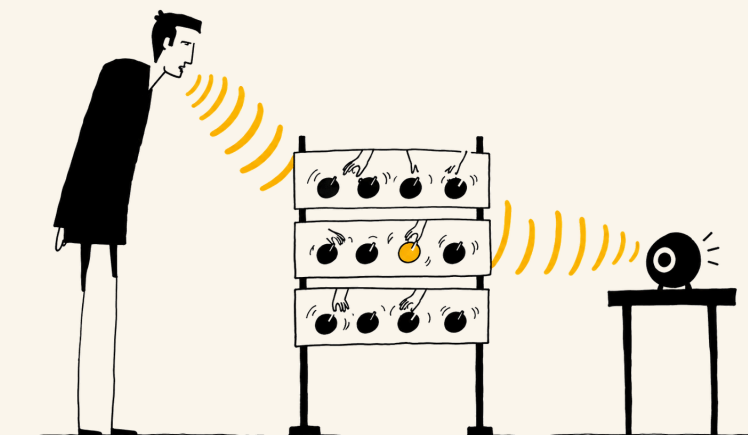


Sumérgete en el aprendizaje profundo



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 en lugar de reglas

De qué aprende la máquina

Puntuar a la máquina

Tanteando el camino hacia abajo

Cuánto, cuántos

Cuál, y con qué seguridad

Muchas etiquetas, y lo mejor primero

Su sección de recomendaciones, y lo que se desarrolla en el tiempo

Darle la vuelta a las cosas, y aprender sin respuestas

Cuando la máquina actúa

Puntos ciegos, apuestas y raíces antiguas

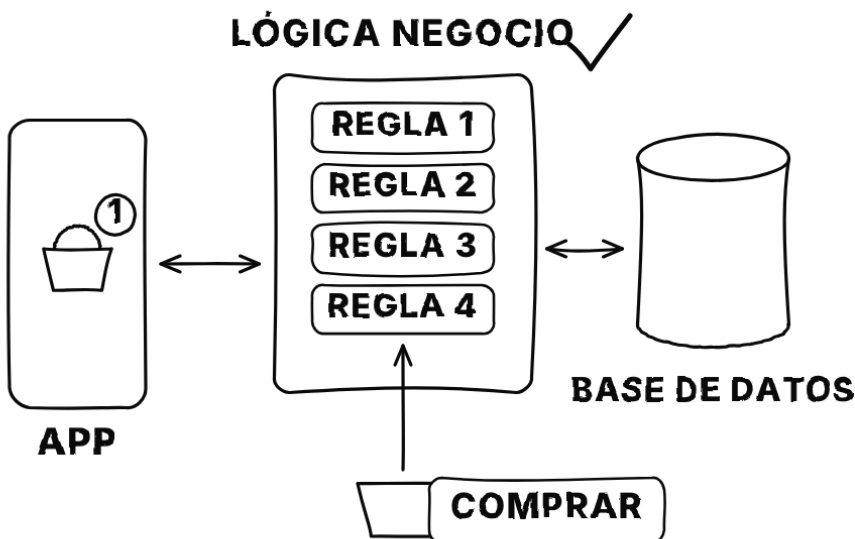
De un callejón sin salida a una explosión

Éxitos silenciosos y preocupaciones reales

Qué lo hace profundo

Reglas que nunca aprenden

Abra cualquier aplicación de compras en su teléfono, ponga un par de calcetines en la cesta y observe cómo el numerito del icono de la cesta salta de cero a uno. Puede hacerlo cien veces, y saltará cada vez, exactamente de la misma manera. En algún lugar, un programador escribió una regla para ese momento: cuando esta persona pulse «añadir», escribir en una tabla una línea nueva que una a esta persona con estos calcetines. Esa regla no cambia en nada porque usted la haya usado. La aplicación no mejora añadiendo calcetines; simplemente hace lo que le dijeron.



Una tienda hecha de reglas

Durante casi toda la historia de la informática, casi todos los programas que usted tocaba en un día estaban hechos así, como una larga lista de reglas escritas por personas. El libro imagina a un equipo frente a una pizarra, planeando una tienda en internet. Deciden que los clientes verán la tienda en un navegador o en una aplicación del teléfono, que una base de datos recordará a cada cliente y cada compra pasada, y que en medio estará la parte que los autores llaman la lógica de negocio, el cerebro de la tienda. Ese cerebro es un conjunto de reglas que le dicen al programa qué hacer en cada situación que al equipo se le ocurra. Luego el equipo sale a la caza de casos incómodos, como un cliente que pulsa «comprar» con la cesta vacía, y añade una regla para cada uno. Rara vez funciona a la perfección la primera vez, pero tras algunas pruebas funciona, y la tienda puede abrir antes de que haya entrado un solo cliente real. Los autores lo llaman una hazaña notable del pensamiento humano, y añaden un consejo certero: si usted puede escribir reglas que acierten el cien por cien de las veces, no necesita aprendizaje automático en absoluto.

Ahora imagine al mismo equipo brillante ante la misma pizarra, pero con otras tareas. Predecir el tiempo de mañana a partir de mapas, imágenes de satélite y las mediciones de las últimas semanas. Leer una pregunta escrita con palabras corrientes y responderla correctamente. Mirar una fotografía, encontrar a cada persona que aparece y trazar un contorno alrededor de cada una. Mostrar a un comprador cosas que probablemente le encantarán pero con las que nunca se habría topado navegando. La última tarea es la que su sección de recomendaciones intenta hacer cada vez que la abre. Para tareas como estas, incluso a los mejores programadores del mundo les costaría escribir las reglas desde cero.

BLANCO MÓVIL

ATRAPADO

WIN \$

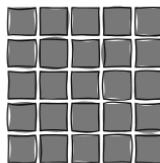
REGLA

WIN \$\$

SE
ESCAPA

MUY ENREDADO

FOTO



¿CÓMO?

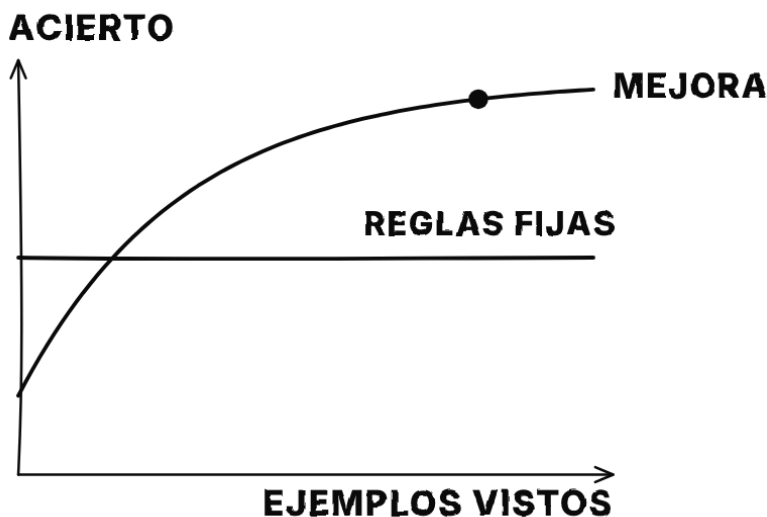
GATO

Dos razones por las que fallan las reglas

El libro da dos razones. La primera es que la respuesta correcta puede ir cambiando. Lo que los compradores quieren esta primavera no es lo que querían la primavera pasada, así que cualquier conjunto fijo de reglas se queda viejo, y una buena solución tiene que adaptarse a medida que cambia el mundo. Un filtro de spam lo sabe bien: en cuanto aprende a atrapar un tipo de correo basura, los remitentes cambian la redacción, y la regla de ayer deja de funcionar. La segunda razón es que la relación entre lo que entra y lo que debería salir puede ser demasiado enredada para escribirla. Una fotografía no es más que una cuadrícula de puntos de colores, y la palabra «gato» es una idea; entre las dos puede haber millones de pequeños cálculos que siguen principios que nadie ha formulado nunca. Usted reconoce sin esfuerzo la cara de un amigo entre la multitud, pero no podría escribir los

pasos que dio su cerebro para hacerlo. La habilidad está ahí, pero oculta a su propia mente consciente.

Aquí es donde entra una idea distinta. El aprendizaje automático es el estudio de los métodos que aprenden de la experiencia. Experiencia aquí suele significar datos, es decir, muchos ejemplos registrados, o los resultados de actuar en algún entorno y ver qué pasa. Cuanta más experiencia reúne un método así, mejor funciona. Compárelo con el botón de los calcetines. La tienda en línea sigue la misma lógica de negocio por muchos millones de clientes que pasen por ella, y solo cambia cuando sus programadores aprenden algo y deciden reescribirla. Un sistema que aprende, en cambio, mejora con los mismos ejemplos que ve. Cuando un asistente de voz entiende mejor su acento, ningún programador escribió una regla especial para su acento; mejoró porque se entrenó con grandes cantidades de voz grabada.



Reglas fijas frente a aprendizaje

El libro se propone enseñar los fundamentos del aprendizaje automático y, sobre todo, una potente familia de sus métodos, llamada aprendizaje profundo. Son los métodos detrás de gran parte del progreso actual en enseñar a los ordenadores a ver, a trabajar con el lenguaje humano, a ayudar en medicina y a leer información genética. El chatbot al que usted escribe preguntas es un producto del aprendizaje profundo: nadie escribió sus respuestas como reglas; aprendió los patrones del lenguaje a partir de una cantidad enorme de texto. Así que la línea divisoria es fácil de enunciar. A un lado están los programas cuyo comportamiento fijan las personas de antemano; al otro, los programas cuyo comportamiento moldean los ejemplos.

Para recordar

La mayoría de los programas son listas de reglas escritas por personas, y se comportan igual por mucho que se usen. Las reglas funcionan bien cuando las personas pueden prever cada caso y acertar siempre, y entonces no hace falta aprendizaje automático. Algunas tareas se resisten a las reglas, o porque la respuesta correcta no deja de cambiar o porque el camino de la entrada a la respuesta es demasiado enredado para escribirlo, aunque nuestro propio cerebro lo recorra sin esfuerzo. El aprendizaje automático es el estudio de los métodos que mejoran con la experiencia, normalmente a partir de datos. El aprendizaje profundo es la familia de esos métodos de la que trata este libro.

Preguntas

1. ¿Por qué el botón de «añadir a la cesta» de la tienda no mejora, por muchos clientes que lo usen?
 - a. Porque sigue reglas escritas de antemano, que solo cambian cuando los programadores las reescriben
 - b. Porque todavía no ha reunido suficientes clientes de los que aprender
 - c. Porque la base de datos olvida cada compra en cuanto se hace
 - d. Porque comprar es una tarea que cambia demasiado deprisa para aprenderla

2. Un equipo puede escribir reglas para una tarea que aciertan absolutamente siempre. ¿Qué aconseja el libro?
 - a. Entrenar de todos modos un sistema que aprenda, porque lo hará aún mejor
 - b. Mezclar las reglas con un sistema que aprenda, por si acaso
 - c. Usar las reglas; para esa tarea no hace falta aprendizaje automático
 - d. Reunir primero datos y luego decidir qué reglas conservar
3. Reconocer la cara de un amigo le resulta fácil. ¿Por qué sigue siendo difícil escribirlo como reglas?
 - a. Porque las caras cambian demasiado a menudo para cualquier programa
 - b. Porque los pasos que da su cerebro están ocultos a su mente consciente
 - c. Porque los ordenadores no pueden guardar fotografías
 - d. Porque hay muy pocas caras de las que aprender
4. Además de la relación enredada entre la entrada y la respuesta, ¿cuál es la otra razón que da la lección para que fallen las reglas?
 - a. Las reglas tardan demasiado en ejecutarse en un ordenador
 - b. Los programadores rara vez prueban sus reglas
 - c. Las bases de datos no pueden guardar suficientes reglas
 - d. La respuesta correcta no deja de cambiar, así que las reglas fijas se quedan viejas

Respuestas

1. **a)** La tienda sigue la misma lógica de negocio por muchos clientes que pasen; solo cambia cuando sus programadores deciden reescribirla.
2. **c)** Si usted puede escribir reglas que acierten el cien por cien de las veces, no necesita aprendizaje automático en absoluto.
3. **b)** Usted reconoce la cara sin esfuerzo, pero no podría escribir los pasos que dio su cerebro; la habilidad está oculta a su mente consciente.
4. **d)** A veces la respuesta correcta va cambiando, así que cualquier conjunto fijo de reglas se queda viejo y una buena solución tiene que adaptarse a medida que cambia el mundo.

Ejercicios

1. Tome las cuatro tareas de la pizarra: el tiempo de mañana, responder a una pregunta escrita, trazar el contorno de cada persona en una foto y sugerir productos que un comprador no encontraría navegando. Para cada una, diga cuál de las dos razones hace difíciles las reglas escritas a mano: la respuesta no deja de cambiar, o la relación entre la entrada y la respuesta es demasiado enredada para escribirla.

Pista: Pregúntese en cada tarea: ¿la respuesta correcta se desplaza con el tiempo, o sería usted incapaz de enumerar los pasos aunque la respuesta sea fija?

Respuesta: El tiempo y los gustos de los compradores cambian con el tiempo, así que las reglas fijas se quedan viejas; además, el tiempo depende de demasiadas mediciones enredadas. Responder preguntas libres y señalar a las personas de una foto son sobre todo del tipo enredado: nosotros lo hacemos con facilidad, pero no sabemos detallar los pasos. Varias tareas tienen las dos dificultades a la vez.

2. Explique con sus palabras por qué el caso de la cesta vacía pertenece a la programación corriente y no al aprendizaje automático.

