar los parámetros del modelo de modo que se optimice la función objetivo.

Quizá huelga decirlo, pero no se puede hacer ciencia de datos sin datos. Podríamos perder cientos de páginas reflexionando sobre qué son exactamente los datos, pero por ahora nos centraremos en las propiedades clave de los conjuntos de datos que nos interesarán. En general, nos interesa una colección de ejemplos. Para trabajar con los datos de forma útil, normalmente necesitamos una representación numérica adecuada. Cada ejemplo (también llamado punto de datos, instancia o muestra) suele consistir en un conjunto de atributos

llamados características (a veces covariables o entradas), a partir de los cuales el modelo debe hacer sus predicciones. En los problemas de aprendizaje supervisado, nuestro objetivo es predecir el valor de un atributo especial, llamado etiqueta (u objetivo), que no forma parte de la entrada del modelo.



Características y una etiqueta

Si trabajáramos con imágenes, cada ejemplo podría consistir en una fotografía (las características) y un número que indica la categoría a la que pertenece la fotografía (la etiqueta). La fotografía se representaría numéricamente como tres cuadrículas de valores que indican el brillo de la luz roja, verde y azul en cada píxel. Por ejemplo, una fotografía en color de 200×200 píxeles constaría de $200 \times 200 \times 3 = 120\,000$ valores numéricos.

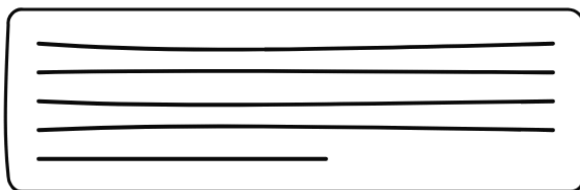
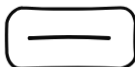
O bien podríamos trabajar con historias clínicas electrónicas y abordar la tarea de predecir la probabilidad de que un paciente sobreviva los próximos 30 días. En ese caso, nuestras características podrían ser un conjunto de atributos fácilmente disponibles y de mediciones que se registran a menudo, como la edad, las constantes vitales, las enfermedades concomitantes, la medicación actual y los procedimientos recientes. La etiqueta disponible para el entrenamiento sería un valor binario que indicara si cada paciente de los datos históricos sobrevivió dentro del plazo de 30 días.

En casos así, cuando cada ejemplo se caracteriza por el mismo número de características numéricas, decimos que las entradas son vectores de longitud fija, y llamamos dimensionalidad de los datos a la longitud (constante) de los vectores. Como puede imaginar, las entradas de longitud fija resultan cómodas: son una complicación menos de la que preocuparse. Sin embargo, no todos los datos pueden representarse fácilmente como vectores de longitud fija. Aunque cabe esperar que las imágenes de microscopio procedan de un equipo estándar, no podemos esperar que todas las imágenes extraídas de Internet tengan la misma resolución o forma. En el caso de las imágenes, podríamos recortarlas a un tamaño estándar, pero esa estrategia tiene un límite. Corremos el riesgo de perder información en las partes recortadas. Además, los datos de texto se resisten con aún más obstinación a las representaciones de longitud fija. Piense en las reseñas que dejan los clientes en sitios de comercio electrónico como Amazon, IMDb y TripAdvisor. Algunas son breves: «¡es un asco!». Otras se extienden durante páginas. Una gran ventaja del aprendizaje profundo sobre los métodos tradicionales es la relativa soltura con la que los modelos modernos manejan datos de longitud variable.

RECORTE



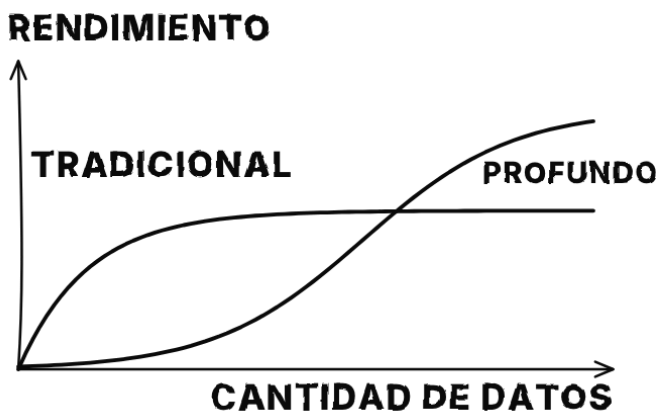
IMÁGENES



RESEÑAS

Datos que no encajan en fila

En general, cuantos más datos tenemos, más fácil es nuestro trabajo. Con más datos, podemos entrenar modelos más potentes y depender menos de suposiciones previas. El paso de datos (relativamente) pequeños a grandes datos ha contribuido en gran medida al éxito del aprendizaje profundo moderno. Para dejarlo claro: muchos de los modelos más interesantes del aprendizaje profundo no funcionan sin grandes conjuntos de datos. Otros podrían funcionar con pocos datos, pero no son mejores que los enfoques tradicionales.

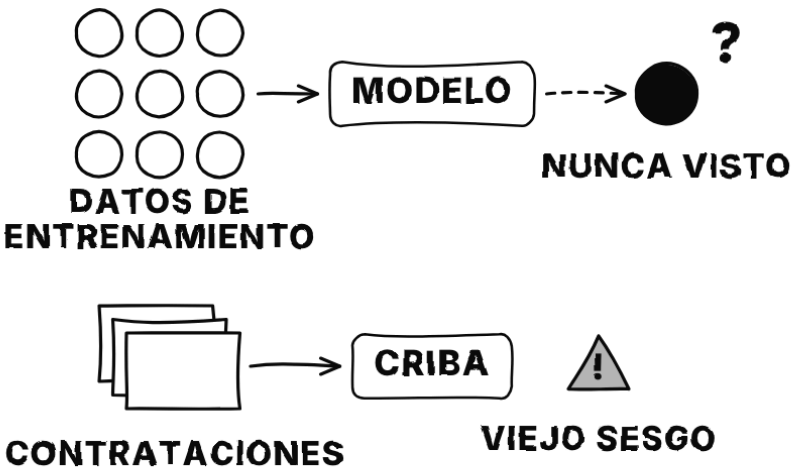


ENTRA BASURA, SALE BASURA

Más datos, modelos mayores

Por último, no basta con tener muchos datos y procesarlos con inteligencia. Necesitamos los datos adecuados. Si los datos están llenos de errores, o si las características elegidas no permiten predecir la magnitud objetivo que nos interesa, el aprendizaje va a fracasar. Lo resume bien el tópico: si entra basura, sale basura. Además, un mal rendimiento predictivo no es la única consecuencia posible. En aplicaciones sensibles del aprendizaje automático, como la vigilancia policial predictiva, la criba de currículos y los modelos de riesgo para conceder préstamos, debemos estar especialmente atentos a las consecuencias de los datos basura. Un fallo frecuente tiene que ver con conjuntos de datos en los que algunos grupos de personas no están representados en los datos de entrenamiento. Imagine aplicar un sistema de reconocimiento de cáncer de piel que nunca hubiera

visto una piel negra. También puede haber fallos cuando los datos no solo infrarrepresentan a algunos grupos, sino que reflejan prejuicios sociales. Por ejemplo, si se usan decisiones de contratación pasadas para entrenar un modelo predictivo que servirá para cribar currículos, los modelos de aprendizaje automático podrían captar y automatizar sin querer injusticias históricas. Tenga en cuenta que todo esto puede ocurrir sin que el científico de datos conspire activamente, ni siquiera sea consciente de ello.



Grupos ausentes, resultados injustos

Preguntas

1. En un problema de aprendizaje supervisado con fotografías, ¿qué parte de un ejemplo es la etiqueta?
 - a. Los valores de brillo de la luz roja, verde y azul
 - b. El número que indica la categoría de la fotografía
 - c. El ancho y el alto de la fotografía en píxeles
 - d. La cuadrícula de posiciones de los píxeles
2. ¿Por qué recortar las imágenes de Internet a un tamaño estándar tiene un límite?
 - a. Las imágenes recortadas no pueden representarse numéricamente
 - b. Recortar cambia los colores de los píxeles
 - c. Corremos el riesgo de perder información en las partes recortadas
 - d. Las imágenes recortadas siempre son demasiado pequeñas para aprender de ellas
3. ¿Por qué el texto es aún más difícil que las imágenes de representar como vectores de longitud fija?
 - a. Las reseñas van de unas pocas palabras a muchas páginas
 - b. El texto no contiene información numérica
 - c. El texto siempre es más corto que una imagen
 - d. Las reseñas se escriben en muchos alfabetos
4. Un modelo de contratación se entrena con decisiones de contratación pasadas que reflejaban prejuicios. ¿Qué advierte el texto que puede ocurrir?
 - a. El modelo se negará a hacer predicciones
 - b. El modelo corregirá las injusticias pasadas

- c. El modelo solo fallará si el científico de datos lo pretende
- d. El modelo puede captar y automatizar injusticias históricas

Respuestas

1. **b)** Cada ejemplo podría consistir en una fotografía, las características, y un número que indica la categoría a la que pertenece, la etiqueta, que no forma parte de la entrada del modelo.
2. **c)** Recortar da entradas de longitud fija, pero corremos el riesgo de perder información en las partes recortadas.
3. **a)** Las reseñas de los clientes varían muchísimo: algunas son tan breves como «¡es un asco!», otras se extienden durante páginas.
4. **d)** Si se usan decisiones de contratación pasadas para entrenar un modelo que criba currículos, podría captar y automatizar sin querer injusticias históricas, aunque nadie sea consciente de ello.

Ejemplos resueltos

Contar los números de una fotografía Una fotografía en color mide 200×200 píxeles. Representada como tres cuadrículas de valores de brillo para la luz roja, verde y azul, ¿de cuántos valores numéricos consta?

Cada cuadrícula tiene un valor por píxel: $200 \times 200 = 40\,000$ valores.

Hay tres cuadrículas, una para la luz roja, otra para la verde y otra para la azul.

Así que la fotografía son $40\,000 \times 3 = 120\,000$ valores numéricos.

Respuesta: 120 000 valores numéricos

Ejercicios

1. Cada ejemplo de un conjunto de fotografías en color de 100×100 píxeles es un vector de longitud fija. ¿Cuál es la dimensionalidad de los datos?

Pista: Cuente los valores por píxel y el número de píxeles, como con la fotografía de 200×200 .

Respuesta: 30 000: hay $100 \times 100 = 10\,000$ píxeles, cada uno con tres valores para rojo, verde y azul, y la longitud constante de los vectores es la dimensionalidad.

2. Para la tarea de predecir si un paciente sobrevivirá los próximos 30 días, nombre algunas características y la etiqueta, y diga qué tipo de valor es la etiqueta.

Pista: Las características son lo que ve el modelo; la etiqueta es lo que debe predecir.

Respuesta: Las características podrían ser la edad, las constantes vitales, las enfermedades concomitantes, la medicación actual y los procedimientos recientes; la etiqueta es si cada paciente de los datos históricos sobrevivió dentro del plazo de 30 días, un valor binario.

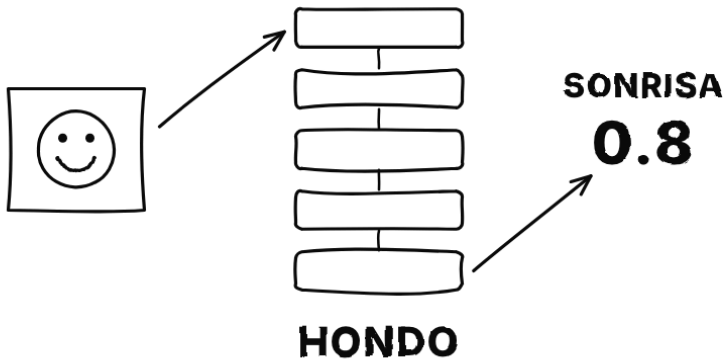
3. Explique con sus propias palabras qué significa «si entra basura, sale basura» para el aprendizaje automático, y dé una consecuencia más allá de las malas predicciones.

Pista: Piense en datos con errores, características que no predicen nada y grupos ausentes de los datos.

Respuesta: Si los datos están llenos de errores o las características no predicen el objetivo, el aprendizaje fracasará por muy inteligente que sea el procesamiento. Más allá de las malas predicciones, un sistema puede fallar injustamente a las personas, por ejemplo un reconocedor de cáncer de piel que nunca vio piel negra, o un cribador de currículos que automatiza prejuicios pasados.

Modelos y funciones objetivo

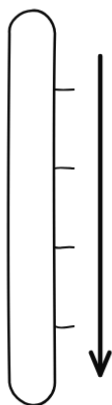
La mayor parte del aprendizaje automático consiste en transformar los datos de algún modo. Podríamos querer construir un sistema que reciba fotos y prediga cuánto sonríe la persona. O podríamos querer recibir un conjunto de lecturas de sensores y predecir hasta qué punto esas lecturas son normales o anómalas. Por modelo entendemos la maquinaria computacional que recibe datos de un tipo y devuelve predicciones de un tipo posiblemente distinto. En particular, nos interesan los modelos estadísticos que pueden estimarse a partir de datos. Aunque los modelos sencillos son perfectamente capaces de resolver problemas igual de sencillos, los problemas en los que se centra este libro llevan al límite a los métodos clásicos. El aprendizaje profundo se distingue de los enfoques clásicos sobre todo por el conjunto de modelos potentes en los que se centra. Estos modelos constan de muchas transformaciones sucesivas de los datos, encadenadas de arriba abajo, de ahí el nombre de aprendizaje profundo. De camino a los modelos profundos, también trataremos algunos métodos más tradicionales.



Un modelo, superficial y profundo

Antes presentamos el aprendizaje automático como aprender de la experiencia. Por aprender entendemos aquí mejorar en alguna tarea con el tiempo. Pero ¿quién dice qué es una mejora? Podría imaginar que proponemos actualizar nuestro modelo y que algunas personas no están de acuerdo en que nuestra propuesta sea una mejora.

PÉRDIDA



MEJOR

puntuación $\uparrow$

pérdida = $-$ puntuación $\downarrow$

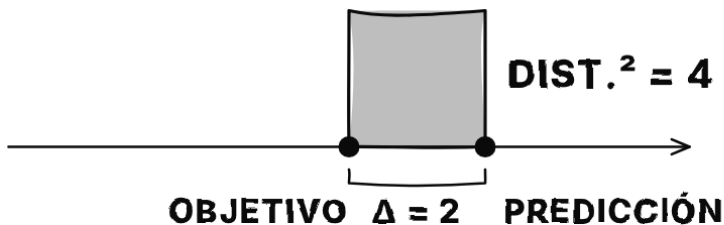
Menos es mejor

Para desarrollar un sistema matemático formal de máquinas que aprenden, necesitamos medidas formales de lo buenos (o malos) que son nuestros modelos. En el aprendizaje automático, y en la optimización en general, las llamamos funciones objetivo. Por convención, solemos definir las funciones objetivo de modo que menos sea mejor. Es solo una convención. Puede tomar cualquier función en la que más es mejor y convertirla, cambiando el signo, en una nueva función cualitativamente idéntica en la que menos es mejor. Como elegimos que menos sea mejor, a estas funciones se las llama a veces funciones de pérdida.

Al predecir valores numéricos, la función de pérdida más común es el error cuadrático, es decir, el cuadrado de la diferencia entre la

predicción y el valor objetivo real. En la clasificación, el objetivo más común es minimizar la tasa de error, es decir, la fracción de ejemplos en los que nuestras predicciones no coinciden con la realidad.

Algunos objetivos (por ejemplo, el error cuadrático) son fáciles de optimizar, mientras que otros (por ejemplo, la tasa de error) son difíciles de optimizar directamente, por no ser diferenciables o por otras complicaciones. En esos casos, lo habitual es optimizar en su lugar un objetivo sustituto.

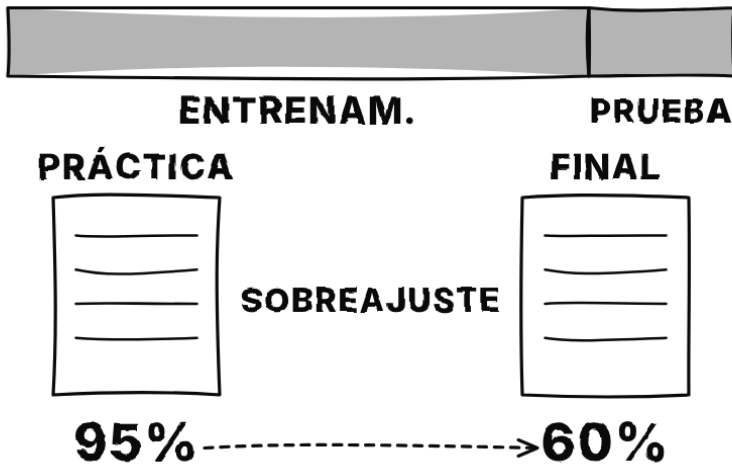


ERROR 3/10
USAR UN SUSTITUTO

Error cuadrático y tasa de error

Durante la optimización, vemos la pérdida como una función de los parámetros del modelo y tratamos el conjunto de datos de entrenamiento como una constante. Aprendemos los mejores valores de los parámetros del modelo minimizando la pérdida en un conjunto

formado por cierto número de ejemplos reunidos para el entrenamiento. Sin embargo, hacerlo bien con los datos de entrenamiento no garantiza hacerlo bien con datos no vistos. Por eso normalmente querremos dividir los datos disponibles en dos partes: el conjunto de datos de entrenamiento (o conjunto de entrenamiento), para aprender los parámetros del modelo, y el conjunto de datos de prueba (o conjunto de prueba), que se reserva para la evaluación. Al final, solemos informar de cómo funcionan nuestros modelos en ambas partes. Puede pensar en el rendimiento de entrenamiento como en las notas que saca un estudiante en los exámenes de práctica con los que se prepara para un examen final de verdad. Aunque los resultados sean alentadores, no garantizan el éxito en el examen final. Mientras estudia, el estudiante podría empezar a memorizar las preguntas de práctica, aparentando dominar el tema, pero titubear ante preguntas nunca vistas en el examen final real. Cuando un modelo funciona bien en el conjunto de entrenamiento pero no generaliza a datos no vistos, decimos que se está sobreajustando a los datos de entrenamiento.



