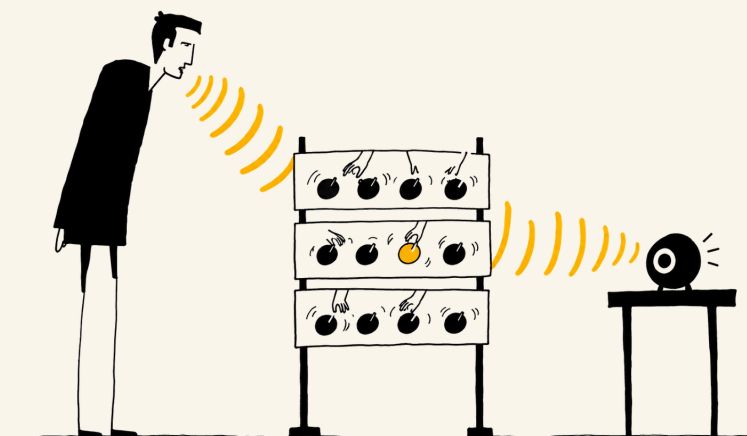


Sumérgete en el aprendizaje profundo



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

Reglas que nunca aprenden

Perillas que aprenden una palabra

Programar con datos

Los datos adecuados y una puntuación justa

Bajar la cuesta y luego aprender de las respuestas

¿Cuánto? Regresión

¿Cuál de ellos? Clasificación

Muchas etiquetas a la vez, y lo mejor primero

El muro que le aprende

Cuando el orden importa: secuencias

Aprender sin profesor

Cuando el mundo responde

Aprender actuando: aprendizaje por refuerzo

Raíces antiguas: de las puertas de la iglesia a las neuronas

El camino hacia el aprendizaje profundo

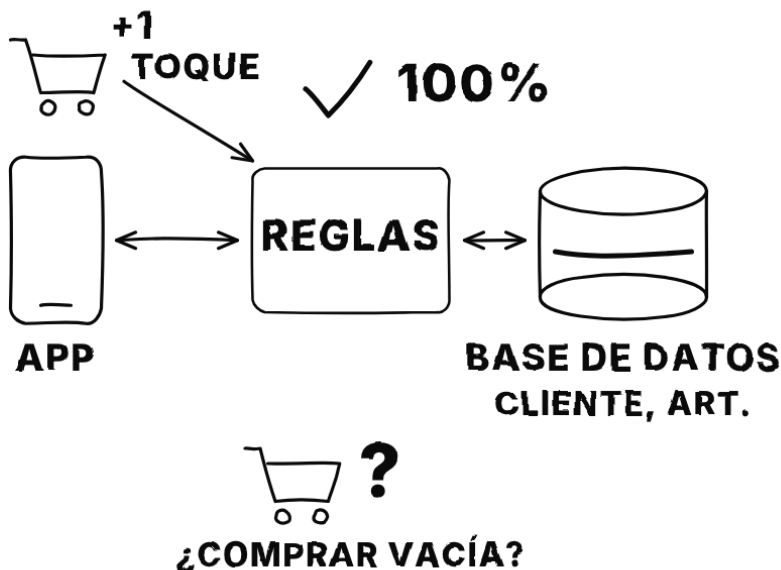
Éxitos ocultos y éxitos visibles

Lo que la IA no es, y lo que hace profundo al aprendizaje

Pruébalo, compártalo

Reglas que nunca aprenden

Abra cualquier aplicación de compras en su teléfono, ponga un par de calcetines en la cesta y observe lo que ocurre. Aparece un numerito en el icono de la cesta, los calcetines se muestran en la lista y el precio se suma al total. Nadie en la tienda vio su toque; todo lo hizo un programa, y lo hizo igual para usted que para los millones de clientes anteriores. Ese programa se escribió como un conjunto de reglas: cuando pase esto, haz aquello.



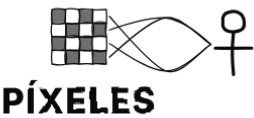
Una tienda en línea hecha de reglas

El libro empieza justamente con un programa así, una tienda en línea, y se pregunta cómo la construirían sus creadores. Pasarían unas horas frente a una pizarra y dividirían el trabajo en tres partes. Primero está la cara de la tienda, las páginas que usted ve en un navegador web o en la aplicación. Segundo, hay una base de datos, una tabla grande y bien cuidada que recuerda quién es usted, qué hay en su cesta y todo lo que alguien ha comprado alguna vez. Tercero, en el centro está lo que los programadores llaman la lógica de negocio, y lo que los autores, medio en broma, llaman el cerebro de la aplicación. El cerebro es una larga lista de reglas que dicen, para cada situación que alguien pueda imaginar, qué debe hacer el programa.

Aquí tiene una de esas reglas, y es tan aburrida como suena. Cuando un cliente toca «añadir a la cesta», el programa escribe una nueva fila en la tabla de cestas, emparejando el número del cliente con el número del producto. Luego el equipo repasa uno por uno los casos límite más raros. ¿Qué debe pasar si alguien pulsa «comprar» con la cesta vacía? También escriben una regla para eso, la prueban, arreglan lo que falla y vuelven a probar. Lo mismo ocurre con los ajustes sencillos de su teléfono: «si la batería baja del veinte por ciento, atenúa la pantalla» es una regla que alguien escribió una vez y que nunca cambiará por sí sola.

Lo notable, dice el libro, es que esto funciona. Las personas pueden diseñar estos sistemas en su cabeza, a menudo para situaciones que nadie ha vivido antes, y ponerlos en marcha antes de que llegue un solo cliente real. Y los autores añaden un consejo sencillo: si usted puede escribir reglas que aciertan el 100% de las veces, no necesita el aprendizaje automático en absoluto. Una regla que siempre acierta es más barata, más clara y más segura que cualquier cosa aprendida.

Ahora imagine al mismo equipo brillante ante la misma pizarra, pero con otra tarea. El libro propone cuatro. Predecir el tiempo de mañana a partir de mapas, imágenes de satélite y el tiempo de los últimos días. Responder cualquier pregunta breve sobre un dato que alguien escriba con sus propias palabras. Mirar una foto y dibujar un contorno alrededor de cada persona que aparece en ella. Mostrar a cada comprador productos que probablemente le encantarán pero que nunca habría encontrado navegando. Con las dos últimas se encuentra usted a diario: la galería de fotos del teléfono encuentra las caras de sus amigos y una aplicación de vídeos llena su muro de cosas que usted no buscó.



Cuatro tareas sin reglas

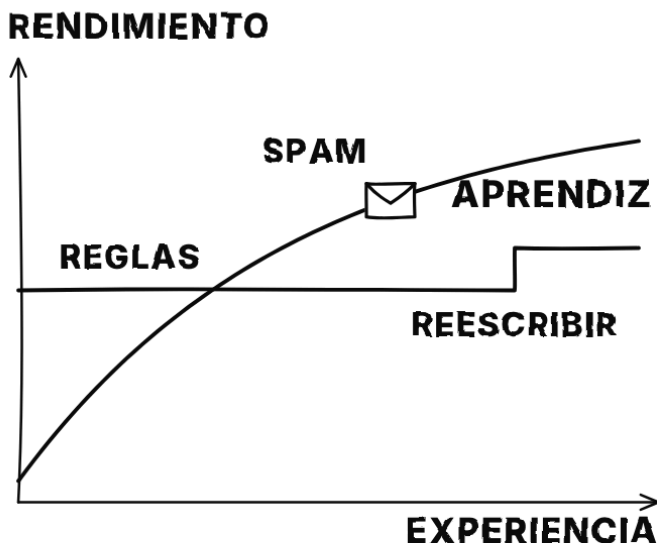
Intente escribir reglas para cualquiera de ellas y se atascará casi enseguida. El libro da dos razones. A veces la respuesta correcta no deja de moverse: lo que gusta a los compradores esta primavera no es lo que les gustaba la anterior, así que una lista fija de reglas quedaría anticuada. Cualquier buena solución tiene que seguir a un mundo que cambia. Otras veces, la relación entre lo que entra y lo que sale es demasiado enredada. Una foto no es más que una cuadrícula de puntos de color, llamados píxeles, y el paso de esos puntos a la idea «esto es una persona» puede exigir miles o millones de pequeños cálculos que nadie sabe escribir. Usted reconoce la cara de un amigo en un instante, pero no podría explicar, paso a paso, cómo lo hizo.

Pruébelo

Pida a alguien que escriba, en forma de reglas, cómo distinguir en una foto un gato de un perro. Empezará por las orejas, el tamaño y el hocico, y en un minuto tropezará con un perro peludo de orejas puntiagudas o con un gato fotografiado de espaldas. Cada regla que añade falla con alguna foto nueva, aunque esa persona nunca confunde un gato con un perro. Esa brecha, entre lo que sabemos hacer y lo que sabemos escribir, es justo donde entra el aprendizaje automático.

Así que el libro da su primera definición. El aprendizaje automático es el estudio de los programas que aprenden de la experiencia: cuanta más experiencia reúne un programa así, mejor hace su trabajo. La experiencia suele llegar en forma de datos, como millones de fotos etiquetadas, o de probar cosas en algún entorno y ver qué pasa. Un filtro de spam es el caso cotidiano: cada vez que alguien marca un mensaje como spam, el filtro tiene un poco más de qué aprender, y la

semana que viene atrapa más basura que esta. Compárelo con la tienda en línea. Sus reglas siguen exactamente iguales por muchos clientes que pasen, y solo cambian cuando los propios programadores aprenden algo y se sientan a reescribir el código. En la tienda aprenden las personas; en el aprendizaje automático aprende el programa.



Mejor con la experiencia

Después el libro dice hacia dónde va. Enseñará los fundamentos del aprendizaje automático y, sobre todo, una potente familia de sus métodos llamada aprendizaje profundo. El aprendizaje profundo es lo que impulsa los avances recientes en visión por computador, en la comprensión del lenguaje humano, en medicina y en la lectura del genoma. Cuando un chatbot responde a su pregunta con frases

fluidas, o un asistente de voz le entiende pese al ruido de la calle, usted está oyendo el aprendizaje profundo en acción.

Para recordar

La mayoría de los programas que usted usa son listas de reglas escritas por personas, y se comportan siempre igual hasta que alguien los reescribe. Ese enfoque funciona de maravilla cuando las reglas se pueden escribir y siempre aciertan. Falla cuando la respuesta no deja de cambiar, o cuando el camino de la entrada a la salida es demasiado enredado para describirlo, como al reconocer una cara en una foto. El aprendizaje automático es el estudio de los programas que mejoran con la experiencia, normalmente con más datos. Este libro enseña aprendizaje automático, y sobre todo aprendizaje profundo, la familia de métodos detrás de los avances actuales en visión, lenguaje y ciencia.

Preguntas

1. Una regla de la tienda dice: cuando un cliente toque «añadir a la cesta», escribe una nueva fila en la tabla de cestas. ¿Qué le pasa a esta regla después de que la hayan usado un millón de clientes?
 - a. Se ajusta poco a poco a las cestas más comunes
 - b. Sigue exactamente igual hasta que los programadores la reescriben

- c. Se sustituye automáticamente por una regla aprendida
 - d. Deja de funcionar cuando la tabla se llena de filas
2. ¿Por qué una lista fija de reglas para recomendar productos a los compradores quedaría anticuada?
- a. Porque la base de datos olvida las compras antiguas
 - b. Porque las reglas funcionan demasiado despacio en los teléfonos
 - c. Porque lo que gusta a los compradores cambia con el tiempo
 - d. Porque los compradores nunca compran productos recomendados
3. Usted distingue un gato de un perro al instante. ¿Por qué eso no le ayuda a escribir reglas para un programa que haga lo mismo?
- a. No puede explicar, paso a paso, cómo lo hace
 - b. Para una cámara, los gatos y los perros son iguales
 - c. Las fotos tienen demasiados pocos píxeles para mostrar al animal
 - d. Los programas no pueden mirar fotos
4. Según el consejo de los autores, ¿cuándo no merece la pena recurrir al aprendizaje automático?
- a. Cuando el programa lo vayan a usar millones de personas
 - b. Cuando la tarea implica una base de datos
 - c. Cuando solo tiene una pequeña cantidad de datos
 - d. Cuando puede escribir reglas que aciertan siempre

Respuestas

1. **b)** Las reglas de la tienda siguen exactamente iguales por muchos clientes que pasen; solo cambian cuando los propios programadores aprenden algo y reescriben el código.
2. **c)** A veces la respuesta correcta no deja de moverse: lo que gusta a los compradores esta primavera no es lo que les gustaba la anterior, así que una buena solución tiene que seguir a un mundo que cambia.
3. **a)** Reconocemos caras y animales en un instante, pero los pasos están más allá de lo que podemos explicar; cada regla que escribimos falla con alguna foto nueva.
4. **d)** Si usted puede escribir reglas que aciertan el 100% de las veces, no necesita el aprendizaje automático en absoluto.

Ejercicios

1. Su teléfono atenúa la pantalla cuando la batería baja del veinte por ciento, y su aplicación de vídeos llena el muro de cosas que usted no buscó. ¿Cuál de las dos es una regla fija y cuál requiere aprendizaje automático? Explique por qué, con las razones de la lección.

Pista: Pregúntese si la respuesta correcta podría escribirse una vez y seguir siendo correcta para siempre.

Respuesta: El ajuste de la batería es una regla fija: se puede escribir una vez y siempre acierta, así que no hace falta aprender nada. El muro es una tarea de aprendizaje automático: lo que disfrutará cada persona no deja de cambiar, y la relación entre su historial y los vídeos que le gustarán es demasiado enredada para escribirla como reglas, así que el programa tiene que mejorar con la experiencia.

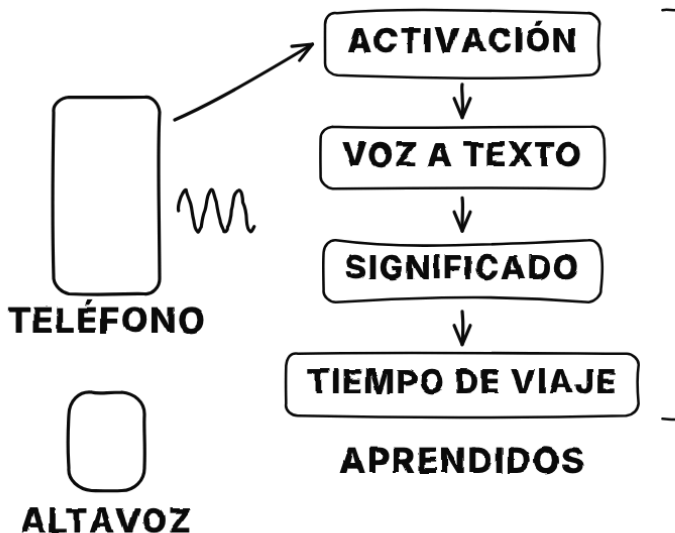
2. Explique con sus propias palabras la frase: «En la tienda aprenden las personas; en el aprendizaje automático aprende el programa».

Pista: Piense en qué cambia el comportamiento de la tienda y qué cambia el comportamiento de un filtro de spam.

Respuesta: El programa de la tienda se comporta igual por mucho que se use; solo mejora cuando sus programadores aprenden algo y reescriben las reglas. Un programa de aprendizaje automático, como un filtro de spam, mejora por sí mismo a medida que reúne experiencia, normalmente más datos, sin que nadie reescriba sus reglas.

Perillas que aprenden una palabra

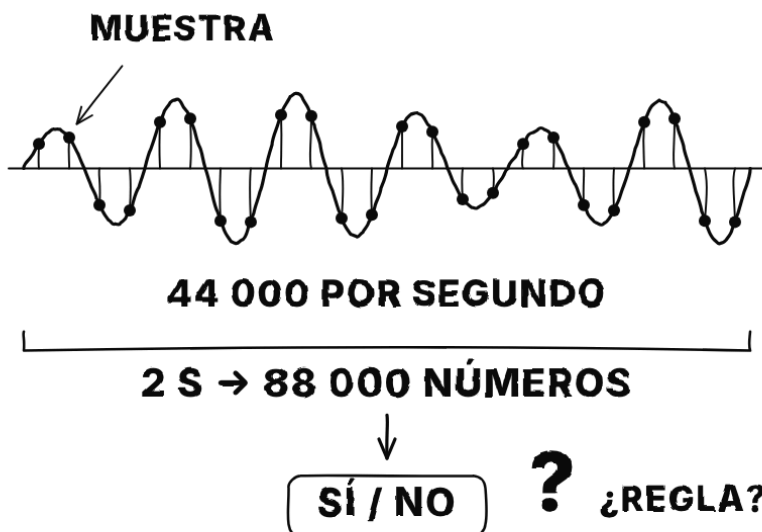
Diga «Hey Siri» u «OK Google» desde el otro lado de una habitación silenciosa y observe cómo se despierta su teléfono. Se ilumina en menos de un segundo, aunque estaba boca abajo y hace un momento parecía no hacer nada. Luego pídale el camino a la cafetería más cercana. Sus palabras aparecen en la pantalla como texto, la aplicación de mapas se abre sola y junto a cada ruta ve cuántos minutos debería llevar.



Cuatro programas aprendidos en un teléfono

Los autores cuentan justo esta historia sobre sí mismos, yendo en coche a por café antes de empezar a escribir el libro, y admiten que es inventada para ilustrar una idea. La idea es que en esos pocos segundos el teléfono usó varios programas que no se escribieron como reglas. Uno oyó la palabra de activación, otro convirtió la voz en texto, otro entendió que usted quería indicaciones y otro estimó el tiempo de viaje. Cada uno aprendió su trabajo a partir de ejemplos. Lo mismo vale para el asistente de voz de un altavoz de cocina: responde a su nombre, pero nadie le escribió nunca una regla que diga cómo suena ese nombre.

Para ver por qué, intente escribir usted mismo el primero de estos programas, el que se despierta con una palabra. Imagínese solo en una habitación con un ordenador y nada más; la figura del libro muestra justo eso, una persona mirando un editor de código y una onda sonora. Empiece por lo que el programa recibe en realidad. El micrófono no oye palabras; mide, una y otra vez, cuánto empuja el aire su diminuta membrana. Cada una de esas medidas se llama muestra, y es simplemente la altura, o amplitud, de la onda sonora en ese instante. Un teléfono toma unas 44 000 muestras por segundo. Así que un fragmento de dos segundos son $2 \times 44\,000 = 88\,000$ números. A partir de esos 88 000 números, el programa debe dar una sola respuesta: sí, la palabra de activación está ahí, o no, no está.

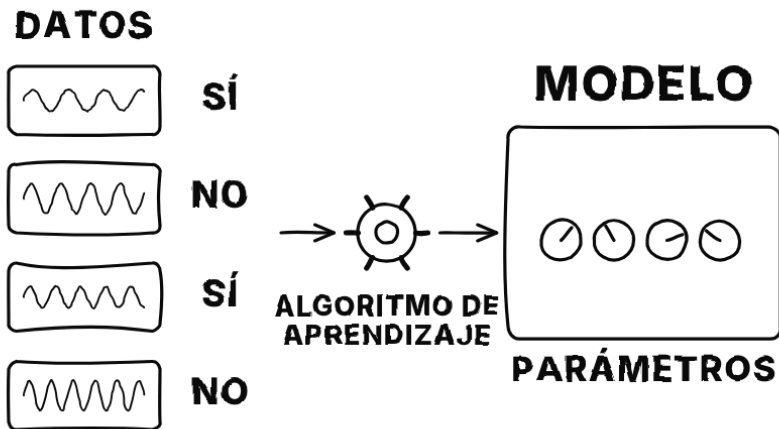


88 000 números, una respuesta

¿Qué regla escribiría? Quizá «si el sonido sube y baja dos veces, di que sí», pero la gente habla alto y bajo, rápido y despacio, resfriada y con acento, con música de fondo. El libro lo admite sin rodeos: los autores tampoco saben escribir un programa así desde cero. Precisamente por eso usamos el aprendizaje automático.

Este es el truco que propone el libro. Usted no puede escribir cómo reconoce la palabra «Alexa», pero la reconoce al instante. Así que puede hacer otra cosa: reunir miles de grabaciones cortas y marcar cada una usted mismo, sí si contiene la palabra y no si no la contiene. Una colección así de ejemplos se llama conjunto de datos, y las marcas se llaman etiquetas. Después, en lugar de escribir el reconocedor, escribe un programa flexible cuyo comportamiento

depende de muchos números ajustables, llamados parámetros. Por último, deja que el conjunto de datos decida qué valores de los parámetros hacen que el programa puntúe mejor según una medida que usted eligió, por ejemplo, con qué frecuencia su sí o no coincide con su etiqueta.

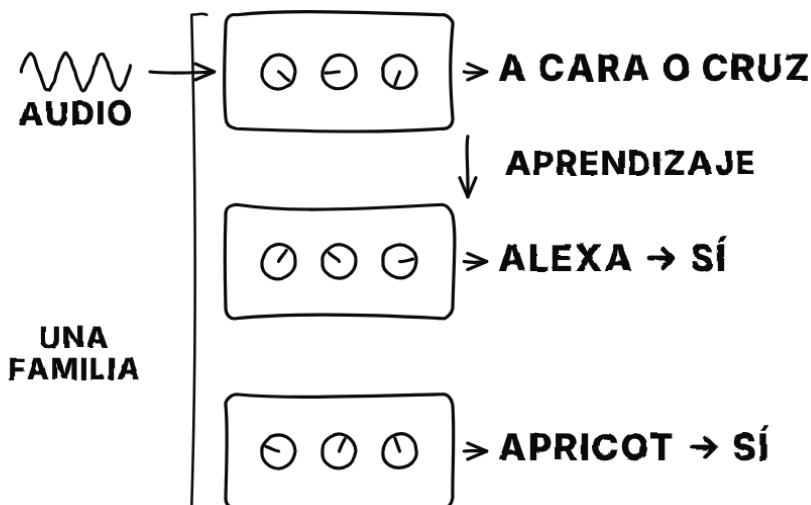


Las etiquetas fijan las perillas

Los autores proponen imaginar los parámetros como las perillas de una mesa de mezclas. Si gira las perillas, el programa se comporta de otra manera; el mismo sonido de entrada da otra respuesta. Una vez fijadas todas las perillas, el libro llama al programa modelo. Todos los programas distintos que podría obtener girando las mismas perillas forman una familia de modelos. Y el programa que mira el conjunto de datos y decide dónde poner las perillas se llama algoritmo de

aprendizaje: un programa que escribe los ajustes de otro programa. Un filtro de spam se construye exactamente así: su familia se fija de antemano, y los mensajes que la gente marca como spam deciden dónde quedan sus perillas.

Antes de que empiece el aprendizaje, alguien tiene que plantear la tarea con exactitud. Aquí la entrada es un fragmento de audio, esas decenas de miles de números, y la salida es una de dos respuestas, sí o no. Después se elige una familia de modelos adecuada para ese tipo de entrada y de salida. Si la familia es buena, existe una posición de las perillas en la que el modelo dice sí cada vez que oye «Alexa». Y como la palabra de activación podría ser cualquiera, una buena familia es lo bastante rica como para que otra posición le haga decir sí solo a «Apricot». Ambas tareas se parecen, así que una sola familia debería servir para las dos. Pero si quisiera pasar de fotografías a pies de foto, o de frases en inglés a frases en chino, las entradas y salidas son de otra naturaleza, y podría necesitar una familia completamente distinta.

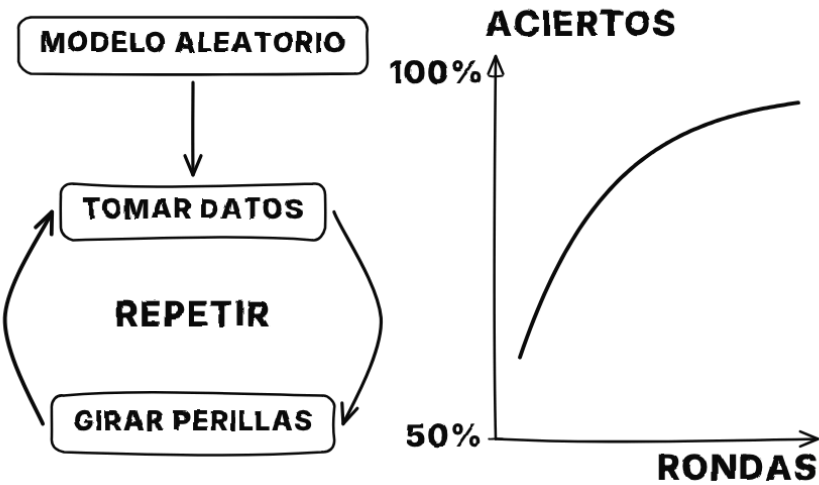


Una familia, muchos ajustes

Ahora bien, ¿dónde empiezan las perillas? Si las pone al azar, el modelo casi seguro no reconocerá «Alexa», ni «Apricot», ni ninguna otra palabra; dirá sí y no más o menos por casualidad. En el aprendizaje automático, aprender significa exactamente esto: encontrar la posición de las perillas que hace que el modelo haga lo que queremos. También decimos que entrenamos el modelo con datos.

La segunda figura del libro dibuja el entrenamiento como un bucle de cuatro pasos. Primero, se empieza con un modelo cuyas perillas están al azar y que no sirve para nada. Segundo, se toma una parte de los datos, por ejemplo un puñado de grabaciones con sus etiquetas de sí o no. Tercero, se giran un poco las perillas para que el modelo acierte

más de esos ejemplos. Cuarto, se vuelve al segundo paso y se sigue dando vueltas hasta que, en palabras de los autores, «el modelo sea estupendo». Imagine que entran unos cientos de grabaciones a la vez, que tras cada lote cada perilla gira un poco, y que la proporción de aciertos sube desde la mitad, que es tirar una moneda, hasta casi todas.



El bucle de entrenamiento

Pruébalo

Pida a un amigo que esconda un objeto en una habitación y le guíe solo con «caliente» y «frío». Empieza a caminar en una dirección al azar, lo cual no sirve de nada; tras cada paso oye una palabra y se corrige un poco; y repite. Nunca aprende una regla sobre dónde está el objeto, pero tras suficientes vueltas acaba justo a su lado. Entrenar un modelo funciona igual, salvo que tiene millones de direcciones en las que moverse a la vez y el «caliente» sale de comparar sus respuestas con las etiquetas.

Para recordar

Su teléfono se despierta con una palabra porque un modelo la aprendió con ejemplos, no porque alguien escribiera una regla para ello. La entrada es enorme, unas 44 000 muestras de amplitud por segundo, y la salida es un único sí o no. No podemos describir cómo oímos una palabra, pero sí podemos etiquetar grabaciones, y ese conjunto de datos etiquetado es lo que enseña al programa. Un modelo es un programa flexible con sus perillas, los parámetros, fijadas en valores concretos; el algoritmo de aprendizaje es lo que encuentra esos valores. El entrenamiento es un bucle: empezar al azar, mirar algunos ejemplos etiquetados, girar las perillas para acertar más en ellos y repetir.