Pista: ¿Puede el equipo prever este caso y escribir la respuesta correcta de una vez por todas?

Respuesta: Pulsar comprar con la cesta vacía es una situación que el equipo puede prever antes de que llegue ningún cliente, y la respuesta correcta, como mostrar un mensaje, se puede escribir y acierta siempre. Las tareas así se resuelven con reglas; el aprendizaje automático es para tareas en las que nadie puede escribir esas reglas.

3. Un filtro de spam se construye de dos maneras: una con una lista fija de frases prohibidas escrita por programadores, la otra entrenada con muchos ejemplos de correos marcados como spam o no spam. ¿Qué le pasa a cada una cuando, mes tras mes, los spammers cambian la redacción?

Pista: ¿Cuál cambia solo cuando una persona la reescribe, y cuál puede mejorar con ejemplos nuevos?

Respuesta: La lista fija sigue igual hasta que alguien la reescribe, así que la nueva redacción se cuela y la lista se queda vieja. El filtro entrenado puede mejorar a medida que recibe nuevos ejemplos marcados, porque un método que aprende mejora con la experiencia.

Perillas en lugar de reglas

Diga «Hey Siri» u «OK Google» desde el otro lado de la habitación y observe cómo se ilumina su teléfono. Se despertó porque distinguió una frase concreta entre todo el sonido que lo rodeaba: la televisión, el hervidor, otras personas hablando. Ahora intente imaginar que escribe usted mismo ese programa, solo en una habitación con un ordenador y nada más. ¿Cuál sería la primera regla?



Entra sonido, sale sí o no

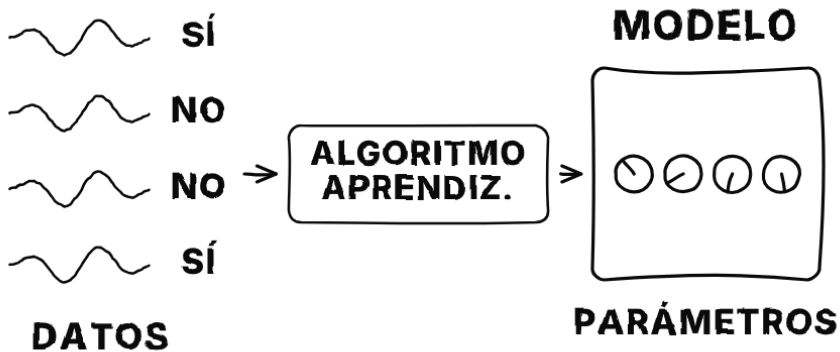
Los autores abren la sección con una pequeña escena que, admiten, se inventaron para ilustrar la idea. De camino a comprar café, uno de

ellos le dice «Hey Siri» a un teléfono, y el teléfono se despierta. Otro pregunta cómo llegar a una cafetería. El teléfono escribe lo que dijo, entiende que quiere indicaciones, abre los mapas, encuentra varias rutas y muestra junto a cada una el tiempo de viaje previsto. En esos pocos segundos trabajaron varios programas aprendidos distintos: uno despertó el teléfono, otro convirtió la voz en texto, otro averiguó qué se quería, otro estimó los tiempos de viaje. Lo mismo ocurre cuando le pide a un asistente de voz que ponga un temporizador: escuchar la frase de activación, escribir sus palabras y averiguar qué quiere decir lo hace, cada cosa, un programa aprendido.

Tomemos solo el primero de estos trabajos, la palabra de activación. El micrófono de un teléfono mide el sonido unas cuarenta y cuatro mil veces por segundo, y cada medición es un solo número: cuánto está empujada la onda sonora hacia arriba o hacia abajo en ese instante. Así que un fragmento de sonido de dos segundos son cuarenta y cuatro mil por dos, unos ochenta y ocho mil números seguidos. El programa debe mirar esos ochenta y ocho mil números y responder solo sí o no: ¿estaba ahí la palabra de activación? Nadie conoce una regla que haga esto de forma fiable. El libro lo admite con franqueza: los autores tampoco saben escribir un programa así desde cero, y precisamente por eso se usa el aprendizaje automático.

Aquí está el truco. No podemos decirle al ordenador cómo reconocer «Alexa», pero nosotros la reconocemos sin esfuerzo. Así que unas personas escuchan una enorme cantidad de grabaciones cortas y marcan cada una con sí o no. Ese montón de grabaciones con sus marcas se llama conjunto de datos, y cada marca es una etiqueta. Luego, en lugar de escribir un reconocedor, escribimos un programa flexible cuyo comportamiento depende de muchos números ajustables, llamados parámetros. El conjunto de datos se usa para

encontrar los valores de los parámetros con los que el programa funciona mejor según alguna medida de rendimiento elegida.



Etiquetas, perillas y un aprendiz

El libro le pide que imagine los parámetros como perillas. Gire las perillas en un sentido y el programa se comporta de una manera; gírelas en otro y se comporta de otra. Una vez fijadas las perillas y sin volver a tocarlas, el programa se llama modelo. Todos los programas que se pueden obtener girando las perillas de un mismo diseño forman una familia de modelos. Y el programa que usa los datos para elegir la posición de las perillas se llama algoritmo de aprendizaje, un programa cuyo resultado es otro programa.

Antes de poner nada en marcha hay que precisar el trabajo: qué entra, qué sale y qué familia de modelos usar. Aquí la entrada es un fragmento de audio y la salida es sí o no. Si la familia está bien elegida, una posición de sus perillas hace que diga sí siempre que oiga «Alexa». Y como la elección de la palabra de activación es arbitraria, una familia lo bastante rica debería tener también otra posición que diga sí solo con «Apricot». Las dos tareas se parecen tanto que la misma familia debería poder con ambas. Pero convertir fotografías en pies de foto, o frases inglesas en chinas, implica entradas y salidas muy distintas, y puede necesitar una familia completamente diferente.

Pruébelo

Pida a alguien que susurre una palabra en una habitación ruidosa, luego que la murmure y luego que la grite desde la habitación de al lado. Usted la reconocerá cada vez, pero si intenta describir con palabras qué hace que el sonido sea «Alexa» y no «Alexis», verá que no hay nada que pueda escribir. Esa distancia entre hacer algo y poder enunciar la regla es la razón de que exista el aprendizaje a partir de ejemplos.

Con las perillas puestas al azar, el modelo no reconoce nada en absoluto. Aprender significa buscar la posición de las perillas con la que el modelo se comporta como queremos, y hacerlo con datos se llama entrenamiento. El libro dibuja el entrenamiento como un ciclo. Se empieza con un modelo cuyas perillas están al azar y que no hace nada útil. Se toma un puñado de ejemplos, grabaciones con sus etiquetas de sí o no. Se retocan las perillas para que el modelo lo haga

un poco mejor con esos ejemplos. Luego se toman más ejemplos y se retoca de nuevo, vuelta tras vuelta, hasta que el modelo es muy bueno.

Así que, en lugar de escribir un reconocedor de la palabra de activación, escribimos un programa que aprende a convertirse en uno cuando se le muestran suficientes ejemplos etiquetados. Los autores lo llaman programar con datos. Del mismo modo se podría «programar» un detector que distinga gatos de perros simplemente mostrándole muchas imágenes de cada uno. Al final da un número positivo grande cuando ve un gato, un número negativo grande cuando ve un perro y un número cercano a cero cuando no está seguro. Su filtro de spam funciona igual: nadie enumeró cada frase de spam; se le mostraron muchos mensajes marcados como spam o no spam, y sus perillas se asentaron donde los separan. El libro añade que el aprendizaje profundo, que viene más adelante, es solo una de muchas formas populares de hacerlo.

Para recordar

Incluso unos pocos segundos con un teléfono pueden implicar varios programas aprendidos, desde despertarse con una frase hasta estimar el tiempo de viaje. Una palabra de activación llega como decenas de miles de números por segundo, y nadie conoce una regla que los convierta de forma fiable en sí o no. Nosotros sí reconocemos la palabra, así que etiquetamos muchos ejemplos y dejamos que un programa flexible con perillas ajustables, sus parámetros, aprenda de ellos. Un modelo es el programa con sus perillas fijadas; un algoritmo de aprendizaje elige las posiciones; el entrenamiento repite pequeños ajustes con ejemplos hasta que el modelo funciona bien. Esto es programar con datos, y el aprendizaje profundo es una forma de hacerlo.

Preguntas

1. ¿Por qué las personas etiquetan las grabaciones con sí o no antes de entrenar un detector de palabra de activación?
 - a. Porque el ordenador no puede guardar sonido sin etiquetar
 - b. Porque nosotros sí reconocemos la palabra aunque no sepamos escribir la regla
 - c. Porque las etiquetas acortan cada grabación
 - d. Porque el micrófono solo graba sonido etiquetado
2. ¿Qué diferencia hay entre un modelo y una familia de modelos?
 - a. Una familia es un modelo entrenado con más datos

- b. Un modelo es el programa con sus perillas fijadas; una familia son todos los programas en que puede convertirse un diseño al girar sus perillas
 - c. Un modelo responde sí o no; una familia responde con números
 - d. Una familia es un grupo de conjuntos de datos usados a la vez
3. ¿Por qué una misma familia de modelos debería poder con «Alexa» y con «Apricot», pero quizá no con poner pies de foto a imágenes?
- a. Porque los pies de foto necesitan más perillas de las que puede tener cualquier familia
 - b. Porque las dos palabras de activación son tareas parecidas con el mismo tipo de entrada y salida, mientras que los pies de foto tienen entradas y salidas muy distintas
 - c. Porque «Apricot» se oye mejor que «Alexa»
 - d. Porque las fotos no se pueden convertir en números
4. En el ciclo de entrenamiento, ¿qué les pasa a las perillas en cada vuelta?
- a. Todas reciben nuevos valores al azar
 - b. Se retocan para que el modelo lo haga un poco mejor con los ejemplos que se acaban de mostrar
 - c. Las fija un programador que escucha los ejemplos
 - d. No se tocan mientras se reúnen más datos

Respuestas

1. **b)** No podemos decirle al ordenador cómo reconocer la palabra, pero nosotros sí la reconocemos, así que marcamos muchas grabaciones y dejamos que el programa aprenda de ellas.
2. **b)** Una vez fijadas las perillas, el programa se llama modelo; todos los programas que se pueden obtener girando las perillas de un mismo diseño forman una familia.
3. **b)** Los dos trabajos de palabra de activación se parecen, así que basta otra posición de las mismas perillas, mientras que pasar de fotos a pies de foto implica entradas y salidas muy distintas.
4. **b)** Se toma un puñado de ejemplos etiquetados y se retocan las perillas para que el modelo lo haga un poco mejor con ellos, y luego se repite.

Ejemplos resueltos

Cuántos números contiene un fragmento Un micrófono mide el sonido unas 44.000 veces por segundo, y cada medición es un número. ¿Cuántos números debe mirar un detector de palabra de activación en un fragmento de tres segundos?

Cada segundo da unas 44.000 mediciones, cada una un número.

Tres segundos dan tres veces más.

44.000 por 3 son 132.000.

Todos esos números deben convertirse en un único sí o no, y por eso ninguna regla sencilla funciona.

Respuesta: unos 132.000 números

Ejercicios

1. Ordene los cuatro pasos del ciclo de entrenamiento y diga por qué el primer modelo es inútil: retocar las perillas; empezar con perillas al

azar; repetir hasta que funcione bien; tomar algunos ejemplos etiquetados.

Pista: ¿Qué puede hacer un modelo antes de haber visto ningún dato?

Respuesta: Empezar con perillas al azar, tomar algunos ejemplos etiquetados, retocar las perillas para que el modelo lo haga mejor con ellos y repetir hasta que funcione bien. El primer modelo es inútil porque sus perillas están al azar y todavía no ha aprendido nada.

2. Un detector de gatos y perros da un número para cada imagen. Da 9 para una imagen, menos 8 para otra y 0,3 para una tercera. ¿Qué cree que muestra cada imagen?

Pista: Positivo grande significa gato, negativo grande significa perro, cerca de cero significa que no está seguro.

Respuesta: La primera es un gato (positivo grande), la segunda un perro (negativo grande), y con la tercera no está seguro, porque el número está cerca de cero.

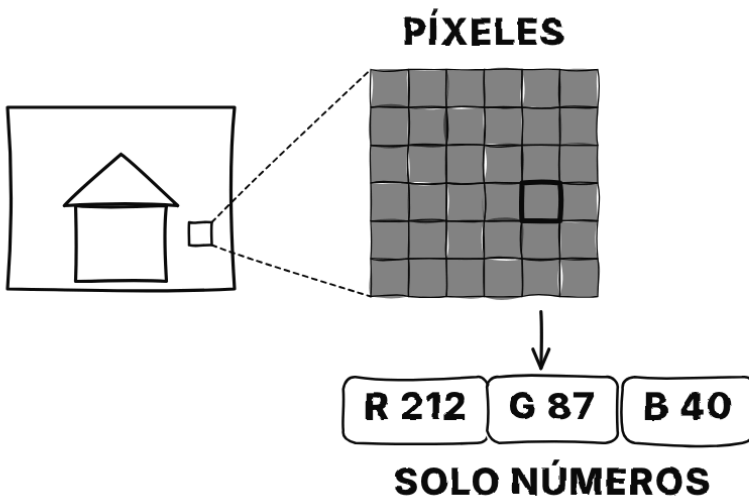
3. Explique con sus palabras por qué los autores llaman al entrenamiento «programar con datos».

Pista: ¿Quién decide el comportamiento del programa, una persona que escribe reglas o algo distinto?