Entrenamiento, prueba y sobreajuste

Preguntas

1. Según el texto, ¿qué distingue principalmente al aprendizaje profundo de los enfoques clásicos?
 - a. No necesita función objetivo
 - b. Sus potentes modelos encadenan muchas transformaciones sucesivas de los datos
 - c. Solo funciona con fotografías
 - d. Nunca usa modelos estadísticos

2. Tiene un objetivo en el que más es mejor, como la exactitud.
¿Cómo puede hacer que encaje en la convención habitual?
 - a. Elevándolo al cuadrado
 - b. Dividiéndolo entre el número de ejemplos
 - c. Cambiándole el signo, para que menos sea mejor
 - d. Sustituyéndolo por la tasa de error en el conjunto de prueba
3. ¿Por qué a menudo se optimiza un objetivo sustituto en lugar de la tasa de error?
 - a. La tasa de error es difícil de optimizar directamente, por no ser diferenciable
 - b. La tasa de error solo sirve para predicciones numéricas
 - c. La tasa de error siempre es mayor que el error cuadrático
 - d. La tasa de error ignora los datos de entrenamiento
4. Un modelo puntúa muy bien en su conjunto de entrenamiento pero mal en el conjunto de prueba reservado. ¿Cómo se llama esto?
 - a. Usar un objetivo sustituto
 - b. Cambiar el signo de la pérdida
 - c. Tratar el conjunto de datos como una constante
 - d. Sobreajustarse a los datos de entrenamiento

Respuestas

1. **b)** El aprendizaje profundo se distingue sobre todo por sus potentes modelos, formados por muchas transformaciones sucesivas de los datos encadenadas, de ahí su nombre.

- 2. c)** Cualquier función en la que más es mejor puede convertirse, cambiando el signo, en otra cualitativamente idéntica en la que menos es mejor.
- 3. a)** Algunos objetivos, como la tasa de error, son difíciles de optimizar directamente por no ser diferenciables o por otras complicaciones, así que se optimiza un sustituto.
- 4. d)** Cuando un modelo funciona bien en el conjunto de entrenamiento pero no generaliza a datos no vistos, se está sobreajustando, como un estudiante que memorizó las preguntas de práctica.

Ejemplos resueltos

Error cuadrático y tasa de error Un modelo predice un valor de 7 donde el valor objetivo real es 4. ¿Cuál es su error cuadrático? Por otra parte, un clasificador falla 3 de 20 ejemplos de prueba; ¿cuál es su tasa de error?

El error cuadrático es el cuadrado de la diferencia entre la predicción y el objetivo: $7 - 4 = 3$.

Al elevar al cuadrado se obtiene $3 \times 3 = 9$.

La tasa de error es la fracción de ejemplos en los que las predicciones no coinciden con la realidad.

Son 3 de 20, es decir, $3 \div 20 = 0,15$.

Respuesta: Error cuadrático 9; tasa de error 0,15 (15 %)

Ejercicios

1. Una predicción de 10 se compara con un valor real de 12. ¿Cuál es el error cuadrático, y una predicción de 14 tendría un error cuadrático mayor o igual?

Pista: Eleve la diferencia al cuadrado; el signo de la diferencia desaparece al elevar al cuadrado.

Respuesta: $10 - 12 = -2$, al cuadrado da 4. Una predicción de 14 difiere en +2, lo que también da 4, así que el error cuadrático es el mismo.

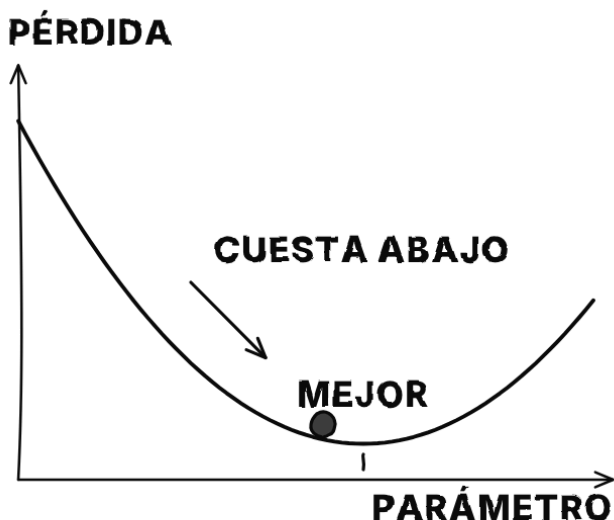
2. Explique con sus propias palabras por qué dividimos los datos en un conjunto de entrenamiento y otro de prueba, usando la comparación del estudiante y el examen.

Pista: ¿Qué es lo que no demuestra una buena nota en los exámenes de práctica?

Respuesta: Hacerlo bien con los datos de entrenamiento no garantiza hacerlo bien con datos no vistos. Un estudiante puede memorizar las preguntas de práctica y luego titubear ante preguntas nuevas del examen; del mismo modo, un modelo se entrena con el conjunto de entrenamiento y se comprueba con un conjunto de prueba reservado, para ver si generaliza en lugar de sobreajustarse.

Algoritmos de optimización y aprendizaje supervisado

Una vez que tenemos una fuente de datos y su representación, un modelo y una función objetivo bien definida, necesitamos un algoritmo capaz de buscar los mejores parámetros posibles para minimizar la función de pérdida. Los algoritmos de optimización más populares para el aprendizaje profundo se basan en un enfoque llamado descenso de gradiente. En pocas palabras, en cada paso este método comprueba, para cada parámetro, cómo cambiaría la pérdida en el conjunto de entrenamiento si se modificara ese parámetro en una cantidad muy pequeña. Después actualiza el parámetro en la dirección que reduce la pérdida.



Descenso de gradiente, paso a paso

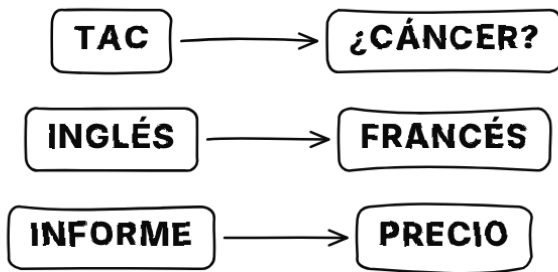
El problema de la palabra de activación de nuestro ejemplo motivador es solo uno de los muchos que puede abordar el aprendizaje automático. Para motivar más al lector y darnos un lenguaje común que nos acompañará a lo largo del libro, ofrecemos ahora una visión general del panorama de los problemas de aprendizaje automático.

El aprendizaje supervisado describe tareas en las que se nos da un conjunto de datos que contiene tanto características como etiquetas y se nos pide producir un modelo que prediga las etiquetas a partir de las características de entrada. Cada par de características y etiqueta se llama ejemplo. A veces, cuando el contexto lo deja claro, podemos usar el término ejemplos para referirnos a una colección de entradas, aunque las etiquetas correspondientes sean desconocidas. La

supervisión entra en juego porque, para elegir los parámetros, nosotros (los supervisores) proporcionamos al modelo un conjunto de datos formado por ejemplos etiquetados. En términos probabilísticos, normalmente nos interesa estimar la probabilidad condicional de una etiqueta dadas las características de entrada. Aunque es solo uno entre varios paradigmas, el aprendizaje supervisado reúne la mayoría de las aplicaciones de éxito del aprendizaje automático en la industria. En parte se debe a que muchas tareas importantes pueden describirse con nitidez como estimar la probabilidad de algo desconocido a partir de un conjunto concreto de datos disponibles.



$$P(\text{etiqueta} \mid \text{rasgos})$$



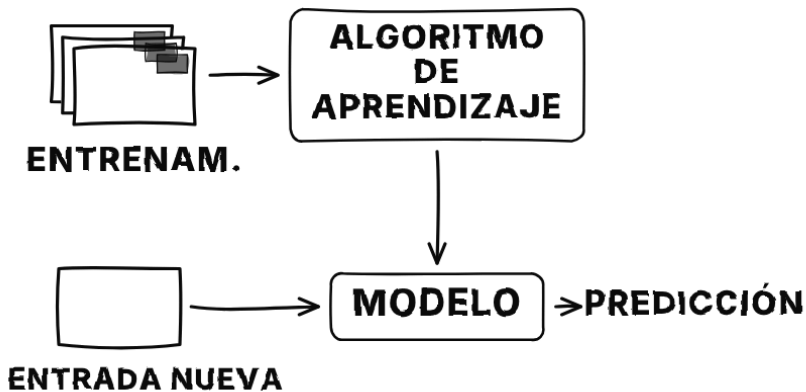
Predecir etiquetas a partir de características

Predecir si hay cáncer o no, dada una imagen de tomografía computarizada. Predecir la traducción correcta al francés, dada una

frase en inglés. Predecir el precio de una acción el mes que viene a partir de los datos de los informes financieros de este mes.

Aunque todos los problemas de aprendizaje supervisado caben en la sencilla descripción «predecir las etiquetas dadas las características de entrada», el propio aprendizaje supervisado puede adoptar formas diversas y exigir montones de decisiones de modelado, según (entre otras cosas) el tipo, el tamaño y la cantidad de las entradas y las salidas. Por ejemplo, usamos modelos distintos para procesar secuencias de longitud arbitraria y representaciones vectoriales de longitud fija. A lo largo de este libro trataremos a fondo muchos de estos problemas.

De manera informal, el proceso de aprendizaje es más o menos así. Primero, se toma una gran colección de ejemplos cuyas características se conocen y se elige de ella un subconjunto aleatorio, obteniendo para cada uno su etiqueta real. A veces estas etiquetas son datos ya recogidos (por ejemplo, ¿murió un paciente durante el año siguiente?) y otras veces hay que recurrir a anotadores humanos para etiquetar los datos (por ejemplo, asignar imágenes a categorías). Juntas, estas entradas y sus etiquetas correspondientes forman el conjunto de entrenamiento. Introducimos el conjunto de datos de entrenamiento en un algoritmo de aprendizaje supervisado, una función que recibe como entrada un conjunto de datos y devuelve otra función: el modelo aprendido. Por último, podemos dar al modelo aprendido entradas que no ha visto antes y usar sus salidas como predicciones de la etiqueta correspondiente. El proceso completo se dibuja en la figura.



De datos etiquetados a un modelo

Preguntas

1. En cada paso del descenso de gradiente, ¿qué comprueba el método para cada parámetro?
 - a. Si el parámetro es mayor que todos los demás
 - b. Cómo cambiaría la pérdida de entrenamiento si el parámetro se modificara ligeramente
 - c. Si el parámetro aparece en el conjunto de prueba
 - d. Con cuántos ejemplos se entrenó el parámetro

2. ¿De dónde viene la «supervisión» del aprendizaje supervisado?
 - a. De un agente que actúa en un entorno
 - b. De un conjunto de prueba reservado para la evaluación
 - c. De los ejemplos etiquetados que proporcionamos para elegir los parámetros
 - d. Del modelo, que comprueba sus propias respuestas
3. Según el texto, ¿qué recibe como entrada y qué devuelve un algoritmo de aprendizaje supervisado?
 - a. Recibe un conjunto de datos y devuelve otra función, el modelo aprendido
 - b. Recibe una sola entrada y devuelve una etiqueta
 - c. Recibe un modelo y devuelve un conjunto de datos
 - d. Recibe etiquetas y devuelve características

Respuestas

1. **b)** En cada paso, el método comprueba para cada parámetro cómo cambiaría la pérdida en el conjunto de entrenamiento ante una pequeña modificación, y luego lo actualiza en la dirección que reduce la pérdida.
2. **c)** Nosotros, los supervisores, proporcionamos al modelo un conjunto de datos de ejemplos etiquetados para elegir sus parámetros.
3. **a)** Un algoritmo de aprendizaje supervisado es una función que recibe como entrada un conjunto de datos y devuelve otra función: el modelo aprendido.

Ejercicios

1. Para cada tarea que enumera el texto, diga cuáles son las características de entrada y cuál es la etiqueta: detección de cáncer,

traducción y predicción del precio de una acción.

Pista: Cada tarea se formula como «predecir ..., dado ...».

Respuesta: Cáncer: las características son una imagen de tomografía computarizada y la etiqueta es cáncer o no cáncer. Traducción: las características son una frase en inglés y la etiqueta es su traducción correcta al francés. Acción: las características son los datos de los informes financieros de este mes y la etiqueta es el precio de la acción el mes que viene.

2. Describa con sus propias palabras los pasos del proceso de aprendizaje supervisado tal como los expone el texto, incluidas las dos formas de obtener las etiquetas.

Pista: Empiece por una gran colección de ejemplos cuyas características se conocen.

Respuesta: Se toma una gran colección de ejemplos con características conocidas, se elige un subconjunto aleatorio y se obtienen sus etiquetas reales, bien de datos ya recogidos, como si un paciente murió en el plazo de un año, bien de anotadores humanos que, por ejemplo, asignan imágenes a categorías. Estas entradas y etiquetas forman el conjunto de entrenamiento, que entra en un algoritmo de aprendizaje supervisado que devuelve un modelo aprendido. Después se dan al modelo entradas nuevas, no vistas, y sus salidas se usan como predicciones de sus etiquetas.

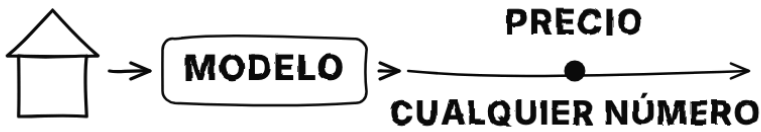
Regresión

Quizá la tarea de aprendizaje supervisado más sencilla de entender sea la regresión. Consideremos, por ejemplo, un conjunto de datos extraído de una base de datos de ventas de viviendas. Podríamos construir una tabla en la que cada fila corresponda a una casa distinta y cada columna a algún atributo relevante, como la superficie de la casa en pies cuadrados, el número de dormitorios, el número de baños y el número de minutos (a pie) hasta el centro de la ciudad. En este conjunto de datos, cada ejemplo sería una casa concreta, y el vector de características correspondiente sería una fila de la tabla. Si vive en Nueva York o San Francisco y no es el director ejecutivo de Amazon, Google, Microsoft o Facebook, el vector de características (superficie, número de dormitorios, número de baños, distancia a pie) de su vivienda podría ser algo así: [600, 1, 1, 60]. En cambio, si vive en Pittsburgh, podría parecerse más a [3000, 4, 3, 10]. Los vectores de características de longitud fija como este son esenciales para la mayoría de los algoritmos clásicos de aprendizaje automático.

UNA CASA				
	PIES ²	DORM.	BAÑOS	MIN
NUEVA YORK	600	1	1	60
PITTSBURGH	3000	4	3	10
LONG. FIJA				

Casas como filas de números

Lo que convierte un problema en una regresión es, en realidad, la forma del objetivo. Supongamos que está buscando una casa nueva. Podría querer estimar el valor justo de mercado de una casa a partir de características como las anteriores. Aquí los datos podrían ser anuncios históricos de viviendas y las etiquetas, los precios de venta observados. Cuando las etiquetas toman valores numéricos arbitrarios (aunque sea dentro de un intervalo), lo llamamos problema de regresión. El objetivo es producir un modelo cuyas predicciones se aproximen mucho a los valores reales de las etiquetas.



¿CUÁNTO? ¿CUÁNTOS?

NOTA DE PELÍCULA

DÍAS DE HOSPITAL

HORAS DE OPERACIÓN

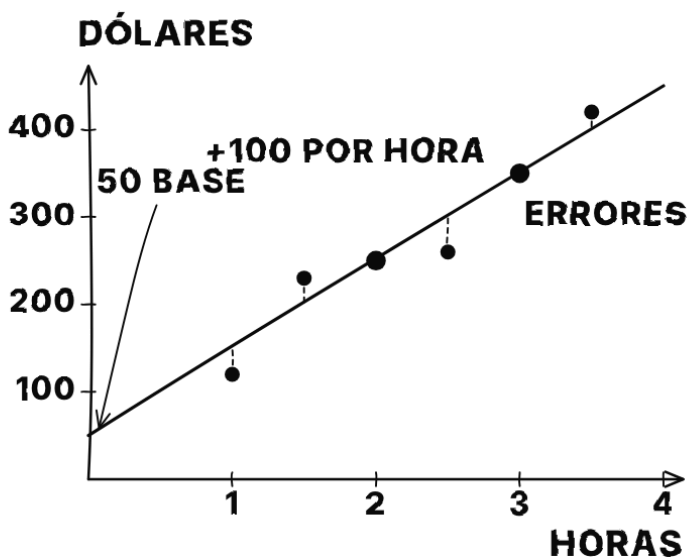
LLUVIA

Regresión: la etiqueta es un número

Muchos problemas prácticos se describen fácilmente como problemas de regresión. Predecir la valoración que un usuario dará a una película puede verse como un problema de regresión, y si en 2009 hubiera diseñado un gran algoritmo para lograrlo, podría haber ganado el premio Netflix de un millón de dólares. Predecir cuánto tiempo pasarán los pacientes en el hospital también es un problema de regresión. Una buena regla práctica es que cualquier problema de tipo ¿cuánto? o ¿cuántos? probablemente sea de regresión. Por ejemplo: ¿cuántas horas durará esta operación? ¿Cuánta lluvia caerá en esta ciudad en las próximas seis horas?

Aunque nunca haya trabajado con aprendizaje automático, probablemente ya haya resuelto un problema de regresión de manera

informal. Imagine, por ejemplo, que le arreglaron los desagües y que el fontanero pasó 3 horas quitando porquería de las tuberías. Luego le envió una factura de 350 dólares. Ahora imagine que un amigo contrató al mismo fontanero durante 2 horas y recibió una factura de 250 dólares. Si alguien le preguntara cuánto esperar en su próxima factura de limpieza, usted podría hacer algunas suposiciones razonables, como que más horas de trabajo cuestan más dólares. También podría suponer que hay un cargo base y que después el fontanero cobra por hora. Si estas suposiciones fueran ciertas, con estos dos ejemplos ya podría identificar la estructura de precios del fontanero: 100 dólares por hora más 50 dólares por presentarse en su casa. Si ha seguido hasta aquí, ya entiende la idea general de la regresión lineal.



La recta de precios del fontanero

En este caso, podíamos obtener los parámetros que coincidían exactamente con los precios del fontanero. A veces esto no es posible, por ejemplo, si parte de la variación se debe a factores ajenos a sus dos características. En esos casos, intentaremos aprender modelos que minimicen la distancia entre nuestras predicciones y los valores observados. En la mayoría de nuestros capítulos nos centraremos en minimizar la función de pérdida del error cuadrático. Como veremos más adelante, esta pérdida corresponde a suponer que nuestros datos estaban alterados por ruido gaussiano.

Preguntas

1. Según el texto, ¿qué convierte un problema en un problema de regresión?
 - a. Las características forman una tabla con una fila por ejemplo
 - b. Las etiquetas toman valores numéricos arbitrarios
 - c. El modelo debe ser lineal
 - d. Los datos proceden de una base de datos de ventas
2. Según la regla práctica del texto, ¿cuál de estas preguntas es más probablemente un problema de regresión?
 - a. ¿Es este correo spam o no?
 - b. ¿Cuál de diez animales aparece en esta foto?
 - c. ¿Cuánta lluvia caerá en esta ciudad en las próximas seis horas?
 - d. ¿Contiene este fragmento la palabra de activación?