Preguntas

1. Nadie sabe escribir cómo reconocer la palabra «Alexa». ¿Qué permite, aun así, enseñar a un programa a hacerlo?
 - a. Las personas reconocen la palabra y pueden etiquetar grabaciones con sí o no
 - b. El micrófono mide la palabra directamente
 - c. La palabra siempre tiene el mismo patrón de volumen
 - d. El teléfono guarda una grabación de cada voz posible
2. Un programa de palabra de activación tiene todas sus perillas al azar. ¿Qué cabe esperar?
 - a. Reconoce «Alexa» pero no otras palabras
 - b. Se niega a responder
 - c. Responde sí y no más o menos por casualidad
 - d. Reconoce todas las palabras igual de bien
3. ¿Por qué la misma familia de modelos podría servir para «Alexa» y «Apricot», pero no para convertir fotografías en pies de foto?
 - a. Los pies de foto son más largos que las palabras de activación
 - b. De fotografías a pies de foto es otro tipo de entrada y de salida
 - c. «Apricot» no es una palabra de activación real
 - d. El algoritmo de aprendizaje no puede tratar fotografías
4. ¿Cuál de estos es el algoritmo de aprendizaje en el ejemplo de la palabra de activación?

- a. El micrófono que recoge las muestras
- b. El modelo terminado, con sus perillas fijadas
- c. Las grabaciones etiquetadas con sí y no
- d. El programa que usa el conjunto de datos para fijar las perillas

Respuestas

1. **a)** Usted no puede escribir cómo reconoce la palabra, pero la reconoce al instante, así que puede etiquetar miles de grabaciones y dejar que ese conjunto de datos enseñe al programa.
2. **c)** Con las perillas al azar, el modelo casi seguro no reconocerá ninguna palabra; aprender es encontrar la posición que le hace hacer lo que queremos.
3. **b)** Las dos palabras de activación se parecen, así que una familia sirve para ambas; de fotos a pies de foto, las entradas y salidas son de otra naturaleza y puede hacer falta otra familia.
4. **d)** El programa que mira el conjunto de datos y decide dónde poner las perillas es el algoritmo de aprendizaje, un programa que escribe los ajustes de otro programa.

Ejemplos resueltos

Cuántos números oye el modelo de palabra de activación El micrófono de un teléfono toma unas 44 000 muestras por segundo. ¿Cuántos números recibe un modelo de palabra de activación de un fragmento de 3 segundos, y cuántas respuestas debe dar?

Cada muestra es un número, la altura de la onda sonora en ese instante.

En un segundo hay unas 44 000 muestras.

En 3 segundos: $3 \times 44\,000 = 132\,000$ muestras.

La tarea se plantea como audio de entrada y sí o no de salida, así que todos estos números llevan a una sola respuesta.

Respuesta: Unos 132 000 números de entrada y una sola respuesta de sí o no

Ejercicios

1. Ordene los cuatro pasos del entrenamiento y diga qué hace cada uno: repetir; ajustar las perillas; empezar con un modelo aleatorio; tomar algunos datos etiquetados.

Pista: El bucle no puede empezar hasta que haya un modelo, por inútil que sea.

Respuesta: Empezar con un modelo cuyas perillas están al azar y que no sirve para nada; tomar algunos datos con etiquetas; ajustar las perillas para que el modelo acierte más de esos ejemplos; repetir los dos últimos pasos hasta que el modelo sea bueno.

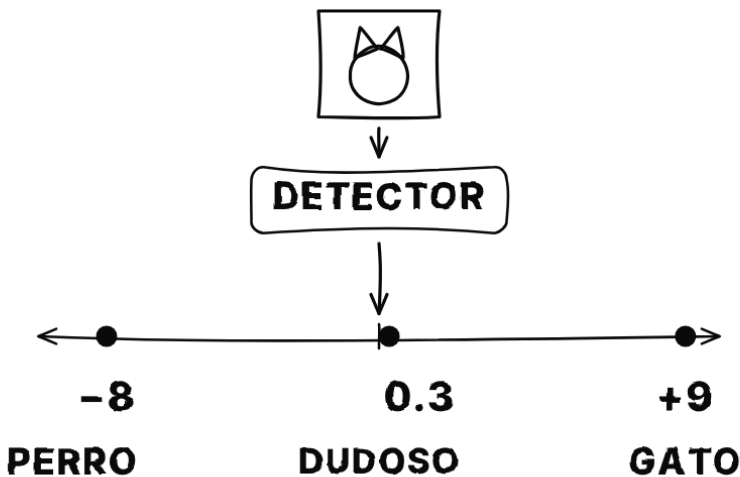
2. Un fragmento entregado a un modelo de palabra de activación contiene 88 000 muestras. ¿Cuánto dura el fragmento?

Pista: Divida entre el número de muestras que se toman en un segundo.

Respuesta: $88\,000 \div 44\,000 = 2$, así que el fragmento dura unos 2 segundos.

Programar con datos

Haga una foto a su gato con el teléfono, luego abra la galería y escriba «gato» en el buscador. La foto aparece, junto con todos los demás gatos que haya fotografiado alguna vez, y ningún perro. Nadie escribió para esa búsqueda una regla que describa bigotes y orejas. En algún lugar, a un programa le mostraron muchísimas fotos de gatos y perros, cada una marcada con la respuesta correcta, y aprendió de ellas la diferencia.



La puntuación de un detector de gatos

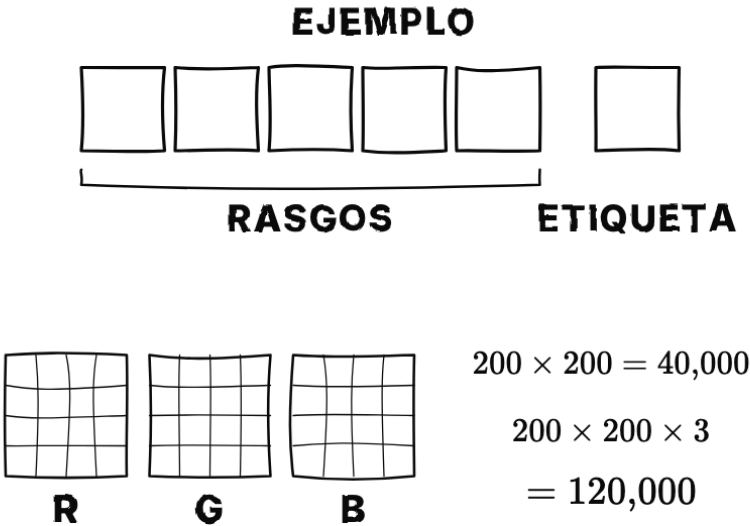
Para eso servía el bucle de entrenamiento de la parte anterior. En lugar de escribir un reconocedor de palabras de activación, escribimos un programa que puede aprender a reconocerlas cuando se le da un gran conjunto de datos etiquetados. Los autores lo llaman programar con datos: el comportamiento del programa no lo deciden líneas de código, sino los ejemplos que le mostramos. Del mismo modo podemos «programar» un detector de gatos simplemente dándole muchas imágenes de gatos y perros.

El libro dice qué acaba haciendo un detector así. Para cada imagen da un número. Un número positivo muy grande significa «gato», un número negativo muy grande significa «perro», y algo cercano a cero significa «no estoy seguro». Así que si da +9 para una imagen, -8 para otra y 0,3 para una tercera, está seguro de un gato, seguro de un perro e indeciso sobre la última, quizá un zorro borroso. El libro añade que esto apenas roza la superficie, y que el aprendizaje profundo es solo una de muchas formas populares de resolver estos problemas.

Ahora el libro da un paso atrás y nombra el tipo de problema que hemos estado resolviendo. Teníamos grabaciones cuya respuesta se conocía, sí o no, y queríamos predecir la respuesta desconocida para grabaciones nuevas. Aprender a predecir una etiqueta desconocida elegida a partir de entradas conocidas, usando ejemplos cuyas etiquetas se conocen, se llama aprendizaje supervisado. Es solo un tipo de aprendizaje automático, y en partes posteriores aparecerán otros. Pero sea cual sea el tipo, dicen los autores, siempre aparecen cuatro ingredientes. Están los datos de los que aprender. Está el modelo, que convierte los datos en una respuesta. Está la función objetivo, una puntuación que dice lo bien o lo mal que lo hace el

modelo. Y está el algoritmo que gira las perillas del modelo para que la puntuación mejore.

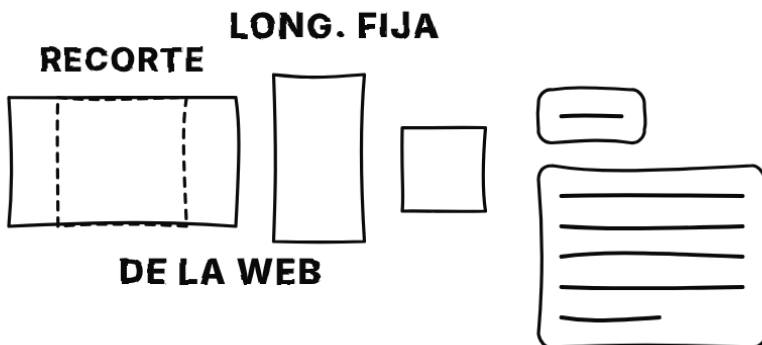
Esta parte se ocupa del primer ingrediente, los datos. Un conjunto de datos es una colección de ejemplos, y el ordenador solo puede trabajar con ellos una vez convertidos en números. Cada ejemplo tiene un conjunto de atributos, llamados características, a partir de los cuales el modelo debe hacer su predicción. En el aprendizaje supervisado hay además un atributo especial, la etiqueta, que es lo que queremos predecir; no se le da al modelo como entrada. El libro menciona que las mismas cosas tienen también otros nombres: a un ejemplo se le llama también punto de datos o muestra, y a las características, entradas o covariables.



Una foto como 120 000 números

En las imágenes, las características son la propia fotografía, escrita como números, y la etiqueta es un número que nombra la categoría, por ejemplo 1 para gato y 2 para perro. Una fotografía en color se convierte en tres cuadrículas de números: una para el brillo de la luz roja en cada píxel, otra para la verde y otra para la azul. El libro hace la cuenta para una foto de 200×200 píxeles. Una cuadrícula tiene $200 \times 200 = 40\,000$ números, y hay tres cuadrículas, así que la foto entera son $200 \times 200 \times 3 = 120\,000$ números. Una sola imagen pequeña ya es una lista de ciento veinte mil valores. Una foto recién sacada de un teléfono moderno tiene millones de píxeles, así que se convierte en muchos millones de números antes de que ningún programa pueda aprender de ella.

Los datos no tienen por qué ser imágenes. El segundo ejemplo del libro son historias clínicas, donde la tarea es predecir si un paciente sobrevivirá los próximos 30 días. Las características son cosas que se registran de todos modos: la edad, constantes vitales como el pulso y la tensión arterial, otras enfermedades, los medicamentos que toma, operaciones recientes. La etiqueta, para cada paciente de los registros pasados, es simplemente sí o no: ¿sobrevivió ese paciente los 30 días?



LONG. VARIABLE

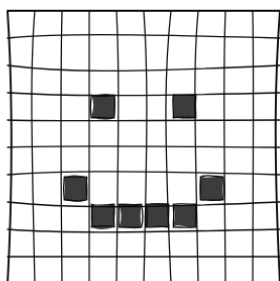
Longitud fija frente a longitud variable

En ambos ejemplos, cada caso tiene el mismo número de características. Cada foto de 200×200 píxeles da exactamente 120 000 números, y cada historia clínica da la misma lista de mediciones. Estas entradas se llaman vectores de longitud fija, siendo un vector una lista ordenada de números, y la longitud de esa lista se llama dimensionalidad de los datos. Así que la dimensionalidad de nuestros datos de fotos es 120 000. La longitud fija es cómoda, dice el libro, porque es una complicación menos de la que preocuparse.

Los datos reales a menudo se niegan a ser tan ordenados. Las imágenes de un microscopio salen de la misma máquina y comparten tamaño, pero las imágenes recogidas de Internet vienen en todos los tamaños y formas. Podríamos recortarlas todas a un cuadrado

estándar, pero lo que cortemos se pierde. El texto es aún más obstinado. Piense en las reseñas de una tienda de aplicaciones: una dice solo «¡es un asco!», la siguiente se extiende durante páginas sobre cada función. Una reseña no tiene una longitud fija natural. El libro cuenta como una de las grandes ventajas del aprendizaje profundo que sus modelos manejan datos de longitud variable con relativa soltura.

Por último, el libro dice algo sobre la cantidad. En general, cuantos más datos tenemos, más fácil es el trabajo. Más datos nos permiten entrenar modelos más potentes y apoyarnos menos en suposiciones inventadas de antemano. El paso de conjuntos de datos bastante pequeños a otros muy grandes es una de las principales razones del éxito del aprendizaje profundo moderno. Muchos de sus modelos más interesantes simplemente no funcionan sin grandes conjuntos de datos, y algunos que sí funcionan con pocos datos no son mejores que los métodos antiguos.



0 0 0 0 0 0 0 0 0 0

⋮

0 0 0 1 0 0 1 0 0 0

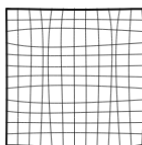
⋮

0 0 0 1 1 1 1 0 0 0

⋮

100 NÚMEROS

DIMENSIÓN 100



144

≠ 100

Una carita como 100 números

Pruébalo

Tome una hoja cuadriculada, de diez cuadros por diez, y coloree una carita sonriente sencilla en unos cuantos cuadros. Escriba 1 en cada cuadro coloreado y 0 en cada cuadro vacío, y lea los números fila a fila. Ahora tiene una lista de 100 números que es la imagen, un vector de longitud fija de dimensionalidad 100. Dibuje una segunda carita en una hoja de doce por doce y obtendrá 144 números, que ya no caben en la misma lista; ese es el problema de la longitud variable en miniatura.

Para recordar

Programar con datos significa decidir el comportamiento de un programa con los ejemplos que le mostramos, no con reglas que escribimos. Todo problema de aprendizaje automático tiene cuatro ingredientes: datos, un modelo, una función objetivo que puntúa el modelo y un algoritmo que mejora la puntuación. En el aprendizaje supervisado, cada ejemplo tiene características, que el modelo ve, y una etiqueta, que debe predecir. Una foto en color de 200×200 son 120 000 números; cuando cada ejemplo tiene el mismo número de características, ese número es la dimensionalidad. El aprendizaje profundo maneja bien los datos de longitud variable, y necesita grandes conjuntos de datos para mostrar su fuerza.

Preguntas

1. Un detector de gatos da -7 para una foto. ¿Qué le indica eso?
 - a. Está bastante seguro de que la foto muestra un perro
 - b. Está bastante seguro de que la foto muestra un gato
 - c. No está seguro de qué muestra la foto
 - d. La foto tiene demasiados pocos píxeles para juzgar
2. En un conjunto de datos hospitalarios para predecir la supervivencia a 30 días, ¿cuál de estos es la etiqueta y no una característica?
 - a. La edad del paciente

- b. Los medicamentos que toma el paciente
 - c. Si el paciente sobrevivió los 30 días
 - d. La tensión arterial del paciente
3. ¿Por qué es difícil convertir las reseñas de una tienda de aplicaciones en vectores de longitud fija?
- a. No contienen números
 - b. Van de dos palabras a muchas páginas
 - c. Están escritas en idiomas distintos
 - d. Son demasiado breves para aprender de ellas
4. ¿Cuál de estos NO es uno de los cuatro ingredientes de todo problema de aprendizaje automático?
- a. Una función objetivo que puntúa el modelo
 - b. Datos de los que aprender
 - c. Un algoritmo que mejora la puntuación
 - d. Una regla escrita a mano para cada caso

Respuestas

1. **a)** Un número positivo muy grande significa gato, un número negativo muy grande significa perro, y algo cercano a cero significa no estoy seguro.
2. **c)** Las características son cosas que se registran de todos modos, como la edad, las constantes vitales y los medicamentos; la etiqueta es si cada paciente pasado sobrevivió los 30 días.
3. **b)** Una reseña dice solo «¡es un asco!» y otra se extiende durante páginas, así que una reseña no tiene una longitud fija natural.
4. **d)** Los cuatro ingredientes son datos, un modelo, una función objetivo y un algoritmo que gira las perillas; las reglas escritas a mano son justo lo que

el aprendizaje automático sustituye.

Ejemplos resueltos

Los números de una foto de 200×200 Una fotografía en color de 200×200 píxeles se escribe como tres cuadrículas de números, para el rojo, el verde y el azul. ¿Cuántos números son, y cuál es la dimensionalidad de un conjunto de datos de fotos así?

Una cuadrícula contiene un valor de brillo por píxel: $200 \times 200 = 40\,000$ números.

Hay tres cuadrículas, una por cada color de luz: $40\,000 \times 3 = 120\,000$.

Cada foto de este tamaño da el mismo número de valores, así que las entradas son vectores de longitud fija.

La dimensionalidad es la longitud de esos vectores.

Respuesta: 120 000 números por foto; la dimensionalidad es 120 000

Ejercicios

1. Una foto de 300×200 píxeles se escribe como tres cuadrículas para el rojo, el verde y el azul. ¿Cuántos números son?

Pista: Primero cuente los píxeles y luego multiplique por el número de cuadrículas de color.

Respuesta: $300 \times 200 = 60\,000$ píxeles, por 3 cuadrículas da 180 000 números.

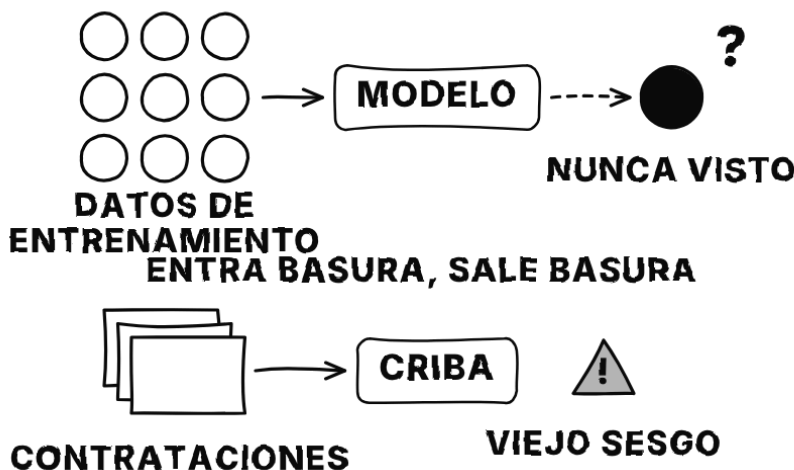
2. Una carita sonriente se colorea en una hoja cuadrículada de 10 por 10 y se escribe con unos y ceros. Una segunda se dibuja en una de 12 por 12. ¿Por qué no pueden ir las dos en el mismo conjunto de vectores de longitud fija sin cambiar algo?

Pista: Cuente los números que da cada hoja.

Respuesta: La primera da $10 \times 10 = 100$ números y la segunda $12 \times 12 = 144$, así que los vectores tienen longitudes distintas; habría que recortar o cambiar el tamaño, con riesgo de perder información, o usar un modelo que maneje longitudes variables.

Los datos adecuados y una puntuación justa

Imagine una aplicación para revisar la piel que mira la foto de un lunar y dice si debería verlo un médico. Se entrenó con muchos miles de fotos y en las pruebas era muy precisa. Ahora alguien con piel oscura fotografía un lunar y recibe una respuesta claramente errónea. El libro usa exactamente este caso, un sistema de reconocimiento de cáncer de piel que nunca había visto piel negra, y da la razón en pocas palabras: los datos eran erróneos.



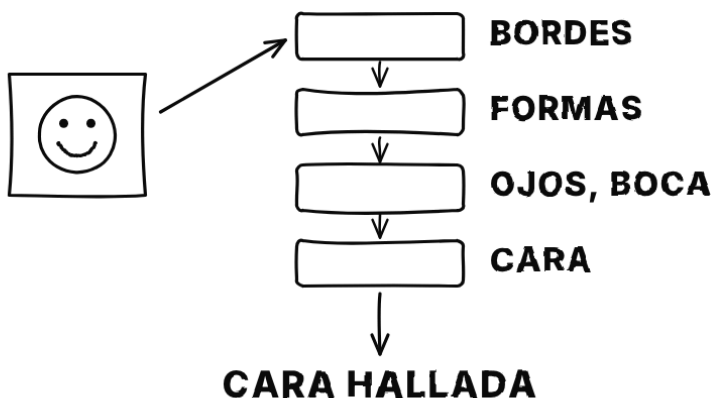
Grupos ausentes, viejos prejuicios

La parte anterior trataba de tener suficientes datos y convertirlos en números. Aquí los autores añaden que eso no basta: necesitamos los datos adecuados. Si los ejemplos están llenos de errores, o si las características que elegimos simplemente no dicen nada de lo que queremos predecir, el aprendizaje fracasa por ingenioso que sea el método. Los programadores tienen un tópico para esto: «si entra basura, sale basura». Y una mala puntuación no es lo peor que puede pasar. El libro nombra ámbitos en los que los datos erróneos hacen daño real: predecir delitos para la policía, cribar solicitudes de empleo y decidir quién obtiene un préstamo.

Hay dos formas habituales de equivocarse. La primera es que algún grupo de personas simplemente falta en los datos de entrenamiento, como la piel oscura en nuestra aplicación. El modelo nunca ha visto casos así, de modo que no tiene en qué basarse. La segunda es más sutil: los datos incluyen a todo el mundo, pero recogen viejos prejuicios. Entrene un programa con años de decisiones de contratación pasadas y, si esas decisiones fueron injustas, el programa aprenderá la injusticia y la repetirá automáticamente y a gran escala. Un chatbot entrenado con texto de Internet puede adquirir los mismos sesgos que contiene ese texto, y por eso sus creadores lo prueban con casos así. El libro subraya que nada de esto necesita un villano: todo puede ocurrir sin que el científico de datos lo pretenda, ni siquiera lo note.

El segundo ingrediente es el modelo. Casi todo el aprendizaje automático transforma datos de un tipo en una predicción de otro tipo. Un sistema podría recibir fotos y decir cuánto sonríe cada persona, o recibir lecturas de sensores de una máquina y decir lo normales o lo raras que parecen. El modelo es esa maquinaria, de la entrada a la predicción, y en este libro es un modelo estadístico, uno

cuyos ajustes se estiman a partir de datos. Los modelos sencillos son perfectamente válidos para problemas sencillos. Pero los problemas que interesan a este libro van más allá de lo que pueden hacer los métodos clásicos, y el aprendizaje profundo se distingue sobre todo por los potentes modelos que usa. Esos modelos hacen pasar los datos por muchas transformaciones seguidas, cada una alimentando a la siguiente, apiladas de arriba abajo; de esa profundidad viene el nombre de aprendizaje profundo. Una aplicación de fotos que encuentra caras hace esto: las primeras etapas detectan bordes, las siguientes formas, las siguientes ojos y bocas, y la última dice «cara».

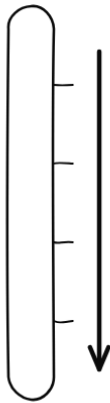


Muchas etapas, una tras otra

El tercer ingrediente responde a una pregunta que suena filosófica. Dijimos que aprender significa mejorar en una tarea. Pero ¿quién

decide qué es «mejor»? Si una persona propone un cambio en el modelo, otra bien puede sostener que lo ha empeorado. Para zanjar esas discusiones, el aprendizaje automático necesita un número que mida lo bueno o lo malo que es un modelo. Ese número se llama función objetivo.

PÉRDIDA



MEJOR

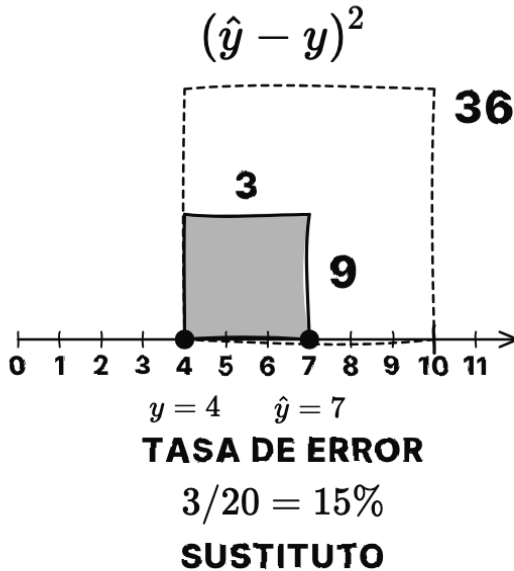
puntuación $\uparrow$

pérdida = $-$ puntuación $\downarrow$

Menos es mejor

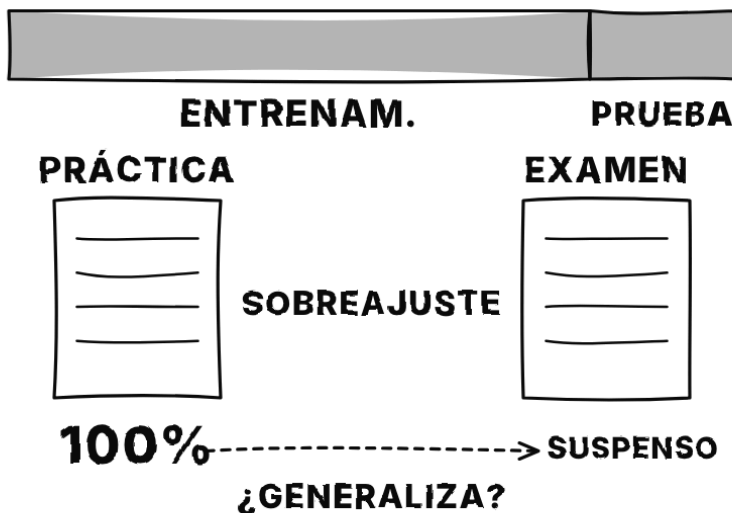
Por convención, las funciones objetivo se plantean de modo que menos sea mejor. Es solo una costumbre, dice el libro. Si tiene una puntuación en la que más es mejor, como la proporción de aciertos, basta con ponerle un signo menos delante, y ahora menos es mejor sin que haya cambiado nada más. Como menos es mejor, estas funciones también se llaman funciones de pérdida: cuanto menor la pérdida, mejor el modelo.

El libro nombra las dos pérdidas más comunes. Cuando predecimos un número, la pérdida habitual es el error cuadrático: tomar la predicción, restarle el valor real y elevar la diferencia al cuadrado. Por escrito, si la predicción es $\hat{y}$ y el valor real es y , el error cuadrático es $(\hat{y} - y)^2$. Supongamos que un modelo predice que un envío tardará 7 días y tarda 4. La diferencia es $7 - 4 = 3$, y el error cuadrático es $3^2 = 9$. Elevar al cuadrado hace que la pérdida sea positiva sea cual sea el lado por el que fallemos, y castiga un fallo grande mucho más que uno pequeño: fallar por 6 cuesta 36, cuatro veces lo que cuesta fallar por 3. Cuando clasificamos cosas en clases, la puntuación natural es la tasa de error, la proporción de ejemplos en los que fallamos; 3 fallos de 20 son $3/20 = 0,15$, es decir, el 15 por ciento. El libro advierte que algunas pérdidas, como el error cuadrático, son fáciles de reducir paso a paso, mientras que otras, como la tasa de error, son incómodas, porque mover un poco las perillas normalmente no cambia ninguna respuesta y la tasa solo salta de vez en cuando. En esos casos se optimiza un sustituto con el que es más fácil trabajar, llamado objetivo sustituto.