Respuesta: El comportamiento del programa no lo escribe una persona regla a regla; se fija mostrándole al programa ejemplos etiquetados y dejando que el algoritmo de aprendizaje elija la posición de las perillas. Los datos ocupan el lugar de las reglas escritas a mano.

De qué aprende la máquina

Haga una foto con su teléfono y ampliela todo lo que se pueda. En algún momento la imagen se deshace en cuadraditos de colores, los píxeles. Cada cuadradito es en realidad tres números: cuán brillante es su luz roja, cuán brillante la verde y cuán brillante la azul. Para un programa que aprende, eso es todo lo que es su foto: una larga lista de números.

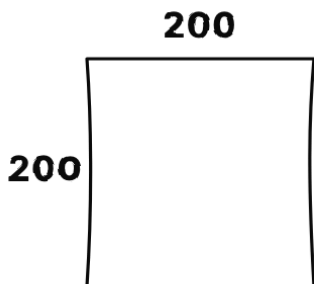


Una foto no es más que números

El detector de palabra de activación de la parte anterior aprendió de grabaciones marcadas con sí o no. Ese tipo de tarea tiene nombre:

aprendizaje supervisado. Tenemos muchos ejemplos cuyas respuestas correctas conocemos, y queremos un programa que prediga la respuesta correcta para ejemplos nuevos en los que no se conoce. Es solo uno de muchos tipos de problema de aprendizaje automático, pero el libro nombra primero cuatro piezas que aparecen en todos. Están los datos de los que aprender. Hay un modelo, que convierte los datos en una respuesta. Hay una función objetivo, una puntuación que dice lo bien o lo mal que lo está haciendo el modelo. Y hay un algoritmo que ajusta los parámetros del modelo, las perillas, para que esa puntuación sea lo mejor posible. Esta parte trata de la primera de las cuatro, los datos.

Los datos son una colección de ejemplos, y para que sirvan cada ejemplo tiene que convertirse en números. Lo que el modelo puede ver de un ejemplo se llama sus características. En el aprendizaje supervisado hay además una cosa especial que el modelo debe predecir y que no se le muestra, llamada etiqueta. En un clasificador de fotos, la foto son las características y un número que nombra su categoría es la etiqueta. El libro cuenta en qué se convierte una foto. Una imagen en color de doscientos píxeles de ancho y doscientos de alto tiene doscientos por doscientos píxeles, que son cuarenta mil. Cada píxel lleva tres números, para el rojo, el verde y el azul, así que cuarenta mil por tres dan ciento veinte mil números para una sola imagen pequeña.

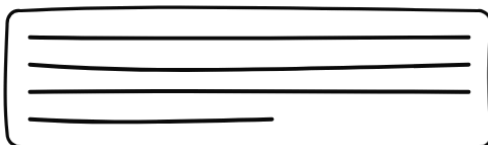


$$200 \times 200 = 40,000$$

$$40,000 \times 3 = 120,000$$

¡MALÍSIMO!

CORTA



LARGA

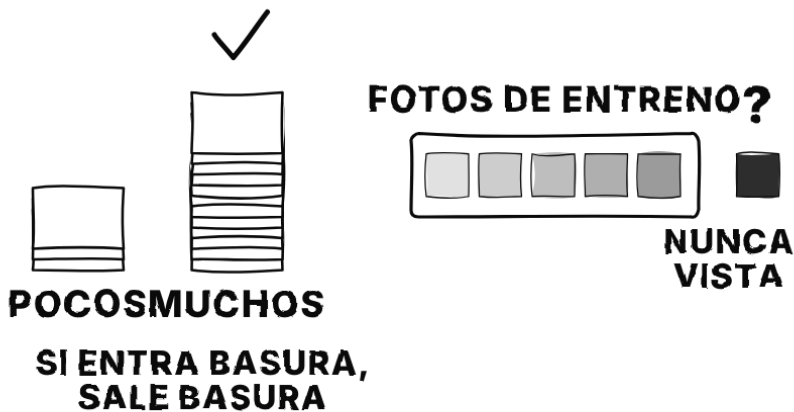
$$200 \times 200 \times 3 = 120.000$$

El segundo ejemplo del libro viene de un hospital. La tarea es predecir si un paciente sobrevivirá los próximos treinta días. Las características son cosas que ya están en el historial del paciente: edad, pulso y tensión arterial, otras enfermedades, medicamentos actuales, operaciones recientes. La etiqueta, para el entrenamiento, es simplemente si cada paciente anterior sobrevivió esos treinta días, un sí o un no. El mismo esquema está debajo de su filtro de spam: las características describen cada mensaje, y la etiqueta es spam o no spam.

Cuando cada ejemplo se describe con el mismo número de características, los ejemplos se llaman de longitud fija, y ese número es la dimensionalidad de los datos. Para las fotos pequeñas es ciento

veinte mil; para los historiales de pacientes, el número de datos que contenga cada historial. La longitud fija facilita la vida, pero muchos datos se niegan a encajar. Las imágenes de un microscopio pueden salir todas del mismo tamaño, pero las imágenes recogidas de internet vienen de todas las formas y tamaños. Se pueden recortar a un tamaño estándar, pero lo que se corta se pierde. El texto es aún más terco. Una reseña en una web de compras puede tener dos palabras, «¡es malísimo!», o extenderse durante páginas. El libro cuenta como una gran ventaja del aprendizaje profundo que sus modelos se las arreglan relativamente bien con ejemplos de longitud variable. El chatbot al que usted escribe es un buen ejemplo: acepta su pregunta tenga tres palabras o tres párrafos.

Más datos suelen facilitar el trabajo. Con más ejemplos se pueden entrenar modelos más potentes y apoyarse menos en suposiciones hechas de antemano. El paso de conjuntos de datos bastante pequeños a otros muy grandes es, dice el libro, una de las razones principales del éxito del aprendizaje profundo. Muchos de sus modelos más emocionantes no funcionan en absoluto sin enormes conjuntos de datos, y algunos de los que sí funcionan con pocos no son mejores que métodos más antiguos.



Más datos, los datos adecuados

Pero la cantidad no basta; los datos tienen que ser los adecuados. Si están llenos de errores, o las características no tienen nada que ver con lo que queremos predecir, el aprendizaje fracasará. El libro cita el viejo dicho «si entra basura, sale basura». Y en algunos usos el daño va más allá de una respuesta errónea: al predecir delitos, filtrar solicitudes de empleo o decidir quién recibe un préstamo, los datos malos pueden perjudicar a las personas. Un fallo frecuente es que algún grupo de personas simplemente falta en los ejemplos. Imagine un sistema para detectar el cáncer de piel al que nunca se le mostró piel oscura durante el entrenamiento; no ha aprendido nada sobre las personas con las que fallará. Otro fallo es más sutil: los datos sí incluyen a todos, pero arrastran los prejuicios del pasado. Si un modelo se entrena con las decisiones de contratación pasadas de una

empresa y luego se usa para clasificar solicitudes nuevas, puede copiar y automatizar en silencio viejas injusticias. Los autores subrayan que esto puede ocurrir sin que nadie lo pretenda, ni siquiera lo note.

Para recordar

Aprendizaje supervisado significa predecir una etiqueta desconocida a partir de características conocidas, después de aprender de ejemplos cuyas etiquetas se conocen. Todo problema de aprendizaje automático tiene cuatro piezas: datos, un modelo, una función objetivo que puntúa el modelo y un algoritmo que ajusta sus parámetros. Los datos tienen que convertirse en números: una pequeña foto en color de doscientos por doscientos píxeles ya son ciento veinte mil. Los ejemplos de longitud fija son fáciles, pero el texto y las imágenes de internet varían de tamaño, y el aprendizaje profundo maneja relativamente bien esos datos de longitud variable. Más datos ayudan, pero solo los datos adecuados: los grupos ausentes o los prejuicios pasados en los ejemplos se convierten en fallos e injusticias del modelo, aunque nadie lo pretenda.

Preguntas

1. En el ejemplo del hospital, ¿cuál es la etiqueta?
 - a. La edad y el pulso del paciente
 - b. La lista de medicamentos actuales
 - c. Si el paciente sobrevivió los treinta días siguientes

- d. El número de operaciones recientes
2. ¿Por qué el texto es más difícil de manejar que las fotos de un mismo microscopio?
- a. Porque el texto no se puede convertir en números
 - b. Porque los textos tienen longitudes muy distintas, así que no encajan en un número fijo de características
 - c. Porque el texto no tiene etiquetas
 - d. Porque las reseñas siempre son demasiado cortas para aprender de ellas
3. Un detector de cáncer de piel funciona bien en las pruebas pero falla con pacientes de piel oscura. ¿Qué salió mal, según la lección?
- a. El modelo tenía demasiados parámetros
 - b. La función objetivo era la puntuación equivocada
 - c. La piel oscura faltaba en los ejemplos de entrenamiento
 - d. Las imágenes no se recortaron a un tamaño estándar
4. ¿Por qué un modelo entrenado con las decisiones de contratación pasadas de una empresa puede ser injusto aunque nadie lo pretenda?
- a. Porque los datos arrastran prejuicios pasados, y el modelo los aprende y los repite
 - b. Porque los datos de contratación siempre son demasiado pocos
 - c. Porque las solicitudes de empleo son texto de longitud variable
 - d. Porque nunca se le dieron etiquetas al modelo

Respuestas

1. **c)** Las características son lo que ya está en el historial; la etiqueta es si cada paciente anterior sobrevivió los treinta días.
2. **b)** Una reseña puede tener dos palabras o extenderse durante páginas, así que el texto se resiste a una longitud fija aún más que las imágenes de internet.
3. **c)** Un fallo frecuente es que algún grupo de personas simplemente falta en los ejemplos, como en un sistema al que nunca se le mostró piel oscura.
4. **a)** Si los datos arrastran los prejuicios del pasado, un modelo entrenado con ellos puede copiar y automatizar en silencio viejas injusticias.

Ejemplos resueltos

Contar los números de una foto Una foto en color mide 200 píxeles de ancho y 200 de alto, y cada píxel se guarda como tres números, para el rojo, el verde y el azul. ¿Cuántos números describen la foto?

Cuente los píxeles: 200 a lo ancho por 200 a lo alto son 40.000 píxeles.

Cada píxel tiene tres números de brillo, uno por color.

Así que multiplique los píxeles por tres: 40.000 por 3 son 120.000.

Toda foto de este tamaño tiene la misma cuenta, así que su dimensionalidad es 120.000.

Respuesta: 120.000 números

Ejercicios

1. Una foto en color más pequeña mide 100 píxeles de ancho y 100 de alto. ¿Cuántos números la describen, y cómo se compara con la foto de 200 por 200?

Pista: Cuente primero los píxeles y recuerde luego que cada píxel tiene tres números.

Respuesta: 100 por 100 son 10.000 píxeles, por 3 son 30.000 números. Reducir a la mitad el ancho y el alto da una cuarta parte de los píxeles, es decir, una cuarta parte de los 120.000 números.

2. Explique por qué recortar las fotos de internet a un tamaño estándar solo arregla en parte el problema de los datos de longitud variable.

Pista: ¿Qué pasa con las partes de la imagen que quedan fuera del recorte?

Respuesta: Recortar hace que todos los ejemplos tengan la misma longitud, pero lo que se corta se pierde, y puede contener justo lo que importa para la predicción. Y no ayuda nada con el texto, cuya longitud varía todavía más.

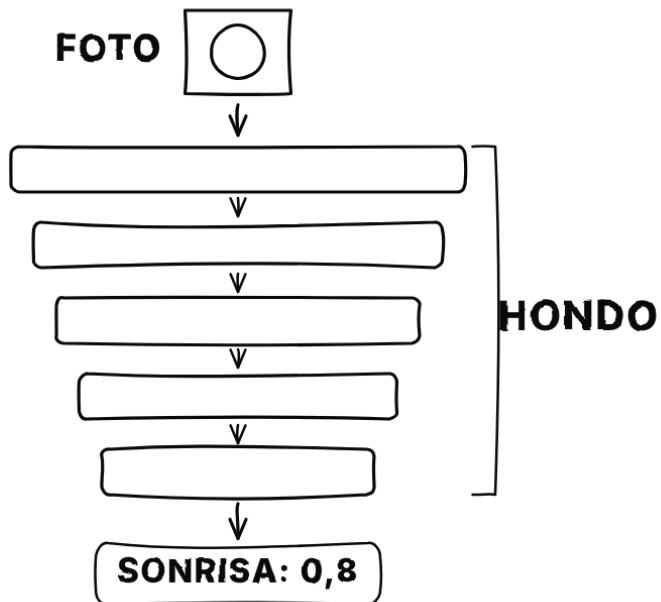
3. Nombre las cuatro piezas que tiene todo problema de aprendizaje automático y diga en una frase qué hace cada una.

Pista: Una es de lo que aprendemos, otra lo convierte en respuestas, otra puntúa, otra ajusta.

Respuesta: Los datos son de lo que aprende la máquina; el modelo convierte los datos en respuestas; la función objetivo puntúa lo bien o lo mal que lo está haciendo el modelo; y el algoritmo ajusta los parámetros del modelo para que esa puntuación sea lo mejor posible.