3. ¿Cuándo no podemos encontrar parámetros que coincidan exactamente con los valores observados?
- a. Cuando parte de la variación se debe a factores ajenos a nuestras características
 - b. Cuando tenemos exactamente dos ejemplos
 - c. Cuando las etiquetas son cantidades en dólares
 - d. Cuando el fontanero cobra un cargo base

Respuestas

1. **b)** Lo que convierte un problema en regresión es la forma del objetivo: cuando las etiquetas toman valores numéricos arbitrarios, lo llamamos regresión.
2. **c)** Una buena regla práctica es que cualquier problema de tipo ¿cuánto? o ¿cuántos? probablemente sea de regresión.
3. **a)** La coincidencia exacta a veces es imposible, por ejemplo si parte de la variación se debe a factores ajenos a las características; entonces minimizamos la distancia entre predicciones y observaciones.

Ejemplos resueltos

La estructura de precios del fontanero Un fontanero cobra 350 dólares por 3 horas de trabajo y 250 dólares por 2 horas. Suponiendo un cargo base más un precio fijo por hora, ¿cuál es la estructura de precios y cuánto costarían 5 horas?

Una hora más (de 2 a 3 horas) añade $350 - 250 = 100$ dólares, así que la tarifa por hora es de 100 dólares.

Para 2 horas, la parte por horas es $2 \times 100 = 200$ dólares, así que el cargo base es $250 - 200 = 50$ dólares.

Comprobación con 3 horas: $50 + 3 \times 100 = 350$ dólares, como en la factura.

Para 5 horas: $50 + 5 \times 100 = 550$ dólares.

Respuesta: 100 dólares por hora más 50 dólares por presentarse; 5 horas costarían 550 dólares

Ejercicios

1. Un fontanero factura 180 dólares por 2 horas y 300 dólares por 4 horas. Suponiendo un cargo base más un precio por hora, halle ambos.

Pista: ¿Cuánto añaden dos horas más?

Respuesta: Dos horas más añaden $300 - 180 = 120$ dólares, es decir, 60 dólares por hora; el cargo base es $180 - 2 \times 60 = 60$ dólares.

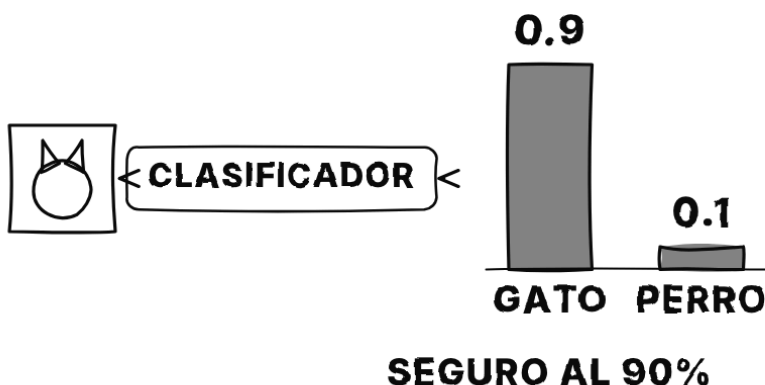
2. Una casa se describe por su superficie en pies cuadrados, dormitorios, baños y minutos a pie hasta el centro. Escriba el vector de características de una casa de 1200 pies cuadrados con 2 dormitorios, 1 baño, a 25 minutos del centro, y diga cuál sería la etiqueta en una regresión sobre ventas de viviendas.

Pista: Mantenga las características en el orden del texto.

Respuesta: El vector de características es $[1200, 2, 1, 25]$; la etiqueta sería el precio de venta observado, un valor numérico.

Clasificación

Aunque los modelos de regresión son estupendos para responder preguntas de tipo ¿cuántos?, muchos problemas no encajan cómodamente en este molde. Consideremos, por ejemplo, un banco que quiere desarrollar una función de escaneo de cheques para su aplicación móvil. Lo ideal sería que el cliente simplemente hiciera una foto del cheque y la aplicación reconociera automáticamente el texto de la imagen. Suponiendo que fuéramos capaces de recortar los trozos de imagen correspondientes a cada carácter manuscrito, la principal tarea pendiente sería determinar qué carácter, de un conjunto conocido, aparece en cada trozo. Este tipo de problemas de ¿cuál de ellos? se llaman clasificación y requieren un conjunto de herramientas distinto del de la regresión, aunque muchas técnicas se trasladan.



PÉRDIDA: ENTROPÍA CRUZADA

¿Cuál de ellos? Probabilidades por clase

En la clasificación, queremos que nuestro modelo mire las características, por ejemplo los valores de los píxeles de una imagen, y prediga a qué categoría (a veces llamada clase), de un conjunto discreto de opciones, pertenece un ejemplo. Para los dígitos manuscritos, podríamos tener diez clases, correspondientes a los dígitos del 0 al 9. La forma más sencilla de clasificación es cuando solo hay dos clases, un problema que llamamos clasificación binaria. Por ejemplo, nuestro conjunto de datos podría consistir en imágenes de animales y nuestras etiquetas podrían ser las clases gato y perro. Mientras que en la regresión buscábamos un regresor que devolviera un valor numérico, en la clasificación buscamos un clasificador, cuya salida es la clase asignada que predice.

Por razones que veremos cuando el libro se vuelva más técnico, puede ser difícil optimizar un modelo que solo puede dar una asignación categórica firme, por ejemplo, o «gato» o «perro». En estos casos, suele ser mucho más fácil expresar el modelo en el lenguaje de las probabilidades. Dadas las características de un ejemplo, nuestro modelo asigna una probabilidad a cada clase posible. Volviendo a nuestro ejemplo de clasificación de animales, en el que las clases son gato y perro, un clasificador podría ver una imagen y dar como salida una probabilidad de 0,9 de que la imagen sea un gato. Podemos interpretar este número diciendo que el clasificador está seguro al 90% de que la imagen muestra un gato. La magnitud de la probabilidad de la clase predicha transmite una idea de incertidumbre. No es la única disponible, y trataremos otras en capítulos dedicados a temas más avanzados.

Cuando hay más de dos clases posibles, llamamos al problema clasificación multiclase. Ejemplos comunes son el reconocimiento de caracteres manuscritos, con las clases 0, 1, 2 y así hasta 9, luego a, b, c, etcétera. Mientras que los problemas de regresión los abordamos intentando minimizar la función de pérdida del error cuadrático, la función de pérdida habitual en los problemas de clasificación se llama entropía cruzada, cuyo nombre quedará claro cuando presentemos la teoría de la información en capítulos posteriores.

Tenga en cuenta que la clase más probable no es necesariamente la que va a usar para tomar su decisión. Suponga que encuentra una seta preciosa en su jardín, como la de la figura.



ORONJA 0,2
SEGURA 0,8

COMER

$$0.2 \times \infty + 0.8 \times 0 = \infty$$

TIRAR ✓

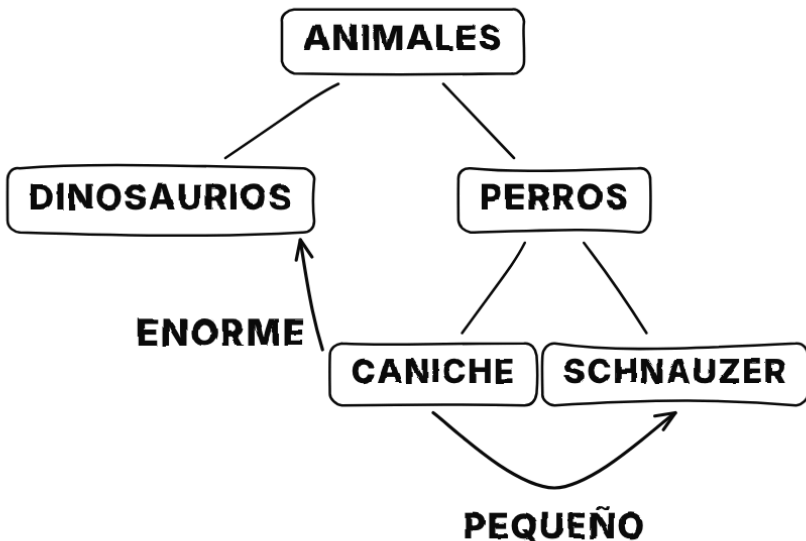
$$0.2 \times 0 + 0.8 \times 1 = 0.8$$

Pérdida esperada de una seta

Suponga ahora que ha construido un clasificador y lo ha entrenado para predecir, a partir de una fotografía, si una seta es venenosa. Digamos que nuestro clasificador de venenos da como salida que la probabilidad de que la figura muestre una oronja verde es 0,2. Dicho de otro modo, el clasificador está seguro al 80% de que nuestra seta no es una oronja verde. Aun así, habría que estar loco para comérsela. Porque el beneficio seguro de una cena deliciosa no compensa un 20% de riesgo de morir por ella. Dicho de otro modo, el efecto del riesgo incierto supera con creces al beneficio. Así, para decidir si comernos la seta, tenemos que calcular el perjuicio esperado de cada acción, que depende tanto de los resultados probables como de los beneficios o daños asociados a cada uno. En este caso, el perjuicio de comer la seta podría ser $0,2 \times \infty + 0,8 \times 0 = \infty$, mientras que la pérdida de tirarla es

$0,2 \times 0 + 0,8 \times 1 = 0,8$. Nuestra prudencia estaba justificada: como nos diría cualquier micólogo, la seta de la figura es en realidad una oronja verde.

La clasificación puede ser mucho más complicada que la simple clasificación binaria o multiclase. Por ejemplo, hay variantes de la clasificación que tratan con clases estructuradas jerárquicamente. En esos casos no todos los errores son iguales: si hemos de equivocarnos, preferiríamos confundirlo con una clase emparentada antes que con una lejana. Normalmente esto se llama clasificación jerárquica. Para inspirarse, puede pensar en Linneo, que organizó la fauna en una jerarquía.



Errores cercanos y lejanos

En la clasificación de animales, quizá no sea tan grave confundir un caniche con un schnauzer, pero nuestro modelo pagaría una enorme penalización si confundiera un caniche con un dinosaurio. Qué jerarquía es relevante puede depender de cómo piense usar el modelo. Por ejemplo, las serpientes de cascabel y las culebras de jarretera pueden estar cerca en el árbol filogenético, pero confundir una de cascabel con una de jarretera podría tener consecuencias mortales.

Preguntas

1. Leer los caracteres manuscritos de un cheque es clasificación y no regresión. ¿Por qué?
 - a. Porque el cheque es una imagen
 - b. Porque el importe del cheque es un número
 - c. Porque cada trozo debe asignarse a un carácter de un conjunto conocido
 - d. Porque el banco tiene muchos clientes
2. ¿Por qué los clasificadores suelen dar una probabilidad para cada clase en lugar de una respuesta firme?
 - a. Un modelo que solo da asignaciones firmes puede ser difícil de optimizar
 - b. Las probabilidades ocupan menos memoria que los nombres de las clases
 - c. Las respuestas firmes nunca son correctas
 - d. Las probabilidades eliminan toda incertidumbre

3. Un clasificador dice que una seta es una oronja verde con probabilidad 0,2. ¿Por qué no debería comérsela de todos modos?
 - a. Porque el clasificador se equivoca el 80% de las veces
 - b. Porque 0,2 es la clase más probable
 - c. Porque los clasificadores no pueden juzgar setas
 - d. Porque el daño esperado de comerla supera el beneficio de la cena
4. En la clasificación jerárquica, ¿qué error es el más costoso para un caniche?
 - a. Llamarlo schnauzer
 - b. Llamarlo dinosaurio
 - c. Los dos cuestan exactamente lo mismo
 - d. Ninguno, porque solo cuentan los errores binarios

Respuestas

1. **c)** La tarea pendiente es determinar qué carácter, de un conjunto conocido, aparece en cada trozo: un problema de ¿cuál de ellos?, llamado clasificación.
2. **a)** Puede ser difícil optimizar un modelo que solo puede dar una asignación categórica firme, así que es más fácil expresar el modelo en el lenguaje de las probabilidades.
3. **d)** El beneficio seguro de una cena no compensa un 20% de riesgo de morir; el perjuicio esperado de comerla es infinito, mientras que tirarla solo cuesta 0,8.
4. **b)** No todos los errores son iguales: confundir un caniche con un schnauzer quizá no sea tan grave, pero confundirlo con un dinosaurio cuesta una penalización enorme.

Ejemplos resueltos

Perjuicio esperado de la seta Un clasificador da una probabilidad de 0,2 de que una seta sea una oronja verde. Comer una oronja verde cuesta ∞ (la muerte) y comer una seta segura cuesta 0; tirar una oronja verde cuesta 0 y tirar una seta segura cuesta 1 (una cena perdida). Calcule el perjuicio esperado de cada acción.

La probabilidad de una seta segura es $1 - 0,2 = 0,8$.

Comer: $0,2 \times \infty + 0,8 \times 0 = \infty$, ya que cualquier posibilidad de un daño infinito hace infinito el total.

Tirar: $0,2 \times 0 + 0,8 \times 1 = 0,8$.

El menor perjuicio esperado es tirarla, así que tiramos la seta.

Respuesta: Comer: ∞ ; tirar: 0,8. Tírela.

Ejercicios

1. Otro clasificador dice que una seta es una oronja verde con probabilidad 0,05. Con los mismos costes (comer una oronja ∞ , tirar una segura 1), ¿cuáles son los perjuicios esperados de comerla y de tirarla?

Pista: Primero halle la probabilidad de que la seta sea segura.

Respuesta: Segura con probabilidad 0,95. Comer: $0,05 \times \infty + 0,95 \times 0 = \infty$. Tirar: $0,05 \times 0 + 0,95 \times 1 = 0,95$. Tirarla sigue siendo la mejor opción, porque cualquier riesgo de un daño infinito supera a la cena.

2. Diga para cada tarea si es clasificación binaria o multiclase, y cuántas clases tiene: gato o perro; dígitos manuscritos; dígitos manuscritos y letras minúsculas de la a a la z.

Pista: Cuente las categorías posibles.

Respuesta: Gato o perro: binaria, 2 clases. Dígitos: multiclase, 10 clases (del 0 al 9). Dígitos y letras minúsculas: multiclase, $10 + 26 = 36$ clases.

Etiquetado, búsqueda y sistemas de recomendación

Algunos problemas de clasificación encajan perfectamente en los esquemas de clasificación binaria o multiclase. Por ejemplo, podríamos entrenar un clasificador binario normal para distinguir gatos de perros. Con el estado actual de la visión por computador, podemos hacerlo fácilmente con herramientas listas para usar. Aun así, por muy preciso que llegue a ser nuestro modelo, podríamos vernos en apuros cuando el clasificador se encuentre con una imagen de los Músicos de Bremen, un popular cuento alemán protagonizado por cuatro animales, como en la figura.

MULTIETIQUETA



GATO



PERRO



BURRO



GALLO



NUBE

-

AWS

AA

-

GPU

Muchas etiquetas a la vez

Como puede ver, en la foto aparecen un gato, un gallo, un perro y un burro, con algunos árboles al fondo. Si prevemos encontrarnos con imágenes así, la clasificación multiclase podría no ser el planteamiento adecuado del problema. En su lugar, quizá queramos dar al modelo la opción de decir que la imagen muestra un gato, un perro, un burro y un gallo.

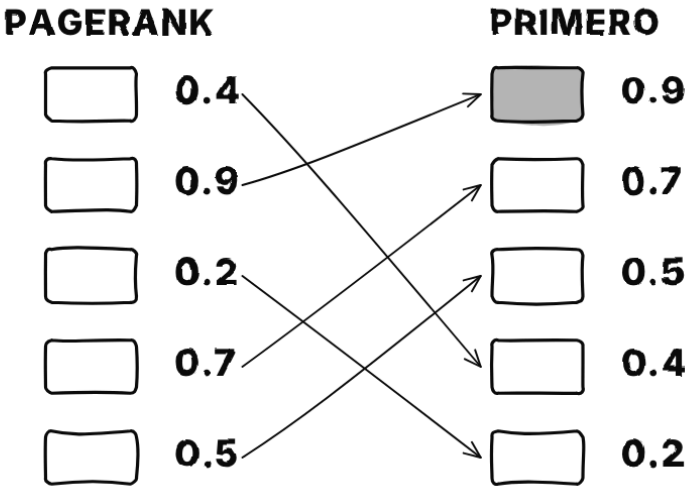
El problema de aprender a predecir clases que no son mutuamente excluyentes se llama clasificación multietiqueta. Los problemas de etiquetado automático suelen describirse mejor como clasificación multietiqueta. Piense en las etiquetas que la gente pone a las entradas de un blog técnico, por ejemplo, «aprendizaje automático», «tecnología», «gadgets», «lenguajes de programación», «Linux»,

«computación en la nube», «AWS». Un artículo típico puede llevar de 5 a 10 etiquetas. Normalmente, las etiquetas muestran cierta estructura de correlación. Es probable que las entradas sobre «computación en la nube» mencionen «AWS» y que las entradas sobre «aprendizaje automático» mencionen «GPU».

A veces estos problemas de etiquetado recurren a conjuntos de etiquetas enormes. La Biblioteca Nacional de Medicina de Estados Unidos emplea a muchos anotadores profesionales que asocian cada artículo que se indexa en PubMed con un conjunto de etiquetas extraídas de la ontología Medical Subject Headings (MeSH), una colección de unas 28 000 etiquetas. Etiquetar correctamente los artículos es importante porque permite a los investigadores hacer revisiones exhaustivas de la bibliografía. Es un proceso que lleva mucho tiempo, y normalmente pasa un año entre el archivo y el etiquetado. El aprendizaje automático puede proporcionar etiquetas provisionales hasta que cada artículo tenga una revisión manual adecuada. De hecho, durante varios años la organización BioASQ ha organizado competiciones sobre esta tarea.

En el campo de la recuperación de información, a menudo ordenamos por rango conjuntos de elementos. Tomemos como ejemplo la búsqueda web. El objetivo no es tanto determinar si una página concreta es relevante para una consulta, sino cuál, de entre un conjunto de resultados relevantes, debe mostrarse de forma más destacada a un usuario concreto. Una forma de hacerlo sería asignar primero una puntuación a cada elemento del conjunto y luego recuperar los elementos mejor puntuados. PageRank, la salsa secreta original del buscador de Google, fue un ejemplo temprano de este tipo de sistema de puntuación. Curiosamente, la puntuación de PageRank no dependía de la consulta en sí. En su lugar, se usaba un filtro de

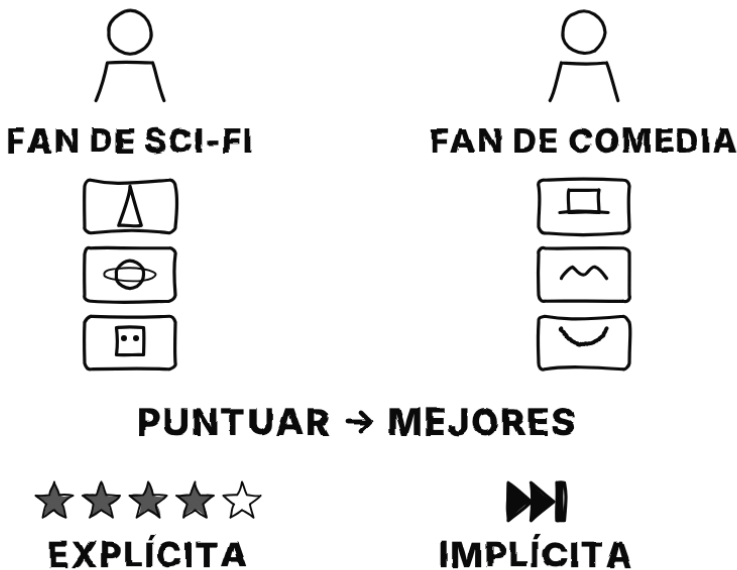
relevancia sencillo para identificar el conjunto de candidatos relevantes y luego PageRank daba prioridad a las páginas con más autoridad. Hoy en día, los buscadores usan aprendizaje automático y modelos de comportamiento para obtener puntuaciones de relevancia que dependen de la consulta. Hay congresos académicos enteros dedicados a este tema.