Error cuadrático y tasa de error

Mientras entrenamos, vemos la pérdida como algo que depende de las perillas del modelo, los parámetros, y mantenemos fijos los datos. Reunimos un conjunto de ejemplos para el entrenamiento y buscamos las posiciones de las perillas que hacen la pérdida en ellos lo más pequeña posible. Pero una pérdida pequeña en esos ejemplos no promete buenas respuestas en otros nuevos. Por eso los datos se dividen en dos: un conjunto de entrenamiento, que se usa para fijar las perillas, y un conjunto de prueba, que se aparta y se usa solo para comprobar el resultado. Al final se informa de ambas puntuaciones.



Entrenamiento, prueba y sobreajuste

Pruébalo

Piense en un estudiante que prepara un examen final con un cuadernillo de preguntas de práctica. Al cabo de unas semanas saca la nota máxima en el cuadernillo cada vez, porque se ha aprendido las respuestas de memoria. Luego llega el examen con preguntas nuevas, y suspende. La nota del cuadernillo era la puntuación de entrenamiento; el examen era la prueba.

Esa historia es la propia imagen que da el libro de lo que sale mal. Cuando un modelo lo hace bien en el conjunto de entrenamiento pero no puede trasladarlo a datos que no ha visto, decimos que se está

sobreajustando a los datos de entrenamiento. Lo que de verdad queremos es lo contrario, un modelo que generalice, que lo haga más o menos igual de bien con ejemplos nuevos que con aquellos de los que aprendió. Un filtro de spam que solo bloquea exactamente los mensajes que se le mostraron estaría sobreajustado; uno útil también bloquea la nueva estafa de la semana que viene.

Para recordar

Muchos datos no bastan; tienen que ser los adecuados, porque si entra basura, sale basura. Los datos que dejan fuera a personas, o que arrastran viejos prejuicios, producen modelos que fallan o tratan injustamente a la gente, aunque nadie lo pretendiera. Un modelo convierte una entrada en una predicción, y los modelos del aprendizaje profundo encadenan muchas transformaciones una tras otra. Una función objetivo, o pérdida, puntúa el modelo, y menos es mejor; el error cuadrático $(\hat{y} - y)^2$ es la habitual para números, la tasa de error para clases. Entrenamos con un conjunto de entrenamiento y comprobamos con un conjunto de prueba aparte, porque hacerlo bien solo con los datos de entrenamiento es sobreajuste.

Preguntas

1. Un modelo de contratación se entrena con años de decisiones pasadas que incluían a todo el mundo pero eran injustas. ¿Qué dice la lección que puede ocurrir?

- a. El modelo aprende la injusticia y la repite automáticamente
 - b. El modelo falla porque falta algún grupo
 - c. El modelo corrige la injusticia por sí mismo
 - d. Nada, salvo que el científico de datos lo pretendiera
2. Un modelo predice 10 donde el valor real es 13. ¿Cuál es su error cuadrático?
- a. 3
 - b. 6
 - c. 9
 - d. -9
3. ¿Por qué a menudo se optimiza un objetivo sustituto en lugar de la tasa de error?
- a. La tasa de error siempre es demasiado grande
 - b. La tasa de error no puede calcularse en un conjunto de prueba
 - c. La tasa de error premia el sobreajuste
 - d. Los giros pequeños de las perillas no suelen cambiar la tasa de error, así que es incómoda de mejorar paso a paso
4. Un filtro de spam bloquea todos los mensajes con los que se entrenó, pero deja pasar la mayoría de las estafas nuevas. ¿Qué ocurre?
- a. Usa una función de pérdida equivocada
 - b. Se está sobreajustando a los datos de entrenamiento
 - c. Su conjunto de entrenamiento era demasiado pequeño para contener spam
 - d. Se puntuó con menos es mejor

Respuestas

1. **a)** Los datos pueden incluir a todo el mundo y aun así recoger viejos prejuicios; un programa entrenado con decisiones pasadas injustas los aprende y los repite, aunque nadie lo pretenda.
2. **c)** El error cuadrático es la predicción menos el valor real, al cuadrado: $10 - 13 = -3$, y $(-3)^2 = 9$; elevar al cuadrado lo hace positivo sea cual sea el lado por el que fallemos.
3. **d)** Mover un poco las perillas normalmente no cambia ninguna respuesta, así que la tasa de error solo salta de vez en cuando; por eso se optimiza un sustituto con el que es más fácil trabajar.
4. **b)** Hacerlo bien en el conjunto de entrenamiento pero no trasladarlo a datos no vistos es sobreajuste, como el estudiante que memorizó el cuadernillo de práctica.

Ejemplos resueltos

Dos fallos, dos errores cuadráticos Un modelo de tiempo de entrega predice 7 días para un paquete que tarda 4 días, y 7 días para otro que tarda 1 día. ¿Cuáles son los dos errores cuadráticos y cómo se comparan?

El error cuadrático es $(\text{predicción} - \text{valor real})^2$.

Primer paquete: $7 - 4 = 3$, así que el error cuadrático es $3^2 = 9$.

Segundo paquete: $7 - 1 = 6$, así que el error cuadrático es $6^2 = 36$.

El segundo fallo es el doble de grande pero cuesta $36 \div 9 = 4$ veces más: elevar al cuadrado castiga mucho más los fallos grandes.

Respuesta: 9 y 36; el fallo mayor cuesta cuatro veces más

Ejercicios

1. Un clasificador se prueba con 50 fotos y falla 4. ¿Cuál es su tasa de error?

Pista: La tasa de error es la proporción de ejemplos predichos mal.

Respuesta: $4 \div 50 = 0,08$, es decir, el 8 por ciento.

2. Su puntuación es la «proporción de aciertos», en la que más es mejor. ¿Cómo la convierte en una pérdida en la que menos es mejor, sin cambiar lo que mide?

Pista: Piense en el signo.

Respuesta: Poniéndole un signo menos delante: una proporción de 0,9 pasa a $-0,9$ y 0,8 pasa a $-0,8$, así que el mejor modelo tiene ahora el número más bajo y nada más ha cambiado.

3. Explique con sus propias palabras por qué una aplicación para revisar la piel que era muy precisa en las pruebas puede aun así fallar gravemente con algunas personas.

Pista: Pregúntese quién aparecía en las fotos de entrenamiento y de prueba.

Respuesta: Si las personas de piel oscura faltaban en las fotos con las que aprendió, y también en las pruebas, el modelo nunca vio casos así y no tenía en qué basarse; la buena puntuación de las pruebas solo abarcaba al tipo de personas que había en los datos. Si entra basura, sale basura.

Bajar la cuesta y luego aprender de las respuestas

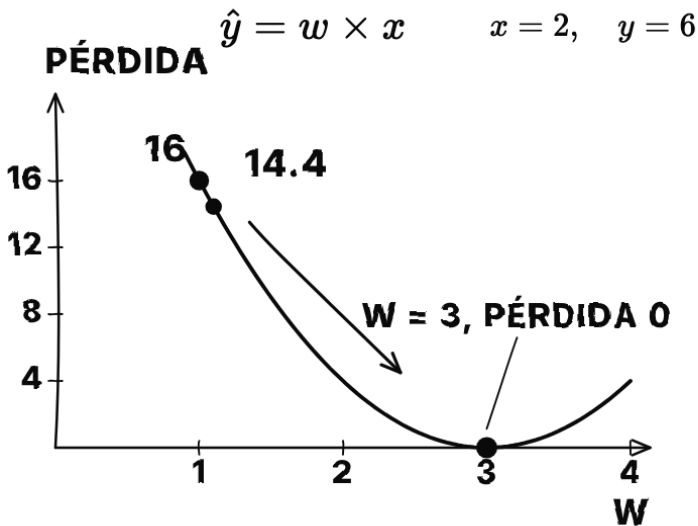
Póngase con los ojos vendados en un campo con colinas e intente llegar al punto más bajo. No ve el valle, pero siente el suelo bajo los pies. Tantee con un pie un poco hacia delante, un poco hacia atrás, un poco a cada lado, y note hacia dónde baja el terreno. Dé un pequeño paso en esa dirección y vuelva a tantear. Paso a paso llega al fondo sin haberlo visto nunca.



Tantear la pendiente cuesta abajo

Ese es el cuarto ingrediente del aprendizaje automático, el algoritmo que encuentra buenas posiciones de las perillas. Ya tenemos los datos, un modelo con sus perillas y una pérdida que dice lo mal que lo hace el modelo. Lo que aún falta es una forma de buscar los parámetros que hagan la pérdida lo más pequeña posible. Los métodos más usados en el aprendizaje profundo se basan en una sola idea, llamada descenso de gradiente. En cada paso se pregunta, para cada parámetro por turno: si moviera este un poquito, ¿subiría o bajaría la pérdida en el conjunto de entrenamiento, y cuánto? Luego mueve un poco cada parámetro en la dirección que reduce la pérdida. La altura del campo es la pérdida, su posición es el ajuste de las perillas, y la pendiente que nota con el pie es lo que mide el método.

Hagámoslo una vez con números, usando un modelo con una sola perilla w . El modelo predice un tiempo de entrega multiplicando la distancia por w , así que su predicción es $\hat{y} = w \times x$. Tenemos un ejemplo de entrenamiento: una distancia de $x = 2$ que tardó $y = 6$ días. Empezamos con $w = 1$: la predicción es 2, el fallo es $2 - 6 = -4$ y el error cuadrático es 16. Ahora movemos la perilla a $w = 1,1$: la predicción pasa a 2,2, el fallo a $-3,8$ y la pérdida a unos 14,4. La pérdida ha bajado, así que aumentar w es la dirección cuesta abajo, y el método da un paso en ese sentido. Repítalo, y w va subiendo poco a poco hacia 3, donde la predicción es exactamente 6 y la pérdida es cero. Un modelo real hace lo mismo con millones de perillas a la vez, cada una movida en su propia dirección cuesta abajo en cada paso. El modelo de lenguaje que hay detrás de un chatbot se entrenó exactamente así, con miles de millones de estos pequeños pasos cuesta abajo sobre sus perillas.

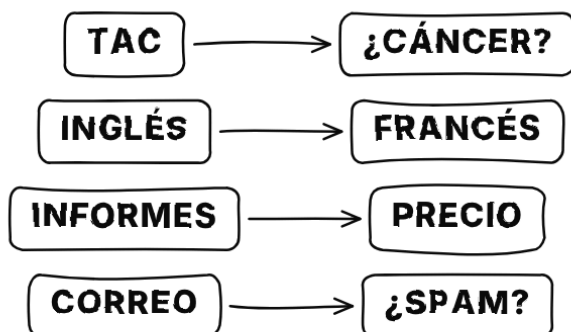


Una perilla, un ejemplo

Con los cuatro ingredientes en la mano, el libro pasa a un mapa de los tipos de problemas que puede resolver el aprendizaje automático. La palabra de activación era solo uno de ellos, y los autores quieren darnos un vocabulario común para el resto del libro.



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



Aprendizaje supervisado

El primer tipo, y el más común, es el aprendizaje supervisado. Se nos da un conjunto de datos en el que cada ejemplo viene como un par, características junto con una etiqueta, y debemos construir un modelo que prediga la etiqueta a partir de las características. Se llama supervisado porque nosotros, los supervisores, entregamos al modelo estos ejemplos etiquetados mientras se ajustan sus perillas. En el lenguaje de la probabilidad, lo que solemos querer es la probabilidad de cada etiqueta posible dadas las características que vemos. El aprendizaje supervisado no es el único tipo, pero reúne la mayoría de los usos con éxito del aprendizaje automático en la industria. La razón es que muchísimas tareas importantes encajan en el patrón «predecir lo desconocido a partir de los datos que tenemos». El libro da tres: cáncer o no cáncer a partir de un TAC; la traducción correcta al

francés de una frase en inglés; el precio de una acción el mes que viene a partir de los informes financieros de este mes. El filtro de spam de su correo también encaja: las características son el mensaje y la etiqueta es «spam» o «no spam».

Una breve descripción abarca todo el aprendizaje supervisado, pero detrás hay muchas decisiones, que dependen de cómo son las entradas y salidas, de lo grandes que son y de cuántas hay. Una secuencia de cualquier longitud, como una frase, requiere un tipo de modelo distinto del de una lista de números de longitud fija.

Después el libro esboza el proceso de principio a fin. Reunir muchos ejemplos cuyas características se conocen, elegir un subconjunto al azar y obtener la etiqueta real de cada uno. A veces las etiquetas ya existen en los registros, como si un paciente murió durante el año siguiente; a veces hay que pagar a personas para que las pongan, por ejemplo clasificando imágenes en categorías. Esas entradas con sus etiquetas forman el conjunto de entrenamiento. El conjunto de entrenamiento entra en un algoritmo de aprendizaje supervisado, que es una función poco corriente: usted le da un conjunto de datos entero y ella le devuelve otra función, el modelo entrenado. Por último, al modelo le llegan entradas nuevas que nunca ha visto, y lo que sale se toma como la predicción de sus etiquetas. La figura del libro lo dibuja como una línea: datos etiquetados, luego el algoritmo de aprendizaje, luego el modelo, y datos nuevos que pasan por el modelo hasta convertirse en predicciones.

Para recordar

El descenso de gradiente encuentra buenos parámetros tanteando la pendiente: para cada perilla, comprueba cómo cambiaría la pérdida de entrenamiento tras un pequeño empujón, y luego mueve cada perilla un poco cuesta abajo. Con una perilla y un ejemplo, unos pocos pasos así llevan la pérdida de 16 hacia cero. El aprendizaje supervisado consiste en predecir etiquetas a partir de características, aprendiendo de ejemplos que vienen en pares de características y etiqueta. La mayoría de los éxitos prácticos del aprendizaje automático son de este tipo. El algoritmo toma un conjunto de datos etiquetado y devuelve un modelo, que después predice etiquetas para entradas nuevas.

Preguntas

1. En el descenso de gradiente, mover un poco una perilla hacia arriba hace que suba la pérdida de entrenamiento. ¿Qué hace el método con esa perilla?
 - a. La deja exactamente donde está
 - b. La sube mucho, para escapar de la subida
 - c. La baja un poco, en la dirección que reduce la pérdida
 - d. La pone a un valor aleatorio
2. En el paseo con los ojos vendados por un campo con colinas, ¿qué representa la pérdida?
 - a. La altura del suelo donde está usted

- b. La dirección hacia la que mira
 - c. El tamaño de su paso
 - d. El número de pasos que ha dado
3. ¿Por qué la tarea «spam o no spam a partir del mensaje» es un ejemplo de aprendizaje supervisado?
- a. Porque el filtro descubre grupos de mensajes sin ninguna etiqueta
 - b. Porque el filtro aprende de recompensas que reúne con el tiempo
 - c. Porque los mensajes tienen una longitud fija
 - d. Porque aprende de ejemplos que emparejan un mensaje con su etiqueta
4. ¿Qué devuelve un algoritmo de aprendizaje supervisado cuando recibe un conjunto de datos etiquetado?
- a. Una sola predicción
 - b. Un modelo entrenado, que es a su vez una función
 - c. Una versión limpia del conjunto de datos
 - d. Una lista de etiquetas para los ejemplos de entrenamiento

Respuestas

- 1. c)** Para cada parámetro, el método comprueba cómo cambiaría la pérdida tras un pequeño empujón y luego lo mueve un poco en la dirección que la hace menor.
- 2. a)** La altura del campo es la pérdida, su posición es el ajuste de las perillas, y la pendiente que nota es lo que mide el método.
- 3. d)** El aprendizaje supervisado aprende a predecir etiquetas a partir de características usando ejemplos en pares de características y etiqueta; aquí

el mensaje son las características y spam o no spam es la etiqueta.

4. b) Usted le da un conjunto de datos entero y le devuelve otra función, el modelo entrenado, que después predice etiquetas para entradas nuevas.

Ejemplos resueltos

Un paso de descenso de gradiente a mano Un modelo predice $\hat{y} = w \times x$. El único ejemplo de entrenamiento es $x = 2$, $y = 6$. Partiendo de $w = 1$, compare el error cuadrático en $w = 1$ y en $w = 0,9$ y decida en qué sentido mueve w el descenso de gradiente.

En $w = 1$ la predicción es $1 \times 2 = 2$, el fallo es $2 - 6 = -4$ y el error cuadrático es 16.

En $w = 0,9$ la predicción es $0,9 \times 2 = 1,8$, el fallo es $-4,2$ y el error cuadrático es $4,2^2 = 17,64$.

Bajar w hizo subir la pérdida, de 16 a 17,64.

Así que la dirección cuesta abajo es hacia arriba: el descenso de gradiente aumenta w un poco.

Respuesta: La pérdida sube cuando w disminuye, así que w se mueve hacia arriba, hacia 3

Ejercicios

1. Con el mismo modelo $\hat{y} = w \times x$ y el ejemplo $x = 2$, $y = 6$, ¿cuál es el error cuadrático en $w = 2$ y en $w = 4$? ¿Dónde está el fondo del valle?

Pista: Calcule primero la predicción, luego el fallo, y elévelo al cuadrado.

Respuesta: En $w = 2$ la predicción es 4, el fallo -2 y la pérdida 4. En $w = 4$ la predicción es 8, el fallo $+2$ y la pérdida también 4. La pérdida es cero en $w = 3$, donde la predicción es exactamente 6, a medio camino.

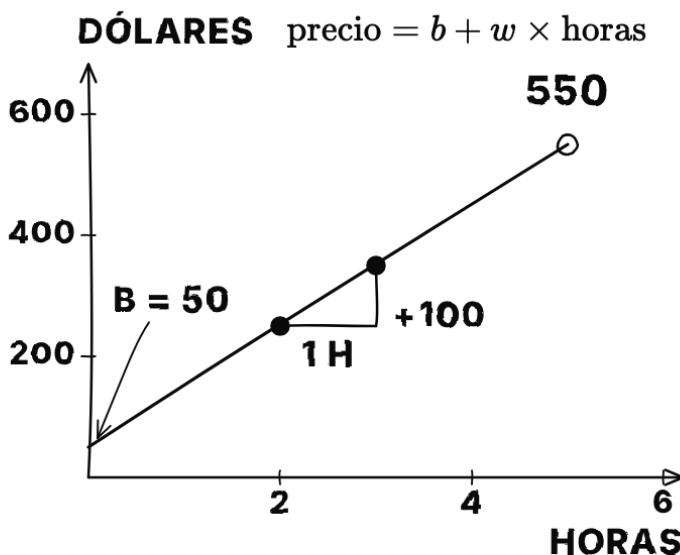
2. Describa con sus propias palabras el camino que va de un montón de ejemplos a una predicción en el aprendizaje supervisado.

Pista: Empiece por reunir ejemplos cuyas características se conocen.

Respuesta: Reunir ejemplos con características conocidas, elegir un subconjunto al azar y obtener sus etiquetas reales, de registros existentes o de personas que las ponen. Estos pares forman el conjunto de entrenamiento. El algoritmo de aprendizaje toma el conjunto de entrenamiento y devuelve un modelo entrenado, y las entradas nuevas que se dan a ese modelo salen como predicciones de sus etiquetas.

¿Cuánto? Regresión

Se le atascan los desagües y un fontanero pasa 3 horas desatascándolos; la factura dice 350 dólares. Un amigo llama al mismo fontanero, que trabaja 2 horas y cobra 250 dólares. Ahora un vecino le pregunta cuánto cuesta una visita. Usted nunca ha estudiado aprendizaje automático, pero ya puede responder, y el libro dice que justo de eso se trata: acaba de hacer una regresión.



Dos facturas fijan la recta

Así es como lo hizo, en tres pasos. Supuso que más horas cuestan más dinero, y que el fontanero cobra una cantidad fija por desplazarse más

una cantidad fija por hora. Escrita como fórmula, la factura es $\text{precio} = b + w \times \text{horas}$, donde b es el cargo por desplazamiento y w la tarifa por hora. La hora extra, de 2 a 3, añadió $350 - 250 = 100$ dólares, así que $w = 100$. Entonces 2 horas costaron 200 de los 250 dólares, lo que deja $b = 50$ por desplazarse. Compruébelo con la primera factura: $50 + 3 \times 100 = 350$, exacto. Así que un trabajo de 5 horas debería costar $50 + 5 \times 100 = 550$ dólares. Los dos números b y w son las perillas del modelo, y encontrarlos a partir de ejemplos es aprender; como la fórmula dibuja una línea recta, se llama regresión lineal.

El libro dice que la regresión es quizá la tarea supervisada más sencilla de entender, y la presenta con una tabla de ventas de viviendas. Cada fila es una casa; cada columna es una característica: la superficie en pies cuadrados, el número de dormitorios, el número de baños y los minutos que se tarda a pie hasta el centro. Una casa es un ejemplo, y su fila es su vector de características. En Nueva York o San Francisco, bromean los autores, salvo que dirija uno de los gigantes tecnológicos, su vivienda podría ser $[600, 1, 1, 60]$: pequeña, un dormitorio, un baño, a una hora a pie del centro. En Pittsburgh podría ser $[3000, 4, 3, 10]$. Cada casa da cuatro números en el mismo orden, un vector de longitud fija, y la mayoría de los métodos clásicos necesitan justo eso.

[pies², dorm., baños, min]

UNA CASA

NUEVA YORK

[600, 1, 1, 60]

PITTSBURGH

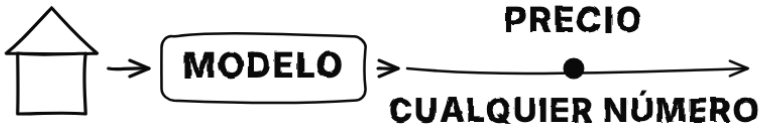
[3000, 4, 3, 10]

LONG. FIJA: 4 NÚMEROS

Casas como vectores de características

Pero lo que convierte un problema en regresión no son las características, sino la etiqueta. Si va a comprar una casa y quiere saber su precio justo, los ejemplos son ventas pasadas y las etiquetas, los precios realmente pagados. Cuando la etiqueta puede ser cualquier número, aunque sea dentro de un rango, el problema es una regresión, y el objetivo es un modelo cuyas predicciones se acerquen a los valores reales. El libro da una regla útil: cualquier pregunta que empiece por «cuánto» o «cuántos» probablemente sea una regresión. ¿Cuántas horas durará esta operación? ¿Cuánta lluvia caerá en esta ciudad en las próximas seis horas? ¿Cuántos días estará este paciente en el hospital? Los autores recuerdan también el premio Netflix de 2009, un millón de dólares para quien mejor predijera las estrellas que los usuarios darían a las películas. Una aplicación de reparto que

le dice «llega en 23 minutos» está respondiendo a una pregunta de regresión cada vez que la abre.



¿CUÁNTO? ¿CUÁNTOS?

HORAS DE OPERACIÓN

LLUVIA

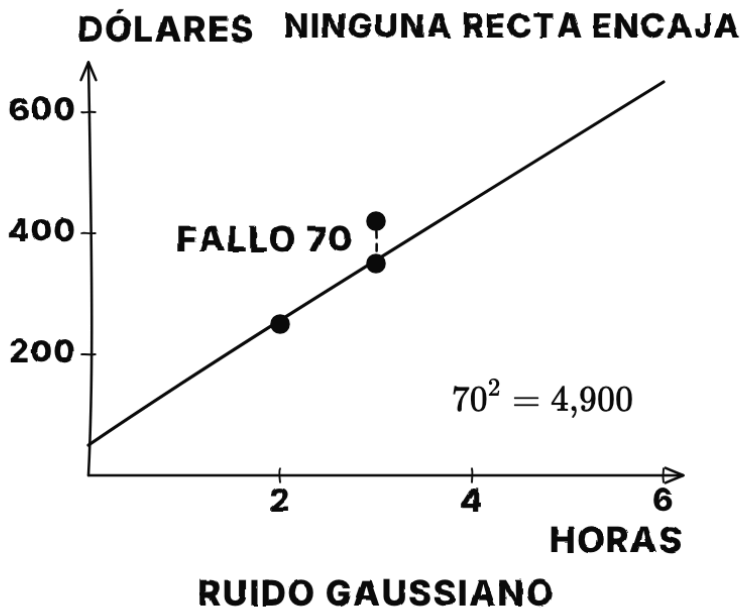
DÍAS DE HOSPITAL

NOTA DE PELÍCULA

La etiqueta es un número

El fontanero era un caso fácil, porque dos facturas fijaban exactamente las dos perillas. Los datos reales rara vez son tan amables. Una tercera factura podría haber sido de 420 dólares por 3 horas, porque las tuberías eran más viejas o el trabajo cayó en domingo, cosas que las horas por sí solas no recogen. Entonces ninguna recta pasa por todas las facturas, y nos conformamos con la recta que falla en todas lo menos posible. El libro dice que en la mayoría de sus capítulos ese «lo menos posible» se medirá con el error cuadrático, el fallo elevado al cuadrado y sumado en todos los ejemplos. Si la recta predice 350 para ese trabajo de 420 dólares, el

fallo es de 70 y suma $70^2 = 4900$ a la pérdida. Los autores hacen una promesa para más adelante: elegir el error cuadrático equivale a suponer que los datos estaban alterados por un ruido aleatorio de un tipo concreto en forma de campana, llamado ruido gaussiano.



Una factura fuera de la recta

Pruébalo

Busque dos recibos del mismo taxi o aplicación de viajes con distancias distintas. Reste los precios y divida entre la diferencia de kilómetros para obtener el precio por kilómetro, y luego reste para hallar la bajada de bandera fija. Use su fórmula para predecir un tercer viaje y compárelo con el recibo real. Si no cuadra, una tarifa dinámica o un atasco es el factor ajeno a sus características, y el tamaño del fallo es lo que mediría el error cuadrático.

Para recordar

La regresión consiste en predecir un número: cualquier pregunta de «cuánto» o «cuántos». Cada ejemplo es un vector de características, como la superficie de una casa, sus habitaciones y la distancia al centro, y su etiqueta es un número, como el precio de venta. Con dos facturas del fontanero encontramos el modelo $\text{precio} = 50 + 100 \times \text{horas}$: un cargo de 50 dólares y 100 dólares por hora, que es una regresión lineal. Cuando ninguna recta se ajusta a todos los ejemplos, elegimos la que tiene el menor error cuadrático.

Preguntas

1. ¿Qué decide si un problema es una regresión?
 - a. Que las características sean números

- b. Que la etiqueta pueda ser cualquier número
 - c. Que los datos vengan en una tabla
 - d. Que el modelo sea una línea recta
2. Según la regla práctica del libro, ¿cuál de estas preguntas es una regresión?
- a. ¿Es venenosa esta seta?
 - b. ¿Qué dígito está escrito aquí?
 - c. ¿Es spam este mensaje?
 - d. ¿Cuántos días estará este paciente en el hospital?
3. Una tercera factura del fontanero no está en la recta que pasa por las dos primeras. ¿Qué hacemos?
- a. Elegir la recta que falla en todas las facturas lo menos posible, según el error cuadrático
 - b. Tirar la tercera factura
 - c. Usar solo la factura más reciente
 - d. Trazar una recta distinta para cada factura

Respuestas

- 1. b)** Lo que convierte un problema en regresión no son las características sino la etiqueta: cuando la etiqueta puede ser cualquier número, el problema es una regresión.
- 2. d)** Cualquier pregunta que empiece por «cuánto» o «cuántos» probablemente sea una regresión.
- 3. a)** Cuando ninguna recta pasa por todas las facturas, nos conformamos con la que falla en todas lo menos posible, medido con el error cuadrático.