Puntuar a la máquina

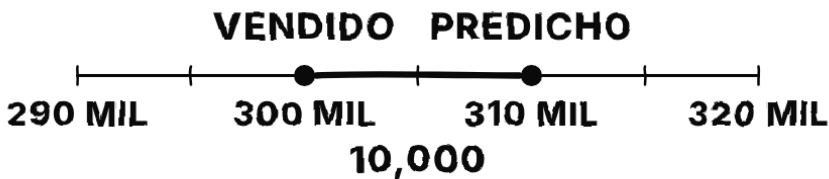
Piense en una estudiante que repasa para un examen final con los exámenes de práctica del año pasado. Al cabo de una semana saca la nota máxima en todos ellos. Luego, en el examen de verdad, ante preguntas que nunca ha visto, le cuesta. No había aprendido la asignatura; había aprendido las respuestas de esos exámenes concretos. Téngala presente, porque las máquinas que aprenden pueden fallar exactamente de la misma manera.



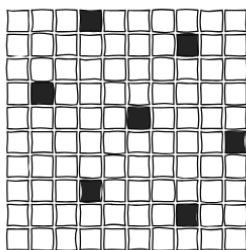
Profundo significa muchos pasos

Primero, ¿qué es un modelo? La mayor parte del aprendizaje automático toma datos de un tipo y los convierte en una predicción, a menudo de otro tipo. Los ejemplos del libro son un sistema que recibe fotos y dice cuánto sonríe la persona, y otro que recibe lecturas de sensores y dice lo normales o lo raras que son. El modelo es la maquinaria que hace esa conversión, y los modelos que interesan al libro son estadísticos, con perillas que se fijan a partir de los datos. Los modelos simples sirven para problemas simples, pero los problemas de este libro van más allá de lo que pueden hacer los métodos clásicos. Lo que distingue al aprendizaje profundo es su familia de modelos potentes contruidos como muchas transformaciones de los datos, una tras otra, encadenadas de arriba abajo. Esa pila de pasos es la razón de que se llame profundo. Por el camino, el libro también verá algunos métodos más antiguos y sencillos.

Describimos el aprendizaje como mejorar en una tarea con la experiencia. Pero ¿quién decide qué cuenta como mejor? Una persona puede pensar que un cambio en el modelo es una mejora, y otra no. Para construir máquinas que aprendan necesitamos un número preciso que diga lo bueno o lo malo que es un modelo. Ese número se llama función objetivo. Por costumbre se plantea de modo que menos sea mejor, y entonces también se llama función de pérdida, una medida de cuánto se pierde. Es solo una convención: cualquier puntuación en la que más es mejor se puede convertir en una en la que menos es mejor simplemente poniéndole un signo menos delante.



$$10,000^2 = 100,000,000$$



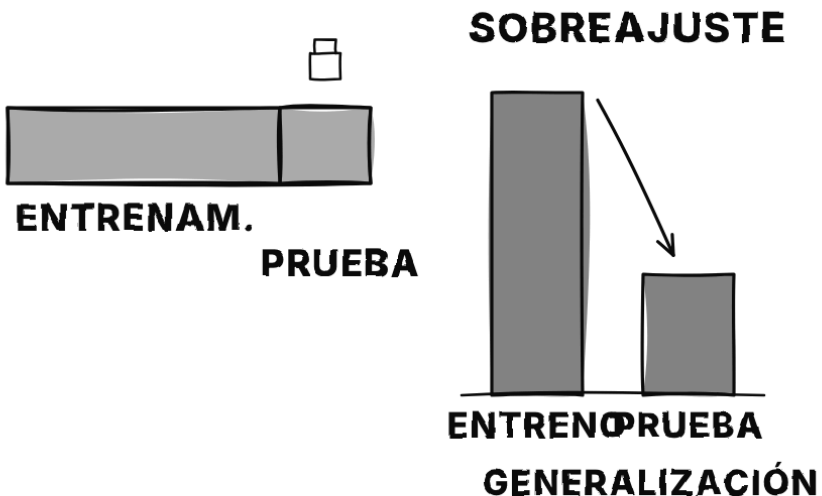
7 MAL = 7 %

Cuánto se pierde

El libro nombra las dos pérdidas más comunes. Cuando el modelo predice un número, la pérdida habitual es el error cuadrático: se toma la diferencia entre la predicción y el valor verdadero y se multiplica por sí misma. Supongamos que un modelo predice que un piso se venderá por trescientos diez mil, y se vende por trescientos mil. La diferencia es diez mil; diez mil por diez mil son cien millones, y esa es la pérdida para ese único ejemplo. Cuando el modelo clasifica cosas en tipos, la medida habitual es la tasa de error, la proporción de ejemplos en que se equivoca: siete errores de cada cien es una tasa de error del siete por ciento. El error cuadrático es fácil de optimizar, pero la tasa de error es difícil de mejorar directamente, así que en la práctica a menudo se minimiza una puntuación sustituta, llamada objetivo sustituto, que tiende a moverse en el mismo sentido. A un

filtro de spam se le juzga al final por la proporción de correo que clasifica mal, aunque se entrene con una puntuación sustituta más suave.

Mientras se entrena el modelo, los ejemplos se mantienen fijos y la pérdida se trata como si dependiera solo de las perillas. Entrenar significa encontrar las posiciones de las perillas que hacen la pérdida en los ejemplos de entrenamiento lo más pequeña posible. Pero hacerlo bien con esos ejemplos no garantiza hacerlo bien con ejemplos nuevos. Así que los datos suelen dividirse en dos montones. El conjunto de entrenamiento se usa para fijar las perillas; el conjunto de prueba se guarda aparte, sin que el modelo lo vea, y solo se usa para comprobar el resultado. Se informa de los resultados en ambos.



Conjunto de entrenamiento y de prueba

Aquí vuelve la estudiante. La puntuación en el conjunto de entrenamiento es su nota en los exámenes de práctica: alentadora, pero no una promesa. Si el modelo, en la práctica, ha memorizado sus ejemplos de entrenamiento, brilla con ellos y tropieza con cualquier cosa nueva. Hacerlo bien con ejemplos que nunca ha visto se llama generalización. Hacerlo bien en el conjunto de entrenamiento pero no generalizar se llama sobreajuste. Cuando su sección de recomendaciones parece conocer a la perfección sus clics pasados pero no le sugiere nada que de verdad quiera después, ese es el tipo de fallo que un conjunto de prueba debe detectar.

Para recordar

Un modelo convierte datos de un tipo en predicciones, y los modelos del aprendizaje profundo son largas cadenas de transformaciones, por eso se llaman profundos. Para decir si un modelo mejoró necesitamos un número, la función objetivo, normalmente planteada de modo que menos sea mejor y llamada entonces pérdida. Para predecir números, la pérdida común es la diferencia al cuadrado respecto a la verdad; para clasificar en tipos, es la proporción de errores, a menudo entrenada a través de un sustituto más suave. Las perillas se ajustan con un conjunto de entrenamiento y se comprueban con un conjunto de prueba aparte, porque el éxito con ejemplos conocidos no es éxito con los nuevos. Un modelo que brilla con sus ejemplos de entrenamiento pero falla con los nuevos está sobreajustado, como una estudiante que memorizó los exámenes de práctica.

Preguntas

1. ¿Por qué el aprendizaje profundo se llama profundo?
 - a. Porque sus modelos necesitan conjuntos de datos muy grandes
 - b. Porque sus modelos son muchas transformaciones de los datos encadenadas una tras otra
 - c. Porque estudia propiedades ocultas de los datos
 - d. Porque sus pérdidas son muy pequeñas
2. Una puntuación en la que más es mejor se convierte en una pérdida. ¿Cómo?
 - a. Elevándola al cuadrado
 - b. Dividiéndola por el número de ejemplos
 - c. Poniéndole un signo menos delante
 - d. Midiéndola en el conjunto de prueba
3. ¿Por qué se usa a menudo un objetivo sustituto en lugar de la tasa de error?
 - a. Porque la tasa de error es difícil de mejorar directamente, mientras que un sustituto más suave tiende a moverse en el mismo sentido
 - b. Porque la tasa de error siempre es cero con los datos de entrenamiento
 - c. Porque la tasa de error solo sirve para números
 - d. Porque el sustituto se mide en el conjunto de prueba

4. Un modelo puntúa muy bien en su conjunto de entrenamiento pero mal en el de prueba. ¿Cómo se llama esto?
- a. Generalización
 - b. Un objetivo sustituto
 - c. Una tasa de error alta en el entrenamiento
 - d. Sobreajuste

Respuestas

1. **b)** Los modelos del aprendizaje profundo se construyen como muchas transformaciones de los datos encadenadas de arriba abajo, y esa pila de pasos es la razón de que se llame profundo.
2. **c)** Cualquier puntuación en la que más es mejor se puede convertir en una en la que menos es mejor simplemente poniéndole un signo menos delante.
3. **a)** La tasa de error es difícil de mejorar directamente, así que a menudo se minimiza una puntuación sustituta que tiende a moverse en el mismo sentido.
4. **d)** Hacerlo bien en el conjunto de entrenamiento pero no generalizar a datos no vistos se llama sobreajuste.

Ejemplos resueltos

Un error cuadrático Un modelo predice que un piso se venderá por 310.000, y en realidad se vende por 300.000. ¿Cuál es el error cuadrático de este ejemplo?

Halle la diferencia entre la predicción y la verdad: 310.000 menos 300.000 son 10.000.

Elévela al cuadrado, es decir, multiplíquela por sí misma: 10.000 por 10.000.

Eso da 100.000.000.

La pérdida es grande porque elevar al cuadrado hace que los fallos grandes cuenten mucho más que los pequeños.

Respuesta: 100.000.000 (cien millones)

Ejercicios

1. Un modelo que clasifica fotos en gatos y perros se equivoca en 14 de 200 fotos de prueba. ¿Cuál es su tasa de error?

Pista: La tasa de error es la proporción de ejemplos en que se equivoca.

Respuesta: 14 entre 200 es 0,07, una tasa de error del siete por ciento.

2. Un modelo predice 8 cuando la verdad es 5, y en otro ejemplo 2 cuando la verdad es 3. ¿Qué ejemplo aporta más a la pérdida de error cuadrático, y cuánto aporta cada uno?

Pista: Tome cada diferencia y multiplíquela por sí misma.

Respuesta: El primero falla por 3, así que su error cuadrático es 9; el segundo falla por 1, así que su error cuadrático es 1. El primero aporta nueve veces más.

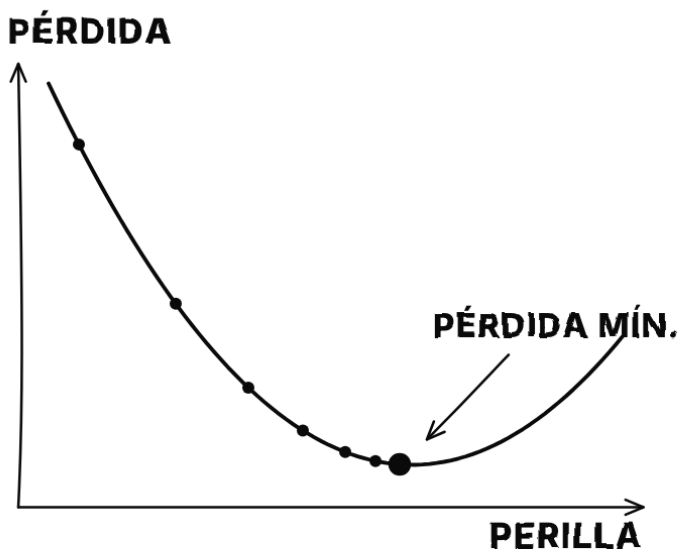
3. Explique con la estudiante y los exámenes de práctica por qué mantenemos un conjunto de prueba aparte del de entrenamiento.

Pista: ¿Qué le dice una nota perfecta en exámenes de práctica memorizados sobre el examen real?

Respuesta: Un modelo ajustado con el conjunto de entrenamiento puede limitarse a memorizarlo, igual que la estudiante memorizó los exámenes de práctica. Solo los ejemplos que nunca ha visto, el conjunto de prueba, muestran si ha aprendido de verdad algo que se traslada, es decir, si generaliza.

Tanteando el camino hacia abajo

Póngase en una ladera en medio de una niebla espesa, queriendo llegar al fondo del valle. No ve más allá de un paso, así que no ve dónde está el valle. Pero sí nota la pendiente bajo sus pies: hacia un lado el terreno baja, hacia el otro sube. Dé un pequeño paso hacia donde baja, vuelva a tantear y dé otro paso. Tarde o temprano estará en el fondo.

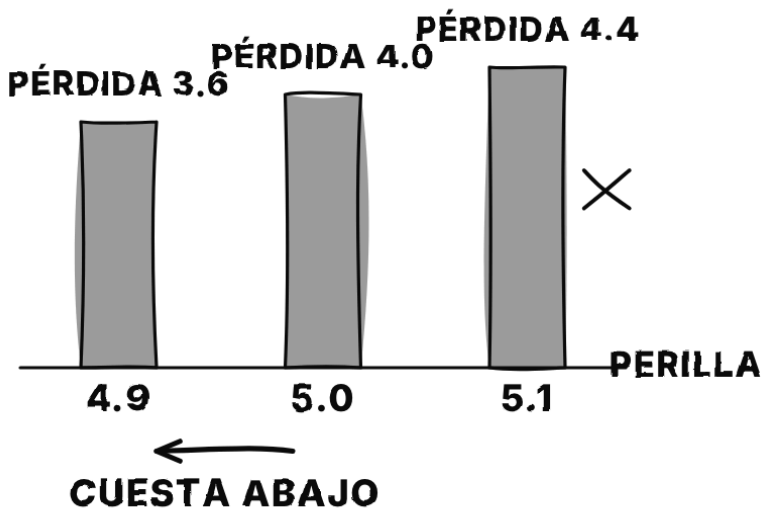


Bajar por la pérdida

Esa es la idea del método que el libro nombra ahora, la última de las cuatro piezas. Tenemos datos, un modelo con perillas y una pérdida

que dice lo mal que lo hace; ahora necesitamos una forma de encontrar las posiciones de las perillas que hacen la pérdida lo más pequeña posible. La mayoría de los métodos usados en aprendizaje profundo se basan en el descenso de gradiente. En cada paso se pregunta, para cada perilla por turno, cómo cambiaría la pérdida en el conjunto de entrenamiento si esa perilla se girara solo un poquito. Luego gira cada perilla un poco en la dirección que baja la pérdida. La pérdida es la altura de la ladera, y las posiciones de las perillas son el lugar donde usted está.