Puntuar y luego ordenar

Los sistemas de recomendación son otro planteamiento de problema relacionado con la búsqueda y la ordenación. Los problemas se parecen en que el objetivo es mostrar un conjunto de elementos relevantes para el usuario. La diferencia principal es el énfasis en la personalización para usuarios concretos en el contexto de los sistemas de recomendación. Por ejemplo, en las recomendaciones de películas,

la página de resultados de un aficionado a la ciencia ficción y la de un entendido en las comedias de Peter Sellers podrían diferir mucho. Problemas parecidos aparecen en otros ámbitos de recomendación, por ejemplo en productos de venta al por menor, música y noticias.



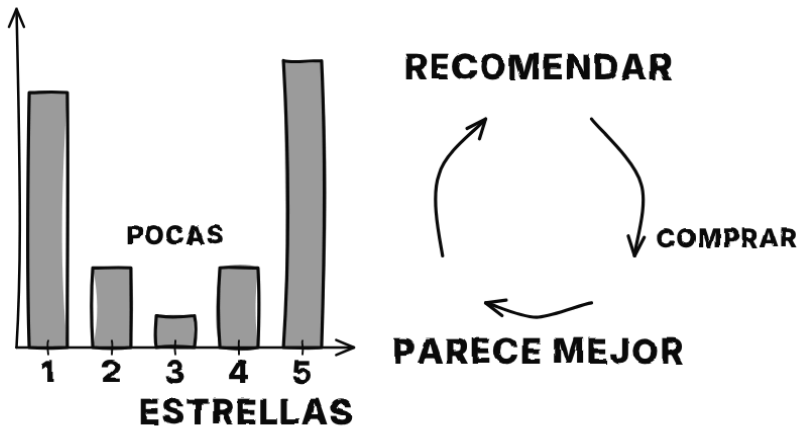
La misma tienda, usuarios distintos

En algunos casos, los clientes dan una opinión explícita y comunican cuánto les gustó un producto concreto (por ejemplo, las valoraciones y reseñas de productos en Amazon, IMDb o Goodreads). En otros casos dan una opinión implícita, por ejemplo saltándose canciones de una lista de reproducción, lo que podría indicar insatisfacción o quizá solo que la canción no venía al caso. En los planteamientos más sencillos, estos sistemas se entrenan para estimar alguna puntuación, como el

número de estrellas esperado o la probabilidad de que un usuario dado compre un artículo concreto.

Con un modelo así, para cualquier usuario podríamos recuperar el conjunto de objetos con las puntuaciones más altas y recomendárselos. Los sistemas en producción son bastante más avanzados y tienen en cuenta la actividad detallada del usuario y las características de los artículos al calcular esas puntuaciones. La figura muestra los libros de aprendizaje profundo que recomienda Amazon basándose en algoritmos de personalización ajustados para captar las preferencias de Aston.

A pesar de su enorme valor económico, los sistemas de recomendación contruidos ingenuamente sobre modelos predictivos sufren graves defectos conceptuales. Para empezar, solo observamos opiniones censuradas: los usuarios valoran preferentemente las películas que les despiertan sentimientos fuertes. Por ejemplo, en una escala de cinco puntos, puede notar que los artículos reciben muchas valoraciones de una y de cinco estrellas, pero llamativamente pocas de tres estrellas. Además, los hábitos de compra actuales suelen ser resultado del algoritmo de recomendación vigente, pero los algoritmos de aprendizaje no siempre tienen en cuenta este detalle. Así, pueden formarse bucles de retroalimentación en los que un sistema de recomendación promociona preferentemente un artículo, que luego se considera mejor (porque se compra más) y a su vez se recomienda aún con más frecuencia. Muchos de estos problemas, sobre cómo tratar la censura, los incentivos y los bucles de retroalimentación, son importantes cuestiones de investigación abiertas.



Valoraciones censuradas y bucles de retroalimentación

Preguntas

1. ¿Por qué la clasificación multiclase es un planteamiento equivocado para una foto de los Músicos de Bremen?
 - a. Porque la foto tiene árboles al fondo
 - b. Porque los animales están dibujados, no fotografiados
 - c. Porque hay varias clases a la vez, y la multiclase elige solo una
 - d. Porque hay demasiado pocas clases

2. ¿Qué tenía de curioso la forma en que PageRank puntuaba las páginas?
 - a. Sus puntuaciones no dependían de la consulta en sí
 - b. Puntuaba las páginas solo por su longitud
 - c. Ordenaba las páginas al azar
 - d. No usaba ningún filtro de relevancia
3. ¿Qué distingue principalmente a los sistemas de recomendación de la búsqueda y la ordenación?
 - a. Nunca usan puntuaciones
 - b. Solo funcionan con películas
 - c. Muestran elementos que no son relevantes
 - d. Su énfasis en la personalización para usuarios concretos
4. ¿Cómo puede formarse un bucle de retroalimentación en un sistema de recomendación?
 - a. Los usuarios dejan de valorar por completo
 - b. Un artículo que promociona el sistema se compra más, se considera mejor y se recomienda aún más
 - c. El sistema olvida las compras antiguas
 - d. Los usuarios valoran todos los artículos con tres estrellas

Respuestas

1. **c)** En la foto aparecen un gato, un gallo, un perro y un burro, así que queremos que el modelo pueda nombrarlos a todos: las clases no son mutuamente excluyentes.
2. **a)** La puntuación de PageRank no dependía de la consulta; un filtro de relevancia sencillo elegía los candidatos y luego PageRank daba prioridad a los que tenían más autoridad.

3. d) Ambos muestran elementos relevantes; la diferencia principal es el énfasis en la personalización, de modo que un aficionado a la ciencia ficción y uno de Peter Sellers ven páginas distintas.

4. b) Las compras se deben en parte a las recomendaciones vigentes; un artículo promocionado preferentemente se compra más, se considera mejor y, a su vez, se recomienda aún más.

Ejercicios

1. Explique con sus propias palabras la diferencia entre la clasificación multiclase y la multitiqueta, usando el ejemplo de las etiquetas de un blog.

Pista: Pregúntese a cuántas clases puede pertenecer un ejemplo a la vez.

Respuesta: En la clasificación multiclase, cada ejemplo pertenece exactamente a una clase de entre varias. En la multitiqueta, las clases no son mutuamente excluyentes, así que un ejemplo puede tener muchas; una entrada de blog puede llevar de 5 a 10 etiquetas a la vez, como computación en la nube y AWS juntas.

2. Explique qué significa «opinión censurada» para un recomendador de películas y qué patrón de valoraciones con estrellas produce.

Pista: ¿Quién se molesta en valorar una película?

Respuesta: Los usuarios valoran sobre todo las películas que les despiertan sentimientos fuertes, así que el sistema solo ve una parte de las opiniones. En una escala de cinco puntos hay muchas valoraciones de una y cinco estrellas y llamativamente pocas de tres, de modo que las valoraciones no dan una imagen justa de lo que siente la gente.

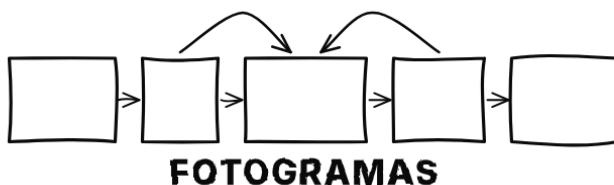
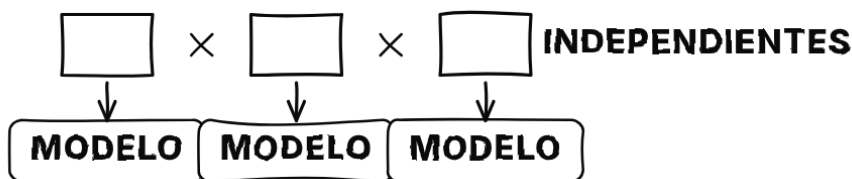
3. Dé un ejemplo de opinión explícita y otro de implícita, y explique por qué la implícita puede ser ambigua.

Pista: Piense en las valoraciones y en saltarse una canción.

Respuesta: Una opinión explícita es una valoración con estrellas o una reseña, que dice directamente cuánto le gustó a alguien un artículo. Una opinión implícita es un comportamiento, como saltarse una canción de una lista; puede significar insatisfacción, o solo que la canción no encajaba en ese momento.

Aprendizaje de secuencias

Hasta ahora hemos visto problemas en los que tenemos un número fijo de entradas y producimos un número fijo de salidas. Por ejemplo, consideramos predecir precios de viviendas a partir de un conjunto fijo de características: superficie, número de dormitorios, número de baños y tiempo de trayecto al centro. También hablamos de pasar de una imagen (de dimensión fija) a las probabilidades predichas de que pertenezca a cada una de un número fijo de clases, y de predecir las valoraciones con estrellas de las compras solo a partir del ID del usuario y del ID del producto. En estos casos, una vez entrenado el modelo, cada ejemplo de prueba se olvida en cuanto se introduce en el modelo. Supusimos que las observaciones sucesivas eran independientes, así que no hacía falta conservar ese contexto.



ENTRA SECUENCIA, SALE SECUENCIA

Entradas independientes frente a secuencias

Pero ¿cómo tratar los fragmentos de vídeo? En este caso, cada fragmento puede tener un número distinto de fotogramas. Y nuestra conjetura sobre lo que ocurre en cada fotograma podría ser mucho más firme si tenemos en cuenta los fotogramas anteriores o siguientes. Lo mismo pasa con el lenguaje. Por ejemplo, un problema popular de aprendizaje profundo es la traducción automática: recibir frases en una lengua de origen y predecir su traducción a otra lengua.

Estos problemas también aparecen en medicina. Podríamos querer un modelo que vigile a los pacientes de la unidad de cuidados intensivos y lance alertas cada vez que su riesgo de morir en las próximas 24 horas supere cierto umbral. Aquí no tiraríamos cada hora todo lo que

sabemos del historial del paciente, porque quizá no queramos hacer predicciones basadas solo en las mediciones más recientes.

Preguntas como estas están entre las aplicaciones más apasionantes del aprendizaje automático y son casos de aprendizaje de secuencias. Requieren que un modelo reciba secuencias de entradas o emita secuencias de salidas (o ambas cosas). En concreto, el aprendizaje secuencia a secuencia considera problemas en los que tanto las entradas como las salidas son secuencias de longitud variable. Algunos ejemplos son la traducción automática y la transcripción de voz a texto. Aunque es imposible considerar todos los tipos de transformaciones de secuencias, merece la pena mencionar los siguientes casos especiales.

Etiquetado y análisis sintáctico. Consiste en anotar una secuencia de texto con atributos. Aquí las entradas y las salidas están alineadas, es decir, hay el mismo número y aparecen en un orden correspondiente. Por ejemplo, en el etiquetado gramatical anotamos cada palabra de una frase con su categoría correspondiente, como «sustantivo» o «complemento directo». O bien podríamos querer saber qué grupos de palabras contiguas se refieren a entidades con nombre, como personas, lugares u organizaciones. En el ejemplo, exageradamente sencillo, de abajo, quizá solo queramos indicar si cada palabra de la frase forma parte o no de una entidad con nombre (etiquetada como «Ent»). En la frase «Tom has dinner in Washington with Sally», las palabras Tom, Washington y Sally se etiquetan «Ent», y todas las demás reciben un guion.

ALINEADAS

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



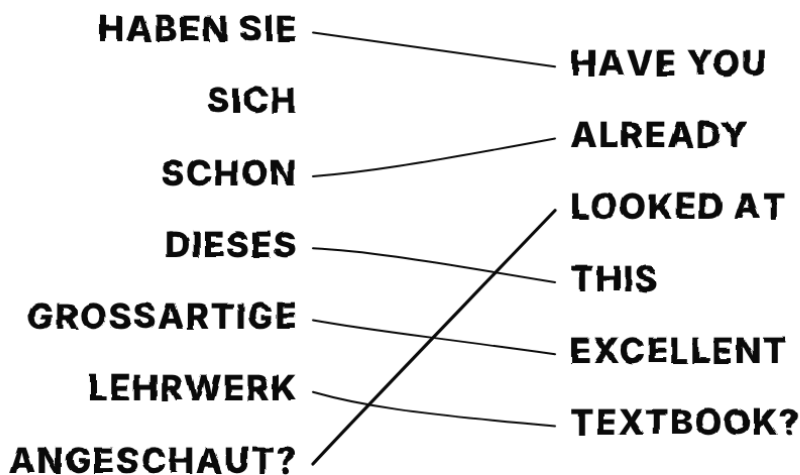
HELLO THERE EVERYONE

MUCHAS MUESTRAS → POCAS PALABRAS

Etiquetas alineadas, audio no alineado

Reconocimiento automático del habla. En el reconocimiento del habla, la secuencia de entrada es una grabación de audio de una persona que habla, como en la figura, y la salida es una transcripción de lo que dijo. La dificultad está en que hay muchos más fotogramas de audio (el sonido suele muestrearse a 8 kHz o 16 kHz) que texto, es decir, no hay una correspondencia uno a uno entre el audio y el texto, ya que miles de muestras pueden corresponder a una sola palabra hablada. Son problemas de aprendizaje secuencia a secuencia en los que la salida es mucho más corta que la entrada. Aunque los humanos son notablemente buenos reconociendo el habla, incluso en grabaciones de baja calidad, conseguir que los ordenadores hagan lo mismo es un reto formidable.

Texto a voz. Es lo inverso del reconocimiento automático del habla. Aquí la entrada es texto y la salida es un archivo de audio. En este caso, la salida es mucho más larga que la entrada.



La traducción reordena las palabras

Traducción automática. A diferencia del reconocimiento del habla, en el que las entradas y salidas correspondientes aparecen en el mismo orden, en la traducción automática los datos no alineados plantean un nuevo reto. Aquí las secuencias de entrada y salida pueden tener longitudes distintas, y las zonas correspondientes de cada secuencia pueden aparecer en otro orden. Consideremos el siguiente ejemplo ilustrativo de la peculiar tendencia del alemán a colocar los verbos al final de la frase. En alemán: «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» En inglés: «Have you already looked at this

excellent textbook?» Con la alineación equivocada: «Have you yourself already this excellent textbook looked at?»

Muchos problemas relacionados surgen en otras tareas de aprendizaje. Por ejemplo, determinar el orden en que un usuario lee una página web es un problema de análisis de maquetación bidimensional. Los problemas de diálogo presentan todo tipo de complicaciones adicionales, porque decidir qué decir a continuación requiere tener en cuenta el conocimiento del mundo real y el estado previo de la conversación a lo largo de grandes distancias temporales. Estos temas son áreas activas de investigación.

Preguntas

1. ¿Por qué un modelo que vigila a pacientes en cuidados intensivos no debería olvidar cada hora el historial del paciente?
 - a. Porque el hospital exige conservar los registros
 - b. Porque las predicciones basadas solo en las últimas mediciones pueden pasar por alto lo importante
 - c. Porque cada medición es independiente de las demás
 - d. Porque el modelo tiene un número fijo de entradas
2. En el etiquetado gramatical, ¿por qué se dice que las entradas y las salidas están alineadas?
 - a. Hay el mismo número y aparecen en un orden correspondiente
 - b. Las dos están escritas en inglés

- c. La salida es mucho más corta que la entrada
 - d. Pueden aparecer en órdenes distintos
3. ¿Qué hace del reconocimiento del habla un problema secuencia a secuencia sin correspondencia uno a uno?
- a. La transcripción es más larga que la grabación
 - b. El habla siempre se graba con baja calidad
 - c. Miles de muestras de audio pueden corresponder a una sola palabra hablada
 - d. Cada fotograma de audio corresponde exactamente a una letra
4. ¿Qué nuevo reto plantea la traducción automática que no plantea el reconocimiento del habla?
- a. La entrada es audio en lugar de texto
 - b. La salida siempre es más corta que la entrada
 - c. Las secuencias tienen una longitud fija
 - d. Las partes correspondientes pueden aparecer en otro orden

Respuestas

1. **b)** No tiraríamos todo lo que sabemos del historial del paciente, porque quizá no queramos predecir solo a partir de las mediciones más recientes.
2. **a)** En el etiquetado y el análisis sintáctico, entradas y salidas están alineadas: una etiqueta por palabra, en el mismo orden que las palabras.
3. **c)** Hay muchos más fotogramas de audio que texto, ya que miles de muestras pueden corresponder a una sola palabra, así que la salida es mucho más corta que la entrada.
4. **d)** En la traducción automática los datos no están alineados: las secuencias pueden tener longitudes distintas y las zonas correspondientes

pueden aparecer en otro orden, como los verbos alemanes al final.

Ejemplos resueltos

Cuántas muestras hay en una palabra El sonido se muestrea a 16 kHz, es decir, 16 000 muestras por segundo. Una persona dice una palabra en medio segundo. ¿Cuántas muestras de audio corresponden a esa sola palabra?

16 kHz significa 16 000 muestras cada segundo.

Medio segundo da $16\,000 \times 0,5 = 8000$ muestras.

Las 8000 muestras corresponden a una sola palabra de la transcripción, y por eso no hay correspondencia uno a uno entre el audio y el texto.

Respuesta: 8000 muestras para una palabra

Ejercicios

1. Para cada tarea, diga si la salida es más o menos igual de larga, mucho más corta o mucho más larga que la entrada: etiquetado gramatical, reconocimiento del habla, texto a voz.

Pista: Compare el número de elementos que entran y que salen.

Respuesta: Etiquetado gramatical: igual de larga, una etiqueta por palabra. Reconocimiento del habla: mucho más corta, miles de muestras se convierten en unas pocas palabras. Texto a voz: mucho más larga, unas pocas palabras se convierten en miles de muestras de audio.

2. Etiquete la frase «Sally visits Paris with Tom» con el sencillo esquema de entidades con nombre del texto, usando «Ent» para las palabras que forman parte de una entidad con nombre y un guion para las demás.

Pista: Las personas y los lugares son entidades con nombre.

Respuesta: Sally: Ent, visits: guion, Paris: Ent, with: guion, Tom: Ent.