Ejemplos resueltos

Los precios del fontanero Un fontanero cobra 350 dólares por 3 horas y 250 dólares por 2 horas. Suponiendo que $\text{precio} = b + w \times \text{horas}$, halle el cargo por desplazamiento b y la tarifa por hora w , y prediga la factura de 5 horas.

La hora extra de 2 a 3 añade $350 - 250 = 100$ dólares, así que $w = 100$.

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

Comprobación con la primera factura: $50 + 3 \times 100 = 350$, que coincide.

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

Respuesta: $b = 50$ dólares, $w = 100$ dólares por hora; 5 horas cuestan 550 dólares

Ejercicios

1. Un trayecto en taxi de 4 km cuesta 9 euros y uno de 10 km cuesta 18 euros. Suponiendo una bajada de bandera más un precio por kilómetro, halle ambos y prediga un trayecto de 6 km.

Pista: ¿Cuánto añaden seis kilómetros más?

Respuesta: 6 km más añaden $18 - 9 = 9$ euros, es decir, 1,5 euros por km; la bajada de bandera es $9 - 4 \times 1,5 = 3$ euros; un trayecto de 6 km cuesta $3 + 6 \times 1,5 = 12$ euros.

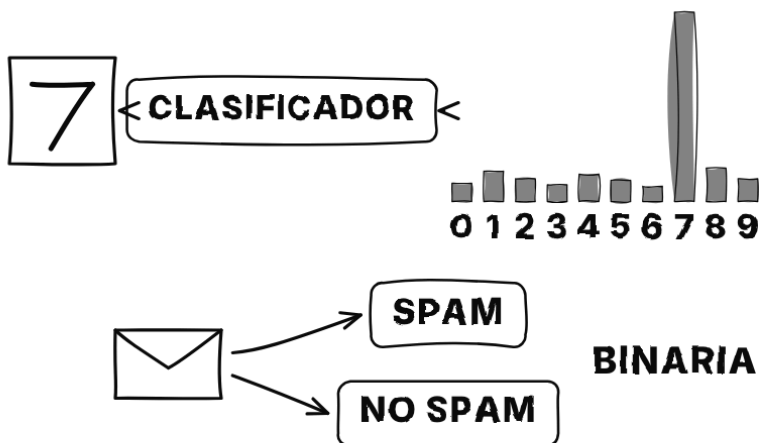
2. El modelo del fontanero predice 350 dólares para un trabajo de 3 horas que en realidad se facturó en 420 dólares. ¿Cuánto añade este trabajo al error cuadrático, y qué sugiere el fallo?

Pista: Reste y luego eleve al cuadrado.

Respuesta: El fallo es $420 - 350 = 70$, así que añade $70^2 = 4900$ a la pérdida. Sugiere un factor ajeno a las horas, como tuberías viejas o un trabajo en domingo, que las características del modelo no recogen.

¿Cuál de ellos? Clasificación

Acerque un cheque a la cámara de la aplicación de su banco y, un momento después, el importe y el nombre se rellenan solos. En algún punto intermedio, la aplicación cortó la letra manuscrita en pequeños trozos, un carácter cada uno, y para cada trozo respondió a una pregunta: ¿qué carácter es este? Esa pregunta no es «cuánto». No tiene sentido decir que un 7 es «un poco más» que un 1 mal escrito; el trozo es uno de un conjunto fijo de caracteres, y la aplicación tiene que elegir el correcto.

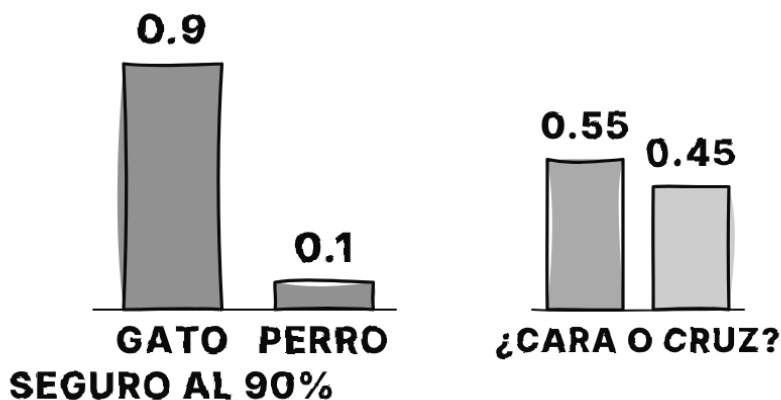


¿Cuál de ellos? Diez clases

El libro llama a estos problemas de «¿cuál de ellos?» clasificación, y los sitúa junto a los problemas de «¿cuánto?» de la regresión. Muchas de las herramientas se trasladan, pero la tarea es distinta. El modelo sigue mirando características, por ejemplo los valores de brillo de los píxeles del trozo. Pero en lugar de un número, debe nombrar una categoría, también llamada clase, de una lista de opciones fijada de antemano. Para los dígitos manuscritos hay diez clases, los dígitos del 0 al 9. Un modelo que da un número se llamaba regresor; uno que da una clase se llama clasificador.

El caso más sencillo solo tiene dos clases, y se llama clasificación binaria. El ejemplo del libro son imágenes de animales etiquetadas gato o perro. El ejemplo cotidiano es su correo: cada mensaje que llega se coloca en una de dos clases, spam o no spam. Cuando hay más de dos clases, el problema es de clasificación multiclase, como leer letra manuscrita, donde las clases van del 0 al 9 y siguen por la a, la b, la c y el resto del alfabeto.

Ahora, un giro que importa más adelante. Resulta que es difícil entrenar un modelo que solo puede decir un «gato» firme o un «perro» firme; el libro promete explicar por qué cuando se vuelva más técnico. Es mucho más fácil si el modelo habla en probabilidades. Dada una imagen, el modelo da a cada clase una probabilidad, números entre 0 y 1 que suman 1. Podría decir $P(\text{gato}) = 0,9$, y por tanto $P(\text{perro}) = 0,1$. Lo leemos así: el clasificador está seguro al 90 por ciento de que la imagen muestra un gato. Lo grande que es la probabilidad ganadora nos dice lo seguro que está el modelo; 0,9 es confiado, 0,55 es apenas algo más que tirar una moneda. Para entrenar un modelo así, dice el libro, la pérdida habitual no es el error cuadrático sino una llamada entropía cruzada, cuyo nombre tendrá sentido cuando el libro llegue a la teoría de la información.



PÉRDIDA: ENTROPÍA CRUZADA

Probabilidades, no veredictos

Después el libro señala algo fácil de pasar por alto: la clase más probable no siempre es aquella según la cual hay que actuar. Su figura muestra una seta preciosa que crece en un jardín. Suponga que ha entrenado un clasificador con fotos para distinguir setas venenosas de setas seguras, y para esta dice que la probabilidad de una oronja verde es 0,2. Así que está seguro al 80 por ciento de que la seta es segura. ¿Se la comería? La respuesta de los autores: habría que estar loco.



ORONJA 0,2
SEGURA 0,8

COMER

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

TIRAR ✓

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

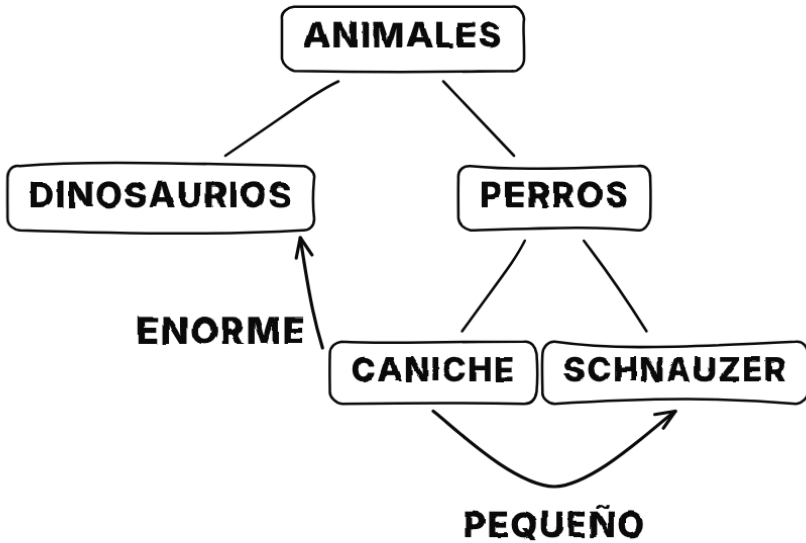
Coste esperado de una seta

Para ver por qué, pese cada acción por lo que podría costar, multiplicado por lo probable que es. Dé un coste a cada resultado. Comerse una oronja verde le mata, un coste tan grande que el libro lo escribe como infinito, ∞ . Comerse una seta segura no cuesta nada, 0; tirar una segura cuesta una cena perdida, digamos 1; tirar una oronja verde cuesta 0. El coste esperado de comerla es $0,2 \times \infty + 0,8 \times 0 = \infty$. El coste esperado de tirarla es $0,2 \times 0 + 0,8 \times 1 = 0,8$. Menos de una cena perdida frente a un desastre seguro: la tira. La decisión depende tanto de las probabilidades como de lo bueno o malo que sea cada resultado. Y el libro añade que cualquier experto en setas lo confirmaría: la de la figura es de verdad una oronja verde.

Pruébelo

Tome una aplicación del tiempo que dice 30 por ciento de probabilidad de lluvia y decida si llevar paraguas. Dé un coste a cada resultado: empaparse con un buen traje podría costar 10, llevar un paraguas que no necesitaba cuesta 1. No llevarlo cuesta de media $0,3 \times 10 = 3$; llevarlo cuesta como mucho 1. Así que se lleva el paraguas, aunque «no llueve» sea la clase más probable, exactamente como con la seta.

La clasificación puede complicarse aún más. A veces las clases forman un árbol genealógico, y entonces no todos los errores son igual de malos. Si el modelo tiene que equivocarse, preferiríamos que eligiera un pariente cercano y no algo lejano. Esto se llama clasificación jerárquica, y el libro menciona a Linneo, que ordenó los animales en un árbol así de especies y familias. Confundir un caniche con un schnauzer, dos razas de perro, es un pequeño desliz; confundir un caniche con un dinosaurio debería costarle caro al modelo.



Errores cercanos y lejanos

Pero qué árbol cuenta depende de para qué sirve el modelo. Las serpientes de cascabel y las culebras de jarretera están muy cerca en el árbol de la vida. Sin embargo, un modelo que llame a una serpiente de cascabel inofensiva culebra de jarretera podría costarle la vida a alguien, así que para una aplicación de senderismo esas dos deben estar muy separadas. Una aplicación del teléfono para identificar plantas se enfrenta a la misma elección: confundir dos tipos de margarita apenas importa, pero confundir el ajo de oso, comestible, con el lirio de los valles, venenoso, sí importa.

Para recordar

La clasificación responde a «¿cuál de ellos?»: coloca cada ejemplo en una de un conjunto fijo de clases, mientras que la regresión responde a «¿cuánto?». Con dos clases es clasificación binaria, como spam o no spam; con más es multiclase, como leer dígitos y letras. Los clasificadores suelen dar una probabilidad para cada clase, y el tamaño de la probabilidad más alta indica lo seguros que están; su pérdida habitual es la entropía cruzada. La clase más probable no siempre es la que hay que seguir: pese cada acción por sus probabilidades y sus costes, como con la seta que es segura al 80 por ciento y aun así no merece la pena comer. Cuando las clases forman una jerarquía, algunos errores son peores que otros, y cuáles depende de cómo se vaya a usar el modelo.

Preguntas

1. ¿Por qué leer un dígito manuscrito es un problema de clasificación y no de regresión?
 - a. Porque la respuesta debe ser uno de un conjunto fijo de caracteres, no una cantidad
 - b. Porque la imagen tiene demasiados píxeles
 - c. Porque los dígitos siempre se escriben mal
 - d. Porque la regresión no puede usar los valores de los píxeles como características

2. Un clasificador da $P(\text{gato}) = 0,55$ y $P(\text{perro}) = 0,45$. ¿Qué le indica esto?
- a. La imagen es sin duda un gato
 - b. La imagen muestra un gato y un perro
 - c. Se inclina por el gato, pero está apenas algo más seguro que al tirar una moneda
 - d. El clasificador está roto, porque los números deberían ser 0 o 1
3. Un clasificador de setas está seguro al 80 por ciento de que una seta es segura. ¿Por qué tirarla de todos modos?
- a. Porque el clasificador se entrenó con demasiado pocas fotos
 - b. Porque el 80 por ciento está por debajo de la tasa de error
 - c. Porque las setas seguras son raras
 - d. Porque el coste esperado de comerla, con la muerte como uno de los resultados, es mucho mayor que una cena perdida
4. ¿Por qué una aplicación de serpientes para senderistas podría tratar como muy separadas a la serpiente de cascabel y la culebra de jarretera, aunque sean parientes cercanos?
- a. Porque no se parecen en nada
 - b. Porque confundirlas podría costarle la vida a alguien, y la jerarquía útil depende del propósito
 - c. Porque la clasificación binaria no puede tratar serpientes
 - d. Porque el árbol de la vida no es una jerarquía

Respuestas

1. a) No tiene sentido decir que un 7 es un poco más que un 1 mal escrito; el trozo es uno de un conjunto fijo de caracteres, lo que es una pregunta de

cuál de ellos.

2. c) El tamaño de la probabilidad ganadora indica lo seguro que está el modelo; 0,9 es confiado, 0,55 es apenas algo más que tirar una moneda.

3. d) Comerla tiene un coste esperado de $0,2 \times \infty + 0,8 \times 0 = \infty$, mientras que tirarla cuesta $0,2 \times 0 + 0,8 \times 1 = 0,8$; la decisión depende de las probabilidades y de los costes a la vez.

4. b) Qué árbol cuenta depende de para qué sirve el modelo; llamar a una serpiente de cascabel inofensiva culebra de jarretera podría ser mortal.

Ejemplos resueltos

La decisión sobre la seta Un clasificador da una probabilidad de 0,2 de que una seta sea una oronja verde. Costes: comer una oronja ∞ , comer una segura 0, tirar una oronja 0, tirar una segura 1. Halle el coste esperado de comerla y de tirarla.

La probabilidad de que la seta sea segura es $1 - 0,2 = 0,8$.

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

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

Se elige la acción con el menor coste esperado.

Respuesta: Comer ∞ , tirar 0,8: tírela

Ejercicios

1. Una aplicación del tiempo da un 30 por ciento de probabilidad de lluvia. Empaparse cuesta 10, llevar un paraguas innecesario cuesta 1 y quedarse seco con paraguas bajo la lluvia cuesta 1. ¿Cuál es el coste esperado de cada opción?

Pista: Sin paraguas solo paga si llueve; con paraguas paga 1 pase lo que pase.

Respuesta: Sin paraguas: $0,3 \times 10 + 0,7 \times 0 = 3$. Con paraguas: $0,3 \times 1 + 0,7 \times 1 = 1$. Llévase el paraguas, aunque sea más probable que no llueva.

2. Un filtro de spam da $P(\text{spam}) = 0,7$ para un mensaje. ¿Cuánto vale $P(\text{no spam})$, y es clasificación binaria o multiclase?

Pista: Las probabilidades de todas las clases suman 1.

Respuesta: $P(\text{no spam}) = 1 - 0,7 = 0,3$, y con dos clases es clasificación binaria.

Muchas etiquetas a la vez, y lo mejor primero

Muestre a un clasificador de gato o perro una imagen del viejo cuento de los Músicos de Bremen. En la figura del libro hay un burro, con un perro sobre el lomo, un gato sobre el perro y un gallo encima, todos delante de unos árboles. El clasificador solo puede responder «gato» o «perro», así que diga lo que diga se equivoca; la respuesta correcta es «gato, y perro, y burro, y gallo». Hacer el modelo más preciso no ayuda, señala el libro: el problema estaba mal planteado.



MULTIETIQUETA

- GATO** ✓
- PERRO** ✓
- BURRO** ✓
- GALLO** ✓

NUBE - **AWS**

AA - **GPU**

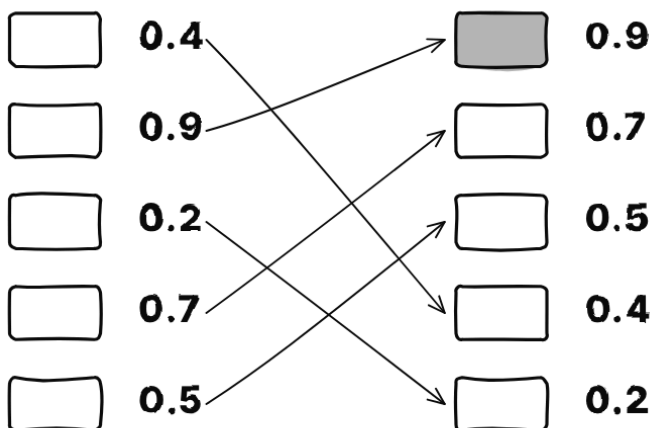
Cuando un ejemplo puede pertenecer a varias clases a la vez, la tarea se llama clasificación multietiqueta. Las clases ya no son excluyentes; el modelo responde sí o no para cada etiqueta por separado. Es la forma natural del etiquetado. Una entrada de un blog de tecnología puede llevar de cinco a diez etiquetas, como «aprendizaje automático», «lenguajes de programación», «Linux» o «computación en la nube». Además, las etiquetas van juntas: una entrada etiquetada «computación en la nube» suele mencionar AWS, el servicio en la nube de Amazon, y una etiquetada «aprendizaje automático» suele mencionar GPU. Un buen etiquetador puede aprovechar esos patrones. Cuando una aplicación de fotos etiqueta una imagen a la vez como «playa», «puesta de sol» y «perro», está haciendo clasificación multietiqueta.

A veces la lista de etiquetas posibles es enorme. La Biblioteca Nacional de Medicina de Estados Unidos hace que expertos den a cada artículo de su base de datos PubMed un conjunto de etiquetas de una lista llamada MeSH, Medical Subject Headings, con unas 28 000 entradas. Unas buenas etiquetas permiten a los investigadores encontrar todo lo escrito sobre una cuestión, pero etiquetar a mano es lento, y los artículos suelen esperar alrededor de un año. El aprendizaje automático puede proponer etiquetas provisionales hasta que una persona las revise, y una competición llamada BioASQ lleva años celebrándose justo sobre esta tarea.

El siguiente tipo de problema trata del orden, no de las clases. Cuando busca en la web, puede haber miles de páginas relevantes. La verdadera pregunta es cuáles poner arriba para usted. Una forma natural es dar a cada candidato una puntuación y mostrar primero las puntuaciones más altas.

PAGERANK

PRIMERO



Puntuar y luego ordenar

El libro recuerda PageRank, el ingrediente secreto original del buscador de Google, como uno de los primeros sistemas de puntuación. Curiosamente, su puntuación no dependía de lo que usted escribía. Un filtro sencillo elegía primero las páginas que coincidían con la consulta, y PageRank ponía después las páginas con más autoridad, más o menos las que enlazan muchas otras, por delante del resto. Hoy, dice el libro, los buscadores aprenden sus puntuaciones con aprendizaje automático y modelos de cómo se comporta la gente, así que la puntuación depende de la consulta, y hay congresos académicos enteros dedicados al tema.

Pruébalo

Busque en su correo una palabra como «factura» y fíjese en el orden de los resultados. Después busque «factura marzo» y vea cómo cambia el orden. El conjunto de mensajes que coinciden se reduce, pero también cambia la ordenación, porque la puntuación depende ahora de más partes de su consulta. Esa es la diferencia entre filtrar lo relevante y ordenar lo que más importa.

Para recordar

Cuando un ejemplo puede pertenecer a varias clases a la vez, como una foto de cuatro animales, la tarea es clasificación multietiqueta, y el etiquetado es su forma habitual. Las etiquetas tienden a aparecer juntas, y la lista de etiquetas puede ser enorme, como los 28 000 descriptores médicos con los que se indexa PubMed. La búsqueda consiste en ordenar: dar a cada candidato relevante una puntuación y mostrar primero el más alto. PageRank puntuaba las páginas sin mirar la consulta; la búsqueda moderna aprende puntuaciones que dependen de la consulta.

Preguntas

1. A un clasificador de gato o perro se le muestran los Músicos de Bremen. ¿Por qué no ayuda hacerlo más preciso?

- a. La imagen tiene demasiados árboles
 - b. El problema está mal planteado: la imagen contiene varias clases a la vez
 - c. El clasificador nunca se entrenó con cuentos
 - d. La precisión solo importa en la regresión
2. En la clasificación multietiqueta, ¿qué hace el modelo con cada etiqueta posible?
- a. Responde sí o no para esa etiqueta por separado
 - b. Elige la única etiqueta más probable
 - c. Ordena las etiquetas y se queda solo con la primera
 - d. Predice un número con cuántas etiquetas hay
3. Según la lección, ¿cuál es la pregunta principal de la búsqueda web?
- a. Si alguna página coincide con la consulta
 - b. Cuántas páginas contiene la web
 - c. Cuáles de las muchas páginas relevantes poner arriba
 - d. Qué etiquetas poner a cada página
4. ¿En qué se diferenciaba PageRank de las puntuaciones que aprenden hoy los buscadores?
- a. Usaba aprendizaje automático, y las puntuaciones actuales no
 - b. Ordenaba las páginas por su longitud
 - c. Solo funcionaba con artículos médicos
 - d. Su puntuación no dependía de la consulta, mientras que las actuales sí

Respuestas

1. **b)** La respuesta correcta es gato, perro, burro y gallo; un modelo que debe elegir una sola clase se equivoca diga lo que diga, así que la tarea debería ser clasificación multietiqueta.
2. **a)** Las clases ya no son excluyentes; el modelo responde sí o no para cada etiqueta por separado.
3. **c)** Puede haber miles de páginas relevantes; la verdadera pregunta es cuáles poner arriba, y eso se hace puntuando a los candidatos y mostrando primero los de puntuación más alta.
4. **d)** La puntuación de PageRank no dependía de lo que usted escribía; un filtro elegía las páginas que coincidían y PageRank ponía primero las de más autoridad. Las puntuaciones aprendidas de hoy dependen de la consulta.

Ejercicios

1. Para cada tarea, diga si es clasificación multiclase o multietiqueta, y por qué: leer qué dígito está escrito en una casilla; etiquetar una noticia con temas; enumerar todos los animales de una foto de una granja.

Pista: Pregúntese si un ejemplo puede tener más de una respuesta correcta.

Respuesta: El dígito es multiclase: cada casilla contiene exactamente un dígito. Etiquetar una noticia es multietiqueta: puede tratar de varios temas. La foto de la granja es multietiqueta: puede haber varios animales a la vez.

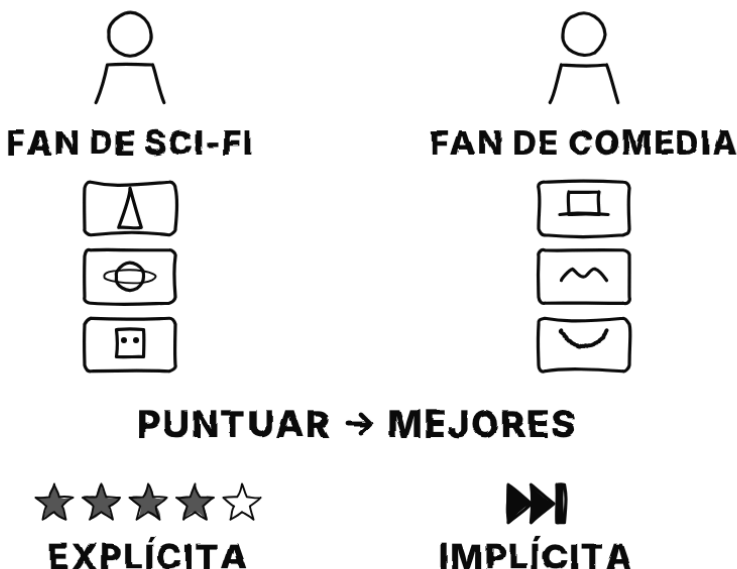
2. Explique con sus propias palabras la diferencia entre filtrar y ordenar en una búsqueda, con un ejemplo propio.

Pista: Lo uno decide qué entra en la lista; lo otro, el orden.

Respuesta: Filtrar elige qué elementos son relevantes, por ejemplo todos los correos que contienen la palabra factura. Ordenar da a cada elemento relevante una puntuación y pone primero los mejores, por ejemplo la factura más reciente o la que mejor coincide arriba del todo.

El muro que le aprende

Abra una aplicación de vídeos en su teléfono y en el de un amigo, uno al lado del otro. Los buscadores son iguales, pero las páginas de inicio no se parecen en nada. La suya puede estar llena de cocina y fútbol, y la de su amigo de gatos y películas antiguas, aunque nadie haya escrito nada.



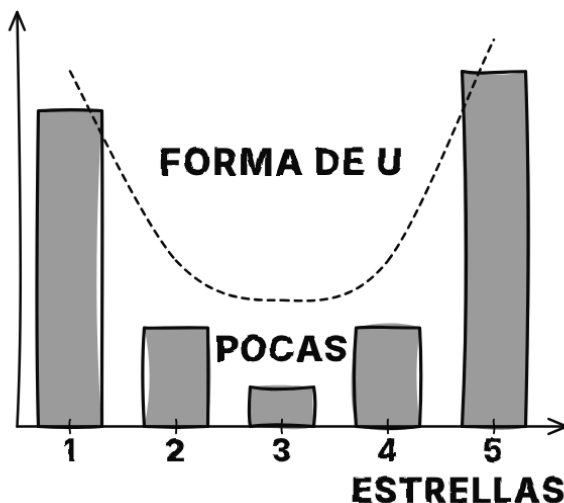
La misma aplicación, muros distintos

Son sistemas de recomendación, el siguiente tipo de problema del libro. Son parientes cercanos de la búsqueda: en ambos, el objetivo es mostrar una lista de elementos que importan al usuario. La diferencia

es la personalización. En el caso de las películas, dicen los autores, la página de un aficionado a la ciencia ficción y la de un amante de las comedias de Peter Sellers deberían ser muy distintas, y lo mismo vale para tiendas, música y noticias.

El sistema aprende de lo que le dice la gente, de dos maneras. La opinión explícita se dice en voz alta: valoraciones con estrellas y reseñas en Amazon, IMDb o Goodreads. La opinión implícita se lee en el comportamiento, como saltarse una canción de una lista de reproducción. Es más difícil de interpretar: saltársela puede significar que no le gusta la canción, o solo que no encajaba en ese momento. En la versión más sencilla, el sistema aprende a predecir una puntuación para cada par de persona y elemento, por ejemplo las estrellas que usted le daría o la probabilidad de que lo compre. Después, para usted, elige los elementos con las puntuaciones predichas más altas y le muestra esos. Los sistemas reales usan mucho más, como registros detallados de lo que usted hace y de cómo es cada elemento. La figura del libro muestra los libros de aprendizaje profundo que Amazon recomendó a uno de los autores, Aston.