Aquí va un paso con números, para un modelo con una sola perilla. Digamos que la perilla está en cinco, y la pérdida ahí es cuatro. Gírela un poquito hacia arriba, a cinco coma uno, y la pérdida sube a unos cuatro coma cuatro; así que hacia arriba es el sentido equivocado. Gírela en cambio un poquito hacia abajo, a cuatro coma nueve, y la pérdida baja a unos tres coma seis. Así que el método mueve la perilla hacia abajo y luego vuelve a comprobar la pendiente desde su nuevo sitio. Un modelo real de aprendizaje profundo tiene millones de perillas, pero cada paso las trata a todas de la misma manera.



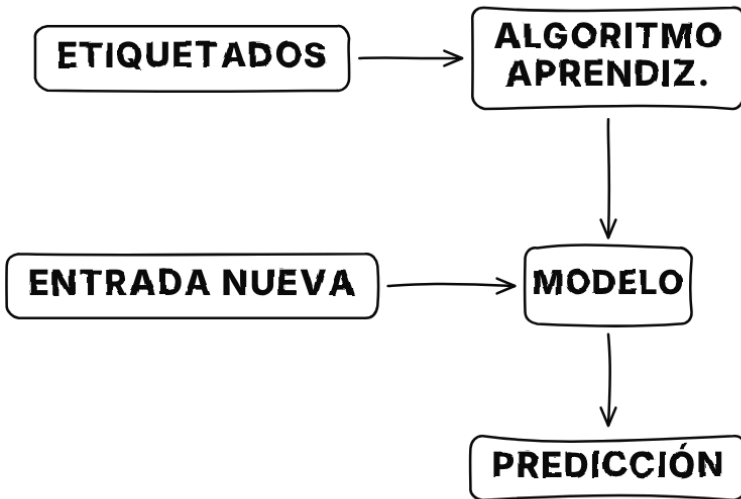
Un paso con números

Con las cuatro piezas en su sitio, el libro pasa a los tipos de problema que puede abordar el aprendizaje automático, empezando por el más común, el aprendizaje supervisado. Aquí se nos dan ejemplos cuyas características se conocen y cuyas etiquetas también, y el trabajo es construir un modelo que prediga la etiqueta a partir de las características. Cada par de características y etiqueta es un ejemplo. Se llama supervisado porque nosotros, los supervisores, le damos al modelo ejemplos etiquetados con los que elegir sus perillas. En el lenguaje de la probabilidad, lo que queremos es cuán probable es cada etiqueta, dadas las características que vemos.

El aprendizaje supervisado es solo uno de varios enfoques, pero supone la mayoría de los usos exitosos del aprendizaje automático en

la industria. Muchas tareas importantes encajan en su esquema sencillo: adivinar algo desconocido a partir de los datos disponibles. ¿Hay cáncer o no, dada una imagen de un escáner médico? ¿Cuál es la traducción francesa correcta de esta frase inglesa? ¿Cuál será el precio de una acción el mes que viene, dados los informes financieros de este mes? Su filtro de spam sigue el mismo esquema: dado un mensaje, predecir spam o no spam. El esquema es sencillo, pero las formas varían muchísimo; las secuencias de cualquier longitud, por ejemplo, necesitan modelos distintos de las entradas de tamaño fijo.

El libro esboza el proceso y lo dibuja: los ejemplos etiquetados entran en un método de aprendizaje, sale un modelo, y las entradas nuevas entran en ese modelo y salen como predicciones. Primero se reúne una gran colección de ejemplos cuyas características se conocen, se eligen algunos al azar y se obtiene la etiqueta verdadera de cada uno. A veces las etiquetas ya están registradas, como si un paciente murió en el año siguiente; otras veces hay que pagar a personas para que etiqueten los datos a mano, por ejemplo clasificando imágenes en categorías. Estos ejemplos con sus etiquetas forman el conjunto de entrenamiento. El conjunto de entrenamiento entra en el algoritmo de aprendizaje, que es a su vez una función cuyo resultado es otra función, el modelo entrenado. Por último, se introducen en el modelo entradas que nunca ha visto, y sus salidas se toman como predicciones de sus etiquetas.



De ejemplos etiquetados a predicciones

Para recordar

El descenso de gradiente es la forma de ajustar las perillas: para cada perilla, comprobar si un giro minúsculo sube o baja la pérdida, luego girarla un poco hacia donde baja, y repetir. Es como encontrar el valle en la niebla tanteando la pendiente bajo los pies. Aprendizaje supervisado significa aprender a predecir etiquetas a partir de características, usando ejemplos cuyas etiquetas se dan al modelo. La mayoría de las aplicaciones exitosas en la industria son de este tipo, desde las imágenes médicas hasta la traducción. El proceso va de los ejemplos etiquetados, a través de un algoritmo de aprendizaje, a un modelo entrenado que hace predicciones con entradas nuevas.

Preguntas

1. En la imagen del descenso de gradiente por una ladera con niebla, ¿qué representa la altura del terreno?
 - a. El número de perillas
 - b. La pérdida en el conjunto de entrenamiento
 - c. El tamaño del conjunto de datos
 - d. El número de pasos dados
2. Un giro minúsculo de una perilla hacia arriba hace subir la pérdida. ¿Qué hace a continuación el descenso de gradiente?
 - a. Gira la perilla todavía más hacia arriba
 - b. Deja la perilla quieta para siempre

- c. Gira la perilla un poco hacia abajo y luego vuelve a comprobar la pendiente
 - d. Devuelve la perilla a un valor al azar
3. ¿Por qué este tipo de aprendizaje se llama supervisado?
- a. Porque una persona vigila el modelo mientras funciona
 - b. Porque le damos al modelo ejemplos etiquetados con los que elegir sus perillas
 - c. Porque los errores del modelo se corrigen a mano después
 - d. Porque el modelo solo funciona bajo el control de un supervisor
4. ¿Qué da al final el algoritmo de aprendizaje?
- a. Un modelo entrenado, que es a su vez una función que convierte entradas nuevas en predicciones
 - b. Una lista de etiquetas para el conjunto de entrenamiento
 - c. Una copia limpia del conjunto de datos
 - d. Un solo número, la pérdida

Respuestas

1. **b)** La pérdida es la altura de la ladera, y las posiciones de las perillas son el lugar donde usted está.
2. **c)** Si hacia arriba es el sentido equivocado, el método mueve la perilla hacia abajo y luego vuelve a comprobar la pendiente desde su nuevo sitio.
3. **b)** Se llama supervisado porque nosotros, los supervisores, le damos al modelo ejemplos etiquetados con los que elegir sus perillas.
4. **a)** El algoritmo de aprendizaje es una función cuyo resultado es otra función, el modelo entrenado.

Ejemplos resueltos

Un paso cuesta abajo Un modelo tiene una sola perilla puesta en 5, y su pérdida ahí es 4. Girar la perilla a 5,1 hace que la pérdida sea de unos 4,4; girarla a 4,9 hace que sea de unos 3,6. ¿Hacia dónde mueve la perilla el descenso de gradiente?

Compare cada giro minúsculo con la pérdida actual, que es 4.

Subir a 5,1 eleva la pérdida a unos 4,4, así que hacia arriba es cuesta arriba.

Bajar a 4,9 reduce la pérdida a unos 3,6, así que hacia abajo es cuesta abajo.

El descenso de gradiente gira la perilla un poco hacia abajo y luego vuelve a comprobar la pendiente.

Respuesta: hacia abajo, hacia valores más pequeños de la perilla

Ejercicios

1. Para cada tarea, nombre las características y la etiqueta: detectar cáncer en una imagen médica; traducir una frase inglesa al francés; predecir el precio de una acción el mes que viene a partir de los informes financieros de este mes.

Pista: Las características son lo que se le muestra al modelo; la etiqueta es lo que debe predecir.

Respuesta: Imagen médica: las características son la imagen, la etiqueta es cáncer o no cáncer. Traducción: las características son la frase inglesa, la etiqueta es su traducción francesa. Acciones: las características son los informes financieros de este mes, la etiqueta es el precio de la acción el mes que viene.

2. Ordene el proceso del aprendizaje supervisado: introducir entradas nuevas en el modelo; obtener la etiqueta verdadera de cada ejemplo

elegido; reunir ejemplos y elegir algunos al azar; introducir el conjunto de entrenamiento en el algoritmo de aprendizaje.

Pista: Las etiquetas tienen que existir antes de poder aprender nada.

Respuesta: Reunir ejemplos y elegir algunos al azar; obtener la etiqueta verdadera de cada uno; introducir el conjunto de entrenamiento etiquetado en el algoritmo de aprendizaje, que da un modelo entrenado; luego introducir entradas nuevas en el modelo y usar sus salidas como predicciones.

Cuánto, cuántos

Un fontanero pasa tres horas desatascando sus desagües y le manda una factura de trescientos cincuenta dólares. Un amigo suyo contrata al mismo fontanero dos horas y paga doscientos cincuenta. Ahora un vecino pregunta cuánto costará un trabajo de cuatro horas. Antes de seguir leyendo, intente responder.

3 H → 350 \$ · 2 H → 250 \$ · 4 H → ?

SUPERFICIE DORM. BAÑOS A PIE

600

1

1

60

3000

4

3

10



VECTOR DE RASGOS

Facturas, y casas como filas

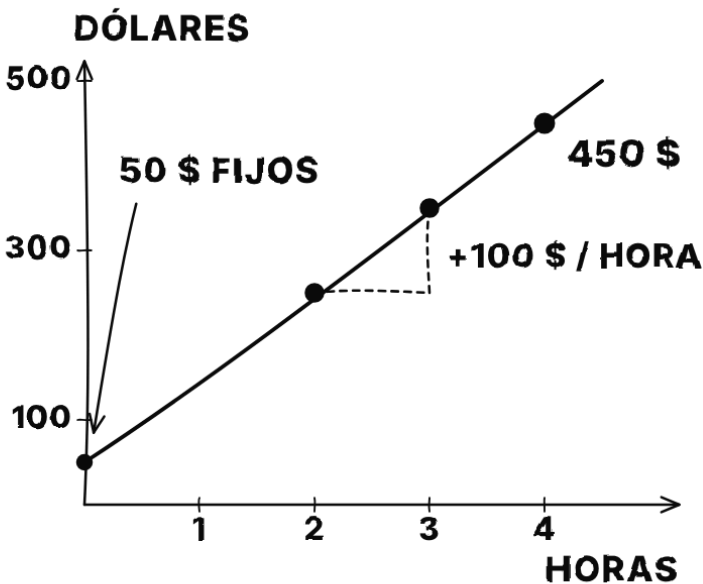
El tipo de aprendizaje supervisado más fácil de entender se llama regresión. Imagine una tabla de ventas de casas, una fila por casa y una columna por cada dato sobre ella: superficie, número de

dormitorios, número de baños y minutos a pie hasta el centro. Cada casa es un ejemplo, y su fila de datos es su vector de características, una lista fija de números en un orden fijo. El libro bromea con que en Nueva York o San Francisco, a menos que dirija una de las gigantescas empresas tecnológicas, su vivienda podría decir: seiscientos pies cuadrados, un dormitorio, un baño, sesenta minutos a pie. En Pittsburgh podría decir: tres mil pies cuadrados, cuatro dormitorios, tres baños, diez minutos a pie. La mayoría de los métodos de aprendizaje clásicos necesitan que cada ejemplo llegue como una lista así, siempre de la misma longitud.

Lo que hace que un problema sea de regresión no son las características, sino el tipo de respuesta. Si busca casa, quizá quiera estimar un precio justo a partir de datos como estos, y un modelo podría aprenderlo de anuncios pasados, con los precios a los que realmente se vendieron las casas como etiquetas. Cuando la etiqueta puede ser cualquier número, aunque sea dentro de cierto intervalo, el problema es de regresión, y el objetivo es un modelo cuyas estimaciones se acerquen a los valores verdaderos. Una regla práctica útil: una pregunta que empieza por «¿cuánto?» o «¿cuántos?» es probablemente de regresión. ¿Cuántas horas durará esta operación? ¿Cuánta lluvia caerá sobre este pueblo en las próximas seis horas? ¿Cuánto tiempo pasará este paciente en el hospital? Adivinar cuántas estrellas le daría usted a una película también es regresión, y el libro recuerda que hacerlo bien en 2009 podía hacerle ganar el premio de un millón de dólares de Netflix.