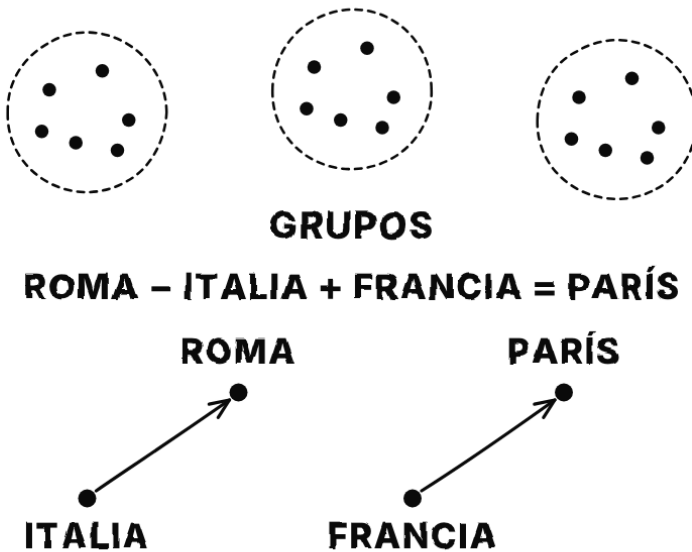
Aprendizaje no supervisado y autosupervisado

Los ejemplos anteriores se centraban en el aprendizaje supervisado, en el que damos al modelo un conjunto de datos gigantesco que contiene tanto las características como los valores de las etiquetas correspondientes. Podría pensar que el aprendiz supervisado tiene un trabajo extremadamente especializado y un jefe extremadamente dictatorial. El jefe está detrás del aprendiz y le dice exactamente qué hacer en cada situación hasta que aprende a pasar de las situaciones a las acciones. Trabajar para un jefe así suena bastante triste. Por otro lado, contentar a un jefe así es bastante fácil. Basta con reconocer el patrón lo antes posible e imitar lo que hace el jefe.

En la situación opuesta, puede ser frustrante trabajar para un jefe que no tiene ni idea de lo que quiere que usted haga. Sin embargo, si piensa ser científico de datos, más le vale acostumbrarse. ¡El jefe podría simplemente entregarle un montón gigantesco de datos y decirle que haga ciencia de datos con ellos! Suena vago porque es vago. A esta clase de problemas la llamamos aprendizaje no supervisado, y el tipo y número de preguntas que podemos plantear solo están limitados por nuestra creatividad. Trataremos las técnicas de aprendizaje no supervisado en capítulos posteriores. Para abrirle el apetito por ahora, describimos algunas de las preguntas que podría plantearse.

¿Podemos encontrar un pequeño número de prototipos que resuman con precisión los datos? Dado un conjunto de fotos, ¿podemos agruparlas en paisajes, fotos de perros, bebés, gatos y cumbres? Del

mismo modo, dada una colección de actividades de navegación de usuarios, ¿podemos agruparlos en usuarios con un comportamiento parecido? Este problema se conoce normalmente como agrupamiento.



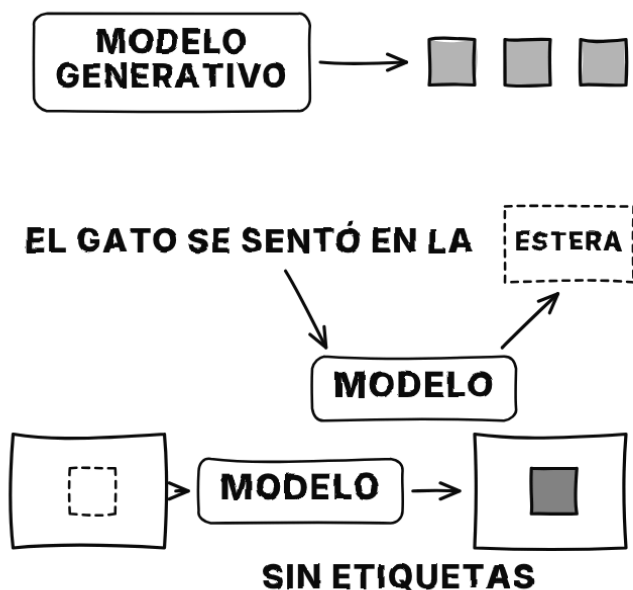
Grupos y aritmética de palabras

¿Podemos encontrar un pequeño número de parámetros que capten con precisión las propiedades relevantes de los datos? Las trayectorias de una pelota se describen bien por su velocidad, su diámetro y su masa. Los sastres han desarrollado un pequeño número de parámetros que describen con bastante precisión la forma del cuerpo humano para ajustar la ropa. Estos problemas se denominan estimación de subespacios. Si la dependencia es lineal, se llama análisis de componentes principales.

¿Existe una representación de objetos (con estructura arbitraria) en el espacio euclídeo de modo que sus propiedades simbólicas encajen bien? Esto puede usarse para describir entidades y sus relaciones, como «Roma» – «Italia» + «Francia» = «París».

¿Existe una descripción de las causas profundas de buena parte de los datos que observamos? Por ejemplo, si tenemos datos demográficos sobre precios de la vivienda, contaminación, delincuencia, ubicación, educación y salarios, ¿podemos descubrir cómo se relacionan basándonos solo en datos empíricos? De estas preguntas se ocupan los campos de la causalidad y de los modelos gráficos probabilísticos.

Otro avance reciente importante y apasionante del aprendizaje no supervisado es la llegada de los modelos generativos profundos. Estos modelos estiman la densidad de los datos, de forma explícita o implícita. Una vez entrenado, un modelo generativo puede usarse para puntuar ejemplos según lo probables que son, o para generar ejemplos sintéticos a partir de la distribución aprendida. Los primeros avances del aprendizaje profundo en el modelado generativo llegaron con la invención de los autocodificadores variacionales y continuaron con el desarrollo de las redes generativas antagónicas. Entre los avances más recientes están los flujos normalizadores y los modelos de difusión.



Aprender de los propios datos

Otro paso en el aprendizaje no supervisado ha sido el auge del aprendizaje autosupervisado, técnicas que aprovechan algún aspecto de los datos sin etiquetar para proporcionar supervisión. En el caso del texto, podemos entrenar modelos para «rellenar huecos», prediciendo palabras ocultas al azar a partir de las palabras que las rodean (sus contextos) en grandes corpus, ¡sin ningún esfuerzo de etiquetado! En el caso de las imágenes, podemos entrenar modelos para que indiquen la posición relativa entre dos regiones recortadas de la misma imagen, para que predigan una parte tapada de una imagen a partir del resto, o para que predigan si dos ejemplos son versiones alteradas de la misma imagen de partida. Los modelos autosupervisados suelen aprender representaciones que luego se

aprovechan afinando los modelos resultantes en alguna tarea posterior que nos interese.

Preguntas

1. En la analogía del jefe, ¿qué diferencia al aprendizaje no supervisado del supervisado?
 - a. El jefe comprueba cada respuesta del aprendiz
 - b. El aprendiz no recibe etiquetas, solo un montón de datos y una petición vaga
 - c. Al aprendiz se le dice exactamente qué hacer en cada situación
 - d. El aprendiz imita lo que hace el jefe
2. Agrupar usuarios por un comportamiento de navegación parecido, sin etiquetas, ¿es un ejemplo de qué?
 - a. Agrupamiento
 - b. Análisis de componentes principales
 - c. Clasificación
 - d. Afinado
3. Según el texto, ¿para qué puede usarse un modelo generativo entrenado?
 - a. Solo para agrupar fotos
 - b. Solo para predecir etiquetas
 - c. Para puntuar ejemplos según lo probables que son, o para generar ejemplos sintéticos

- d. Para descubrir las causas profundas de los precios de la vivienda
4. ¿Cómo obtiene supervisión el aprendizaje autosupervisado sin esfuerzo de etiquetado?
- a. Usa un jefe dictatorial
 - b. Pide a anotadores humanos unas pocas etiquetas
 - c. Solo funciona con datos totalmente etiquetados
 - d. Usa algún aspecto de los propios datos sin etiquetar, como palabras ocultas

Respuestas

- 1. **b)** El jefe no supervisado simplemente entrega un montón gigantesco de datos y dice que se haga ciencia de datos con ellos; no hay etiquetas que digan qué hacer.
- 2. **a)** Encontrar un pequeño número de prototipos, como grupos de usuarios con un comportamiento parecido, se conoce como agrupamiento.
- 3. **c)** Una vez entrenado, un modelo generativo puede puntuar ejemplos según lo probables que son o generar ejemplos sintéticos a partir de la distribución aprendida.
- 4. **d)** El aprendizaje autosupervisado aprovecha los datos sin etiquetar para proporcionar supervisión, por ejemplo prediciendo palabras ocultas al azar a partir de sus contextos.

Ejercicios

- 1. Explique con sus propias palabras qué muestra el ejemplo «Roma – Italia + Francia = París» sobre representar palabras en el espacio euclídeo.

Pista: Piense en qué es Roma para Italia.

Respuesta: Si las palabras se colocan como puntos en el espacio de modo que las relaciones encajen, al quitar Italia de Roma queda algo como «capital de», y al sumar Francia se obtiene la capital de Francia, París. La representación capta relaciones simbólicas entre entidades.

2. Describa dos tareas autosupervisadas para imágenes que menciona el texto, y diga de dónde viene la supervisión.

Pista: La propia imagen proporciona la respuesta.

Respuesta: Una tarea es indicar la posición relativa de dos regiones recortadas de la misma imagen; otra es predecir una parte tapada a partir del resto de la imagen (o decir si dos ejemplos son versiones alteradas de la misma imagen). Las respuestas vienen de las propias imágenes, así que no hacen falta etiquetas humanas.

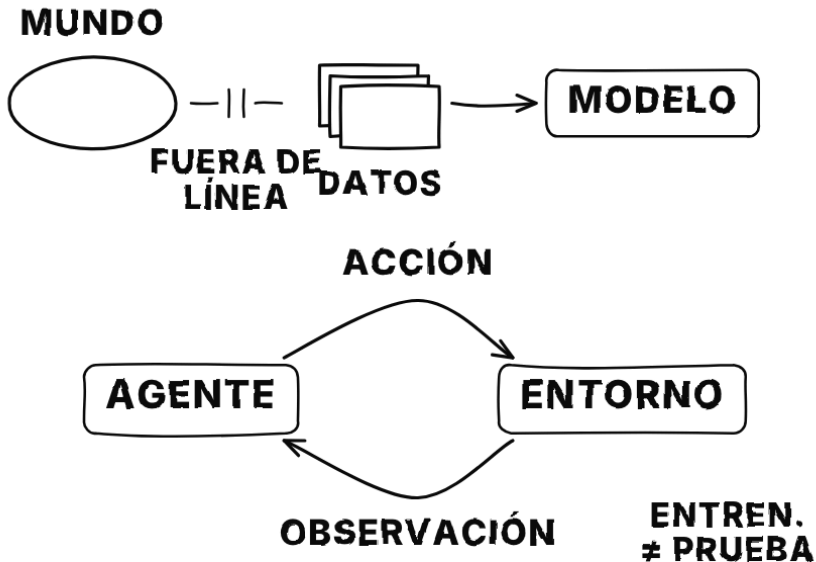
3. Explique la idea de la estimación de subespacios con el ejemplo del sastre.

Pista: ¿Cuántos números necesita un sastre?

Respuesta: Un cuerpo tiene infinidad de detalles, pero los sastres usan un pequeño número de parámetros, unas pocas medidas, que describen la forma del cuerpo lo bastante bien como para ajustar la ropa. La estimación de subespacios busca un conjunto pequeño de parámetros así que capte las propiedades relevantes de los datos.

Interacción con un entorno y aprendizaje por refuerzo

Hasta ahora no hemos hablado de dónde vienen realmente los datos ni de qué ocurre realmente cuando un modelo de aprendizaje automático genera una salida. Eso se debe a que el aprendizaje supervisado y el no supervisado no tratan estas cuestiones de una forma muy elaborada. En ambos casos, tomamos de entrada un gran montón de datos y luego ponemos en marcha nuestras máquinas de reconocimiento de patrones sin volver a interactuar nunca con el entorno. Como todo el aprendizaje tiene lugar después de que el algoritmo se desconecta del entorno, a veces se llama aprendizaje fuera de línea. Por ejemplo, el aprendizaje supervisado supone el sencillo patrón de interacción que muestra la figura.



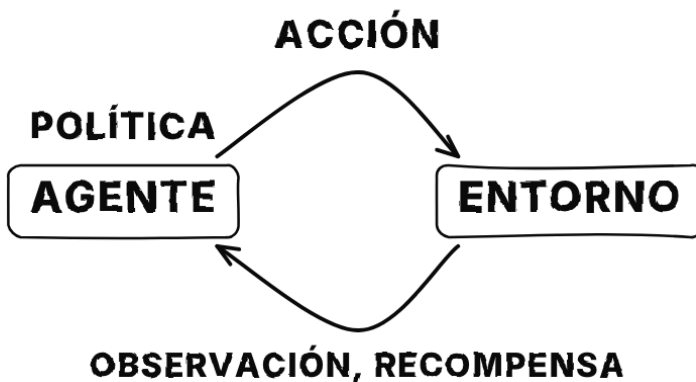
Aprendizaje fuera de línea frente a actuar

Esta sencillez del aprendizaje fuera de línea tiene su encanto. La ventaja es que podemos ocuparnos del reconocimiento de patrones de forma aislada, sin preocuparnos por las complicaciones que surgen al interactuar con un entorno dinámico. Pero este planteamiento del problema es limitado. Si crecí leyendo las novelas de robots de Asimov, probablemente se imagine agentes artificialmente inteligentes capaces no solo de hacer predicciones, sino también de actuar en el mundo. Queremos pensar en agentes inteligentes, no solo en modelos predictivos. Eso significa que tenemos que pensar en elegir acciones, no solo en hacer predicciones. A diferencia de las simples predicciones, las acciones sí influyen en el entorno. Si queremos entrenar a un agente inteligente, debemos tener en cuenta

cómo sus acciones pueden influir en sus observaciones futuras, de modo que el aprendizaje fuera de línea no es adecuado.

Tener en cuenta la interacción con un entorno abre toda una serie de nuevas preguntas de modelado. Las siguientes son solo algunos ejemplos. ¿Recuerda el entorno lo que hicimos antes? ¿Quiere el entorno ayudarnos, como un usuario que dicta texto a un reconocedor de voz? ¿Quiere el entorno vencernos, como los spammers que adaptan sus correos para esquivar los filtros de spam? ¿Tiene el entorno una dinámica cambiante? Por ejemplo, ¿se parecerán siempre los datos futuros a los pasados, o cambiarán los patrones con el tiempo, ya sea de forma natural o en respuesta a nuestras herramientas automáticas?

Estas preguntas plantean el problema del cambio de distribución, en el que los datos de entrenamiento y de prueba son distintos. Un ejemplo de esto que muchos habremos vivido es hacer exámenes escritos por el profesor cuando los deberes los preparaban sus ayudantes. A continuación describimos brevemente el aprendizaje por refuerzo, un marco rico para plantear problemas de aprendizaje en los que un agente interactúa con un entorno.



POLÍTICA: OBSERVACIÓN → ACCIÓN

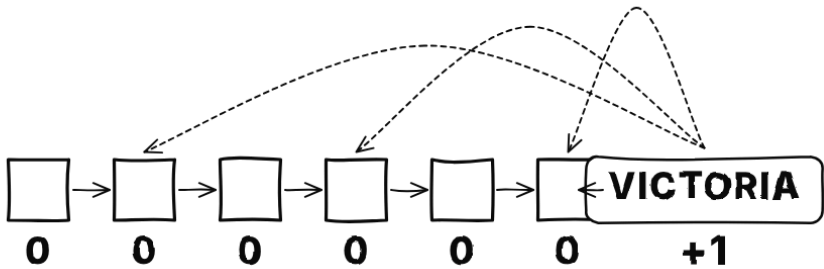
El bucle del aprendizaje por refuerzo

Si le interesa usar el aprendizaje automático para desarrollar un agente que interactúe con un entorno y realice acciones, probablemente acabará centrándose en el aprendizaje por refuerzo. Esto puede incluir aplicaciones en robótica, en sistemas de diálogo e incluso en el desarrollo de inteligencia artificial (IA) para videojuegos. El aprendizaje por refuerzo profundo, que aplica el aprendizaje profundo a problemas de aprendizaje por refuerzo, se ha disparado en popularidad. La revolucionaria red Q profunda, que venció a humanos en juegos de Atari usando solo la entrada visual, y el programa AlphaGo, que destronó al campeón del mundo del juego de tablero go, son dos ejemplos destacados.

El aprendizaje por refuerzo da un planteamiento muy general de un problema en el que un agente interactúa con un entorno a lo largo de una serie de pasos de tiempo. En cada paso de tiempo, el agente recibe del entorno alguna observación y debe elegir una acción que después se transmite de vuelta al entorno mediante algún mecanismo (a veces llamado actuador), y tras cada vuelta el agente recibe una recompensa del entorno. Este proceso se ilustra en la figura. Después el agente recibe una nueva observación, elige una nueva acción, y así sucesivamente. El comportamiento de un agente de aprendizaje por refuerzo lo rige una política. En pocas palabras, una política es simplemente una función que asigna acciones a las observaciones del entorno. El objetivo del aprendizaje por refuerzo es producir buenas políticas.

Es difícil exagerar la generalidad del marco del aprendizaje por refuerzo. Por ejemplo, el aprendizaje supervisado puede replantearse como aprendizaje por refuerzo. Supongamos que tuviéramos un problema de clasificación. Podríamos crear un agente de aprendizaje por refuerzo con una acción por cada clase. Después podríamos crear un entorno que diera una recompensa exactamente igual a la función de pérdida del problema de aprendizaje supervisado original.

¿MÉRITO?



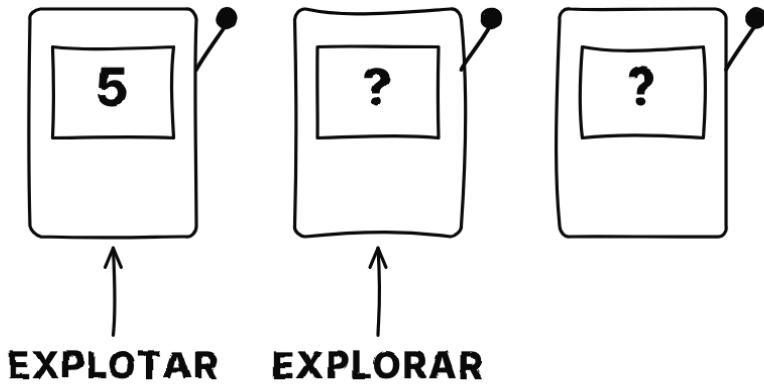
Recompensa solo al final

Además, el aprendizaje por refuerzo también puede abordar muchos problemas que el aprendizaje supervisado no puede. Por ejemplo, en el aprendizaje supervisado siempre esperamos que la entrada de entrenamiento venga acompañada de la etiqueta correcta. Pero en el aprendizaje por refuerzo no suponemos que, para cada observación, el entorno nos diga la acción óptima. En general, solo recibimos alguna recompensa. Además, puede que el entorno ni siquiera nos diga qué acciones llevaron a la recompensa.