Los sistemas de recomendación valen muchísimo dinero, pero el libro advierte que construirlos ingenuamente sobre un modelo predictivo tiene defectos profundos. El primero es que la opinión está censurada: solo vemos parte de las opiniones. La gente valora sobre todo lo que le despierta sentimientos fuertes. En una escala de una a cinco estrellas hay muchos unos y muchos cincos, y extrañamente pocos treses, no porque haya pocas películas normales, sino porque la gente rara vez se molesta en valorar una película normal.



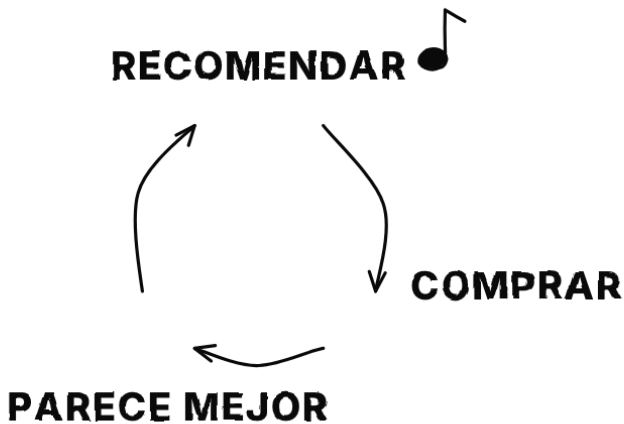
Pocos valoran el término medio

Pruébalo

Abra las reseñas de cualquier producto popular de una tienda en línea y mire el gráfico de barras de las valoraciones. Cuente la proporción de reseñas de una y de cinco estrellas, y la de tres estrellas. Normalmente verá que el gráfico se hunde en el centro, con forma de U. Ese hundimiento es la opinión censurada de la que habla el libro: el centro silencioso nunca habla.

El segundo defecto es un bucle. Lo que la gente compra hoy es en parte lo que le mostró el recomendador actual. Supongamos que el sistema promociona un artículo un poco más que los demás. Lo ve

más gente, así que lo compra más gente, así que la siguiente ronda de entrenamiento lo considera mejor, así que se recomienda aún más. El artículo sube porque se promocionó, no porque fuera el mejor. Una canción que una aplicación de música no deja de meter en su mezcla se vuelve «popular» en parte porque la aplicación la puso ahí. El libro dice que cómo tratar la censura, los incentivos y estos bucles de retroalimentación sigue siendo una cuestión de investigación abierta.



Un bucle de retroalimentación

Para recordar

Los sistemas de recomendación son como la búsqueda, pero personalizados: personas distintas ven listas distintas. Aprenden de la opinión explícita, como las valoraciones, y de la implícita, como los saltos, y predicen una puntuación para cada persona y elemento. La opinión está censurada, porque la gente valora sobre todo lo que ama u odia, y por eso escasean las valoraciones de tres estrellas. Las recomendaciones moldean lo que compra la gente, así que un sistema puede crear un bucle de retroalimentación que siga promocionando los mismos artículos.

Preguntas

1. ¿Qué separa principalmente a un sistema de recomendación de un buscador?
 - a. Un recomendador nunca usa puntuaciones
 - b. Un recomendador personaliza la lista para cada usuario
 - c. Un recomendador solo muestra elementos que nadie ha valorado
 - d. Un recomendador solo funciona con películas
2. ¿Por qué es más difícil aprender de saltarse una canción que de una valoración de una estrella?
 - a. Los saltos nunca se registran
 - b. Los saltos solo ocurren en aplicaciones de música

- c. Un salto puede significar desagrado, o solo que la canción no encajaba en ese momento
 - d. Un salto siempre significa que a la persona le encantó la canción
3. ¿Por qué escasean extrañamente las valoraciones de tres estrellas?
- a. La gente rara vez se molesta en valorar cosas que solo le parecen regulares
 - b. Hay pocos productos normales
 - c. Los recomendadores borran las valoraciones de tres estrellas
 - d. La escala de valoración no tiene valor central
4. El recomendador de una tienda promociona un artículo un poco más que los demás. ¿Qué puede pasar tras varias rondas de entrenamiento?
- a. El artículo desaparece de la tienda
 - b. Las valoraciones del artículo quedan censuradas
 - c. Nada, ya que el entrenamiento solo usa valoraciones
 - d. El artículo se compra más, se considera mejor y se promociona aún más

Respuestas

- 1. b)** Ambos muestran una lista de elementos que importan; la diferencia es la personalización, de modo que un aficionado a la ciencia ficción y uno de Peter Sellers ven páginas distintas.
- 2. c)** La opinión implícita se lee en el comportamiento, y saltarse una canción puede significar que no le gusta o solo que no encajaba en ese momento.

- 3. a)** La opinión está censurada: la gente valora sobre todo lo que le despierta sentimientos fuertes, así que el centro silencioso rara vez habla.
- 4. d)** Lo ve más gente, así que más lo compran, así que la siguiente ronda lo considera mejor y lo recomienda aún más: un bucle de retroalimentación.

Ejercicios

1. Clasifique como opinión explícita o implícita: una reseña de cinco estrellas de un libro; parar una película a los diez minutos; añadir una canción a favoritos pulsando un corazón; comprar un producto.

Pista: La opinión explícita expresa una opinión; la implícita hay que deducirla del comportamiento.

Respuesta: Explícita: la reseña de cinco estrellas y pulsar el corazón, porque ambas expresan directamente una opinión. Implícita: parar la película y comprar el producto, porque la opinión hay que deducirla de lo que se hizo.

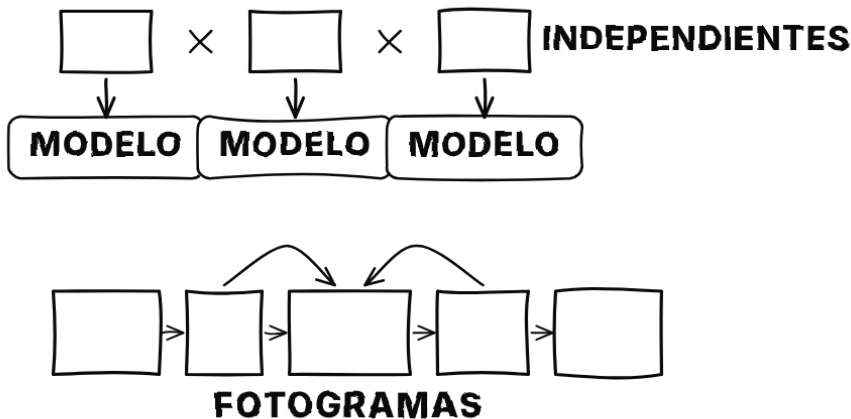
2. Explique con sus propias palabras por qué un recomendador entrenado con lo que la gente compró el mes pasado puede acabar aprendiendo simplemente sus propias elecciones anteriores.

Pista: Pregúntese de dónde salieron las compras del mes pasado.

Respuesta: Muchas compras del mes pasado ocurrieron porque el recomendador mostró esos artículos. Entrenar con ellas hace que esos artículos parezcan mejores de lo que son, así que el sistema los recomienda aún más: aprende de datos que él mismo moldeó, un bucle de retroalimentación.

Cuando el orden importa: secuencias

Detenga un partido de fútbol en un solo fotograma: un jugador en el aire, el balón junto a su pie. ¿Está tirando a puerta, pasando o cayéndose tras una falta? Con un fotograma no se sabe; con el segundo anterior y el posterior, lo sabe enseguida. El significado de cada momento depende de los momentos que lo rodean.



ENTRA SECUENCIA, SALE SECUENCIA

Entradas independientes frente a secuencias

Todo lo visto en el curso hasta ahora lo ignoraba. Una casa venía con un conjunto fijo de características, cuatro números, y recibía un precio. Una imagen de tamaño fijo recibía una probabilidad por cada

clase de una lista fija. Una valoración con estrellas se predecía solo a partir de un número de usuario y un número de producto. Y una vez que el modelo había respondido para un ejemplo, lo olvidaba por completo antes de que llegara el siguiente. El libro dice que esto se basa en una suposición: un ejemplo no nos dice nada del siguiente, así que no hay motivo para recordar.

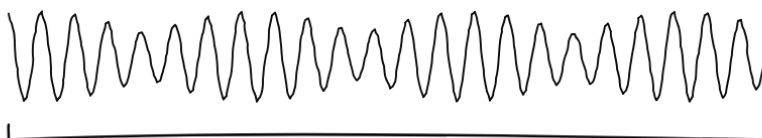
El vídeo rompe esa suposición dos veces. Cada clip puede tener un número distinto de fotogramas, así que no hay longitud fija. Y nuestra conjetura sobre un fotograma es mucho mejor si miramos los fotogramas anteriores y posteriores. Con el lenguaje pasa lo mismo: una frase puede tener cualquier longitud, y una palabra se entiende por las que la rodean. Y también con la medicina. Imagine un modelo en una unidad de cuidados intensivos que debe dar la alarma cada vez que el riesgo de que un paciente muera en las próximas 24 horas suba por encima de cierto nivel. Sería absurdo que olvidara cada hora el historial del paciente y juzgara solo por la última tensión y el último pulso.

Problemas como estos se llaman aprendizaje de secuencias, y el libro los cuenta entre los usos más apasionantes del aprendizaje automático. El modelo tiene que recibir una secuencia de entradas, o producir una secuencia de salidas, o ambas cosas. Cuando tanto la entrada como la salida son secuencias de longitud variable, se llama aprendizaje secuencia a secuencia; traducir frases y convertir voz en texto son los ejemplos típicos. El libro destaca algunos casos especiales.

ALINEADAS

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



48 000 MUESTRAS



SEIS PALABRAS

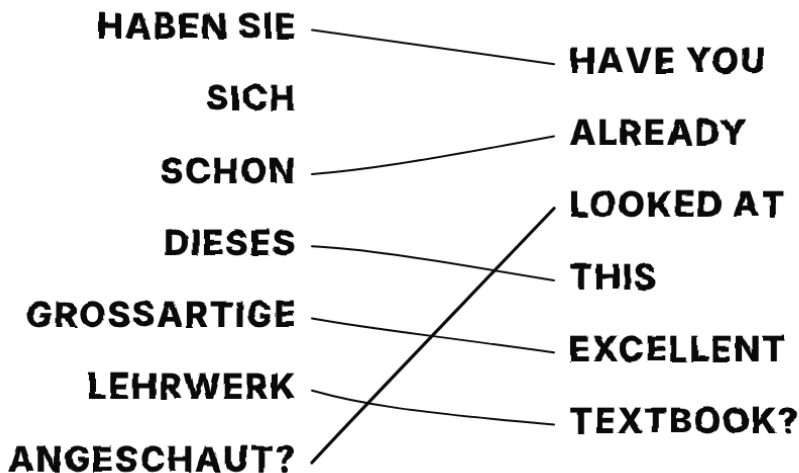
Etiquetas alineadas, audio no alineado

El primero es el etiquetado y el análisis sintáctico, en el que cada palabra de un texto recibe una marca. Aquí la entrada y la salida están alineadas: hay una marca por palabra, en el mismo orden que las palabras. En el etiquetado gramatical, cada palabra recibe su función gramatical, como sustantivo o verbo. En una versión más sencilla, cada palabra se marca según forme parte o no del nombre de una persona, un lugar o una organización. El ejemplo del libro es la frase «Tom has dinner in Washington with Sally», en la que Tom, Washington y Sally se marcan como nombres y las otras cuatro palabras no. Entran siete palabras y salen siete marcas.

El segundo es el reconocimiento automático del habla: entra una grabación y salen las palabras escritas. Aquí las longitudes son

tremendamente distintas. El sonido suele grabarse a 8 kHz o 16 kHz, es decir, ocho o dieciséis mil muestras por segundo. Una frase de seis palabras dicha en tres segundos a 16 kHz son $3 \times 16\,000 = 48\,000$ muestras. Así que 48 000 números deben convertirse en seis palabras, unas 8000 muestras por palabra, y no hay una correspondencia limpia uno a uno entre trozos de sonido y letras. La figura del libro muestra las palabras «Deep Learning» estiradas a lo largo de una onda sonora, con cada sonido extendido sobre un largo tramo de ella. Las personas entienden el habla con facilidad, incluso por una mala línea telefónica; para los ordenadores, dice el libro, es un reto formidable. El asistente de voz que le entiende pese al ruido de la calle está resolviendo exactamente este problema.

El tercer caso es el mismo problema al revés: texto a voz. Entra un texto corto y sale un archivo de sonido largo. Cuando una aplicación de navegación dice «dentro de doscientos metros, gire a la izquierda», unas pocas palabras se han convertido en decenas de miles de muestras.



La traducción reordena las palabras

El cuarto es la traducción automática, y trae una dificultad nueva. En el habla, los sonidos y las palabras al menos llegan en el mismo orden. En la traducción, las dos frases pueden tener longitudes distintas, y las partes correspondientes pueden estar en otro orden. El libro lo muestra con el alemán, al que le gusta poner el verbo al final. «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» significa «¿Ya ha mirado este excelente libro de texto?». Tradúzcalo palabra por palabra, manteniendo el orden alemán, y obtendrá «Have you yourself already this excellent textbook looked at?», que es la alineación equivocada. El verbo «looked at» tiene que viajar desde el final de la frase alemana hasta el medio de la inglesa. Un chatbot que le traduce un mensaje tiene que aprender ese tipo de reordenación para cada par de idiomas.

Pruébalo

Tome una frase corta en un idioma que conozca además del español, o pídasela a un amigo. Escríbala palabra por palabra en español, manteniendo el orden original, y debajo escriba una buena traducción al español. Trace líneas entre las palabras que significan lo mismo. Donde las líneas se cruzan, el orden ha cambiado; ese cruce es lo que hace que la traducción no esté alineada.

El libro termina con un vistazo a casos más difíciles. Averiguar en qué orden lee una persona una página web es un problema en dos dimensiones, no a lo largo de una línea. Y en un diálogo, decidir qué decir a continuación requiere conocimiento del mundo y memoria de lo que se dijo mucho antes. Estos, dicen los autores, siguen siendo áreas activas de investigación.

Para recordar

Muchas tareas anteriores tenían entradas de tamaño fijo y trataban cada ejemplo como independiente, olvidándolo tras responder. El aprendizaje de secuencias trata entradas o salidas que son secuencias, como vídeo, frases o el historial de un paciente, donde el contexto y la longitud importan. En el etiquetado, entrada y salida están alineadas: una marca por palabra, en orden. El reconocimiento del habla convierte decenas de miles de muestras por segundo en unas pocas palabras; el texto a voz hace lo contrario. La traducción automática no está alineada: las longitudes difieren, y las partes correspondientes pueden ir en otro orden, como el verbo alemán al final.

Preguntas

1. ¿Qué suposición de las tareas anteriores rompe el vídeo?
 - a. Que cada ejemplo es independiente y puede olvidarse tras responder
 - b. Que las características son números
 - c. Que las etiquetas de los ejemplos de entrenamiento se conocen
 - d. Que los modelos tienen parámetros
2. ¿Por qué etiquetar las palabras de una frase se llama un problema alineado?

- a. La salida es mucho más corta que la entrada
 - b. Las palabras pueden aparecer en cualquier orden
 - c. Hay una marca por palabra, en el mismo orden que las palabras
 - d. Solo los nombres reciben marcas
3. ¿Por qué no hay correspondencia uno a uno entre sonido y letras en el reconocimiento del habla?
- a. Porque la transcripción es más larga que la grabación
 - b. Porque decenas de miles de muestras se convierten en solo unas pocas palabras
 - c. Porque el habla siempre tiene ruido
 - d. Porque cada palabra dura exactamente una muestra
4. ¿Qué hace que la traducción automática no esté alineada?
- a. La salida siempre es más corta que la entrada
 - b. Cada palabra debe etiquetarse como nombre
 - c. Las traducciones se hacen a partir de sonido
 - d. Las longitudes difieren y las partes correspondientes pueden ir en otro orden

Respuestas

- 1. a)** Las tareas anteriores suponían que un ejemplo no dice nada del siguiente, así que el modelo olvidaba cada uno; en el vídeo el número de fotogramas varía y cada uno se entiende por sus vecinos.
- 2. c)** En el etiquetado, entrada y salida están alineadas: una marca por palabra, en el mismo orden; entran siete palabras y salen siete marcas.

3. b) A 16 kHz, tres segundos de habla son 48 000 muestras que se convierten en seis palabras, unas 8000 muestras por palabra.

4. d) En la traducción, las dos frases pueden tener longitudes distintas y las partes correspondientes pueden estar en otro orden, como cuando el verbo alemán del final pasa al medio en inglés.

Ejemplos resueltos

Muestras por palabra El sonido se graba a 16 kHz, es decir, 16 000 muestras por segundo. Una frase de seis palabras se dice en tres segundos. ¿Cuántas muestras son, y cuántas por palabra aproximadamente?

16 kHz significa 16 000 muestras cada segundo.

Tres segundos dan $3 \times 16\,000 = 48\,000$ muestras.

Seis palabras se las reparten, así que a cada una le tocan unas $48\,000 \div 6 = 8000$ muestras.

Así que miles de números de entrada corresponden a una sola palabra de salida: las longitudes son tremendamente distintas.

Respuesta: 48 000 muestras, unas 8000 por palabra

Ejercicios

1. Un teléfono graba a 8 kHz. ¿Cuántas muestras contiene un mensaje de voz de 5 segundos?

Pista: 8 kHz son 8000 muestras cada segundo.

Respuesta: $5 \times 8000 = 40\,000$ muestras.

2. Marque la frase «Anna flew from Paris to Rome» con el sencillo esquema de etiquetado de nombres, y diga cuántas palabras entran y cuántas marcas salen.

Pista: Las personas y los lugares son nombres.

Respuesta: Anna, Paris y Rome se marcan como nombres; flew, from y to no. Entran seis palabras y salen seis marcas, porque el etiquetado está alineado.

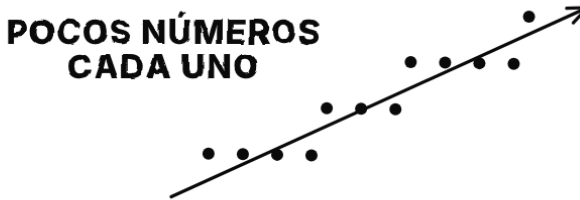
Aprender sin profesor

Mire la galería de fotos de su teléfono y busque «playa». Ahora mire el álbum «Personas», donde su teléfono ha juntado todas las fotos del mismo amigo en un montón. Nadie le dijo nunca el nombre de su amigo ni qué fotos van juntas; encontró los grupos por sí solo. Es un tipo de aprendizaje distinto de todo lo anterior.

El libro explica la diferencia con dos jefes. El aprendizaje supervisado es como trabajar para un jefe muy mandón que está detrás de usted y le dice exactamente qué hacer en cada situación, hasta que aprende a imitarlo. Aburrido, admiten los autores, pero fácil: detecte el patrón e imítelo. El jefe opuesto no tiene ni idea de lo que quiere. Le deja un enorme montón de datos en la mesa y le dice: «¡Haga ciencia de datos con esto!». Suena vago porque es vago, y más le vale a un científico de datos acostumbrarse. Los problemas sin etiquetas como este se llaman aprendizaje no supervisado, y el único límite de las preguntas que puede plantear es su imaginación.

El libro da algunas de esas preguntas. ¿Pueden unos pocos ejemplos típicos resumir todos los datos, por ejemplo clasificar un montón de fotos en paisajes, perros, bebés, gatos y cumbres, o agrupar usuarios según cómo navegan? Eso es el agrupamiento. El álbum «Personas» de su teléfono y una aplicación de música que agrupa a oyentes con gustos parecidos hacen agrupamiento. ¿Pueden unos pocos números captar lo que importa de cada ejemplo? El vuelo de una pelota se describe bien por su velocidad, su tamaño y su masa, y los sastres describen un cuerpo humano para ajustar la ropa con un puñado de medidas. Eso es la estimación de subespacios, y cuando los números

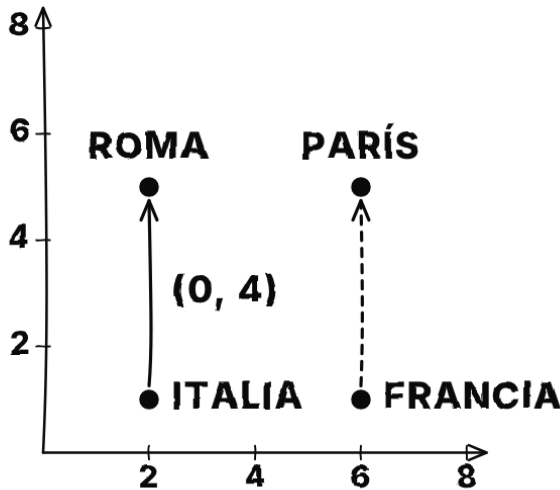
dependen de los datos de forma lineal, se llama análisis de componentes principales.



Grupos hallados sin etiquetas

Una tercera pregunta es si las palabras y las cosas pueden colocarse como puntos en el espacio de modo que sus relaciones se conviertan en aritmética. El ejemplo del libro es Roma – Italia + Francia = París. Aquí tiene una imagen inventada de cómo puede funcionar, con cada palabra como un punto con dos números. Supongamos que Roma está en (2, 5), Italia en (2, 1) y Francia en (6, 1). Roma menos Italia es $(2 - 2, 5 - 1) = (0, 4)$, un paso que significa «la capital de». Suma ese paso a Francia: $(6 + 0, 1 + 4) = (6, 5)$, y si París está en (6, 5), la suma cae justo en él. Los sistemas reales usan cientos de números por palabra, pero la idea es la misma.

$$\text{ROMA} - \text{ITALIA} + \text{FRANCIA} = \text{PARÍS}$$

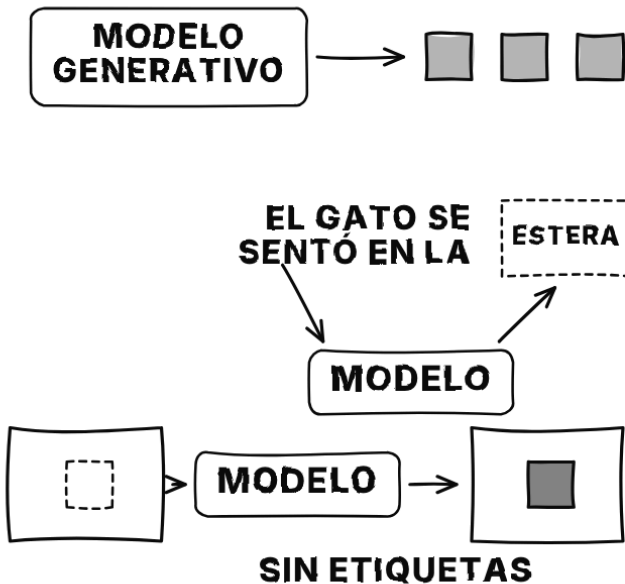


$$\text{Roma} - \text{Italia} + \text{Francia} = \text{París}$$

Una cuarta pregunta trata de las causas. Con datos sobre precios de la vivienda, contaminación, delincuencia, ubicación, educación y salarios, ¿podemos averiguar solo a partir de los números qué causa qué? El estudio de la causalidad y de los modelos gráficos probabilísticos aborda este tipo de preguntas.

El libro destaca una línea reciente y apasionante: los modelos generativos profundos. Aprenden cómo se distribuyen los datos, qué tipos de ejemplos son comunes y cuáles raros, a veces de forma explícita y a veces solo implícita. Una vez entrenado, un modelo así puede hacer dos cosas: decir lo probable que es un ejemplo dado, o producir ejemplos totalmente nuevos que parezcan salidos de los mismos datos. Los primeros avances fueron los autocodificadores

variacionales y las redes generativas antagónicas; entre los más nuevos están los flujos normalizadores y los modelos de difusión. Las aplicaciones que dibujan una imagen a partir de una frase que usted escribe se basan en modelos de difusión.



Etiquetas gratis

Por último, el libro describe el aprendizaje autosupervisado, una forma de obtener etiquetas gratis a partir de datos sin etiquetar. Con texto, se ocultan algunas palabras al azar y se entrena el modelo para rellenar los huecos a partir de las palabras que los rodean. Miles de millones de frases se convierten en ejemplos de entrenamiento sin que nadie etiquete nada, porque la palabra oculta es la etiqueta. Con imágenes, un modelo puede aprender a decir dónde estaban, una respecto a otra, dos piezas recortadas de la misma foto, a rellenar una

parte tapada a partir del resto, o a decir si dos imágenes alteradas proceden del mismo original. Las representaciones que aprende un modelo así por el camino se reutilizan después: el modelo se afina, se ajusta un poco, para la tarea que de verdad nos interesa. El modelo de lenguaje que hay dentro de un chatbot se entrenó primero justo así, adivinando palabras que faltan y palabras siguientes en cantidades enormes de texto.

Pruébalo

Tome un párrafo de un periódico y tape cada séptima palabra con una tira de papel. Léalo e intente adivinar cada palabra oculta, y luego levante las tiras. Acertará la mayoría, porque las palabras de alrededor de cada hueco dicen mucho de él. Haciendo exactamente esto, millones de veces, un modelo autosupervisado aprende un idioma sin una sola etiqueta.

Para recordar

El aprendizaje supervisado tiene un jefe que da todas las respuestas; el no supervisado solo recibe un montón de datos y debe encontrar la estructura por sí mismo. El agrupamiento encuentra grupos, la estimación de subespacios encuentra unos pocos números que describen cada ejemplo, y las representaciones de palabras pueden convertir relaciones en aritmética, como en Roma – Italia + Francia = París. Los modelos generativos profundos aprenden cómo se distribuyen los datos y pueden puntuar ejemplos o crear otros nuevos. El aprendizaje autosupervisado fabrica sus propias etiquetas a partir de los datos, como palabras ocultas, y las representaciones que aprende se afinan después para otras tareas.

Preguntas

1. En la imagen de los dos jefes de la lección, ¿qué hace que un problema sea no supervisado?
 - a. El jefe comprueba cada respuesta
 - b. Solo recibe un montón de datos, sin etiquetas que digan qué hacer
 - c. Imita lo que hace el jefe en cada situación
 - d. Los datos vienen con un número fijo de características
2. Una aplicación de música agrupa a oyentes con gustos parecidos sin que nadie le indique ningún grupo. ¿Qué tipo de

- problema es?
- a. Estimación de subespacios
 - b. Aprendizaje autosupervisado
 - c. Agrupamiento
 - d. Un modelo generativo
3. ¿Qué puede hacer un modelo generativo profundo entrenado?
- a. Decir lo probable que es un ejemplo, o producir ejemplos nuevos parecidos a los datos
 - b. Solo clasificar ejemplos en clases etiquetadas
 - c. Solo encontrar las causas de los datos
 - d. Solo rellenar palabras ocultas
4. En el aprendizaje autosupervisado con texto, ¿de dónde sale la etiqueta?
- a. De personas que leen cada frase
 - b. De un jefe que da todas las respuestas
 - c. De un conjunto de datos etiquetado aparte
 - d. De la propia palabra oculta, sacada del texto

Respuestas

- 1. b)** El jefe opuesto le deja un enorme montón de datos en la mesa y le dice que haga ciencia de datos con ellos; los problemas sin etiquetas como este son aprendizaje no supervisado.
- 2. c)** Encontrar unos pocos grupos típicos que resuman los datos, como oyentes con gustos parecidos, es agrupamiento.
- 3. a)** Una vez entrenado, un modelo así puede decir lo probable que es un ejemplo dado, o producir ejemplos totalmente nuevos que parezcan salidos

de los mismos datos.