Volvamos al fontanero. Probablemente supuso dos cosas: más horas cuestan más, y la factura es un cargo fijo por presentarse más tanto por cada hora. Las dos facturas difieren en una hora y en cien dólares, así que cada hora cuesta cien dólares. Tres horas a cien son

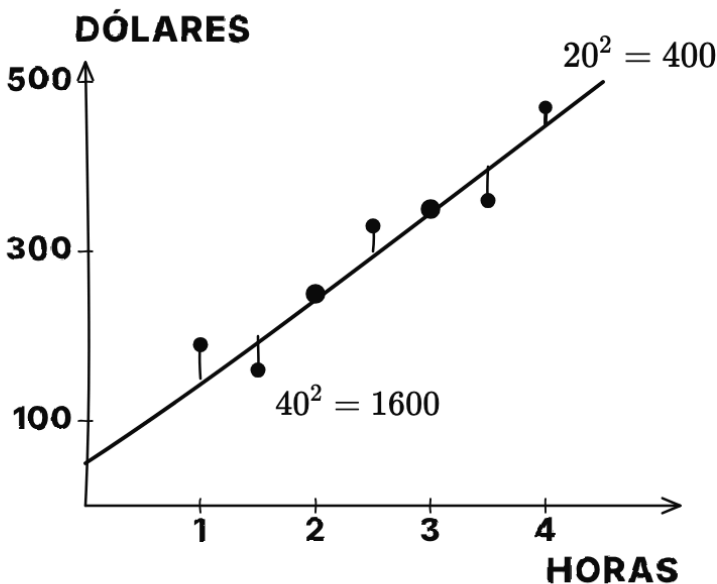
trescientos, y la factura era de trescientos cincuenta, así que presentarse cuesta cincuenta dólares. Compruébelo con la factura de su amigo: cincuenta más dos horas a cien son doscientos cincuenta, exactamente. Así que el trabajo de cuatro horas cuesta cincuenta más cuatrocientos, es decir, cuatrocientos cincuenta dólares. Si lo ha seguido, ya entiende la idea de la regresión lineal: la respuesta es una cantidad fija más una cantidad determinada por cada unidad de cada característica.



Un cargo fijo más una tarifa

Aquí dos parámetros, la tarifa por hora y el cargo por desplazamiento, coincidieron exactamente con las facturas. Los datos reales rara vez son tan ordenados, porque parte de la variación viene de cosas que no están en la tabla: un atasco rebelde, un trayecto largo, el tráfico.

Entonces ninguna posición de las perillas encaja con todos los ejemplos, y en su lugar buscamos la que mantiene las predicciones lo más cerca posible de lo que realmente se observó. El libro mide casi siempre la cercanía con el error cuadrático, el fallo multiplicado por sí mismo. Supongamos que un tercer trabajo, de cuatro horas, se facturó en cuatrocientos setenta: la regla predice cuatrocientos cincuenta, el fallo es de veinte dólares y el error cuadrático es veinte por veinte, cuatrocientos. Un fallo el doble de grande contaría cuatro veces más, así que elevar al cuadrado hace que los fallos grandes pesen mucho más que los pequeños. Los autores señalan que esta elección de pérdida es la natural cuando las variaciones sueltas se comportan como el tipo más común de ruido aleatorio, el de forma de campana.



Fallos al cuadrado

Para recordar

La regresión es aprendizaje supervisado en el que la respuesta es un número; las preguntas «¿cuánto?» y «¿cuántos?» suelen ser de regresión. Cada ejemplo es un vector de características, una lista de datos de longitud fija, como la superficie, los dormitorios, los baños y el tiempo a pie de una casa. Las dos facturas del fontanero revelan un cargo por desplazamiento de cincuenta dólares y una tarifa de cien la hora: la idea de la regresión lineal. Cuando los datos reales no encajan exactamente, elegimos los parámetros que mantienen las predicciones más cerca de la verdad, normalmente haciendo el error cuadrático lo más pequeño posible.

Preguntas

1. ¿Qué hace que un problema sea de regresión y no de otro tipo?
 - a. Que las características sean números
 - b. Que la etiqueta, la respuesta buscada, sea un número
 - c. Que los datos vengan en una tabla
 - d. Que solo haya dos características
2. ¿Por qué un modelo entrenado con facturas reales de fontaneros no suele coincidir exactamente con cada factura?
 - a. Porque cosas que no están en la tabla, como un atasco rebelde o el tráfico, también cambian las facturas
 - b. Porque hay demasiadas características

- c. Porque el error cuadrático no se puede calcular con dinero
 - d. Porque por ley cada factura tiene un cargo por desplazamiento distinto
3. Una predicción falla por 10 y otra por 20. ¿Cómo se comparan sus errores cuadráticos?
- a. El segundo cuenta el doble
 - b. Cuentan lo mismo
 - c. El segundo cuenta cuatro veces más
 - d. El segundo cuenta la mitad

Respuestas

1. **b)** Lo que hace que un problema sea de regresión no son las características sino el tipo de respuesta: cuando la etiqueta puede ser cualquier número, es regresión.
2. **a)** Parte de la variación viene de cosas que no están en la tabla, así que ninguna posición de las perillas encaja con todos los ejemplos.
3. **c)** Un fallo el doble de grande cuenta cuatro veces más, porque el fallo se multiplica por sí mismo.

Ejemplos resueltos

Los precios del fontanero Un fontanero cobró 350 dólares por un trabajo de 3 horas y 250 dólares por uno de 2 horas. Suponiendo un cargo fijo por presentarse más una tarifa por hora, halle el cargo, la tarifa y el coste de un trabajo de 4 horas.

Las facturas difieren en una hora y en 100 dólares, así que la tarifa es de 100 dólares la hora.

Tres horas suponen 300 de los 350 dólares, así que el cargo fijo es de 50 dólares.

Comprobación: 50 más 2 horas a 100 dan 250, lo facturado.

Un trabajo de 4 horas cuesta 50 más 400.

Respuesta: un cargo fijo de 50 dólares y 100 dólares la hora; un trabajo de 4 horas cuesta 450 dólares

Ejercicios

1. Un tercer trabajo de 4 horas se facturó en 470 dólares, mientras que la regla de 50 dólares más 100 la hora predice 450. Un cuarto trabajo, de 1 hora, se facturó en 140 dólares. ¿Cuál es el error cuadrático de la regla en cada trabajo?

Pista: Calcule primero lo que predice la regla para una hora y luego tome cada fallo y multiplíquelo por sí mismo.

Respuesta: En el trabajo de 4 horas el fallo es de 20 dólares, así que el error cuadrático es 400. En el de 1 hora la regla predice 150, el fallo es de 10 dólares y el error cuadrático es 100.

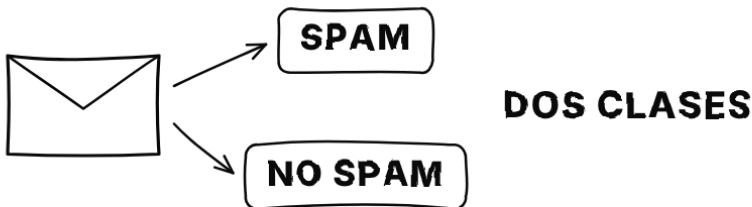
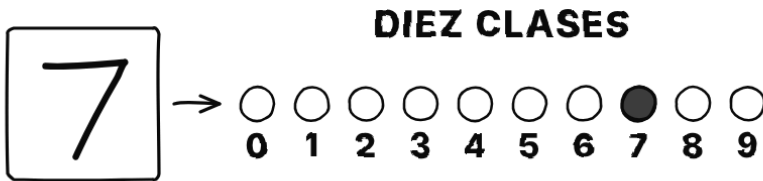
2. Escriba el vector de características de una casa de 1.200 pies cuadrados, 2 dormitorios, 1 baño y 25 minutos a pie del centro, en el orden del libro, y diga qué etiqueta predeciría para ella un modelo de regresión.

Pista: El orden del libro es superficie, dormitorios, baños, minutos a pie.

Respuesta: El vector de características es 1.200, 2, 1, 25. Un modelo de regresión entrenado con ventas pasadas predeciría un número, el precio de la casa.

Cuál, y con qué seguridad

Imagine una seta pálida y hermosa en el jardín de su casa. La fotografía con una aplicación entrenada para detectar setas venenosas, y la aplicación dice estar segura al ochenta por ciento de que la seta es inofensiva. Así que lo más probable es que sea segura. ¿La cocinaría para cenar? Tenga presente su respuesta; el libro vuelve a esta misma seta.



¿Qué clase es?

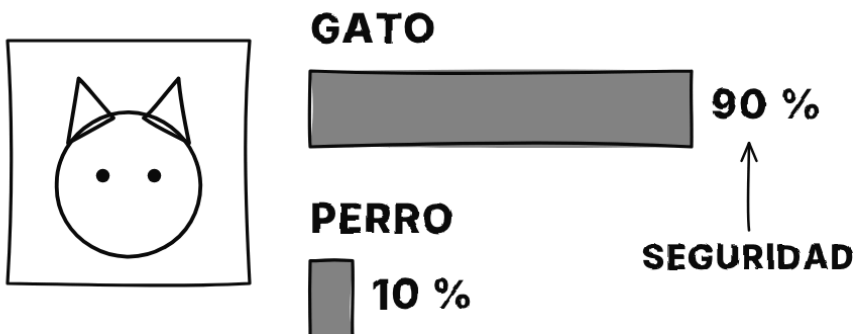
La regresión responde a «¿cuánto?» y «¿cuántos?», pero muchas preguntas son de otro tipo. Pensemos en un banco que quiere que su

aplicación del teléfono lea un cheque a partir de una foto.

Supongamos que la imagen ya se ha cortado en pequeños trozos, uno por cada carácter escrito a mano. Lo que queda es decidir, para cada trozo, qué carácter de un conjunto conocido muestra. Estas preguntas del tipo «¿cuál?» se llaman clasificación. Necesitan herramientas propias, aunque muchas ideas se trasladan de la regresión.

En la clasificación, el modelo mira las características, como los píxeles de una imagen, y dice a qué categoría, también llamada clase, pertenece el ejemplo, dentro de un conjunto fijo de opciones. Para los dígitos escritos a mano hay diez clases, del cero al nueve. Con solo dos clases, como gato y perro, se llama clasificación binaria. Su filtro de spam es clasificación binaria: cada mensaje acaba en una de dos clases, spam o no spam. Un modelo de regresión, un regresor, da un número; un modelo de clasificación, un clasificador, da una clase. Con más de dos clases, como al leer escritura a mano con los diez dígitos y todas las letras, se llama clasificación multiclase.

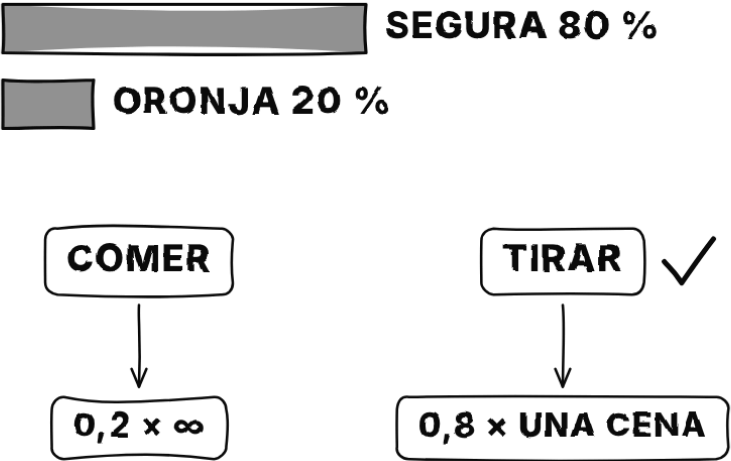
Un modelo que solo puede dar una respuesta firme, «gato» o «perro», resulta difícil de entrenar, por razones que el libro explicará más adelante. Así que normalmente se le hace dar en su lugar una probabilidad para cada clase. Al ver una foto, un clasificador podría dar la probabilidad de gato como cero coma nueve, lo que podemos leer como «seguro al noventa por ciento de que es un gato», y por tanto un diez por ciento para perro. El tamaño de la probabilidad de la clase elegida nos dice lo inseguro que está el modelo, aunque el libro señala que no es la única medida de incertidumbre. La regresión se entrenaba haciendo pequeño el error cuadrático; la clasificación tiene su propia pérdida habitual, llamada entropía cruzada, un nombre que el libro promete explicar cuando llegue a la teoría de la información.



Una probabilidad para cada clase

Ahora, la seta. El clasificador de setas venenosas da una probabilidad de cero coma dos de que sea una oronja verde, así que está seguro al ochenta por ciento de que no lo es. Aun así, solo un insensato se la comería, porque el placer seguro de una buena cena no vale un riesgo de uno entre cinco de morir. Para decidir, el libro sopesa cada acción: para cada resultado posible, se multiplica su probabilidad por el daño que causaría, y se suma todo. Si se come la seta, hay un veinte por ciento de probabilidad de un daño tan grande que podemos llamarlo infinito, y un ochenta por ciento de ningún daño; el veinte por ciento de algo infinito sigue siendo infinito. Si la tira, hay un veinte por ciento de probabilidad de no perder nada, porque de todos modos era venenosa, y un ochenta por ciento de perder una cena, que contamos como uno: cero coma ocho por uno da una pérdida de solo cero coma

ocho. Cero coma ocho contra infinito: tirarla gana con mucho. La clase más probable no es necesariamente la que hay que seguir. Y la cautela era acertada, porque la seta de la imagen del libro es de verdad una oronja verde.