Pensemos en el ajedrez. La única señal de recompensa real llega al final de la partida, cuando ganamos y obtenemos una recompensa de, digamos, 1, o cuando perdemos y recibimos una recompensa de, digamos, menos 1. Así que quienes aprenden por refuerzo deben

resolver el problema de la asignación de mérito: determinar a qué acciones atribuir el mérito o la culpa de un resultado. Lo mismo vale para una empleada a la que ascienden el 11 de octubre. Ese ascenso probablemente refleje una serie de acciones bien elegidas a lo largo del año anterior. Conseguir ascensos en el futuro requiere averiguar qué acciones del camino llevaron a los ascensos anteriores.

Quienes aprenden por refuerzo también pueden tener que lidiar con el problema de la observabilidad parcial. Es decir, la observación actual puede no contarle todo sobre su estado actual. Supongamos que su robot de limpieza se ha quedado atrapado en uno de los muchos armarios idénticos de su casa. Rescatar al robot implica deducir su ubicación exacta, lo que puede exigir tener en cuenta observaciones anteriores a que entrara en el armario.



BANDIDO MULTIBRAZO

Explotar o explorar

Por último, en cualquier momento, quien aprende por refuerzo puede conocer una buena política, pero puede haber muchas otras políticas mejores que el agente nunca ha probado. Quien aprende por refuerzo debe elegir constantemente entre explotar la mejor estrategia conocida (por ahora) como política, o explorar el espacio de estrategias, renunciando quizá a parte de la recompensa a corto plazo a cambio de conocimiento.

El problema general del aprendizaje por refuerzo tiene un planteamiento muy general. Las acciones influyen en las observaciones siguientes. Las recompensas solo se observan para las acciones elegidas. El entorno puede ser observable total o parcialmente. Tener en cuenta toda esta complejidad a la vez puede

ser pedir demasiado. Además, no todo problema práctico presenta toda esta complejidad. Por eso los investigadores han estudiado varios casos especiales de problemas de aprendizaje por refuerzo.

Cuando el entorno es totalmente observable, llamamos al problema de aprendizaje por refuerzo proceso de decisión de Markov. Cuando el estado no depende de las acciones anteriores, lo llamamos problema de bandido contextual. Cuando no hay estado, solo un conjunto de acciones disponibles con recompensas inicialmente desconocidas, tenemos el clásico problema del bandido multibrazo.

Preguntas

1. ¿Por qué el aprendizaje fuera de línea no es adecuado para entrenar a un agente inteligente?
 - a. Porque el aprendizaje fuera de línea no necesita datos
 - b. Porque las acciones del agente influyen en sus observaciones futuras
 - c. Porque los agentes no pueden hacer predicciones
 - d. Porque el aprendizaje fuera de línea es demasiado lento
2. Los spammers cambian sus correos para esquivar los filtros de spam. ¿Qué pregunta sobre el entorno ilustra esto?
 - a. ¿Recuerda el entorno lo que hicimos?
 - b. ¿Quiere el entorno ayudarnos?
 - c. ¿Quiere el entorno vencernos?
 - d. ¿Es el entorno totalmente observable?

3. En el ajedrez, ¿por qué quien aprende por refuerzo debe resolver el problema de la asignación de mérito?
 - a. La única recompensa real llega al final, así que debe averiguar qué jugadas merecen el mérito o la culpa
 - b. El tablero solo es visible en parte
 - c. Cada jugada recibe su propia recompensa
 - d. Las reglas cambian durante la partida
4. Cuando no hay estado, solo un conjunto de acciones con recompensas inicialmente desconocidas, ¿cómo se llama el problema?
 - a. Un proceso de decisión de Markov
 - b. Un problema de bandido contextual
 - c. Aprendizaje fuera de línea
 - d. Un problema de bandido multibrazo

Respuestas

1. **b)** Las acciones sí influyen en el entorno, así que un agente entrenado debe tener en cuenta cómo afectan sus acciones a sus observaciones futuras, algo que el aprendizaje fuera de línea ignora.
2. **c)** Los spammers que adaptan sus correos para esquivar los filtros son el ejemplo del texto de un entorno que quiere vencernos.
3. **a)** La única recompensa real, ganar o perder, llega al final de la partida, así que quien aprende debe determinar a qué acciones atribuir el mérito o la culpa del resultado.
4. **d)** Sin estado, solo con acciones disponibles con recompensas inicialmente desconocidas, tenemos el clásico problema del bandido multibrazo.

Ejercicios

1. Explique con sus propias palabras qué es una política y cuál es el objetivo del aprendizaje por refuerzo.

Pista: Una política conecta lo que ve el agente con lo que hace.

Respuesta: Una política es una función que asigna acciones a las observaciones del entorno; rige cómo se comporta el agente. El objetivo del aprendizaje por refuerzo es producir buenas políticas, que obtengan recompensas altas con el tiempo.

2. Describa cómo puede replantearse un problema de clasificación como aprendizaje por refuerzo.

Pista: Piense en cada clase como una acción.

Respuesta: Se crea un agente con una acción por cada clase y un entorno que da una recompensa igual a la pérdida del problema supervisado original. Elegir la clase correcta equivale entonces a elegir la acción que obtiene la mejor recompensa.

3. Una aplicación de comida a domicilio recomienda siempre el restaurante que mejor ha funcionado hasta ahora. Con los términos del texto, ¿qué está haciendo y qué podría perderse?

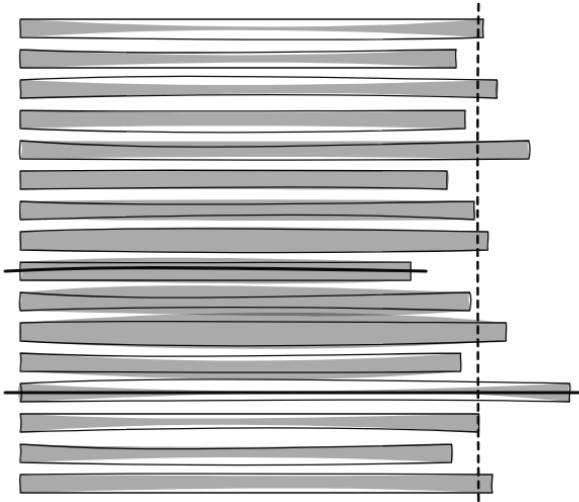
Pista: Piense en explotar y explorar.

Respuesta: Está explotando la mejor política conocida por ahora. Podría perderse restaurantes mejores que nunca ha probado; explorar supondría renunciar a parte de la recompensa a corto plazo a cambio de conocimiento sobre otras opciones.

Raíces

Acabamos de repasar un pequeño subconjunto de los problemas que puede abordar el aprendizaje automático. Para un conjunto diverso de problemas de aprendizaje automático, el aprendizaje profundo ofrece potentes herramientas para resolverlos. Aunque muchos métodos de aprendizaje profundo son inventos recientes, las ideas centrales del aprendizaje a partir de datos se llevan estudiando siglos. De hecho, los seres humanos llevan muchísimo tiempo queriendo analizar datos y predecir resultados futuros, y ese deseo está en la raíz de buena parte de las ciencias naturales y las matemáticas. Dos ejemplos son la distribución de Bernoulli, que debe su nombre a Jacob Bernoulli (1655 a 1705), y la distribución gaussiana, descubierta por Carl Friedrich Gauss (1777 a 1855). Gauss inventó, por ejemplo, el algoritmo de mínimos cuadrados, que todavía hoy se usa en multitud de problemas, desde cálculos de seguros hasta diagnósticos médicos. Herramientas como estas reforzaron el enfoque experimental de las ciencias naturales; por ejemplo, la ley de Ohm, que relaciona la corriente y la tensión en una resistencia, se describe perfectamente con un modelo lineal.

MEDIA RECORTADA



Dieciséis pies, un pie

Incluso en la Edad Media, los matemáticos tenían una aguda intuición de las estimaciones. Por ejemplo, el libro de geometría de Jacob Köbel (1460 a 1533) ilustra cómo promediar la longitud de los pies de 16 hombres adultos para estimar la longitud típica del pie en la población, como en la figura.

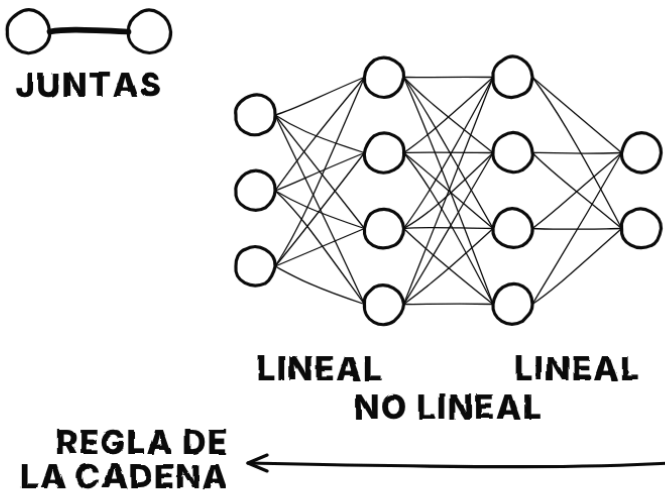
Al salir un grupo de personas de una iglesia, se pidió a 16 hombres adultos que se pusieran en fila y se les midieron los pies. Después, la suma de esas medidas se dividió entre 16 para obtener una estimación de lo que hoy llamamos un pie. Este «algoritmo» se mejoró más tarde para tratar los pies deformes: se despedía a los dos hombres con el pie más corto y el más largo y se promediaba solo sobre el resto. Es uno de los primeros ejemplos de una estimación por media recortada.

La estadística despegó de verdad con la disponibilidad y la recogida de datos. Uno de sus pioneros, Ronald Fisher (1890 a 1962), contribuyó de forma notable a su teoría y también a sus aplicaciones en genética. Muchos de sus algoritmos (como el análisis discriminante lineal) y conceptos (como la matriz de información de Fisher) siguen ocupando un lugar destacado en los fundamentos de la estadística moderna. Incluso sus recursos de datos tuvieron un impacto duradero. El conjunto de datos Iris que Fisher publicó en 1936 todavía se usa a veces para mostrar algoritmos de aprendizaje automático. Fisher también fue partidario de la eugenesia, lo que debería recordarnos que el uso moralmente dudoso de la ciencia de datos tiene una historia tan larga y persistente como su uso productivo en la industria y las ciencias naturales.

Otras influencias para el aprendizaje automático vinieron de la teoría de la información de Claude Shannon (1916 a 2001) y de la teoría de la computación propuesta por Alan Turing (1912 a 1954). Turing planteó la pregunta «¿pueden pensar las máquinas?» en su famoso artículo *Computing Machinery and Intelligence*. Al describir lo que hoy se conoce como prueba de Turing, propuso que una máquina puede considerarse inteligente si a un evaluador humano le resulta difícil distinguir entre las respuestas de una máquina y las de un humano, basándose solo en interacciones por texto.

Otras influencias vinieron de la neurociencia y la psicología. Al fin y al cabo, los seres humanos muestran claramente un comportamiento inteligente. Muchos estudiosos se han preguntado si podría explicarse esa capacidad y, quizá, reproducir su funcionamiento. Uno de los primeros algoritmos inspirados en la biología lo formuló Donald Hebb (1904 a 1985). En su innovador libro *The Organization of Behavior*, postuló que las neuronas aprenden por refuerzo positivo.

Esto se conoció como la regla de aprendizaje de Hebb. Estas ideas inspiraron trabajos posteriores, como el algoritmo de aprendizaje del perceptrón de Rosenblatt, y sentaron las bases de muchos algoritmos de descenso de gradiente estocástico en los que se apoya hoy el aprendizaje profundo: reforzar el comportamiento deseable y debilitar el indeseable para obtener buenos valores de los parámetros de una red neuronal.



Capas y retropropagación

La inspiración biológica es lo que dio su nombre a las redes neuronales. Desde hace más de un siglo (desde los modelos de Alexander Bain, 1873, y James Sherrington, 1890), los investigadores han intentado montar circuitos computacionales parecidos a redes de neuronas que interactúan. Con el tiempo, la interpretación de la

biología se ha vuelto menos literal, pero el nombre se quedó. En su núcleo hay unos pocos principios clave que se encuentran en la mayoría de las redes actuales. El primero es la alternancia de unidades de procesamiento lineales y no lineales, a menudo llamadas capas. El segundo es el uso de la regla de la cadena (también conocida como retropropagación) para ajustar a la vez los parámetros de toda la red.

Tras un rápido progreso inicial, la investigación en redes neuronales languideció aproximadamente de 1995 a 2005. Esto se debió principalmente a dos razones. Primero, entrenar una red es muy costoso computacionalmente. Aunque a finales del siglo pasado la memoria de acceso aleatorio era abundante, la potencia de cálculo escaseaba. Segundo, los conjuntos de datos eran relativamente pequeños. De hecho, el conjunto Iris de Fisher, de 1936, seguía siendo una herramienta popular para probar la eficacia de los algoritmos. El conjunto de datos MNIST, con sus 60 000 dígitos manuscritos, se consideraba enorme.

Ante la escasez de datos y de cálculo, herramientas estadísticas potentes como los métodos de núcleo, los árboles de decisión y los modelos gráficos resultaron empíricamente superiores en muchas aplicaciones. Además, a diferencia de las redes neuronales, no necesitaban semanas de entrenamiento y daban resultados predecibles con sólidas garantías teóricas.

Preguntas

1. ¿Por qué el procedimiento mejorado de medir pies despedía a los hombres con el pie más corto y el más largo?
 - a. Para que la fila fuera más corta
 - b. Para tratar los pies deformes que distorsionarían la media
 - c. Porque solo cabían 14 hombres en la fila
 - d. Para que el resultado fuera un número entero
2. Según el texto, ¿qué dos principios están en el núcleo de la mayoría de las redes neuronales actuales?
 - a. Las medias recortadas y los mínimos cuadrados
 - b. Los métodos de núcleo y los árboles de decisión
 - c. La alternancia de capas lineales y no lineales, y la retropropagación para ajustar todos los parámetros a la vez
 - d. La prueba de Turing y el aprendizaje de Hebb
3. ¿Por qué languideció la investigación en redes neuronales aproximadamente de 1995 a 2005?
 - a. El entrenamiento era costoso computacionalmente y los conjuntos de datos eran pequeños
 - b. La memoria escaseaba demasiado para guardar las redes
 - c. Se había demostrado que las redes neuronales no funcionaban
 - d. Nadie conocía la retropropagación
4. ¿Por qué en aquella época los métodos de núcleo, los árboles de decisión y los modelos gráficos ganaban en muchas aplicaciones?
 - a. Necesitaban muchos más datos que las redes neuronales
 - b. Estaban inspirados en la biología

- c. Eran los únicos métodos con capas
- d. Con pocos datos y poco cálculo funcionaban mejor, se entrenaban rápido y tenían garantías sólidas

Respuestas

- 1. **b)** El algoritmo se mejoró para tratar los pies deformes: se despedía a los dos extremos y se promediaba sobre el resto, una de las primeras medias recortadas.
- 2. **c)** Los principios clave son la alternancia de unidades de procesamiento lineales y no lineales, llamadas capas, y la regla de la cadena, o retropropagación, para ajustar a la vez los parámetros de toda la red.
- 3. **a)** Dos razones: el entrenamiento era muy costoso computacionalmente cuando la potencia de cálculo escaseaba, y los conjuntos de datos eran relativamente pequeños.
- 4. **d)** Ante la escasez de datos y de cálculo resultaron empíricamente superiores, no necesitaban semanas de entrenamiento y daban resultados predecibles con sólidas garantías teóricas.

Ejemplos resueltos

El pie de Köbel, simple y recortado Los pies de dieciséis hombres miden, en centímetros: 27, 28, 29, 30, 30, 31, 31, 31, 32, 32, 33, 33, 34, 34, 35, 39. Halle la media simple y luego la media recortada tras despedir al más corto y al más largo.

Se suman los dieciséis: la suma es 509 centímetros.

Se divide entre 16: $509 \div 16 \approx 31,81$ centímetros.

Se despiden al más corto (27) y al más largo (39): la suma de los otros catorce es $509 - 27 - 39 = 443$.

Se divide entre 14: $443 \div 14 \approx 31,64$ centímetros; el inusual 39 ya no tira de la estimación hacia arriba.

Respuesta: Media simple de unos 31,81 cm; media recortada de unos 31,64 cm

Ejercicios

1. Cinco medidas son 10, 11, 12, 13 y 34. Calcule la media simple y la media recortada que quita la menor y la mayor. ¿Cuál describe mejor un valor típico?

Pista: Quite 10 y 34 para la media recortada.

Respuesta: Media simple: $80 \div 5 = 16$. Media recortada: $(11 + 12 + 13) \div 3 = 12$. La media recortada es más típica, porque el único valor raro, 34, tiraba de la media simple hacia arriba.

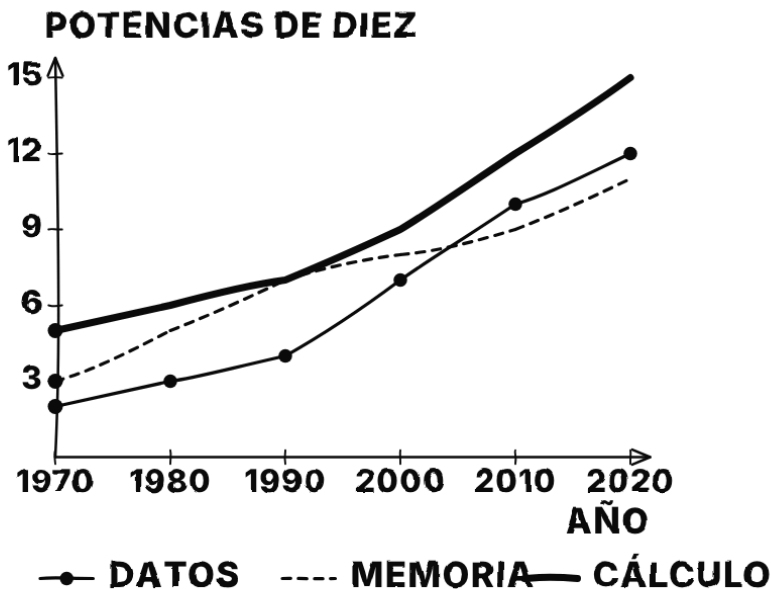
2. Describa la prueba de Turing con sus propias palabras.

Pista: ¿Quién tiene que notar la diferencia, y a partir de qué?

Respuesta: Un evaluador humano intercambia mensajes de texto con una máquina y con un humano. Si le resulta difícil distinguir las respuestas de la máquina de las del humano, basándose solo en el texto, la máquina puede considerarse inteligente.