4. d) Se ocultan algunas palabras y se entrena al modelo para rellenar los huecos; la palabra oculta es la etiqueta, así que nadie tiene que etiquetar nada.

Ejemplos resueltos

Aritmética de palabras en un mapa de juguete En un mapa inventado de dos números, Roma está en (2, 5), Italia en (2, 1), Francia en (6, 1) y París en (6, 5). Calcule $\text{Roma} - \text{Italia} + \text{Francia}$ y compruebe si cae en París.

Reste número a número: $\text{Roma} - \text{Italia} = (2 - 2, 5 - 1) = (0, 4)$, el paso que significa «capital de».

Suma ese paso a Francia: $(6 + 0, 1 + 4) = (6, 5)$.

París está en (6, 5), así que la suma cae justo en París.

Respuesta: (6, 5), que es París

Ejercicios

1. En el mismo tipo de mapa de juguete, Alemania está en (4, 1), Berlín en (4, 5), España en (8, 1) y Madrid en (8, 5). Calcule $\text{Berlín} - \text{Alemania} + \text{España}$.

Pista: Halle primero el paso de Alemania a Berlín.

Respuesta: $\text{Berlín} - \text{Alemania} = (0, 4)$; sumándolo a España se obtiene (8, 5), que es Madrid.

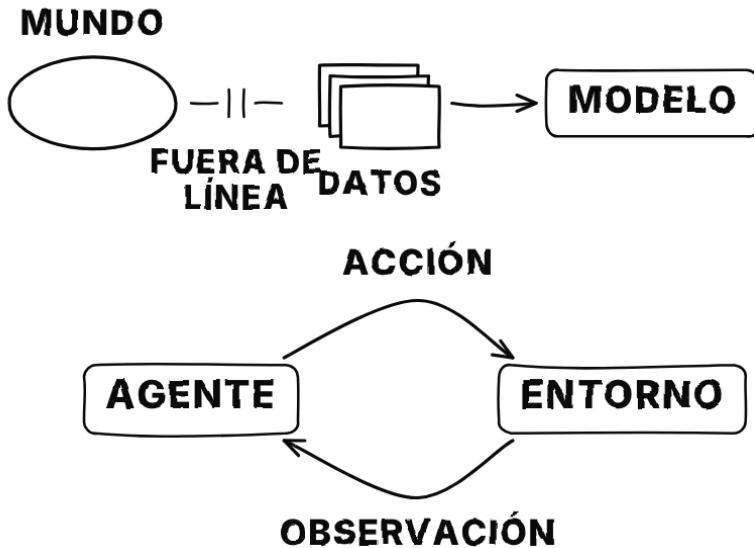
2. Explique con sus propias palabras por qué el aprendizaje autosupervisado puede usar miles de millones de frases y el supervisado normalmente no.

Pista: ¿Quién tiene que poner las etiquetas en cada caso?

Respuesta: El aprendizaje supervisado necesita una etiqueta para cada ejemplo, que normalmente ponen personas, algo lento y caro. El autosupervisado fabrica sus etiquetas a partir de los propios datos, por ejemplo ocultando palabras y pidiendo al modelo que las adivine, así que puede usarse cualquier cantidad de texto sin etiquetar.

Cuando el mundo responde

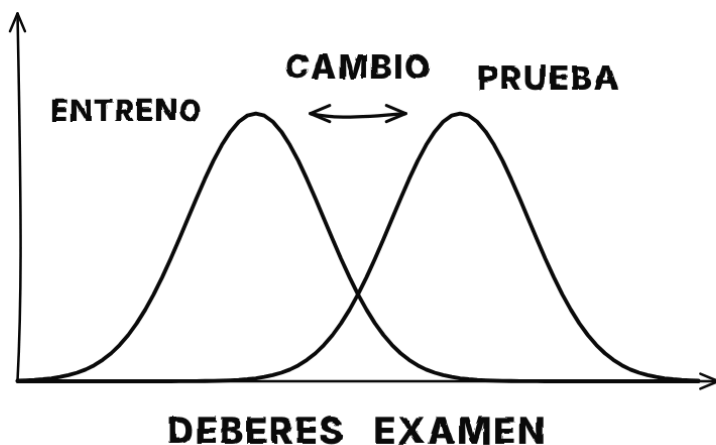
Piense de dónde han salido hasta ahora todos nuestros datos. Alguien reunió una vez un gran montón de ejemplos, y después la máquina de aprendizaje se puso a trabajar con él, sin volver a tocar el mundo. El libro señala que el aprendizaje supervisado y el no supervisado apenas se preguntan de dónde vienen los datos ni qué ocurre después de que el modelo da su respuesta. Como todo el aprendizaje ocurre después de que el programa se ha desconectado de su entorno, esto se llama aprendizaje fuera de línea. La figura del libro lo dibuja de forma sencilla: el entorno aporta datos, los datos se recogen y el modelo aprende del montón.



Aprendizaje fuera de línea frente a actuar

Esto tiene un encanto real. Podemos estudiar el reconocimiento de patrones por sí solo, sin preocuparnos por un mundo que se mueve y reacciona. Pero es limitado. Cualquiera que haya crecido con los relatos de robots de Isaac Asimov imagina máquinas inteligentes que no solo predicen, sino que actúan en el mundo. Así que queremos pensar en agentes, no solo en modelos que predicen, y en elegir acciones, no solo en hacer predicciones. La diferencia es que las acciones cambian el mundo. Las acciones de un agente cambian lo que verá a continuación, así que no puede aprender una sola vez de un montón fijo y darlo por terminado. Un robot aspirador es el caso cotidiano: hacia dónde va decide qué parte de la habitación verá después, y por tanto qué puede aprender de la habitación.

En cuanto el mundo responde, surgen preguntas nuevas, y el libro enumera algunas. ¿Recuerda el entorno lo que hicimos antes? ¿Quiere ayudarnos, como una persona que habla con claridad a un reconocedor de voz? ¿Quiere vencernos, como los spammers que reescriben sin parar sus correos para colarse entre los filtros de spam? Su filtro de spam juega de verdad a este juego: cada vez que aprende un truco, los remitentes cambian el truco. ¿Y cambia el mundo con el tiempo, de modo que el futuro no se parezca al pasado, ya sea por sí solo o por las mismas herramientas que desplegamos?



Entrenado con una, probado con otra

Todas estas preguntas llevan a un problema, el cambio de distribución: los datos con los que se prueba o se usa el modelo son distintos de aquellos con los que se entrenó. Los autores dan un

ejemplo que conocen muchos estudiantes. Los deberes los ponían los profesores ayudantes, así que usted practicaba con su tipo de preguntas. Luego el examen lo escribía el profesor, y las preguntas eran distintas. Se entrenó con una distribución y le examinaron con otra.

Pruébelo

Pida a un amigo que escriba diez preguntas de aritmética con su propio estilo, y a otro diez más con un estilo distinto, por ejemplo problemas con enunciado en lugar de cuentas sueltas. Practique solo con el primer grupo hasta que sea rápido, y luego cronometrese con el segundo. Normalmente irá más lento con el estilo desconocido, aunque la aritmética sea la misma. Esa lentitud es el cambio de distribución en miniatura.

Para recordar

El aprendizaje fuera de línea reúne una vez un montón de datos y aprende de él sin volver a actuar en el mundo. Es cómodo, pero las acciones de un agente cambian el mundo y, por tanto, lo que ve después. El entorno puede recordar, ayudar, contraatacar o ir cambiando con el tiempo. Cuando los datos en el momento de uso difieren de los de entrenamiento, tenemos un cambio de distribución, como los deberes de los ayudantes y el examen del profesor.

Preguntas

1. ¿Por qué no basta el aprendizaje fuera de línea para un agente como un robot aspirador?
 - a. Porque las acciones del agente cambian lo que verá a continuación
 - b. Porque los robots no pueden guardar datos
 - c. Porque el aprendizaje fuera de línea solo funciona con texto
 - d. Porque un agente nunca hace predicciones
2. Los spammers reescriben sin parar sus correos para colarse entre los filtros. ¿Qué tipo de entorno es este?
 - a. Uno que quiere ayudarnos
 - b. Uno que quiere vencernos
 - c. Uno que nunca cambia
 - d. Uno que lo olvida todo
3. Practicó con deberes puestos por los profesores ayudantes y luego se enfrentó a un examen escrito por el profesor. ¿De qué es esto un ejemplo?
 - a. Sobreajuste a un conjunto de prueba
 - b. Aprendizaje fuera de línea
 - c. Cambio de distribución
 - d. Agrupamiento

Respuestas

1. **a)** Las acciones cambian el mundo: hacia dónde va el robot decide qué parte de la habitación verá después, así que no puede aprender una sola vez de un montón fijo.
2. **b)** El ejemplo del libro de un entorno que quiere vencernos son los spammers que reescriben correos para esquivar los filtros de spam.
3. **c)** Se entrenó con una distribución, las preguntas de los ayudantes, y le examinaron con otra, las del profesor: un cambio de distribución.

Ejercicios

1. Un modelo que predice lo que la gente compra en internet se entrenó con datos de antes de unas grandes rebajas navideñas. Explique cómo podría afectarle el cambio de distribución durante las rebajas.

Pista: ¿Se parecen los datos de las rebajas a aquellos con los que se entrenó?

Respuesta: Durante las rebajas la gente compra cosas distintas, en cantidades y a precios distintos, así que los datos en el momento de uso difieren de los de entrenamiento. Las predicciones del modelo pueden ser malas porque aprendió patrones que ya no se cumplen.

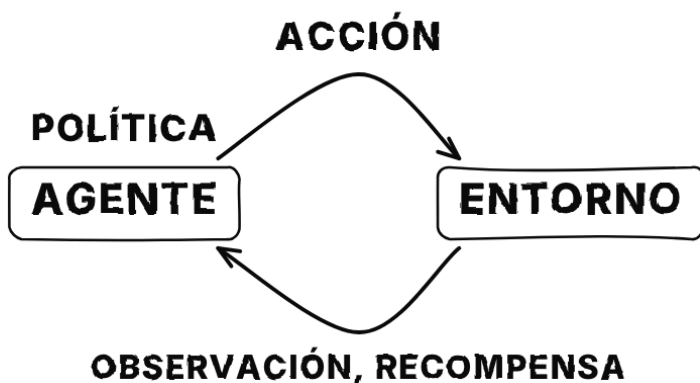
2. Para cada entorno, diga si ayuda, contraataca o simplemente cambia con el tiempo: una persona que habla con claridad a un asistente de voz; estafadores que cambian sus trucos para burlar el detector de un banco; las tendencias de la moda que varían de un año a otro.

Pista: Pregúntese qué quiere el entorno, si es que quiere algo.

Respuesta: La persona que habla con claridad ayuda al sistema. Los estafadores contraatacan, adaptándose para burlar el detector. La moda simplemente cambia con el tiempo, así que los datos futuros no se parecen a los pasados.

Aprender actuando: aprendizaje por refuerzo

Cuando enseña a un perro a sentarse, no le explica nada. Espera a que se siente por casualidad, le da un premio y repite. Tras muchos intentos, el perro averigua qué acciones suyas le traen premios. Nunca se le dice la respuesta correcta; solo descubre, después, si las cosas salieron bien.



POLÍTICA: OBSERVACIÓN → ACCIÓN

El bucle del aprendizaje por refuerzo

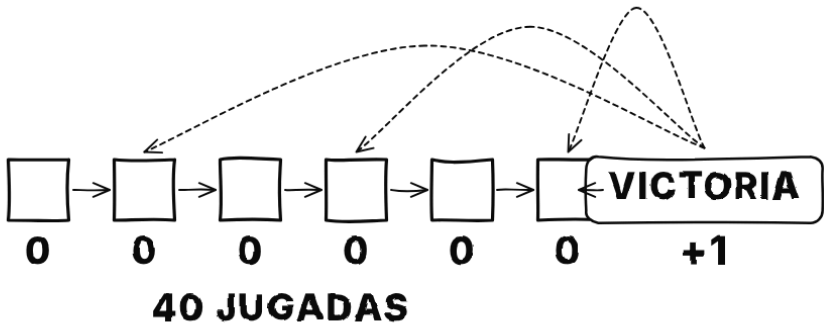
Cuando se usa el aprendizaje automático para construir un agente que actúa en un mundo, dice el libro, lo más probable es que acabe

exactamente con este tipo de aprendizaje, llamado aprendizaje por refuerzo. Entre sus usos hay robots, sistemas de diálogo y jugadores de videojuegos. Combinado con el aprendizaje profundo se convirtió en aprendizaje por refuerzo profundo, y dos veces llegó a los titulares. Un programa llamado red Q profunda aprendió a ganar a personas en videojuegos de Atari a partir solo de la imagen de la pantalla. Y AlphaGo venció al campeón del mundo del juego de tablero go.

El planteamiento es sencillo y muy general. El agente y el entorno se turnan, paso tras paso. En cada paso, el agente recibe una observación, lo que puede ver del mundo. Elige una acción, que se transmite de vuelta al mundo mediante algún mecanismo, a veces llamado actuador, como el motor de un robot. Después recibe una recompensa, un número que dice lo bueno que fue ese turno, y el bucle vuelve a empezar con la siguiente observación. La figura del libro lo dibuja como un círculo: el agente envía acciones al entorno, y el entorno devuelve observaciones y recompensas. Lo que hace el agente lo rige su política, simplemente una regla, una función, que asigna a lo que observa lo que hace. El objetivo del aprendizaje por refuerzo es encontrar buenas políticas. Un robot aspirador que aprende la distribución de su piso está en este bucle: percibe la habitación, elige hacia dónde ir y recibe una recompensa por el suelo limpiado.

El marco es tan general que incluso engloba el aprendizaje supervisado. Tome una tarea de clasificación con diez clases. Dé al agente diez acciones, una por clase, y deje que el entorno pague una recompensa sacada directamente de la función de pérdida del problema original, de modo que la recompensa puntúe cada elección igual que la pérdida puntuaba cada predicción. Elegir acciones para ganar recompensa es entonces lo mismo que clasificar.

¿MÉRITO?



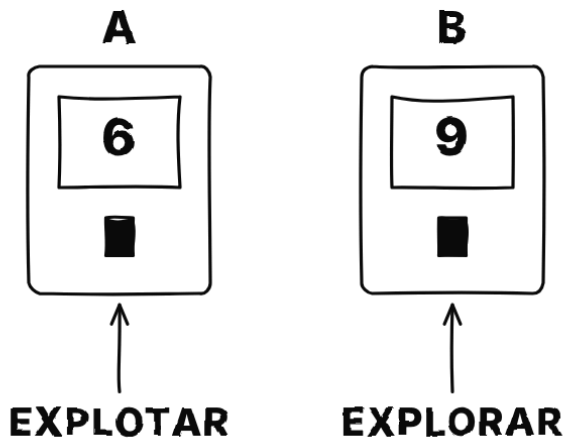
Recompensa solo al final

Y trata cosas que el aprendizaje supervisado no puede. En el aprendizaje supervisado, cada entrada de entrenamiento viene con su etiqueta correcta. En el aprendizaje por refuerzo, nadie le dice al agente la mejor acción para cada observación; solo recibe alguna recompensa. Peor aún, puede que el mundo no le diga qué acciones le valieron esa recompensa.

El libro usa el ajedrez para mostrar el problema. La única recompensa real llega al final del todo, digamos +1 por ganar y -1 por perder. Una partida puede durar 40 jugadas, y el +1 llega solo después de la última. ¿Cuál de esas 40 jugadas ganó la partida, y cuál estuvo a punto de perderla? Averiguar qué acciones merecen el mérito o la culpa se llama el problema de la asignación de mérito. El ejemplo humano del

libro es una empleada a la que ascienden el 11 de octubre: el ascenso refleja muchas buenas decisiones del último año, y para volver a ascender hay que averiguar cuáles importaron.

Una segunda dificultad es la observabilidad parcial: lo que el agente ve ahora puede no decirle dónde está. Imagine a su robot de limpieza atrapado en uno de varios armarios idénticos de su casa. Desde dentro todos parecen iguales; para saber en cuál está, debe recordar lo que vio antes de entrar.



Explotar o explorar

Una tercera es la elección entre aprovechar y buscar. En cualquier momento, el agente puede conocer una buena política, mientras existen otras mejores que nunca ha probado. Puede explotar,

quedándose con la mejor estrategia que conoce, o explorar, probando otras y renunciando ahora a algo de recompensa a cambio de conocimiento.

Pruébelo

Imagine dos máquinas de café en el trabajo que no ha probado. El lunes prueba la máquina A y el café está bien, digamos un 6 sobre 10. Si solo explota, bebe de la A para siempre y nunca descubre que la máquina B hace un café de 9. Si dedica el martes a explorar la B, se arriesga a un mal café ese día, pero cada día siguiente puede ser un 9. Ese intercambio entre un 6 seguro ahora y un posible 9 después es exactamente la elección entre explorar y explotar.

Todo esto junto, acciones que moldean las observaciones posteriores, recompensas vistas solo para las acciones realizadas, un mundo visto entero o solo en parte, es mucho que manejar a la vez. Y no todo problema práctico lo tiene todo. Así que los investigadores estudian casos especiales, y el libro nombra tres. Cuando el agente ve todo el estado del mundo, el problema se llama proceso de decisión de Markov. Cuando el estado no depende de lo que el agente hizo antes, es un problema de bandido contextual. Y cuando no hay ningún estado, solo un conjunto de acciones cuyas recompensas se desconocen al principio, es el clásico problema del bandido multibrazo, llamado así por una fila de máquinas tragaperras, cada una con su propio pago desconocido; nuestras dos máquinas de café son uno diminuto. Una aplicación de noticias que prueba dos titulares con lectores distintos para ver cuál recibe más clics está resolviendo un problema de bandido.

Para recordar

En el aprendizaje por refuerzo, un agente actúa paso a paso: observa, elige una acción y recibe una recompensa. Su política asigna acciones a las observaciones, y el objetivo es una buena política. A diferencia del aprendizaje supervisado, nunca se le dice la acción correcta, y la recompensa puede llegar tarde, así que debe resolver la asignación de mérito, como en el ajedrez con $+1$ o -1 solo al final. Puede que solo vea una parte del mundo, y debe equilibrar explotar lo que funciona con explorar lo que podría funcionar mejor. Entre los casos especiales están los procesos de decisión de Markov, los bandidos contextuales y los bandidos multibrazo.

Preguntas

1. ¿Qué hace la política de un agente de aprendizaje por refuerzo?
 - a. Asigna a lo que observa el agente lo que hace
 - b. Decide lo grande que es la recompensa
 - c. Etiqueta cada ejemplo de entrenamiento
 - d. Guarda cada observación para siempre
2. Un programa de ajedrez gana una partida de 40 jugadas y recibe $+1$ al final. ¿Qué problema debe resolver para aprender de ello?
 - a. La observabilidad parcial
 - b. El cambio de distribución

- c. La asignación de mérito: qué jugadas merecen el mérito
 - d. Agrupar las jugadas en grupos
3. ¿Por qué un robot de limpieza atrapado en uno de varios armarios idénticos se enfrenta a la observabilidad parcial?
- a. Su recompensa es demasiado pequeña
 - b. Lo que ve ahora no le dice en qué armario está
 - c. Tiene demasiadas acciones entre las que elegir
 - d. Se le dice la acción correcta en cada paso
4. Un problema no tiene ningún estado, solo un conjunto de acciones cuyas recompensas se desconocen al principio. ¿Cómo se llama?
- a. Un proceso de decisión de Markov
 - b. Un problema de bandido contextual
 - c. Aprendizaje supervisado
 - d. Un problema de bandido multibrazo

Respuestas

1. **a)** Lo que hace el agente lo rige su política, una función que asigna a lo que observa lo que hace; el objetivo es encontrar buenas políticas.
2. **c)** La recompensa llega solo después de la última jugada, así que quien aprende debe averiguar qué acciones merecen el mérito o la culpa.
3. **b)** Desde dentro todos los armarios parecen iguales; para saber en cuál está, debe recordar lo que vio antes de entrar.
4. **d)** Sin estado, solo con acciones cuyas recompensas se desconocen al principio, es el clásico problema del bandido multibrazo.

Ejemplos resueltos

¿Merece la pena explorar? Durante 10 días de trabajo toma un café al día. La máquina A, probada el día 1, puntúa 6. La máquina B no se ha probado. Compare la puntuación total si siempre usa la A, y si prueba la B el día 2 y luego usa la mejor máquina, cuando la B resulta puntuar 9 y cuando resulta puntuar 2.

Explotar siempre la A: $10 \times 6 = 60$.

Explorar la B el día 2 cuando la B puntúa 9: $6 + 9 + 8 \times 9 = 87$.

Explorar la B el día 2 cuando la B puntúa 2: $6 + 2 + 8 \times 6 = 56$.

Explorar cuesta como mucho 4 puntos si la B es mala, pero gana 27 si es buena: renunciar a un poco de recompensa a corto plazo puede comprar un conocimiento valioso.

Respuesta: Siempre A: 60; explorando: 87 si la B es buena, 56 si es mala

Ejercicios

1. Nombre la observación, la acción y la recompensa de un robot aspirador que limpia un piso.

Pista: ¿Qué percibe, qué hace y por qué recibe recompensa?

Respuesta: La observación es lo que sus sensores le dicen de la habitación que le rodea; la acción es hacia dónde va a continuación; la recompensa es la cantidad de suelo que limpia.

2. Explique con sus propias palabras cómo puede convertirse una tarea de clasificación con diez clases en aprendizaje por refuerzo.

Pista: Haga de cada clase una acción.

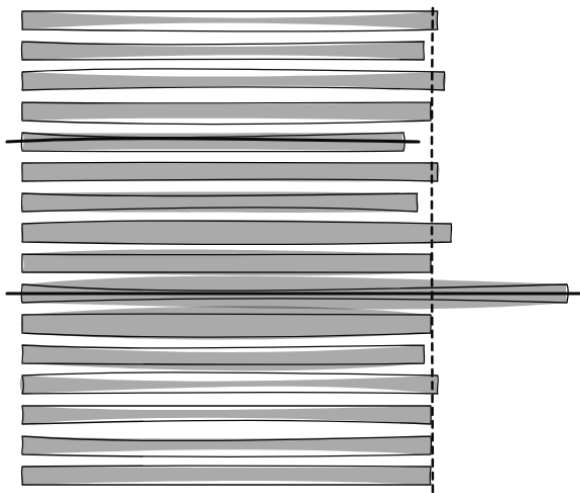
Respuesta: Se dan al agente diez acciones, una por clase, y se deja que el entorno pague una recompensa sacada de la función de pérdida del problema original. Así la recompensa puntúa cada elección de clase igual

que la pérdida puntuaba cada predicción, de modo que aprender a actuar bien es aprender a clasificar.

Raíces antiguas: de las puertas de la iglesia a las neuronas

Imagine un domingo en un pueblo alemán del siglo XVI. Termina la misa y, cuando la gente sale de la iglesia, se pide a dieciséis hombres adultos que se pongan en fila, talón con punta. Se mide la longitud total de sus dieciséis pies y se divide entre dieciséis. El resultado se declara un «pie», la unidad de longitud del pueblo.

30,1, RECORTADA



Dieciséis pies, un pie

El libro toma esta escena de un libro de geometría de Jacob Köbel, que vivió de 1460 a 1533, y su figura muestra a los hombres en fila. La

idea es que aprender de los datos es mucho más antiguo que los ordenadores. La gente siempre ha querido leer los datos y predecir lo que viene, y ese deseo está en la raíz de buena parte de la ciencia y las matemáticas. Incluso esta receta sencilla se mejoró después para tratar los pies raros: se despedía a los hombres del pie más corto y del más largo, y solo se promediaban los otros catorce. Es una de las primeras medias recortadas conocidas, una media que ignora los extremos.

Veamos por qué ayuda, con números inventados. Supongamos que quince hombres tienen pies de unos 30 centímetros, que suman 450, y el decimosexto, un gigante, tiene un pie de 40. La media simple es $(450 + 40) / 16 \approx 30,6$ centímetros, arrastrada hacia arriba por un solo hombre. Quite el pie más largo, 40, y el más corto, digamos 28, y promedie los catorce restantes: $(450 - 28) / 14 \approx 30,1$. Un pie inusual ya no mueve la unidad del pueblo. Las valoraciones de una tienda de aplicaciones responden a la misma preocupación: unas pocas reseñas furiosas o falsas pueden hundir la media, y por eso algunos sitios las recortan o les dan menos peso.

El libro nombra más antepasados. Jacob Bernoulli, de 1655 a 1705, y Carl Friedrich Gauss, de 1777 a 1855, dieron cada uno su nombre a una distribución de probabilidad que todavía se usa en todas partes. Gauss también inventó el método de mínimos cuadrados, que elige la recta que hace mínima la suma de los fallos al cuadrado, y que todavía se usa para todo, desde los seguros hasta el diagnóstico médico. Herramientas como estas reforzaron la ciencia experimental. La ley de Ohm para una resistencia, $V = R \times I$, tensión igual a resistencia por corriente, es un modelo lineal perfecto: duplique la corriente que pasa por una resistencia de 10Ω de 0,5 A a 1 A, y la tensión se duplica de 5 a 10 voltios.

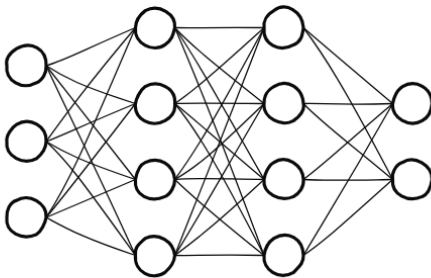
La estadística despegó de verdad cuando pudieron reunirse datos en cantidad. Ronald Fisher, de 1890 a 1962, construyó buena parte de su teoría y la aplicó a la genética; los métodos e ideas que llevan su nombre siguen siendo básicos en el campo. Su conjunto de datos Iris de 1936, medidas de tres tipos de flor de iris, todavía se usa para mostrar algoritmos de aprendizaje. El libro añade una advertencia: Fisher también promovió la eugenesia, un recordatorio de que la ciencia de datos se ha usado para fines moralmente dudosos desde hace tanto como para fines buenos.

Otras raíces están en la teoría de la información de Claude Shannon y en la teoría de la computación de Alan Turing. Turing preguntó «¿pueden pensar las máquinas?» y propuso lo que hoy llamamos prueba de Turing: una máquina cuenta como inteligente si a una persona que intercambia texto con ella le cuesta distinguir sus respuestas de las de un humano. A mucha gente que conversa con un chatbot moderno le resultaría difícil juzgar esa prueba, y en parte por eso se sigue discutiendo.

Otras raíces más vienen del estudio del cerebro y de la mente. Los seres humanos son claramente inteligentes, así que los estudiosos se preguntaron si esa capacidad podría explicarse y copiarse. Donald Hebb, de 1904 a 1985, propuso en su libro *The Organization of Behavior* que las neuronas aprenden por refuerzo positivo: las conexiones que se usan juntas se fortalecen. Esta regla de Hebb inspiró el perceptrón de Rosenblatt, una de las primeras máquinas que aprendían, y está en la base de la idea que hay detrás de los métodos de entrenamiento del aprendizaje profundo actual: reforzar lo que funciona, debilitar lo que no, y así llegar a buenos valores de los parámetros de la red.