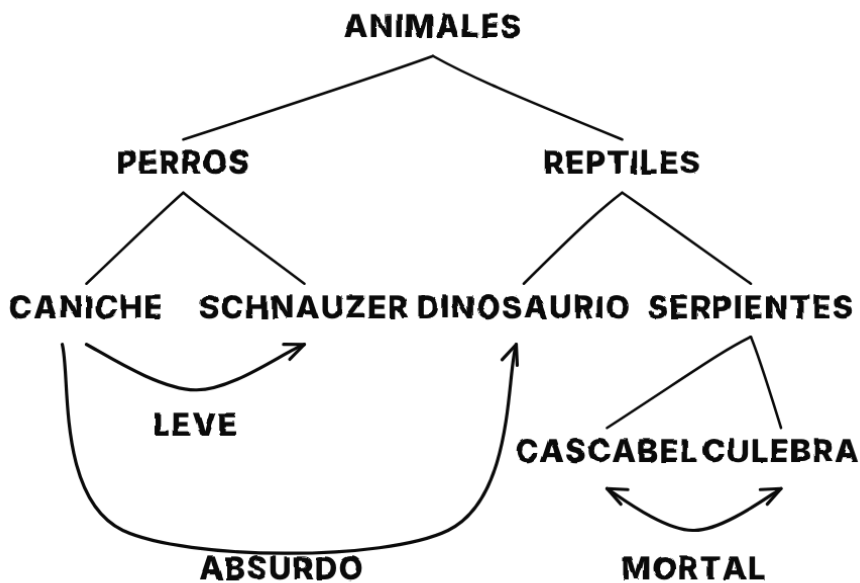


Sopesar cada acción por su daño

Pruébalo

La próxima vez que una aplicación le dé una cifra de confianza, pregúntese cuánto le costaría cada resultado. Un noventa por ciento de «le gustará esta canción» basta de sobra para darle a reproducir; un noventa por ciento de «este medicamento es compatible con sus otras pastillas» no basta, ni de lejos, por sí solo. La misma probabilidad puede pedir acciones opuestas cuando los costes son distintos.

La clasificación puede ser todavía más enrevesada. A veces las clases forman un árbol genealógico, más o menos como el naturalista Linneo ordenó a los animales en grupos anidados. Entonces no todos los errores son iguales: si el modelo tiene que equivocarse, mejor que caiga en un pariente cercano que en uno lejano. Esto se llama clasificación jerárquica. Confundir un caniche con un schnauzer es un pequeño desliz; confundir un caniche con un dinosaurio merece una penalización enorme. Pero qué árbol cuenta depende de para qué sea el modelo. Las serpientes de cascabel y las inofensivas culebras rayadas están juntas en el árbol genealógico de las serpientes, y sin embargo tomar una de cascabel por una culebra rayada podría matarle.



Algunos errores son peores

Para recordar

La clasificación responde a «¿cuál?», colocando cada ejemplo en una clase de un conjunto fijo: binaria con dos clases, multiclase con más. Los clasificadores suelen dar una probabilidad para cada clase, que además indica lo seguros que están, y se entrenan habitualmente con una pérdida llamada entropía cruzada.

Actuar bien significa sopesar cada opción por la probabilidad de cada resultado multiplicada por su daño, no limitarse a elegir la clase más probable. Con la seta, comérsela conlleva un daño esperado infinito y tirarla solo cero coma ocho de una cena.

Cuando las clases forman un árbol, algunos errores son peores que otros, y cuáles depende del uso, como con la serpiente de cascabel y la culebra rayada.

Preguntas

1. Un clasificador dice que una foto es un gato con probabilidad 0,9. ¿Cómo debe leerse esto?
 - a. Nueve de cada diez fotos que ve son gatos
 - b. Está seguro al noventa por ciento de que esta foto muestra un gato
 - c. La foto es noventa por ciento gato y diez por ciento perro
 - d. Acertó en el noventa por ciento de su conjunto de entrenamiento

2. ¿Por qué es sensato tirar una seta que la aplicación considera segura con un ochenta por ciento de probabilidad?
 - a. Porque ochenta por ciento significa que la aplicación está estropeada
 - b. Porque el daño ponderado de comérsela, una pequeña probabilidad de muerte, pesa mucho más que el daño ponderado de perder una cena
 - c. Porque la clase más probable siempre es errónea
 - d. Porque las setas no se pueden clasificar
3. En la clasificación jerárquica, ¿qué error merece la penalización más pequeña?
 - a. Confundir un caniche con un dinosaurio
 - b. Confundir un caniche con un schnauzer
 - c. Confundir un perro con un pez
 - d. Todos los errores merecen la misma penalización
4. ¿Por qué confundir una serpiente de cascabel con una culebra rayada cuenta como un error grave aunque sean parientes cercanas?
 - a. Porque el coste del error depende de para qué se use el modelo, y aquí podría matar
 - b. Porque pertenecen a familias de animales distintas
 - c. Porque el modelo se entrenó con muy pocas serpientes
 - d. Porque las serpientes siempre son multiclase

Respuestas

1. **b)** Una probabilidad de gato de cero coma nueve puede leerse como «seguro al noventa por ciento de que es un gato».

2. b) Sopesando cada acción por la probabilidad de cada resultado multiplicada por su daño, comérsela conlleva un daño esperado infinito y tirarla solo cero coma ocho.

3. b) Si el modelo tiene que equivocarse, mejor que caiga en un pariente cercano: confundir un caniche con un schnauzer es un pequeño desliz.

4. a) Qué árbol cuenta depende de para qué sea el modelo; tomar una de cascabel por una culebra rayada podría ser mortal.

Ejemplos resueltos

Comer o no comer Un clasificador da una probabilidad de 0,2 de que una seta sea una oronja verde y de 0,8 de que no lo sea. Comerse una oronja verde es un daño infinito; comerse una seta buena no cuesta nada. Tirar una oronja verde no cuesta nada; tirar una seta buena cuesta una cena, que contamos como 1. ¿Qué acción tiene el daño esperado menor?

Para cada acción, multiplique el daño de cada resultado por su probabilidad y sume.

Comer: 0,2 por un daño infinito, más 0,8 por 0; solo el primer término ya es infinito.

Tirar: 0,2 por 0, más 0,8 por 1, que es 0,8.

0,8 es muchísimo menor que un daño infinito, así que hay que tirar la seta, aunque «segura» sea la clase más probable.

Respuesta: tirarla: un daño esperado de 0,8 frente a uno infinito si se come

Ejercicios

1. Otra aplicación de setas da solo una probabilidad de 0,01 de oronja verde. Con los mismos daños, ¿cuál es el daño esperado de comerla y de tirarla? ¿Qué le dice esto?

Pista: Cualquier probabilidad positiva por un daño infinito sigue siendo infinita.

Respuesta: Comerla: 0,01 por un daño infinito sigue siendo infinito.

Tirlarla: 0,99 por 1 es 0,99. Tirlarla sigue ganando: cuando un resultado es catastrófico, incluso una pequeña probabilidad de que ocurra pesa más que una pequeña pérdida segura.

2. Diga si cada caso es binario, multiclase o regresión: leer cuál de los dígitos del 0 al 9 hay en un trozo de cheque; decidir gato o perro; predecir el precio de una casa; leer cualquier letra o dígito.

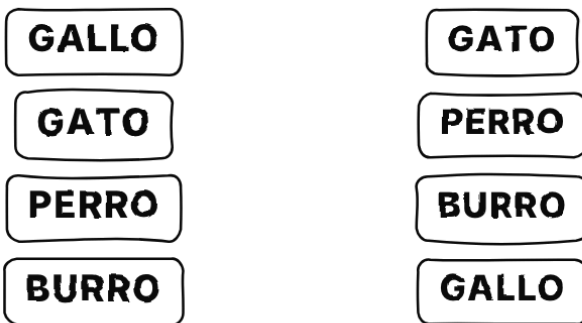
Pista: Cuente las clases, o pregúntese si la respuesta es un número.

Respuesta: Los dígitos del 0 al 9 y las letras más los dígitos son clasificación multiclase; gato o perro es clasificación binaria; el precio de la casa es un número, así que es regresión.

Muchas etiquetas, y lo mejor primero

Imagine a los músicos de Bremen del viejo cuento alemán: un burro, un perro de pie sobre su lomo, un gato sobre el perro y un gallo en lo alto, con árboles detrás. Ahora muestre esa imagen a un clasificador entrenado para responder «¿gato o perro?». Hoy un clasificador así es fácil de construir y puede ser muy preciso. Pero aquí está condenado a equivocarse, porque debe elegir exactamente una respuesta, y la respuesta verdadera es «los dos, y además un burro y un gallo».

MULTIETIQUETA



Una imagen, varias etiquetas

El problema no es el modelo, sino la pregunta. Si se esperan imágenes así, elegir una clase entre muchas es una forma equivocada de plantear el problema. Al modelo debería permitírsele decir: un gato, y un perro, y un burro, y un gallo. Predecir clases que pueden ser ciertas a la vez se llama clasificación multietiqueta.

He aquí una forma de ver cuánto más grande es esta pregunta. Con cuatro animales y la regla de «elegir exactamente uno», hay cuatro respuestas posibles. Si cada animal puede estar o no estar de forma independiente, cada uno tiene dos posibilidades, así que hay dos por dos por dos por dos, que son dieciséis respuestas posibles, desde ningún animal hasta los cuatro. Cada etiqueta adicional vuelve a duplicar la cuenta.

ELEGIR UNO

GATO

PERRO

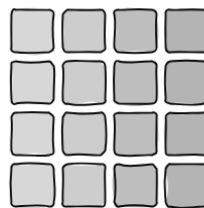
BURRO

GALLO

= 4

CUALQUIERA

$$2 \times 2 \times 2 \times 2 = 16$$



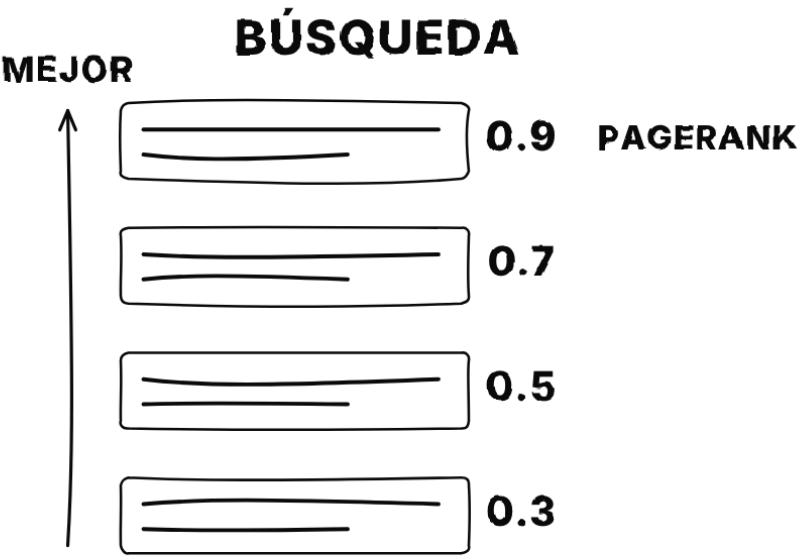
= 16

La clasificación multietiqueta es la descripción natural del etiquetado automático. Piense en las etiquetas de las entradas de un blog de tecnología: «aprendizaje automático», «tecnología», «gadgets», «lenguajes de programación», «computación en la nube». Un artículo típico podría llevar entre cinco y diez. Y las etiquetas van acompañadas: las entradas sobre computación en la nube suelen mencionar el servicio en la nube de Amazon, las de aprendizaje automático suelen mencionar los chips gráficos en los que funciona. Los hashtags que le sugiere una aplicación cuando publica una foto son una estimación de este tipo: varias etiquetas a la vez, no una.

A veces el conjunto de etiquetas es enorme. En la Biblioteca Nacional de Medicina de Estados Unidos, personal formado etiqueta cada artículo que entra en su gran índice médico con términos de una lista de unos veintiocho mil encabezamientos temáticos. Las buenas etiquetas permiten a los investigadores encontrar todo lo escrito sobre una cuestión, pero el trabajo es lento, y un artículo suele esperar alrededor de un año para ser etiquetado. El aprendizaje automático puede aportar etiquetas provisionales hasta que una persona revise cada artículo, y desde hace varios años hay competiciones sobre hacer exactamente esto.

La búsqueda vuelve a darle la vuelta a la pregunta. En una búsqueda en la web suele haber muchas páginas relevantes; la verdadera pregunta es cuáles mostrar primero. Una forma es dar a cada página una puntuación y devolver las de puntuación más alta. PageRank, el ingrediente secreto original del buscador de Google, fue una de las primeras puntuaciones de este tipo. Curiosamente, PageRank no dependía en absoluto de las palabras de la búsqueda: un filtro sencillo seleccionaba primero las páginas relevantes, y PageRank ponía luego arriba las de más autoridad. Hoy los buscadores usan aprendizaje

automático y modelos del comportamiento de las personas para calcular puntuaciones que sí dependen de la búsqueda. Hay congresos académicos enteros dedicados a este único problema.



La búsqueda ordena por puntuación

Para recordar

Un clasificador obligado a elegir una clase falla con una imagen de un burro, un perro, un gato y un gallo juntos. La clasificación multietiqueta permite que varias clases sean ciertas a la vez, y con cuatro etiquetas ya hay dieciséis respuestas posibles. Es la forma natural del etiquetado automático, desde las entradas de blog hasta los veintiocho mil encabezamientos médicos de una gran biblioteca. La búsqueda trata de ordenar: dar una puntuación a los elementos relevantes y mostrar primero los mejores, como hacía PageRank sin mirar siquiera las palabras de la búsqueda.