El camino hacia el aprendizaje profundo

Mucho de esto cambió con la disponibilidad de cantidades masivas de datos, gracias a la World Wide Web, la llegada de empresas que atienden a cientos de millones de usuarios en línea, la difusión de sensores baratos y de alta calidad, el almacenamiento de datos barato (ley de Kryder) y el cálculo barato (ley de Moore). En particular, el panorama del cálculo en el aprendizaje profundo se vio revolucionado por los avances en las GPU, diseñadas originalmente para los videojuegos. De repente, algoritmos y modelos que parecían computacionalmente inviables estaban al alcance. Esto se ilustra mejor en la tabla, que compara el tamaño de los conjuntos de datos con la memoria y la potencia de cálculo de los ordenadores.



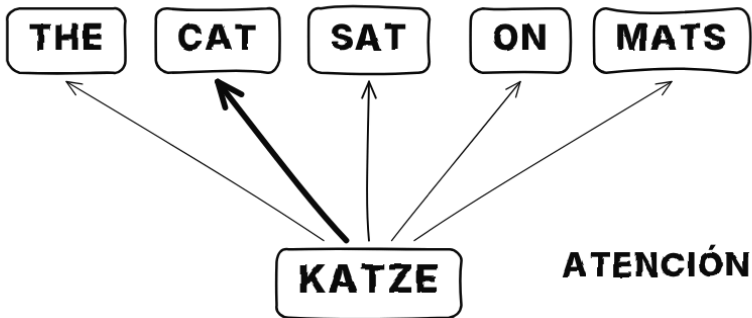
Datos, memoria y cálculo, 1970–2020

En 1970, un conjunto de datos tenía unos 100 ejemplos (los datos Iris), la memoria era de 1 KB y un ordenador hacía unas 100 mil operaciones de coma flotante por segundo (el Intel 8080). En 1980, un conjunto de datos tenía unos 1000 ejemplos (precios de viviendas en Boston), la memoria era de 100 KB y los ordenadores hacían 1 millón de operaciones por segundo (el Intel 80186). En 1990, los conjuntos de datos tenían 10 000 ejemplos (reconocimiento óptico de caracteres), la memoria era de 10 MB y los ordenadores hacían 10 millones de operaciones por segundo (el Intel 80486). En 2000, los conjuntos de datos tenían 10 millones de ejemplos (páginas web), la memoria era de 100 MB y los ordenadores hacían mil millones de operaciones por segundo (el Intel Core). En 2010, los conjuntos de datos tenían 10 000 millones de ejemplos (publicidad), la memoria

era de 1 GB y los ordenadores hacían un billón de operaciones por segundo (la NVIDIA C2050). En 2020, los conjuntos de datos tenían un billón de ejemplos (una red social), la memoria era de 100 GB y los ordenadores hacían mil billones de operaciones por segundo (la NVIDIA DGX-2).

Observe que la memoria de acceso aleatorio no ha seguido el ritmo del crecimiento de los datos. Al mismo tiempo, los aumentos de la potencia de cálculo han superado al crecimiento de los conjuntos de datos. Esto significa que los modelos estadísticos tienen que ser más eficientes en memoria, y así quedan libres para dedicar más ciclos de cálculo a optimizar parámetros, gracias al mayor presupuesto de cálculo. En consecuencia, el punto óptimo del aprendizaje automático y la estadística pasó de los modelos lineales (generalizados) y los métodos de núcleo a las redes neuronales profundas. Esta es también una de las razones por las que muchos de los pilares del aprendizaje profundo, como los perceptrones multicapa, las redes neuronales convolucionales, la memoria a corto y largo plazo y el Q-learning, fueron esencialmente «redescubiertos» en la última década, tras pasar bastante tiempo relativamente dormidos.

El progreso reciente en modelos estadísticos, aplicaciones y algoritmos se ha comparado a veces con la explosión cámbrica: un momento de rápido avance en la evolución de las especies. En efecto, el estado del arte no es una mera consecuencia de aplicar los recursos disponibles a algoritmos con décadas de antigüedad. Tenga en cuenta que la lista de ideas que sigue apenas roza la superficie de lo que ha ayudado a los investigadores a lograr un enorme progreso en la última década.



TRANSFORMER: SOLO ATENCIÓN

La atención como puntero

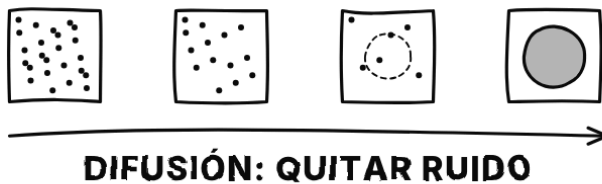
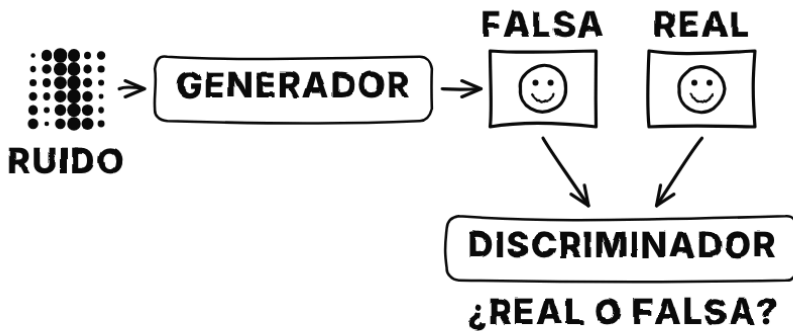
Nuevos métodos de control de la capacidad, como el dropout, han ayudado a mitigar el sobreajuste. En él se inyecta ruido en toda la red neuronal durante el entrenamiento.

Los mecanismos de atención resolvieron un segundo problema que llevaba más de un siglo atormentando a la estadística: cómo aumentar la memoria y la complejidad de un sistema sin aumentar el número de parámetros aprendibles. Los investigadores encontraron una solución elegante usando lo que solo puede verse como una estructura de punteros aprendible. En lugar de tener que recordar una secuencia de texto entera, por ejemplo para la traducción automática, en una representación de dimensión fija, bastaba con guardar un puntero al estado intermedio del proceso de traducción. Esto permitió aumentar

mucho la precisión en secuencias largas, ya que el modelo ya no necesitaba recordar la secuencia entera antes de empezar a generar una nueva.

Construida únicamente sobre mecanismos de atención, la arquitectura Transformer ha mostrado un escalado superior: funciona mejor al aumentar el tamaño del conjunto de datos, el del modelo y la cantidad de cálculo de entrenamiento. Esta arquitectura ha tenido un éxito convincente en una amplia gama de áreas, como el procesamiento del lenguaje natural, la visión por computador, el reconocimiento del habla, el aprendizaje por refuerzo y las redes neuronales de grafos. Por ejemplo, un solo Transformer preentrenado con modalidades tan diversas como texto, imágenes, pares de las articulaciones y pulsaciones de botones puede jugar a Atari, poner pies a imágenes, conversar y controlar un robot.

Al modelar las probabilidades de las secuencias de texto, los modelos de lenguaje pueden predecir texto a partir de otro texto. Escalar los datos, el modelo y el cálculo ha desbloqueado cada vez más capacidades de los modelos de lenguaje para realizar las tareas deseadas generando texto parecido al humano a partir de un texto de entrada. Por ejemplo, al alinear los modelos de lenguaje con la intención humana, ChatGPT de OpenAI permite a los usuarios interactuar con él de forma conversacional para resolver problemas, como depurar código o escribir de forma creativa.



Generador contra discriminador

Los diseños en varias etapas, por ejemplo mediante las redes de memoria y el programador-intérprete neuronal, permitieron a los modeladores estadísticos describir enfoques iterativos del razonamiento. Estas herramientas permiten modificar repetidamente un estado interno de la red neuronal profunda, realizando así los pasos sucesivos de una cadena de razonamiento, igual que un procesador puede modificar la memoria durante un cálculo.

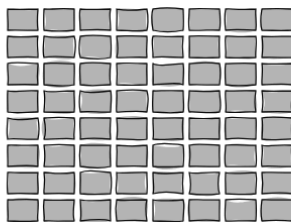
Un avance clave en el modelado generativo profundo fue la invención de las redes generativas antagónicas. Tradicionalmente, los métodos estadísticos de estimación de densidad y los modelos generativos se centraban en encontrar distribuciones de probabilidad adecuadas y algoritmos (a menudo aproximados) para muestrear de ellas. Como

resultado, estos algoritmos estaban muy limitados por la falta de flexibilidad propia de los modelos estadísticos. La innovación crucial de las redes generativas antagónicas fue sustituir el muestreador por un algoritmo arbitrario con parámetros diferenciables. Estos se ajustan después de modo que el discriminador (en la práctica, una prueba de dos muestras) no pueda distinguir los datos falsos de los reales. Gracias a la posibilidad de usar algoritmos arbitrarios para generar datos, la estimación de densidad se abrió a una gran variedad de técnicas. Los ejemplos de cebras al galope y de caras falsas de famosos dan testimonio de este progreso. Incluso aficionados al dibujo pueden producir imágenes fotorrealistas solo a partir de bocetos que describen la composición de una escena.

Además, mientras que el proceso de difusión añade poco a poco ruido aleatorio a las muestras de datos, los modelos de difusión aprenden el proceso de eliminación de ruido para construir poco a poco muestras de datos a partir de ruido aleatorio, invirtiendo el proceso de difusión. Han empezado a sustituir a las redes generativas antagónicas en modelos generativos profundos más recientes, como DALL-E 2 e Imagen, para arte creativo y generación de imágenes a partir de descripciones de texto.

En muchos casos, una sola GPU no basta para procesar las grandes cantidades de datos disponibles para el entrenamiento. En la última década, la capacidad de construir algoritmos de entrenamiento paralelos y distribuidos ha mejorado mucho. Uno de los retos clave al diseñar algoritmos escalables es que el caballo de batalla de la optimización en el aprendizaje profundo, el descenso de gradiente estocástico, se basa en procesar minilotes de datos relativamente pequeños. Al mismo tiempo, los lotes pequeños limitan la eficiencia de las GPU. Por tanto, entrenar con 1024 GPU con un tamaño de

minilote de, digamos, 32 imágenes por lote equivale a un minilote total de unas 32 000 imágenes. Los trabajos, primero de Li (2017) y después de You y colaboradores (2017) y de Jia y colaboradores (2018), llevaron el tamaño hasta 64 000 observaciones, reduciendo el tiempo de entrenamiento del modelo ResNet-50 en el conjunto de datos ImageNet a menos de 7 minutos. En comparación, al principio los tiempos de entrenamiento eran del orden de días.



1024 GPU

$$1,024 \times 32 \approx 32,000$$

IMÁGENES POR PASO

DÍAS



MINUTOS

THEANO → TENSORFLOW → PYTORCH, JAX

Muchas GPU, un gran lote

La capacidad de paralelizar el cálculo también ha contribuido al progreso del aprendizaje por refuerzo. Esto ha llevado a avances importantes en los que los ordenadores logran un rendimiento sobrehumano en tareas como el go, los juegos de Atari, Starcraft y las simulaciones físicas (por ejemplo, con MuJoCo) donde hay

simuladores del entorno disponibles. Véase, por ejemplo, Silver y colaboradores (2016) para una descripción de estos logros en AlphaGo. En pocas palabras, el aprendizaje por refuerzo funciona mejor si se dispone de muchas tuplas (estado, acción, recompensa). La simulación ofrece esa vía.

Los marcos de aprendizaje profundo han desempeñado un papel crucial en la difusión de ideas. La primera generación de marcos de código abierto para modelar redes neuronales la formaron Caffe, Torch y Theano. Muchos artículos fundamentales se escribieron con estas herramientas. Hoy han sido sustituidos por TensorFlow (que a menudo se usa mediante su API de alto nivel Keras), CNTK, Caffe 2 y Apache MXNet. La tercera generación de marcos la forman las llamadas herramientas imperativas para el aprendizaje profundo, una tendencia que probablemente inició Chainer, que usaba una sintaxis parecida a Python NumPy para describir los modelos. Esta idea la adoptaron PyTorch, la API Gluon de MXNet y JAX.

Preguntas

1. Como la memoria creció más despacio que los datos pero el cálculo creció rápido, ¿qué tuvieron que hacer los modelos estadísticos?
 - a. Usar menos cálculo y más memoria
 - b. Ser más eficientes en memoria y dedicar más ciclos de cálculo a optimizar parámetros
 - c. Dejar de usar grandes conjuntos de datos
 - d. Volver a los modelos lineales

2. Según el texto, ¿qué problema resolvieron los mecanismos de atención?
 - a. Cómo inyectar ruido durante el entrenamiento
 - b. Cómo entrenar con muchas GPU a la vez
 - c. Cómo aumentar la memoria y la complejidad de un sistema sin aumentar el número de parámetros aprendibles
 - d. Cómo muestrear de una distribución de probabilidad
3. En una red generativa antagónica, ¿cómo se ajustan los parámetros del generador?
 - a. De modo que el discriminador no pueda distinguir los datos falsos de los reales
 - b. De modo que el discriminador detecte siempre las falsificaciones
 - c. De modo que coincidan con una distribución de probabilidad fija
 - d. De modo que añada ruido paso a paso
4. ¿Por qué ayuda la simulación al aprendizaje por refuerzo?
 - a. Elimina la necesidad de recompensas
 - b. Sustituye a la política
 - c. Hace que los datos sean independientes
 - d. Proporciona muchas tuplas de estado, acción y recompensa

Respuestas

1. b) Los modelos estadísticos tienen que ser más eficientes en memoria, y así quedan libres para dedicar más ciclos de cálculo a optimizar parámetros, gracias al mayor presupuesto de cálculo.

2. c) La atención resolvió cómo aumentar la memoria y la complejidad sin más parámetros aprendibles, usando una estructura de punteros aprendible en lugar de recordar la secuencia entera.

3. a) El muestreador se sustituye por un algoritmo con parámetros diferenciables, que se ajustan para que el discriminador no pueda distinguir los datos falsos de los reales.

4. d) El aprendizaje por refuerzo funciona mejor si se dispone de muchas tuplas de estado, acción y recompensa, y la simulación ofrece esa vía.

Ejemplos resueltos

El minilote total El entrenamiento se hace con 1024 GPU, y cada una procesa un minilote de 32 imágenes. ¿Cuántas imágenes forman el minilote total?

Cada GPU procesa 32 imágenes por paso.

Hay 1024 GPU trabajando a la vez.

$1024 \times 32 = 32\,768$ imágenes, que el texto redondea a unas 32 000.

Respuesta: 32 768 imágenes, unas 32 000

Ejercicios

1. Según la tabla, ¿por qué factor creció el tamaño de los conjuntos de datos de 1970 (100 ejemplos) a 2020 (un billón de ejemplos), y por qué factor creció la memoria de 1 KB a 100 GB?

Pista: Cuente potencias de diez; 1 GB es un millón de KB.

Respuesta: Los datos crecieron de 10^2 a 10^{12} , un factor de 10^{10} , diez mil millones. La memoria creció de 1 KB a 100 GB, es decir, 100 millones de KB, un factor de 10^8 . Así que la memoria creció cien veces menos que los datos, y por eso no ha seguido el ritmo.

2. Explique con sus propias palabras cómo crea una imagen un modelo de difusión.

Pista: Empiece por lo que hace el proceso de difusión con los datos.

Respuesta: El proceso de difusión añade poco a poco ruido aleatorio a los datos hasta que solo queda ruido. Un modelo de difusión aprende a deshacerlo, quitando ruido paso a paso, de modo que puede partir de ruido aleatorio y construir poco a poco una nueva muestra de datos, como una imagen que encaje con una descripción de texto.

Historias de éxito

El reparto del trabajo entre los investigadores de sistemas, que construyen mejores herramientas, y los modeladores estadísticos, que construyen mejores redes neuronales, ha simplificado mucho las cosas. Por ejemplo, entrenar un modelo lineal de regresión logística era antes un problema de deberes nada trivial, digno de darse a los nuevos doctorandos de aprendizaje automático de la Universidad Carnegie Mellon en 2014. Hoy esta tarea puede hacerse con menos de 10 líneas de código, lo que la pone de lleno al alcance de cualquier programador.

La inteligencia artificial tiene una larga historia de logros que habrían sido difíciles de conseguir de otro modo. Por ejemplo, desde los años noventa se usan sistemas de clasificación de correo con reconocimiento óptico de caracteres. De ahí procede, al fin y al cabo, el famoso conjunto de datos de dígitos manuscritos MNIST. Lo mismo ocurre con la lectura de cheques para depósitos bancarios y la evaluación de la solvencia de los solicitantes. Las transacciones financieras se comprueban automáticamente en busca de fraude. Esto es la columna vertebral de muchos sistemas de pago del comercio electrónico, como PayPal, Stripe, AliPay, WeChat, Apple, Visa y MasterCard. Los programas de ajedrez llevan décadas siendo competitivos. El aprendizaje automático alimenta la búsqueda, las recomendaciones, la personalización y la ordenación en Internet. Dicho de otro modo, el aprendizaje automático está en todas partes, aunque a menudo oculto a la vista.

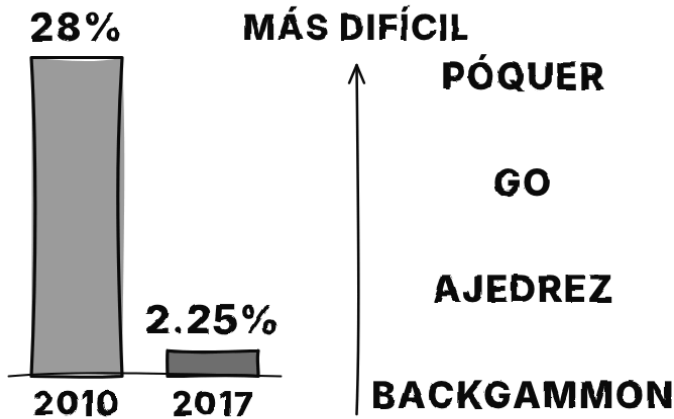
Solo recientemente la IA ha pasado a primer plano, sobre todo gracias a soluciones a problemas que antes se consideraban intratables y que afectan directamente a los consumidores. Muchos de estos avances se atribuyen al aprendizaje profundo.

Los asistentes inteligentes, como Siri de Apple, Alexa de Amazon y el asistente de Google, son capaces de responder a peticiones habladas con un grado de precisión razonable. Esto incluye tareas sencillas, como encender la luz, y otras más complejas, como pedir cita en la peluquería u ofrecer diálogos de atención telefónica. Probablemente sea la señal más visible de que la IA está influyendo en nuestras vidas.

Un ingrediente clave de los asistentes digitales es su capacidad de reconocer el habla con precisión. La precisión de estos sistemas ha ido aumentando hasta igualar a la de los humanos en ciertas aplicaciones.