JUNTAS



LINEAL

LINEAL

NO LINEAL

**REGLA DE
LA CADENA**



Capas y retropropagación

De ahí viene el nombre de red neuronal. Desde hace más de un siglo, desde los modelos de Alexander Bain en 1873 y de James Sherrington en 1890, como los nombra el libro, los investigadores han intentado construir circuitos parecidos a redes de neuronas. Con el tiempo el vínculo con la biología se ha aflojado, pero el nombre se ha quedado. Hay dos principios presentes en la mayoría de las redes actuales. Primero, capas que alternan pasos lineales sencillos con otros no lineales. Segundo, la regla de la cadena del cálculo, conocida en este campo como retropropagación, que ajusta a la vez todos los parámetros de la red entera.

Pruébalo

Sostenga una regla y pida a un amigo que la suelte, y atrápela lo más rápido que pueda, diez veces. Al principio sus capturas son lentas e irregulares; tras unos intentos, se vuelven más seguras. Nadie le dijo a su cerebro qué conexiones nerviosas reforzar; las que llevaron a una buena captura simplemente se volvieron a usar. Ese es el espíritu de la regla de Hebb, que inspiró las primeras máquinas que aprendían.

Tras un rápido progreso inicial, la investigación en redes neuronales se estancó aproximadamente de 1995 a 2005, por dos razones. Entrenar una red exige muchísimo cálculo, y a finales del siglo pasado la memoria abundaba pero la potencia de cálculo escaseaba. Y los conjuntos de datos eran pequeños. Los datos Iris de Fisher, de 1936, con 150 flores, seguían siendo una prueba habitual, y la colección MNIST de 60 000 dígitos manuscritos se consideraba enorme. Con pocos datos y poca potencia de cálculo, otros métodos, los métodos de núcleo, los árboles de decisión y los modelos gráficos, sencillamente funcionaban mejor en muchos usos. No necesitaban semanas de entrenamiento y daban resultados predecibles respaldados por una teoría sólida.

Para recordar

Aprender de los datos tiene siglos: los dieciséis pies de Köbel, promediados y después despojados de los extremos, son una estimación temprana. Bernoulli, Gauss, Fisher, Shannon y Turing pusieron los cimientos, y los mínimos cuadrados de Gauss y los modelos lineales como la ley de Ohm siguen en uso. La idea de Hebb de que las neuronas aprenden por refuerzo llevó al perceptrón y dio su nombre a las redes neuronales. Las redes actuales se apoyan en capas lineales y no lineales alternadas y en la retropropagación. Aproximadamente de 1995 a 2005, las redes neuronales se estancaron porque el cálculo escaseaba y los conjuntos de datos eran pequeños, y métodos más sencillos y rápidos lo hacían mejor.

Preguntas

1. ¿Por qué la receta de Köbel despedía a los hombres del pie más corto y del más largo?
 - a. Para que un solo pie inusual no moviera mucho la unidad del pueblo
 - b. Porque solo cabían catorce hombres en los escalones de la iglesia
 - c. Para que la división fuera más fácil
 - d. Porque los pies más largos eran de forasteros

2. ¿Qué propuso Turing como prueba de si una máquina es inteligente?
 - a. Que pueda ganar a un humano al ajedrez
 - b. Que aprenda por refuerzo
 - c. Que a una persona que intercambia texto con ella le cueste distinguir sus respuestas de las de un humano
 - d. Que pueda medir pies con precisión
3. ¿Qué dos principios se encuentran en la mayoría de las redes neuronales actuales?
 - a. Las medias recortadas y los mínimos cuadrados
 - b. Capas lineales y no lineales alternadas, y la retropropagación
 - c. La regla de Hebb y la prueba de Turing
 - d. Los métodos de núcleo y los árboles de decisión
4. ¿Por qué los métodos de núcleo y los árboles de decisión funcionaron mejor que las redes neuronales aproximadamente de 1995 a 2005?
 - a. Estaban inspirados en el cerebro
 - b. Necesitaban más datos
 - c. Se inventaron más tarde
 - d. Con pocos datos y poca potencia de cálculo funcionaban mejor, se entrenaban rápido y tenían una teoría sólida

Respuestas

1. a) Quitar los extremos antes de promediar, una media recortada, impide que un pie raro tire de la unidad hacia arriba o hacia abajo.

2. c) La prueba de Turing considera inteligente a una máquina si a una persona que intercambia texto con ella le cuesta distinguir sus respuestas de las de un humano.

3. b) La mayoría de las redes tienen capas que alternan pasos lineales sencillos y no lineales, y usan la regla de la cadena, la retropropagación, para ajustar todos los parámetros a la vez.

4. d) Con pocos datos y poca potencia de cálculo, sencillamente funcionaban mejor, no necesitaban semanas de entrenamiento y daban resultados predecibles respaldados por una teoría sólida.

Ejemplos resueltos

Media simple y media recortada Los pies de quince hombres suman 450 centímetros; un decimosexto pie mide 40 centímetros. El pie más corto mide 28 centímetros. Calcule la media simple de los dieciséis y la media recortada tras quitar el 40 y el 28.

Media simple: $(450 + 40) \div 16 = 490 \div 16 \approx 30,6$ centímetros.

Recortar quita el 40 y el 28; el 28 está entre los quince, así que los catorce restantes suman $450 - 28 = 422$.

Media recortada: $422 \div 14 \approx 30,1$ centímetros.

El único pie largo ya no tira del resultado hacia arriba.

Respuesta: Media simple de unos 30,6 cm; media recortada de unos 30,1 cm

Ejercicios

1. Con la ley de Ohm $V = R \times I$, ¿qué tensión hay en una resistencia de 20 ohmios por la que pasan 0,3 amperios? ¿Qué le pasa a la tensión si se triplica la corriente?

Pista: Multiplique la resistencia por la corriente; la ley es lineal.

Respuesta: $V = 20 \times 0,3 = 6$ voltios. Triplicar la corriente a 0,9 amperios triplica la tensión hasta 18 voltios, porque el modelo es lineal.

2. Dé las dos razones por las que la investigación en redes neuronales se estancó aproximadamente entre 1995 y 2005, y diga cómo limitaba cada una a las redes.

Pista: Una tiene que ver con los ordenadores y otra con los datos.

Respuesta: La potencia de cálculo escaseaba, y entrenar una red exige muchísimo cálculo, así que no se podían entrenar bien las redes. Y los conjuntos de datos eran pequeños, siendo típicos los datos Iris y los 60 000 dígitos de MNIST, así que las redes grandes tenían muy poco de lo que aprender.

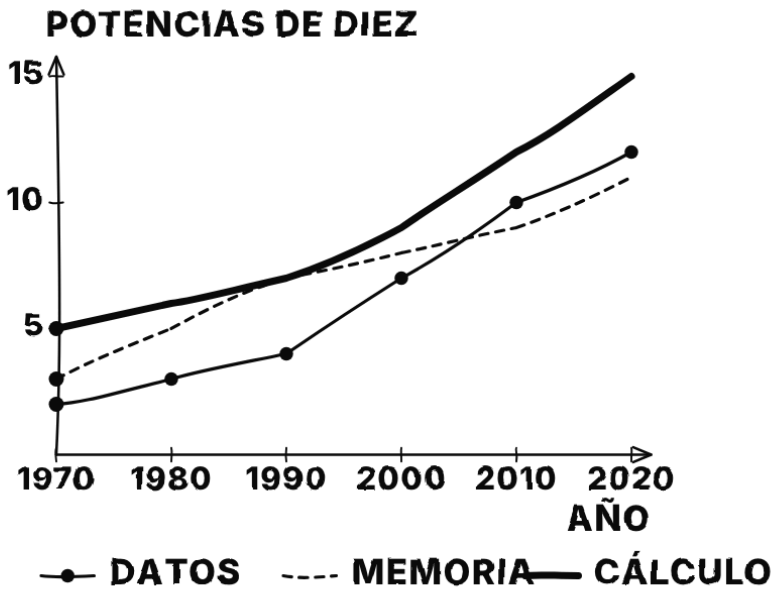
El camino hacia el aprendizaje profundo

Compare el teléfono que lleva en el bolsillo con un ordenador de sobremesa de 1990. Su teléfono guarda miles de veces más fotos de las que podía almacenar aquel ordenador, y hace en un segundo más cálculos que aquella máquina en horas. Piense ahora en cuántas fotos, mensajes y clics produce el mundo entero cada día. El libro dice que este doble cambio, más datos y muchísimo más cálculo, es lo que despertó a las redes neuronales de su largo sueño.

Coincidieron varias cosas. La World Wide Web, y empresas que atienden en línea a cientos de millones de personas, produjeron datos en cantidades nunca vistas. Se extendieron por todas partes sensores baratos y buenos, desde cámaras hasta micrófonos. El almacenamiento se abarató cada vez más, una tendencia que el libro llama ley de Kryder, y el cálculo también, según la ley de Moore. Y un componente cambió las reglas del juego: el procesador gráfico, o GPU, construido para los videojuegos. Una GPU hace miles de cálculos sencillos a la vez, que es justo lo que necesita el entrenamiento de una red neuronal. De repente, modelos que parecían imposiblemente caros de calcular estaban al alcance. El mismo tipo de chip que dibuja el paisaje de un videojuego entrena ahora los modelos que hay detrás de los chatbots y los asistentes de voz.

El libro muestra el cambio en una tabla, década a década, con tres columnas: lo grande que era un conjunto de datos típico, cuánta memoria tenía un ordenador y cuántos cálculos hacía por segundo. En 1970, un conjunto de datos eran unos cien ejemplos, las flores Iris de

Fisher; la memoria era de un kilobyte; un procesador hacía unos cien mil cálculos por segundo. Hacia 1990, los conjuntos de datos tenían unos diez mil ejemplos, como letra manuscrita escaneada; la memoria era de diez megabytes; el procesador hacía diez millones de cálculos por segundo. Hacia 2010, los conjuntos de datos tenían unos diez mil millones de ejemplos, de la publicidad en línea; la memoria era de un gigabyte; una GPU hacía un billón de cálculos por segundo. Y hacia 2020, un conjunto de datos de una red social podía tener un billón de ejemplos, con cien gigabytes de memoria y mil billones de cálculos por segundo.



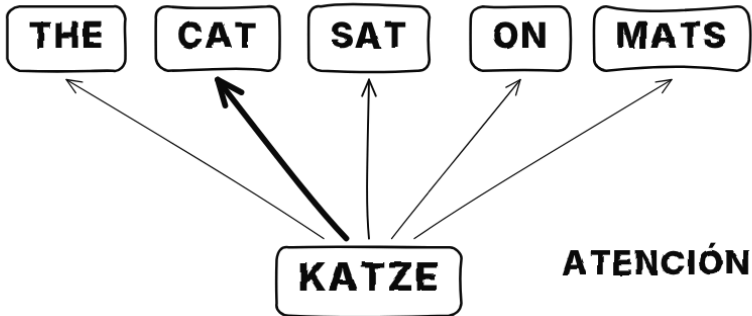
Datos, memoria y cálculo, 1970–2020

Leamos la tabla a lo largo de los cincuenta años. Los datos crecieron de cien ejemplos a un billón, es decir, de 10^2 a 10^{12} , un factor de 10^{10} ,

diez mil millones. La memoria creció de un kilobyte a cien gigabytes, un factor de 10^8 , cien millones. Los cálculos por segundo crecieron de cien mil a mil billones, de nuevo un factor de 10^{10} . Así que la memoria se quedó cien veces por detrás del crecimiento de los datos, mientras que la potencia de cálculo, dice el libro, creció aún más deprisa que los conjuntos de datos. El libro saca la conclusión: los modelos deben usar la memoria con moderación, pero pueden permitirse gastar mucho más cálculo en ajustar sus parámetros. Eso desplazó el punto óptimo del aprendizaje automático lejos de los modelos lineales y los métodos de núcleo, que encajaban con las condiciones antiguas, hacia las redes neuronales profundas. También por eso muchas herramientas clásicas del aprendizaje profundo, como los perceptrones multicapa, las redes convolucionales, la memoria a corto y largo plazo y el Q-learning, se redescubrieron en la práctica en la década de 2010, tras años dormidas.

Pero los autores insisten en que el progreso no fue solo ideas viejas en hardware nuevo. Comparan la última década con la explosión cámbrica, el momento de la historia de la vida en que aparecieron de golpe nuevos tipos de animales. Las ideas nuevas se multiplicaron, y el libro enumera algunas, admitiendo que la lista apenas roza la superficie.

DROPOUT: RUIDO AL ENTRENAR



TRANSFORMER: SOLO ATENCIÓN

La atención como puntero

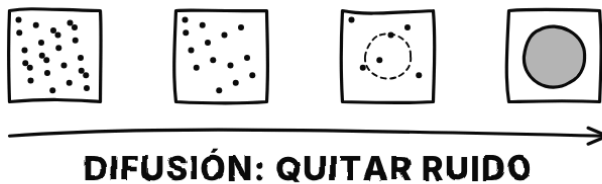
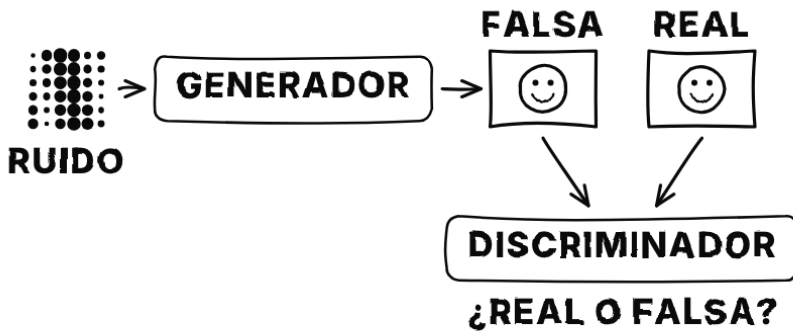
La primera es un mejor control del sobreajuste, con métodos como el dropout. Durante el entrenamiento se inyecta ruido aleatorio por toda la red, por ejemplo apagando partes de ella al azar, para que no pueda limitarse a memorizar los ejemplos de entrenamiento.

La segunda es la atención. La estadística llevaba mucho tiempo con un enigma: cómo dar a un sistema más memoria y más complejidad sin añadir más parámetros aprendibles. Los modelos de traducción antiguos tenían que comprimir una frase entera en un único resumen de tamaño fijo antes de poder empezar a traducirla, y las frases largas salían perjudicadas. La atención permite al modelo guardar en su lugar una especie de puntero aprendible, mirando hacia atrás la parte relevante de la entrada sobre la marcha. La precisión en frases largas

subió de forma notable, porque el modelo ya no tenía que recordarlo todo antes de empezar a producir la traducción.

La tercera nació de la segunda: el Transformer, un diseño construido enteramente sobre la atención. Su propiedad más llamativa es cómo escala: sigue mejorando a medida que crecen los datos, el modelo y el presupuesto de cálculo. Los Transformers triunfan hoy en el lenguaje, la visión, el reconocimiento del habla, el aprendizaje por refuerzo y más. El libro menciona un solo Transformer entrenado con texto, imágenes, movimientos de las articulaciones de un robot y pulsaciones de botones que puede jugar a Atari, describir imágenes, conversar y controlar un brazo robótico.

La cuarta son los modelos de lenguaje, que aprenden la probabilidad del texto y así pueden predecir texto a partir de otro texto. Escalar los datos, el modelo y el cálculo ha desbloqueado cada vez más capacidades. ChatGPT, que nombra el libro, es un modelo de lenguaje ajustado para seguir las intenciones de las personas; puede hablar con él para arreglar un fallo en su código o para esbozar un cuento.



Generador contra discriminador

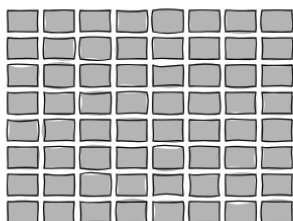
La quinta son los diseños que trabajan en varias etapas, como las redes de memoria y el programador-intérprete neuronal. Permiten a una red cambiar una y otra vez su propio estado interno, paso a paso, igual que un procesador actualiza la memoria durante un cálculo, de modo que puede seguir una cadena de razonamiento.

La sexta es una nueva forma de generar datos, las redes generativas antagónicas. Los métodos antiguos tenían que escribir una fórmula de probabilidad adecuada y luego encontrar la manera de sacar muestras de ella, lo que los limitaba. Una red generativa antagónica sustituye el muestreador por cualquier programa ajustable, el generador, y lo enfrenta a una segunda red, el discriminador, que intenta distinguir los ejemplos falsos de los reales. El generador se ajusta hasta que el

discriminador ya no nota la diferencia. Entre los resultados hay vídeos de cebras al galope y caras convincentes de personas que nunca existieron; incluso aficionados pueden convertir un boceto tosco en una imagen que parece una foto.

La séptima son los modelos de difusión. La difusión añade poco a poco ruido aleatorio a una imagen hasta que solo queda ruido; un modelo de difusión aprende a ejecutar ese proceso al revés, quitando ruido paso a paso, de modo que puede empezar con ruido puro y terminar con una imagen nueva. Estos modelos han empezado a sustituir a las redes antagónicas, por ejemplo en DALL-E 2 e Imagen, que crean imágenes a partir de una descripción escrita. Las aplicaciones que convierten una frase como «un gato con traje espacial» en una imagen se basan en estos modelos.

La octava es entrenar con muchas GPU a la vez. A menudo una sola GPU no basta para todos los datos. El caballo de batalla del entrenamiento, el descenso de gradiente estocástico, trabaja con pequeños lotes de ejemplos cada vez, llamados minilotes, pero los lotes pequeños desaprovechan la potencia de una GPU. Así que se reparte el trabajo: con 1024 GPU, cada una con un minilote de 32 imágenes, un paso procesa $1024 \times 32 = 32\,768$ imágenes, unas treinta y dos mil. Trabajos posteriores lo llevaron a sesenta y cuatro mil imágenes por paso y redujeron el entrenamiento de un conocido modelo de imágenes, ResNet-50, con la colección ImageNet a menos de siete minutos, cuando antes llevaba días.



1024 GPU

$$1,024 \times 32 = 32,768$$

IMÁGENES POR PASO

DÍAS



MINUTOS

THEANO → TENSORFLOW → PYTORCH, JAX

Muchas GPU, un gran lote

La novena es el aprendizaje por refuerzo, que también se benefició del cálculo en paralelo. Los ordenadores alcanzaron un juego sobrehumano en el go, los juegos de Atari y Starcraft, y lo hicieron bien en simulaciones físicas, allí donde existe un simulador del entorno. La razón es sencilla: el aprendizaje por refuerzo funciona mejor con muchos ejemplos de estado, acción y recompensa, y un simulador puede producir tantos como se quiera.

La décima es el software. Los marcos de aprendizaje profundo difundieron las ideas. La primera generación de herramientas de código abierto fue Caffe, Torch y Theano, y con ellas se escribieron muchos artículos emblemáticos. Las sustituyeron TensorFlow, que a menudo se usa a través de su capa más sencilla Keras, y CNTK, Caffe

2 y Apache MXNet. La tercera generación es imperativa: se describe un modelo en Python corriente, paso a paso, igual que se escribiría cualquier cálculo con la biblioteca NumPy; empezó Chainer, y siguieron PyTorch, la interfaz Gluon de MXNet y JAX.

Pruébalo

Tome una hoja de papel y escriba una al lado de otra las tasas de crecimiento de los cincuenta años de la tabla: datos, diez mil millones de veces; memoria, cien millones de veces; cálculo, diez mil millones de veces. Ahora imagine que su casa creciera igual: una habitación cien millones de veces más grande, llena de diez mil millones de veces más cajas. No podría tener cada caja a la vista; tendría que elegir con astucia qué guardar a mano y usar sus piernas, muchísimo más rápidas, para ir a buscar el resto. Esa es la situación que describe el libro, y las redes neuronales profundas encajan en ella mejor que los métodos antiguos.

Para recordar

Las redes neuronales volvieron porque los datos se dispararon, gracias a la web, los sensores y el almacenamiento barato, y el cálculo se abarató, sobre todo gracias a las GPU construidas para los videojuegos. En cincuenta años, los datos y el cálculo crecieron unos diez mil millones de veces y la memoria solo unos cien millones, así que los modelos cambian ahora memoria por cálculo, lo que favorece a las redes profundas. También importaron las ideas nuevas: el dropout contra el sobreajuste, la atención y el Transformer, los modelos de lenguaje, y las redes antagónicas y los modelos de difusión para generar datos. El entrenamiento repartido entre muchas GPU, con pasos de decenas de miles de imágenes, redujo el entrenamiento de días a minutos, y los simuladores alimentaron el aprendizaje por refuerzo. Los marcos de código abierto, de Theano y Caffé a TensorFlow y PyTorch, llevaron las ideas a todo el mundo.

Preguntas

1. ¿Por qué el procesador gráfico, construido para los videojuegos, se adaptó tan bien a las redes neuronales?
 - a. Tiene más memoria que cualquier otro chip
 - b. Hace miles de cálculos sencillos a la vez, que es lo que necesita el entrenamiento
 - c. Se diseñó para guardar grandes conjuntos de datos
 - d. Ejecuta código de Python más rápido

2. La memoria creció mucho más despacio que los datos, mientras que el cálculo creció rápido. ¿Qué supuso esto para los modelos?
 - a. Tenían que guardar cada ejemplo
 - b. Tenían que dejar de usar GPU
 - c. Tenían que usar la memoria con moderación pero podían gastar mucho más cálculo en ajustar parámetros
 - d. Tenían que volverse más pequeños y sencillos, como los modelos lineales
3. ¿Qué problema resolvió la atención para los modelos de traducción?
 - a. Tenían que comprimir una frase entera en un único resumen fijo antes de traducir
 - b. No podían funcionar en GPU
 - c. Necesitaban datos etiquetados
 - d. Se sobreajustaban al conjunto de entrenamiento
4. En una red generativa antagónica, ¿cuándo es suficientemente bueno el generador?
 - a. Cuando copia exactamente los ejemplos de entrenamiento
 - b. Cuando el discriminador detecta todas las falsificaciones
 - c. Cuando escribe una fórmula de probabilidad
 - d. Cuando el discriminador ya no puede distinguir sus falsificaciones de los ejemplos reales

Respuestas

1. **b)** Una GPU hace miles de cálculos sencillos a la vez, que es justo lo que necesita el entrenamiento de una red neuronal.

- 2. c)** Los modelos deben usar la memoria con moderación, pero pueden permitirse mucho más cálculo para ajustar sus parámetros, lo que desplazó el punto óptimo hacia las redes profundas.
- 3. a)** Los modelos antiguos comprimían la frase entera en un resumen de tamaño fijo; la atención guarda un puntero aprendible y mira hacia atrás la parte relevante sobre la marcha.
- 4. d)** El generador se ajusta frente a un discriminador que intenta distinguir lo falso de lo real, hasta que el discriminador ya no nota la diferencia.

Ejemplos resueltos

El minilote total El entrenamiento se reparte entre 1024 GPU, y cada una procesa un minilote de 32 imágenes. ¿Cuántas imágenes procesa un paso de entrenamiento?

Cada GPU procesa 32 imágenes por paso.

Las 1024 GPU trabajan a la vez.

$1024 \times 32 = 32\,768$ imágenes por paso, unas treinta y dos mil.

Respuesta: 32 768 imágenes, unas 32 000

Ejercicios

1. Según la tabla, ¿por qué factor creció la memoria desde un kilobyte en 1970 hasta cien gigabytes en 2020, y por qué factor crecieron los datos de cien ejemplos a un billón?

Pista: Un gigabyte es un millón de kilobytes; cuente potencias de diez.

Respuesta: Memoria: 100 GB son 100 000 000 KB, un factor de 10^8 .

Datos: de 10^2 a 10^{12} , un factor de 10^{10} . Los datos crecieron cien veces más que la memoria.

2. Si 256 GPU procesan cada una un minilote de 64 imágenes, ¿cuántas imágenes trata un paso?

Pista: Multiplique el número de GPU por el tamaño del minilote.

Respuesta: $256 \times 64 = 16\,384$ imágenes por paso.

Éxitos ocultos y éxitos visibles

Escriba una carta, ponga la dirección a mano en el sobre y échela al buzón. En algún centro de clasificación, una máquina lee esa dirección y envía la carta a su destino, y lleva haciéndolo desde los años noventa. Usted no lo ve, pero es aprendizaje automático, y el libro dice que esto es lo típico: la tecnología lleva décadas trabajando en silencio.

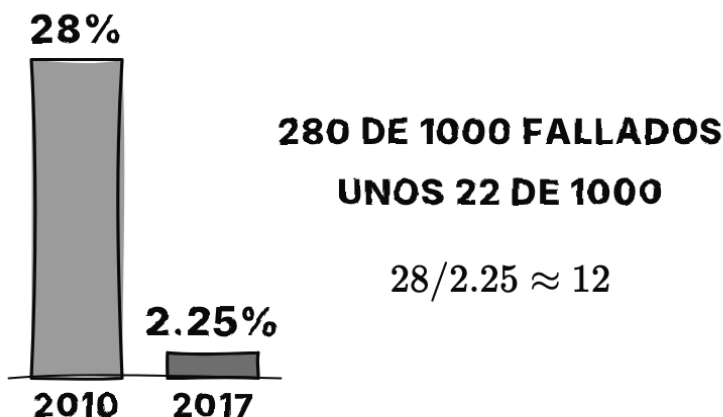
Primero, sin embargo, los autores señalan cuánto más fácil se ha vuelto el propio trabajo. Los investigadores de sistemas construyen ahora las herramientas, y los modeladores estadísticos construyen las redes encima. En 2014, entrenar un modelo sencillo llamado regresión logística era un problema de deberes serio para los nuevos doctorandos de la Universidad Carnegie Mellon. Hoy lleva menos de diez líneas de código, al alcance de cualquier programador.

Ahora, los éxitos ocultos. De la lectura de direcciones manuscritas en el correo salió la famosa colección de dígitos manuscritos MNIST. El mismo tipo de lectura procesa los cheques que se ingresan en los bancos, y modelos parecidos calculan si es probable que los solicitantes de un préstamo lo devuelvan. Los pagos con tarjeta se comprueban automáticamente en busca de fraude, y esa comprobación es la columna vertebral de sistemas de pago como PayPal, Stripe, AliPay, WeChat, Apple, Visa y MasterCard. Los programas de ajedrez llevan décadas siendo fuertes. Y el aprendizaje automático mueve la búsqueda, las recomendaciones, la personalización y la ordenación en todo Internet. En resumen, dice el libro, el aprendizaje automático está en todas partes, aunque casi

siempre fuera de la vista. Cada vez que su tarjeta se bloquea en el extranjero hasta que confirma un mensaje de texto, un modelo de fraude ha mirado ese pago y lo ha encontrado inusual.

Solo recientemente la inteligencia artificial ha salido a escena, sobre todo resolviendo problemas que parecían fuera de alcance y que la gente vive directamente; muchos de estos avances se atribuyen al aprendizaje profundo. El primero y más visible es el asistente de voz. Siri, Alexa y el asistente de Google responden a peticiones habladas con bastante fiabilidad, desde encender una luz hasta pedir cita en la peluquería o atender una llamada de soporte. En su núcleo está el reconocimiento del habla, que ha mejorado hasta igualar, en algunos usos, la precisión humana. Cuando dicta un mensaje en una calle ruidosa y sale bien, está usando esa mejora.