El reconocimiento de objetos también ha recorrido un largo camino. Identificar el objeto de una imagen era una tarea bastante difícil en 2010. En la prueba de referencia ImageNet, investigadores de NEC Labs y de la Universidad de Illinois en Urbana-Champaign lograron una tasa de error top-5 del 28%. Para 2017, esta tasa de error se había reducido al 2,25%. Del mismo modo, se han logrado resultados asombrosos en la identificación del canto de las aves y en el diagnóstico del cáncer de piel.

ERROR IMAGENET



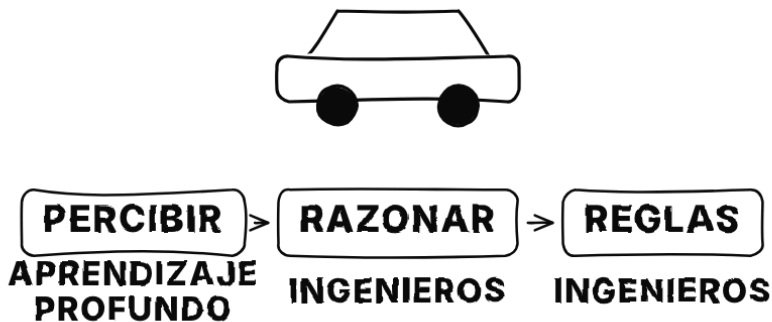
Del 28% al 2,25%, y los juegos

La destreza en los juegos servía antes como vara de medir de la capacidad humana. Empezando por TD-Gammon, un programa para jugar al backgammon que usaba aprendizaje por refuerzo de diferencias temporales, el progreso algorítmico y computacional ha llevado a algoritmos para una amplia gama de aplicaciones.

Comparado con el backgammon, el ajedrez tiene un espacio de estados y un conjunto de acciones mucho más complejos. DeepBlue venció a Garri Kaspárov usando un paralelismo masivo, hardware específico y una búsqueda eficiente en el árbol del juego. El go es aún más difícil, por su enorme espacio de estados. AlphaGo alcanzó el nivel humano en 2015, usando aprendizaje profundo combinado con muestreo de árboles de Montecarlo. La dificultad del póquer era que el espacio de estados es grande y solo parcialmente observable (no

conocemos las cartas de los rivales). Libratus superó el rendimiento humano en el póquer usando estrategias estructuradas de forma eficiente.

Otra señal del progreso de la IA es la llegada de los vehículos autónomos. Aunque la autonomía total aún no está al alcance, se han hecho excelentes avances en esa dirección, con empresas como Tesla, NVIDIA y Waymo que venden productos que permiten una autonomía parcial. Lo que hace tan difícil la autonomía total es que conducir bien requiere la capacidad de percibir, de razonar y de incorporar reglas a un sistema. Por ahora, el aprendizaje profundo se usa sobre todo en el aspecto visual de estos problemas. El resto lo ajustan a fondo los ingenieros.



HECHO PARA UN OBJETIVO

Conducir requiere más que ver

Esto apenas roza la superficie de las aplicaciones importantes del aprendizaje automático. Por ejemplo, la robótica, la logística, la biología computacional, la física de partículas y la astronomía deben algunos de sus avances recientes más impresionantes, al menos en parte, al aprendizaje automático, que se está convirtiendo así en una herramienta omnipresente para ingenieros y científicos.

En artículos no técnicos se plantean a menudo preguntas sobre un próximo apocalipsis de la IA y la verosimilitud de una singularidad. El temor es que, de algún modo, los sistemas de aprendizaje automático adquieran consciencia y tomen decisiones, independientemente de sus programadores, que afecten directamente a la vida de las personas. En cierta medida, la IA ya afecta directamente al sustento de las personas: la solvencia se evalúa automáticamente, los pilotos automáticos guían en su mayor parte los vehículos y las decisiones sobre conceder la libertad bajo fianza usan datos estadísticos. En un plano más frívolo, podemos pedirle a Alexa que encienda la cafetera.

Por suerte, estamos lejos de un sistema de IA consciente que pudiera manipular deliberadamente a sus creadores humanos. En primer lugar, los sistemas de IA se diseñan, entrenan y despliegan de una manera concreta y orientada a un objetivo. Aunque su comportamiento pueda dar la ilusión de una inteligencia general, en la base de su diseño hay una combinación de reglas, heurísticas y modelos estadísticos. En segundo lugar, hoy por hoy simplemente no existen herramientas de inteligencia artificial general capaces de mejorarse a sí mismas, de razonar sobre sí mismas y de modificar, ampliar y mejorar su propia arquitectura mientras intentan resolver tareas generales.

Una preocupación mucho más urgente es cómo se usa la IA en nuestra vida diaria. Es probable que muchas tareas rutinarias que hoy hacen las personas puedan automatizarse y se automaticen. Los robots agrícolas probablemente reducirán los costes de los agricultores ecológicos, pero también automatizarán las cosechas. Esta fase de la revolución industrial puede tener consecuencias profundas para amplias capas de la sociedad, ya que los trabajos poco cualificados dan mucho empleo en muchos países. Además, los modelos estadísticos aplicados sin cuidado pueden llevar a sesgos raciales, de género o de edad, y suscitan preocupaciones razonables sobre la equidad de los procedimientos si se automatizan para guiar decisiones importantes. Es importante asegurarse de que estos algoritmos se usen con cuidado. Con lo que sabemos hoy, esto nos parece una preocupación mucho más urgente que la posibilidad de que una superinteligencia malévola destruya a la humanidad.

Preguntas

1. ¿Por qué el texto dice que el aprendizaje automático está «en todas partes, aunque a menudo oculto a la vista»?
 - a. Porque solo se usa en laboratorios de investigación
 - b. Porque mueve en silencio la clasificación del correo, la detección de fraudes, la búsqueda y las recomendaciones
 - c. Porque la gente se niega a usarlo
 - d. Porque solo fue posible a partir de 2017
2. Según el texto, ¿qué hacía difícil el póquer para los ordenadores?

- a. Su espacio de estados es diminuto
 - b. Requiere hardware específico
 - c. Su espacio de estados es grande y solo parcialmente observable, porque las cartas de los rivales están ocultas
 - d. No tiene recompensas
3. ¿Cuál es la razón principal que dan los autores para no temer a una IA consciente que manipule a sus creadores?
- a. Los sistemas de IA se diseñan para objetivos concretos, y aún no existen herramientas de inteligencia general que se mejore a sí misma
 - b. Los sistemas de IA siempre se apagan por la noche
 - c. La IA nunca se ha usado en decisiones reales
 - d. Las leyes prohíben a la IA tomar decisiones
4. ¿Qué consideran los autores una preocupación más urgente que una superinteligencia malévola?
- a. Que los ordenadores se queden sin memoria
 - b. Que los juegos se vuelvan demasiado fáciles
 - c. Que la IA deje de mejorar
 - d. La automatización de trabajos rutinarios y los modelos sesgados que toman decisiones importantes

Respuestas

1. **b)** La clasificación del correo, la lectura de cheques, la evaluación de solvencia, la detección de fraudes, la búsqueda, las recomendaciones y la ordenación usan aprendizaje automático, casi siempre sin que se vea.
2. **c)** La dificultad del póquer era que el espacio de estados es grande y solo parcialmente observable: no conocemos las cartas de los rivales.

3. a) Los sistemas de IA se diseñan y despliegan de forma concreta y orientada a un objetivo, y no hay herramientas de inteligencia artificial general capaces de mejorarse y modificarse a sí mismas.

4. d) La preocupación urgente es cómo se usa la IA en la vida diaria: la automatización de muchas tareas rutinarias y los modelos aplicados sin cuidado que llevan a sesgos raciales, de género o de edad.

Ejemplos resueltos

El progreso en ImageNet En ImageNet, la tasa de error top-5 era del 28% en 2010 y del 2,25% en 2017. De 1000 imágenes de prueba, ¿cuántas se fallaron aproximadamente cada año, y cuántas veces menor se hizo el error?

En 2010: el 28% de 1000 es $0,28 \times 1000 = 280$ imágenes falladas.

En 2017: el 2,25% de 1000 es $0,0225 \times 1000 = 22,5$, unas 22 o 23 imágenes falladas.

Cociente: $28 \div 2,25 \approx 12,4$.

Respuesta: Unas 280 falladas en 2010 y unas 22 en 2017; el error se hizo unas 12 veces menor

Ejercicios

1. Explique por qué, según el texto, la conducción totalmente autónoma es más difícil que reconocer objetos en imágenes.

Pista: Conducir requiere más que ver.

Respuesta: Conducir bien requiere la capacidad de percibir, de razonar y de incorporar reglas a un sistema. El aprendizaje profundo se usa sobre todo para la parte visual, la percepción; el resto todavía lo ajustan a fondo los ingenieros.

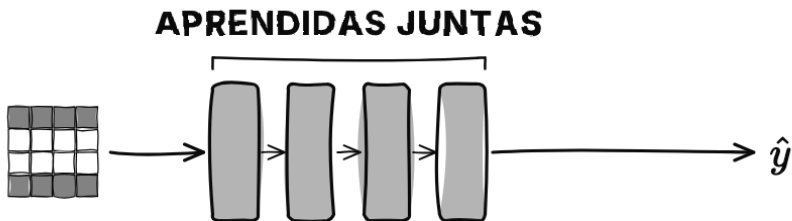
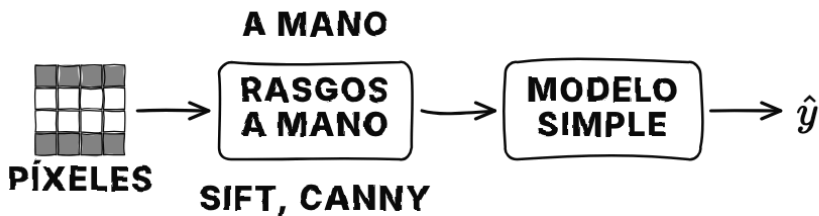
2. Dé tres formas, mencionadas en el texto, en que la IA ya afecta directamente a la vida de las personas.

Pista: Piense en el dinero, los viajes y los tribunales.

Respuesta: La solvencia se evalúa automáticamente, los pilotos automáticos guían en su mayor parte los vehículos y las decisiones sobre conceder la libertad bajo fianza usan datos estadísticos.

La esencia del aprendizaje profundo

Hasta ahora hemos hablado del aprendizaje automático en términos generales. El aprendizaje profundo es la parte del aprendizaje automático que se ocupa de modelos basados en redes neuronales de muchas capas. Es profundo precisamente en el sentido de que sus modelos aprenden muchas capas de transformaciones. Aunque esto pueda sonar limitado, el aprendizaje profundo ha dado lugar a una variedad vertiginosa de modelos, técnicas, planteamientos de problemas y aplicaciones. Se han desarrollado muchas intuiciones para explicar las ventajas de la profundidad. Podría decirse que todo aprendizaje automático tiene muchas capas de cálculo, la primera de las cuales son los pasos de procesamiento de características. Lo que distingue al aprendizaje profundo es que las operaciones de cada una de las muchas capas de representaciones se aprenden conjuntamente a partir de los datos.



DE EXTREMO A EXTREMO

Características hechas a mano frente a extremo a extremo

Los problemas que hemos tratado hasta ahora, como aprender de la señal de audio en bruto, de los valores de píxel en bruto de las imágenes, o pasar de frases de longitud arbitraria a sus equivalentes en otras lenguas, son aquellos en los que el aprendizaje profundo destaca y los métodos tradicionales flaquean. Resulta que estos modelos de muchas capas son capaces de tratar datos perceptivos de bajo nivel de una forma que las herramientas anteriores no podían. Podría decirse que el rasgo común más importante de los métodos de aprendizaje profundo es el entrenamiento de extremo a extremo. Es decir, en lugar de montar un sistema a partir de componentes ajustados por separado, se construye el sistema y luego se ajusta su rendimiento de forma conjunta. Por ejemplo, en visión por computador los científicos solían separar el proceso de ingeniería de

características del proceso de construir modelos de aprendizaje automático. El detector de bordes de Canny y el extractor de características SIFT de Lowe reinaron durante más de una década como algoritmos para convertir imágenes en vectores de características. En otros tiempos, la parte crucial de aplicar el aprendizaje automático a estos problemas consistía en idear formas diseñadas a mano de transformar los datos en algo adecuado para modelos poco profundos. Por desgracia, el ingenio humano tiene un límite comparado con una evaluación sistemática de millones de opciones realizada automáticamente por un algoritmo. Cuando se impuso el aprendizaje profundo, estos extractores de características se sustituyeron por filtros ajustados automáticamente que daban una precisión superior.

Así, una ventaja clave del aprendizaje profundo es que sustituye no solo los modelos poco profundos del final de las cadenas de aprendizaje tradicionales, sino también el laborioso proceso de ingeniería de características. Además, al sustituir buena parte del preprocesamiento específico de cada dominio, el aprendizaje profundo ha eliminado muchas de las fronteras que antes separaban la visión por computador, el reconocimiento del habla, el procesamiento del lenguaje natural, la informática médica y otras áreas de aplicación, y ofrece así un conjunto unificado de herramientas para abordar problemas diversos.

Más allá del entrenamiento de extremo a extremo, estamos viviendo una transición de las descripciones estadísticas paramétricas a modelos totalmente no paramétricos. Cuando los datos escasean, hay que apoyarse en suposiciones simplificadoras sobre la realidad para obtener modelos útiles. Cuando los datos abundan, estas pueden sustituirse por modelos no paramétricos que se ajustan mejor a los

datos. En cierta medida, esto refleja el progreso que vivió la física a mediados del siglo pasado con la llegada de los ordenadores. En lugar de resolver a mano aproximaciones paramétricas del comportamiento de los electrones, ahora se puede recurrir a simulaciones numéricas de las ecuaciones en derivadas parciales correspondientes. Esto ha llevado a modelos mucho más precisos, aunque a menudo a costa de la interpretación.

Otra diferencia respecto a trabajos anteriores es la aceptación de soluciones subóptimas, el trato con problemas de optimización no lineales y no convexos, y la disposición a probar cosas antes de demostrarlas. Este empirismo recién descubierto al tratar problemas estadísticos, unido a una rápida llegada de talento, ha producido un rápido progreso en el desarrollo de algoritmos prácticos, aunque en muchos casos a costa de modificar y reinventar herramientas que existían desde hacía décadas.

Al final, la comunidad del aprendizaje profundo se enorgullece de compartir herramientas más allá de las fronteras académicas y empresariales, publicando como código abierto muchas bibliotecas excelentes, modelos estadísticos y redes entrenadas. Es con este espíritu con el que los cuadernos que forman este libro están disponibles libremente para su distribución y uso. Hemos trabajado mucho para reducir las barreras de acceso para cualquiera que quiera aprender aprendizaje profundo, y esperamos que nuestros lectores se beneficien de ello.

En resumen: el aprendizaje automático estudia cómo los sistemas informáticos pueden aprovechar la experiencia (a menudo datos) para mejorar su rendimiento en tareas concretas. Combina ideas de la estadística, la minería de datos y la optimización. A menudo se usa

como medio para implementar soluciones de IA. Como clase de aprendizaje automático, el aprendizaje de representaciones se centra en cómo encontrar automáticamente la forma adecuada de representar los datos. Entendido como aprendizaje de representaciones en varios niveles mediante el aprendizaje de muchas capas de transformaciones, el aprendizaje profundo sustituye no solo los modelos poco profundos del final de las cadenas tradicionales de aprendizaje automático, sino también el laborioso proceso de ingeniería de características. Buena parte del progreso reciente del aprendizaje profundo se ha debido a la abundancia de datos procedentes de sensores baratos y de aplicaciones a escala de Internet, y a un progreso importante en el cálculo, sobre todo gracias a las GPU. Además, la disponibilidad de marcos de aprendizaje profundo eficientes ha facilitado mucho el diseño y la implementación de la optimización de sistemas completos, y esto es un componente clave para obtener un alto rendimiento.

Preguntas

1. Si podría decirse que todo aprendizaje automático tiene muchas capas de cálculo, ¿qué distingue al aprendizaje profundo?
 - a. Usa una sola capa
 - b. Las operaciones de cada una de sus muchas capas de representaciones se aprenden conjuntamente a partir de los datos
 - c. Nunca usa características
 - d. Siempre necesita que una persona diseñe cada capa

2. ¿Qué significa entrenamiento de extremo a extremo?
 - a. Construir el sistema completo y ajustar su rendimiento de forma conjunta
 - b. Ajustar cada componente por separado y luego montarlos
 - c. Entrenar solo la última capa de un modelo
 - d. Recorrer un conjunto de datos una vez de principio a fin
3. ¿Por qué los filtros ajustados automáticamente superaron a extractores de características diseñados a mano como SIFT?
 - a. Las personas dejaron de trabajar en visión
 - b. Las características hechas a mano eran demasiado rápidas
 - c. Un algoritmo puede evaluar sistemáticamente millones de opciones, más de lo que permite el ingenio humano
 - d. Estaba prohibido usar extractores de características
4. ¿Cuándo pueden sustituirse las suposiciones simplificadoras sobre la realidad por modelos no paramétricos?
 - a. Cuando los datos escasean
 - b. Cuando los modelos deben ser fáciles de interpretar
 - c. Cuando no hay ordenadores
 - d. Cuando los datos abundan

Respuestas

1. **b)** Lo que distingue al aprendizaje profundo es que las operaciones de cada una de las muchas capas de representaciones se aprenden conjuntamente a partir de los datos.
2. **a)** En lugar de montar un sistema a partir de componentes ajustados por separado, se construye el sistema y luego se ajusta su rendimiento de forma conjunta.

3. c) El ingenio humano tiene un límite comparado con una evaluación sistemática de millones de opciones realizada automáticamente por un algoritmo.

4. d) Cuando los datos escasean hay que apoyarse en suposiciones simplificadoras; cuando abundan, estas pueden sustituirse por modelos no paramétricos que se ajustan mejor a los datos.

Ejercicios

1. Explique con sus propias palabras las dos cosas que sustituye el aprendizaje profundo en una cadena tradicional de aprendizaje automático.

Pista: Piense en lo que ocurre antes de la cadena y al final de ella.

Respuesta: Sustituye el modelo poco profundo del final de la cadena, y también la laboriosa ingeniería de características del principio, en la que las personas diseñaban a mano cómo convertir los datos en bruto, como las imágenes, en vectores de características.

2. El texto compara el paso a los modelos no paramétricos con un cambio en la física. Describa la comparación y el precio pagado en ambos casos.

Pista: ¿Qué permitieron hacer los ordenadores a los físicos en lugar de resolver a mano?

Respuesta: Los físicos dejaron de resolver a mano aproximaciones paramétricas simplificadas del comportamiento de los electrones y pasaron a simulaciones numéricas de las ecuaciones de fondo, igual que el aprendizaje automático pasa de modelos paramétricos simplificados a modelos que se ajustan a datos abundantes. Ambos dan modelos mucho más precisos, a menudo a costa de la interpretación.