ERROR TOP-5 IMAGENET



Del 28% al 2,25%

El reconocimiento de objetos en imágenes ha avanzado igual de lejos. El libro da cifras de la prueba ImageNet, en la que un modelo tiene cinco intentos para cada imagen y solo cuenta como fallo si ninguno de los cinco acierta. En 2010, el mejor equipo, de NEC Labs y la Universidad de Illinois, tenía un error top-5 del 28%. Para 2017 había bajado al 2,25%. De mil imágenes, eso son unas $0,28 \times 1000 = 280$ falladas en 2010, y solo unas 22 en 2017. El error se redujo en un factor de $28 / 2,25 \approx 12$ en siete años. Hubo saltos parecidos al identificar aves por su canto y al diagnosticar el cáncer de piel.

Los juegos sirven desde hace mucho como vara de medir de la capacidad humana, y el libro traza una escalera de ellos. Empieza con TD-Gammon, un programa de backgammon que aprendía con una

forma de aprendizaje por refuerzo. El ajedrez es mucho más complejo que el backgammon, y Deep Blue de IBM venció al campeón del mundo Garri Kaspárov con un cálculo en paralelo masivo, hardware especial y una búsqueda astuta entre las jugadas posibles. El go es aún más difícil, porque el número de posiciones posibles es enorme; AlphaGo alcanzó el nivel humano en 2015 combinando el aprendizaje profundo con un método de muestrear continuaciones probables, llamado búsqueda en árbol de Montecarlo. El póquer añadió otra dificultad: el juego es grande y solo visible en parte, ya que no se ven las cartas de los demás jugadores. Un programa llamado Libratus venció aun así a jugadores humanos, usando estrategias cuidadosamente estructuradas.

MÁS DIFÍCIL



PÓQUER  **OCULTAS**

GO

AJEDREZ

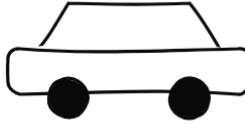
BACKGAMMON

Una escalera de juegos

Pruébalo

Piense en cómo juega al tres en raya con un niño. Ve todo el tablero, así que puede planear cada jugada: eso es como el ajedrez o el go. Piense ahora en un juego de cartas en el que la mano del otro jugador está oculta; debe adivinar por cómo apuesta o juega. Esa información que falta es lo que hizo del póquer un reto propio para los ordenadores.

Los coches autónomos son otra señal de progreso. La autonomía total aún no ha llegado, pero empresas como Tesla, NVIDIA y Waymo venden coches que pueden conducirse solos en parte. Lo difícil es que conducir bien requiere percepción, razonamiento y cumplir reglas, todo en un mismo sistema. Por ahora, dice el libro, el aprendizaje profundo se encarga sobre todo de ver; el resto lo ajustan con cuidado los ingenieros.



LA IA YA DECIDE: CRÉDITO, FIANZA

Conducir requiere más que ver

Y esto es solo una pequeña muestra. La robótica, la logística, el estudio de las células vivas, la física de partículas y la astronomía deben algunos de sus avances recientes al aprendizaje automático, que se está convirtiendo en una herramienta cotidiana de ingenieros y científicos.

Con el éxito llegan las preocupaciones. Los artículos divulgativos se preguntan a menudo si se acerca un «apocalipsis de la IA» o una «singularidad», un punto en el que las máquinas adquirirían consciencia y empezarían a decidir sobre nuestras vidas sin sus programadores. La respuesta del libro empieza con una observación serena: la IA ya afecta directamente a la vida de las personas. Si le conceden un crédito se decide a menudo automáticamente; los pilotos

automáticos hacen la mayor parte del vuelo; y las decisiones sobre conceder la libertad bajo fianza a un acusado usan datos estadísticos. Y, en un tono más ligero, podemos pedirle a Alexa que encienda la cafetera.

Para recordar

Las herramientas han hecho el aprendizaje automático muchísimo más fácil: un modelo que era deberes de doctorado en 2014 ocupa hoy menos de diez líneas de código. Durante décadas ha trabajado fuera de la vista, leyendo direcciones y cheques, evaluando créditos, detectando fraudes y ordenando resultados de búsqueda. El aprendizaje profundo trajo éxitos visibles: asistentes de voz, reconocimiento del habla de nivel casi humano en algunos usos y reconocimiento de imágenes cuyo error en ImageNet bajó del 28% al 2,25%. Los juegos, del backgammon al ajedrez, el go y el póquer, cayeron uno tras otro, y los coches autónomos usan el aprendizaje profundo sobre todo para ver. La IA ya moldea decisiones reales sobre las personas, como los créditos y las fianzas.

Preguntas

1. ¿Por qué dice la lección que el aprendizaje automático está en todas partes aunque casi siempre fuera de la vista?
 - a. Porque solo funciona en laboratorios

- b. Porque durante décadas ha leído el correo, comprobado pagos y ordenado resultados de búsqueda en silencio
 - c. Porque solo fue posible a partir de 2017
 - d. Porque la gente se niega a usarlo
2. En el error top-5 de ImageNet, ¿cuándo cuenta como fallo la respuesta de un modelo?
- a. Cuando falla su primer intento
 - b. Cuando falla cualquiera de sus cinco intentos
 - c. Cuando ninguno de sus cinco intentos acierta
 - d. Cuando tarda más de cinco segundos
3. ¿Qué hizo del póquer un reto distinto del ajedrez o el go?
- a. El juego solo es visible en parte, ya que las cartas de los demás jugadores están ocultas
 - b. Tiene menos posiciones posibles que el ajedrez
 - c. No pueden jugarlo los ordenadores
 - d. No tiene reglas
4. Según la lección, ¿de qué parte de la conducción autónoma se encarga hoy sobre todo el aprendizaje profundo?
- a. De cumplir las normas de tráfico
 - b. De planificar la ruta
 - c. De razonar sobre los demás conductores
 - d. De ver, es decir, de la percepción

Respuestas

1. **b)** La clasificación del correo, la lectura de cheques, la evaluación de créditos, la detección de fraudes, la búsqueda y las recomendaciones usan

aprendizaje automático que nadie ve.

2. c) Un modelo tiene cinco intentos para cada imagen y solo cuenta como fallo si ninguno de los cinco acierta.

3. a) El póquer es grande y solo visible en parte, porque no se ven las cartas de los demás jugadores.

4. d) Conducir bien requiere percepción, razonamiento y reglas; el aprendizaje profundo se encarga sobre todo de ver, y el resto lo ajustan los ingenieros.

Ejemplos resueltos

Cuánto ha avanzado el reconocimiento de imágenes El error top-5 de ImageNet era del 28% en 2010 y del 2,25% en 2017.

¿Cuántas de 1000 imágenes se fallaron cada año, y por qué factor se redujo el error?

2010: $0,28 \times 1000 = 280$ imágenes falladas.

2017: $0,0225 \times 1000 = 22,5$, unas 22 imágenes falladas.

Factor: $28 \div 2,25 \approx 12,4$, así que el error se redujo unas doce veces en siete años.

Respuesta: Unas 280 falladas en 2010, unas 22 en 2017; unas doce veces menor

Ejercicios

1. El error top-5 de un modelo con 2000 imágenes es del 5%.

¿Cuántas imágenes falla?

Pista: Calcule el 5 por ciento de 2000.

Respuesta: $0,05 \times 2000 = 100$ imágenes, aquellas para las que ninguno de sus cinco intentos acierta.

2. Nombre tres formas en que, según la lección, la IA ya afecta directamente a la vida de las personas.

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

Respuesta: Los créditos se deciden a menudo automáticamente, los pilotos automáticos hacen la mayor parte del vuelo y las decisiones sobre conceder la libertad bajo fianza usan datos estadísticos.

Lo que la IA no es, y lo que hace profundo al aprendizaje

Pregunte a un chatbot si planea apoderarse del mundo y quizá le dé una respuesta reflexiva, incluso inquietante. Suena como alguien con mente propia. El veredicto del libro sobre esa impresión es sereno: estamos lejos de cualquier máquina capaz de manipular a sabiendas a las personas que la construyeron.

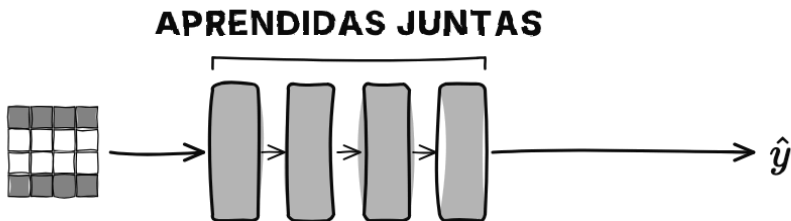
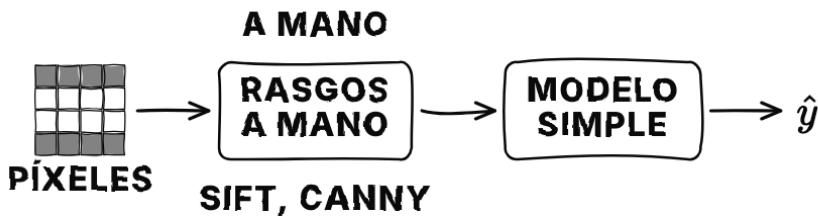
Los autores dan dos razones. Primero, los sistemas de IA se diseñan, entrenan y despliegan para objetivos concretos. Su comportamiento puede dar la ilusión de una inteligencia general, pero por debajo es una mezcla de reglas, reglas prácticas y modelos estadísticos.

Segundo, hoy por hoy simplemente no hay herramientas para la inteligencia artificial general, sistemas capaces de mejorarse a sí mismos, razonar sobre sí mismos y rediseñar su propia estructura mientras intentan resolver cualquier tarea. La respuesta inquietante del chatbot es texto predicho a partir de otro texto, como vimos antes, no un plan.

La verdadera preocupación, dice el libro, es cómo se usa la IA en la vida cotidiana. Muchos trabajos rutinarios que hoy hacen personas pueden automatizarse y se automatizarán. Los robots agrícolas probablemente abaratarán la agricultura ecológica, pero también se harán cargo de la cosecha. Esta etapa de la revolución industrial puede golpear con dureza a amplias capas de la sociedad, porque en muchos países el trabajo manual sencillo da muchísimos empleos. Y los modelos estadísticos aplicados sin cuidado pueden tratar a las personas de forma distinta según su raza, su sexo o su edad, lo que

plantea preguntas justas sobre la equidad cuando esos modelos toman automáticamente decisiones importantes. Estos algoritmos deben usarse con cuidado, y los autores lo consideran mucho más urgente que cualquier superinteligencia malvada que destruya a la humanidad.

Ahora el libro pasa a lo que es realmente el aprendizaje profundo. El aprendizaje profundo es la parte del aprendizaje automático basada en redes neuronales con muchas capas. Es profundo exactamente en ese sentido: sus modelos aprenden muchas capas de transformaciones, una tras otra. Podría decirse que todo aprendizaje automático tiene capas, empezando por los pasos que preparan las características. Lo que distingue al aprendizaje profundo es que cada capa se aprende, y todas se aprenden juntas, a partir de los datos.



DE EXTREMO A EXTREMO

Características hechas a mano frente a extremo a extremo

Los problemas que hemos visto, ondas sonoras, píxeles en bruto, frases de cualquier longitud que traducir, son justo aquellos en los que el aprendizaje profundo brilla y los métodos antiguos tenían dificultades. Los modelos de muchas capas pueden trabajar directamente con datos sensoriales en bruto y de bajo nivel de una forma que las herramientas anteriores no podían. El hilo común, dice el libro, es el entrenamiento de extremo a extremo. En lugar de construir un sistema a partir de piezas, cada una ajustada por su cuenta, se construye el sistema entero y se ajusta todo junto.

La visión por computador es el caso más claro. Durante años, los científicos mantuvieron separados dos trabajos: la ingeniería de características, decidir a mano cómo convertir una imagen en

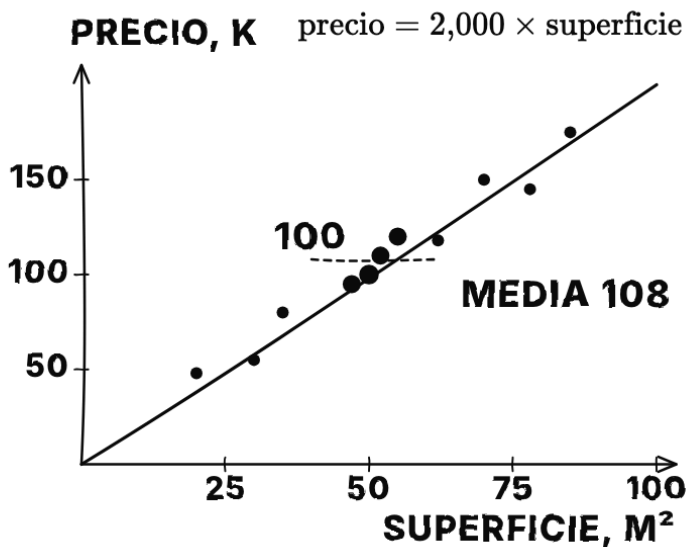
números, y la construcción del modelo que aprendía de esos números. Herramientas hechas a mano como el detector de bordes de Canny y el método SIFT de Lowe reinaron durante más de una década. La parte ingeniosa del trabajo era inventar esas transformaciones para que un modelo sencillo y poco profundo pudiera arreglárselas. Pero el ingenio humano solo puede probar hasta cierto punto, comparado con un algoritmo que prueba millones de opciones de forma sistemática y automática. Cuando llegó el aprendizaje profundo, los extractores de características hechos a mano dejaron paso a filtros ajustados automáticamente, y la precisión subió. Los filtros faciales de una aplicación de videollamadas encuentran sus ojos y su boca con características que nadie dibujó a mano; la red las aprendió de ejemplos.

Así que el aprendizaje profundo sustituye dos cosas a la vez: el modelo poco profundo del final de la antigua cadena y la laboriosa ingeniería de características del principio. Como elimina tanta preparación específica de cada campo, también ha derribado muros. La visión, el reconocimiento del habla, el procesamiento del lenguaje y la informática médica tenían antes sus propios juegos de herramientas por separado; el aprendizaje profundo les da un único conjunto compartido.

Pruébalo

Intente describir con reglas escritas cómo encontrar el borde de una taza de café en una foto: «donde lo oscuro se encuentra con lo claro», quizá. Luego piense en una taza blanca sobre una mesa blanca, o en una taza en la sombra, y vea cómo falla su regla. Un ingeniero de los años dos mil pasaba años refinando reglas así; una red profunda aprende las suyas a partir de miles de fotos de tazas. Ese es el cambio que describe el libro: fuera las características hechas a mano, dentro las capas aprendidas.

El libro nombra un segundo cambio: de los modelos paramétricos a los no paramétricos. Un modelo paramétrico fija de antemano la forma de la respuesta, con unos pocos números por rellenar, como una recta con su pendiente y su valor inicial. Cuando los datos escasean, esas suposiciones simplificadoras son la única forma de obtener un modelo útil. Cuando los datos abundan, se puede prescindir de las suposiciones y el modelo puede seguir a los propios datos. Aquí tiene un ejemplo inventado. Para poner precio a un piso, una regla en línea recta podría decir $\text{precio} = 2000 \times \text{superficie}$, así que un piso de 50 metros cuadrados cuesta 100 000. Un enfoque no paramétrico buscaría en cambio los tres pisos vendidos más parecidos a este y promediaría sus precios, digamos 95, 110 y 120 mil, lo que da $(95 + 110 + 120) / 3 \approx 108$ mil. Con pocos datos, la recta es más segura; con las ventas de toda una ciudad, los vecinos captan cosas que la recta nunca podría, como una calle tranquila o unas vistas.



Una recta frente a los vecinos

Los autores lo comparan con la física de mediados del siglo pasado. Una vez llegaron los ordenadores, los físicos ya no tenían que resolver a mano fórmulas simplificadas del comportamiento de los electrones; podían simular numéricamente las ecuaciones de fondo. Los modelos se volvieron mucho más precisos, pero a menudo más difíciles de interpretar. El aprendizaje profundo paga el mismo precio: los modelos flexibles se ajustan mejor, y son más difíciles de explicar.

Para recordar

Estamos lejos de una IA consciente: los sistemas actuales se construyen para objetivos concretos, y aún no hay herramientas para máquinas que se mejoren y rediseñen a sí mismas. Las preocupaciones urgentes son la automatización del trabajo rutinario y los modelos descuidados que tratan injustamente a las personas. El aprendizaje profundo significa muchas capas de transformaciones, todas aprendidas juntas a partir de los datos y entrenadas de extremo a extremo. Sustituye tanto la ingeniería de características hecha a mano como el modelo poco profundo, y da a la visión, el habla y el lenguaje un conjunto de herramientas compartido. Con datos abundantes, los modelos no paramétricos flexibles sustituyen a las suposiciones simplificadoras, ganando precisión a costa de algo de interpretabilidad.

Preguntas

1. Según el libro, ¿por qué estamos lejos de una IA que pudiera manipular a sabiendas a sus creadores?
 - a. Porque los sistemas de IA se apagan por la noche
 - b. Porque la IA se construye para objetivos concretos y aún no hay herramientas para una inteligencia general que se mejore a sí misma
 - c. Porque los chatbots no saben escribir texto
 - d. Porque las leyes lo prohíben

2. ¿Qué considera el libro más urgente que una superinteligencia malvada?
 - a. Que los ordenadores se queden sin memoria
 - b. Que los chatbots den respuestas inquietantes
 - c. La automatización del trabajo rutinario y los modelos descuidados que tratan injustamente a las personas
 - d. Que los juegos se vuelvan demasiado fáciles para los ordenadores
3. Si podría decirse que todo aprendizaje automático tiene capas, ¿qué distingue al aprendizaje profundo?
 - a. Cada capa se aprende, y todas se aprenden juntas a partir de los datos
 - b. Tiene exactamente dos capas
 - c. Su primera capa siempre se diseña a mano
 - d. No usa datos
4. ¿Qué precio pagan los modelos no paramétricos flexibles, igual que las simulaciones físicas?
 - a. Necesitan menos datos
 - b. Son menos precisos
 - c. No pueden calcularse
 - d. A menudo son más difíciles de interpretar

Respuestas

1. **b)** Los sistemas de IA se diseñan para objetivos concretos, como una mezcla de reglas y modelos estadísticos, y aún no hay herramientas para sistemas que se mejoren y rediseñen a sí mismos.

- 2. c)** La verdadera preocupación es cómo se usa la IA: la automatización de trabajos rutinarios y los modelos estadísticos aplicados sin cuidado que tratan a las personas de forma distinta según su raza, su sexo o su edad.
- 3. a)** Lo que distingue al aprendizaje profundo es que cada capa se aprende, y todas se aprenden juntas, a partir de los datos.
- 4. d)** Simular las ecuaciones hizo los modelos de la física mucho más precisos pero a menudo más difíciles de interpretar; los modelos flexibles del aprendizaje profundo pagan el mismo precio.

Ejemplos resueltos

Una recta frente a los vecinos Una regla en línea recta pone precio a un piso con $\text{precio} = 2000 \times \text{superficie}$. Los tres vecinos vendidos más parecidos a un piso de 50 metros cuadrados se vendieron por 95, 110 y 120 mil. Compare el precio de la recta con la media de los vecinos.

Recta: $2000 \times 50 = 100\,000$.

Vecinos: $(95 + 110 + 120) \div 3 = 325 \div 3 \approx 108,3$ mil.

La recta fija la forma de antemano; la media de los vecinos sigue a los propios datos.

Con muchos datos de ventas, los vecinos pueden captar cosas que la recta pasa por alto, como una calle tranquila.

Respuesta: Recta: 100 000; vecinos: unos 108 000

Ejercicios

- 1.** Con la misma regla en línea recta, $\text{precio} = 2000 \times \text{superficie}$, ¿cuál es el precio de un piso de 65 metros cuadrados? Si sus tres vecinos más cercanos se vendieron por 120, 126 y 138 mil, ¿qué da el enfoque no paramétrico?

Pista: Multiplique para la recta; promedie para los vecinos.

Respuesta: Recta: $2000 \times 65 = 130\,000$. Vecinos: $(120 + 126 + 138) \div 3 = 384 \div 3 = 128$ mil.

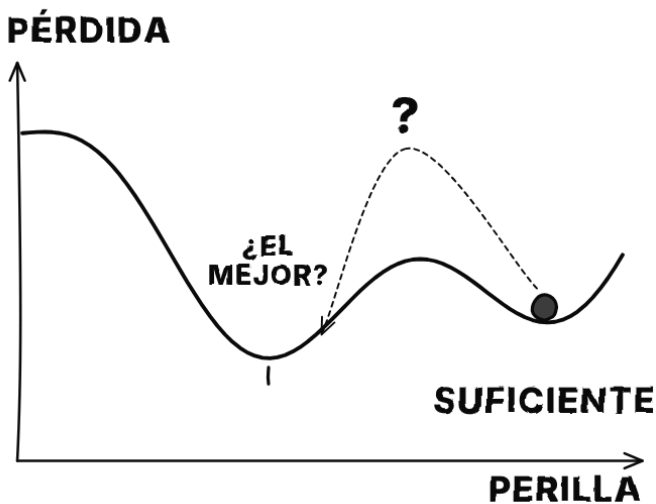
2. Explique con sus propias palabras qué sustituyó el entrenamiento de extremo a extremo en la visión por computador.

Pista: Piense en los dos trabajos separados que hacían antes los científicos.

Respuesta: Los científicos diseñaban antes las características a mano, con herramientas como los detectores de bordes y SIFT, y luego entrenaban por separado un modelo poco profundo con ellas. El entrenamiento de extremo a extremo construye el sistema entero y lo ajusta todo junto, de modo que la red aprende a la vez sus propias características y su modelo final.

Pruébelo, compártalo

Recuerde el descenso por la ladera de antes en el curso, pero ahora el campo está lleno de hondonadas y crestas. Puede acabar en una pequeña hondonada en lugar del valle más profundo, y no puede demostrar que ha encontrado el mejor sitio. El libro dice que el aprendizaje profundo ha hecho las paces con eso.



Hondonadas y crestas

Los trabajos anteriores buscaban problemas en los que la mejor respuesta pudiera encontrarse y demostrarse. El aprendizaje profundo acepta respuestas buenas en lugar de perfectas, trabaja con

paisajes de pérdida enredados y no convexos, llenos de hondonadas, y está dispuesto a probar cosas antes de que nadie pueda demostrar que funcionan. Este nuevo empirismo, junto con una avalancha de recién llegados con talento, ha producido métodos prácticos con rapidez, aunque a menudo rehaciendo o reinventando herramientas que existían desde hacía décadas.

La comunidad también se enorgullece de compartir entre universidades y empresas: se publican abiertamente bibliotecas, modelos y redes totalmente entrenadas. Con el mismo espíritu, los autores hacen que los cuadernos de su libro sean libres de usar y de compartir, para reducir las barreras a cualquiera que quiera aprender. Muchos de los modelos que hay detrás de las aplicaciones de su teléfono partieron de redes compartidas abiertamente de este modo.

El resumen del capítulo ata los cabos. El aprendizaje automático estudia cómo los ordenadores pueden usar la experiencia, normalmente datos, para mejorar en una tarea; se nutre de la estadística, la minería de datos y la optimización, y a menudo es la forma de construir la IA. El aprendizaje de representaciones, una de sus ramas, busca formas buenas de representar los datos automáticamente. El aprendizaje profundo aprende representaciones en muchas capas, y así sustituye tanto el modelo poco profundo como la ingeniería de características hecha a mano. Su progreso reciente se debe a la abundancia de datos, de sensores baratos e Internet, a muchísimo más cálculo, sobre todo gracias a las GPU, y a marcos eficientes que facilitan construir y ajustar sistemas completos.

Así se cierra la introducción: de las reglas que nunca aprenden, pasando por los muchos tipos de problemas de aprendizaje, a los ingredientes que hicieron funcionar al aprendizaje profundo.

Para recordar

El aprendizaje profundo acepta soluciones buenas e imperfectas, trabaja con paisajes de pérdida enredados y prueba ideas antes de demostrarlas. Su comunidad comparte abiertamente herramientas, modelos y redes entrenadas, y este libro es gratuito con el mismo espíritu. El aprendizaje automático usa la experiencia para mejorar; el aprendizaje profundo aprende representaciones en capas y sustituye las características hechas a mano. Los datos, las GPU y unos buenos marcos son las tres fuerzas detrás de su progreso reciente.

Preguntas

1. ¿En qué se diferencia la actitud del aprendizaje profundo ante la optimización de la de los trabajos anteriores?
 - a. Solo acepta soluciones demostradamente óptimas
 - b. Acepta soluciones buenas pero imperfectas y prueba cosas antes de demostrarlas
 - c. Evita los problemas no convexos
 - d. Nunca usa el descenso de gradiente
2. En un paisaje de pérdida lleno de hondonadas y crestas, ¿qué puede ocurrir al bajar la cuesta?
 - a. Siempre se llega al valle más profundo
 - b. No es posible moverse en absoluto
 - c. La pérdida deja de depender de los parámetros

- d. Puede acabar en una pequeña hondonada en lugar del valle más profundo
3. ¿Qué tres fuerzas nombra el resumen detrás del progreso reciente del aprendizaje profundo?
- a. Datos abundantes, más cálculo gracias a las GPU y marcos eficientes
 - b. Características hechas a mano, conjuntos de datos pequeños y demostraciones
 - c. Reglas, bases de datos y navegadores web
 - d. Ajedrez, go y póquer

Respuestas

1. **b)** Los trabajos anteriores buscaban respuestas óptimas demostrables; el aprendizaje profundo acepta respuestas buenas en lugar de perfectas, trabaja con paisajes enredados y prueba ideas antes de demostrarlas.
2. **d)** En un campo lleno de hondonadas uno puede detenerse en una pequeña hondonada y no poder demostrar que ha encontrado el mejor sitio.
3. **a)** El progreso reciente se debe a la abundancia de datos, a muchísimo más cálculo, sobre todo gracias a las GPU, y a marcos eficientes que facilitan construir y ajustar sistemas completos.

Ejercicios

1. Explique con sus propias palabras qué es el aprendizaje de representaciones y cómo lo lleva más lejos el aprendizaje profundo.

Pista: Piense en lo que la ingeniería de características hacía a mano.

Respuesta: El aprendizaje de representaciones busca formas buenas de representar los datos automáticamente, en lugar de que las personas diseñen las características a mano. El aprendizaje profundo aprende esas

representaciones en muchas capas, cada una apoyada en la anterior, lo que le permite sustituir tanto la ingeniería de características hecha a mano como el modelo poco profundo del final.

2. Según el espíritu que describe la lección, ¿por qué compartir abiertamente redes entrenadas podría acelerar el progreso?

Pista: Piense en quién puede construir sobre un modelo compartido.

Respuesta: Cuando las bibliotecas, los modelos y las redes entrenadas se publican abiertamente, cualquiera en una universidad o una empresa puede partir de ellos en lugar de empezar de cero, lo que reduce las barreras de entrada y permite que las ideas se difundan y mejoren deprisa.