Preguntas

1. ¿Por qué incluso un clasificador de gato o perro muy preciso tiene que equivocarse con la imagen de Bremen?
 - a. Porque los animales están dibujados, no fotografiados
 - b. Porque debe elegir exactamente una clase, mientras que la respuesta verdadera son varias a la vez
 - c. Porque hay árboles al fondo
 - d. Porque no había burros en su conjunto de entrenamiento
2. Con cinco etiquetas posibles, cada una de las cuales puede estar o no estar, ¿cuántas respuestas posibles hay?
 - a. 5
 - b. 10

c. 25

d. 32

3. ¿Qué hacía un buscador con PageRank con las palabras que usted escribía?

a. Un filtro sencillo las usaba para seleccionar las páginas relevantes, y luego PageRank ordenaba esas páginas sin mirar las palabras

b. PageRank puntuaba cada página según cuántas veces aparecían sus palabras

c. Las ignoraba y mostraba las mismas páginas a todo el mundo

d. Las usaba para poner varias etiquetas a cada página

Respuestas

1. **b)** Debe elegir exactamente una respuesta, y la respuesta verdadera es los dos, y además un burro y un gallo: el problema es la pregunta, no el modelo.

2. **d)** Cada etiqueta adicional duplica la cuenta: cuatro etiquetas dan dieciséis respuestas, así que cinco dan treinta y dos.

3. **a)** PageRank no dependía de las palabras de la búsqueda: un filtro sencillo seleccionaba las páginas relevantes, y PageRank ponía arriba las de más autoridad.

Ejemplos resueltos

Contar las respuestas posibles Una imagen puede contener cualquiera de cuatro animales: un burro, un perro, un gato y un gallo. ¿Cuántas respuestas posibles hay si el modelo debe elegir exactamente un animal, y cuántas si cada animal puede estar o no estar?

Elegir exactamente uno de cuatro animales da 4 respuestas posibles. Si cada animal puede estar o no estar, cada uno tiene 2 posibilidades. Las opciones de los cuatro animales se combinan: 2 por 2 por 2 por 2. Eso da 16, desde ningún animal hasta los cuatro juntos.

Respuesta: 4 respuestas con una sola clase; 16 con clasificación multietiqueta

Ejercicios

1. Un blog usa 10 etiquetas posibles, y cualquier entrada puede llevar cualquiera de ellas. ¿Cuántos conjuntos distintos de etiquetas podría tener una entrada?

Pista: Cada etiqueta está puesta o no, y cada etiqueta adicional duplica la cuenta.

Respuesta: 2 multiplicado por sí mismo 10 veces, que son 1.024, contando la entrada sin ninguna etiqueta.

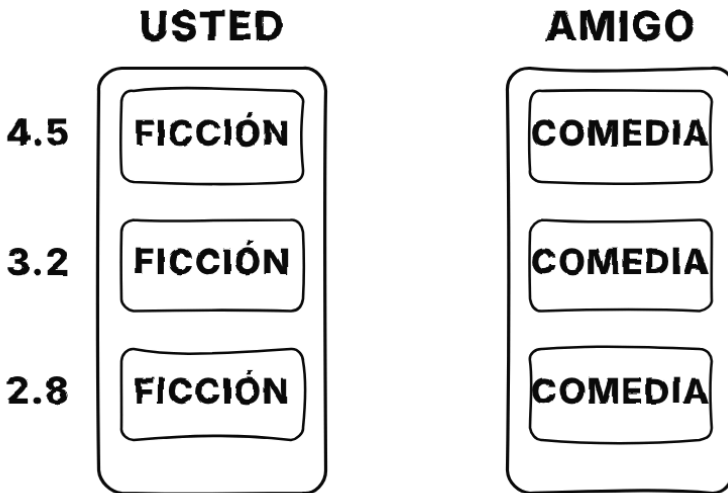
2. Explique por qué un buscador necesita ordenar, y no solo un sí o un no para cada página.

Pista: ¿Cuántas páginas relevantes encuentra una búsqueda típica?

Respuesta: Una búsqueda suele encontrar muchas páginas relevantes, así que saber si cada página es relevante no basta; el buscador debe decidir cuáles mostrar primero. Da a cada página una puntuación y muestra arriba las de puntuación más alta.

Su sección de recomendaciones, y lo que se desarrolla en el tiempo

Abra la aplicación de películas de su teléfono y luego pásesela a un amigo con gustos muy distintos. En su pantalla, la fila de arriba es ciencia ficción; en la suya, comedias antiguas. La misma aplicación, las mismas películas en su catálogo, dos páginas distintas. ¿Cómo decide qué poner primero para cada uno de ustedes?



La misma app, distintas primeras filas

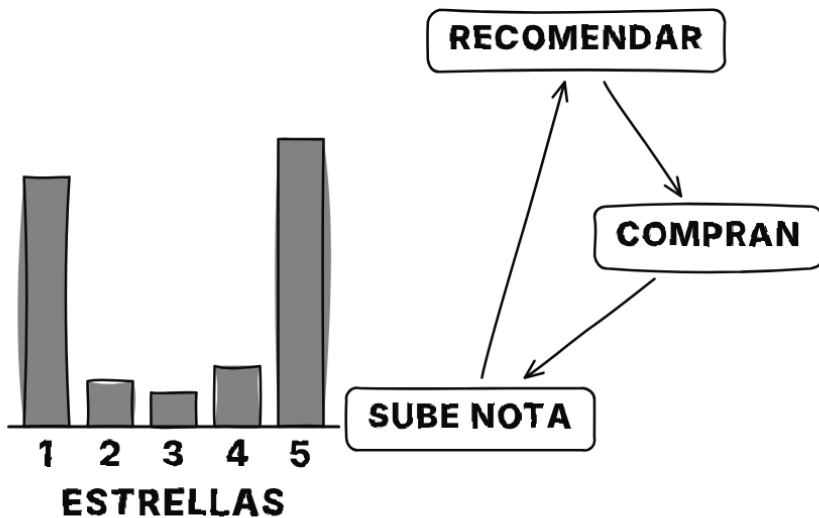
Los sistemas de recomendación son primos hermanos de la búsqueda y la ordenación: ambos eligen un conjunto de elementos y deciden

cuáles mostrar primero. La diferencia es que un sistema de recomendación es personal. Un fan de la ciencia ficción y un amante de las comedias de Peter Sellers deberían recibir páginas de resultados muy distintas, y lo mismo vale para las compras, la música y las noticias.

Para aprender lo que le gusta a la gente, el sistema escucha dos tipos de señal. A veces la gente lo dice abiertamente, con estrellas y reseñas en una tienda, una web de cine o una de libros; eso es retroalimentación explícita. Más a menudo lo dice solo con lo que hace, como saltarse una canción de una lista; eso es retroalimentación implícita. Saltarse una canción puede significar que no gusta, o solo que no encajaba con el momento. Cuando su sección de recomendaciones se reorganiza después de que usted se salte unos cuantos vídeos, está actuando según este tipo de señal silenciosa. En la versión más sencilla, el sistema aprende una puntuación para cada par de persona y elemento, como el número de estrellas que esa persona probablemente le daría, o la probabilidad de que lo compre. Luego, para cada persona, muestra los elementos con las puntuaciones más altas. Supongamos que el modelo espera que usted le dé a una película cuatro coma cinco estrellas, a otra tres coma dos y a una tercera dos coma ocho: las pone en ese orden, la de cuatro y media primero. Los sistemas reales son mucho más elaborados y tienen en cuenta la actividad detallada y las características de cada elemento; el libro muestra los libros de aprendizaje profundo que una gran tienda en línea le sugirió a uno de sus propios autores.

Estos sistemas son enormemente valiosos, pero, contruidos de forma ingenua sobre un modelo predictivo, tienen fallos serios. Primero, la retroalimentación está censurada: la gente valora sobre todo aquello sobre lo que tiene una opinión fuerte. En una escala de cinco estrellas

se ven muchos unos y cinco y extrañamente pocos treses, así que las valoraciones no muestran lo que pensó un espectador medio. Segundo, lo que la gente compra hoy es en parte resultado de lo que el sistema recomendó ayer, y puede que el método de aprendizaje no lo tenga en cuenta. Así puede formarse un bucle de retroalimentación: un artículo se promociona, así que más gente lo compra, así que parece mejor, así que se promociona todavía más. Cómo tratar la retroalimentación censurada, los incentivos y esos bucles son todavía cuestiones de investigación abiertas.



Valoraciones desequilibradas, bucles que se alimentan solos

Todo lo visto hasta ahora tenía una forma fija: la misma lista de datos para cada casa, la misma cuadrícula de píxeles para cada foto, un conjunto fijo de clases, una valoración con estrellas a partir de solo

una persona y un producto. Y una vez que el modelo había respondido a un ejemplo, lo olvidaba, como si cada caso nuevo no tuviera nada que ver con el anterior. Esa suposición, que los ejemplos sucesivos son independientes, a menudo no se cumple. Un vídeo puede tener cualquier número de fotogramas, y lo que pasa en uno está mucho más claro si se miran los fotogramas de antes y de después. Con el lenguaje pasa lo mismo: en la traducción automática entra una frase entera y debe salir una frase entera. En la unidad de cuidados intensivos de un hospital, un modelo podría vigilar a los pacientes y dar la alarma cuando el riesgo de morir en las próximas veinticuatro horas supere cierto umbral, y sería absurdo tirar cada hora la historia del paciente y juzgar solo por las últimas lecturas.

LOS DE ALREDEDOR



CUALQUIER LONGITUD



HISTORIAL

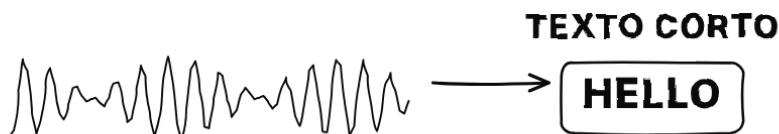
Un fotograma engaña, los fotogramas cuentan

Los problemas así se llaman aprendizaje de secuencias: el modelo recibe una secuencia, o la da, o ambas cosas. Cuando tanto la entrada como la salida son secuencias de cualquier longitud, se llama aprendizaje de secuencia a secuencia, como en la traducción, o al convertir voz en texto. El libro nombra algunos casos especiales.

El primero es el etiquetado y el análisis sintáctico, donde cada palabra de un texto recibe una etiqueta, y la entrada y la salida están alineadas: tantas etiquetas como palabras, en el mismo orden. Se podría etiquetar cada palabra con su categoría gramatical, o marcar qué palabras forman parte de nombres de personas, lugares u organizaciones. En la frase «Tom has dinner in Washington with Sally», las palabras Tom, Washington y Sally se marcarían como nombres, y las demás no.

TOM HAS DINNER IN
ENT — — —

WASHINGTON WITH SALLY
ENT — ENT



MILES DE NÚMEROS

Etiquetas en orden; sonido largo, texto corto

El segundo es el reconocimiento de voz: entra una grabación y sale una transcripción escrita. El sonido suele medirse ocho mil o dieciséis mil veces por segundo. A dieciséis mil por segundo, una palabra que se tarda medio segundo en decir son ocho mil mediciones, y sin embargo en la página puede tener solo cinco o seis letras. Así que no hay una correspondencia uno a uno entre el sonido y el texto, y la salida es mucho más corta que la entrada. La figura del libro muestra las palabras «deep learning» extendidas a lo largo de una onda sonora, con cada letra repartida por un tramo de ella. Cada vez que su asistente de voz escribe su pregunta en la pantalla, ha comprimido miles de números en unas pocas palabras. Las personas lo hacen notablemente bien, incluso con interferencias en la línea; para los ordenadores es un reto formidable.

Para recordar

Los sistemas de recomendación ordenan elementos para cada persona, aprendiendo de la retroalimentación explícita, como las valoraciones, y de la implícita, como los saltos de canción. La versión más sencilla puntúa cada par de persona y elemento, por ejemplo por las estrellas esperadas, y muestra las puntuaciones más altas. Las valoraciones censuradas y los bucles de retroalimentación, en los que las recomendaciones alimentan su propio éxito, son fallos serios que siguen siendo problemas abiertos. Cuando los ejemplos dependen de lo que vino antes, como en el vídeo, el lenguaje o la historia de un paciente, necesitamos aprendizaje de secuencias, y aprendizaje de secuencia a secuencia cuando ambos lados son secuencias de cualquier longitud. El etiquetado mantiene una etiqueta por palabra, mientras que el reconocimiento de voz comprime miles de mediciones de sonido por palabra en unas pocas letras.

Preguntas

1. Se salta una canción de una lista. ¿Por qué el sistema de recomendación no puede estar seguro de que no le gusta?
 - a. Porque saltarla es retroalimentación implícita y puede significar solo que la canción no encajaba con el momento
 - b. Porque los saltos nunca se registran
 - c. Porque solo cuentan como retroalimentación las valoraciones con estrellas

- d. Porque la canción se le recomendó a otra persona
2. ¿Por qué hay extrañamente pocas valoraciones de tres estrellas en una escala de cinco?
- a. Porque el sistema las redondea a una o a cinco
 - b. Porque la retroalimentación está censurada: la gente valora sobre todo aquello sobre lo que tiene una opinión fuerte
 - c. Porque en la mayoría de las webs no se permiten tres estrellas
 - d. Porque los artículos recomendados siempre son excelentes
3. ¿Por qué un monitor de pacientes en cuidados intensivos es un problema de aprendizaje de secuencias?
- a. Porque cada lectura es independiente de la anterior
 - b. Porque da una sola clase
 - c. Porque juzgar solo por las últimas lecturas tiraría la historia del paciente, que importa
 - d. Porque las lecturas llegan como una foto fija
4. ¿Por qué la salida del reconocimiento de voz es mucho más corta que la entrada?
- a. Porque el reconocedor se salta la mitad de las palabras
 - b. Porque cada palabra son muchos miles de mediciones de sonido pero solo unas pocas letras
 - c. Porque las transcripciones son resúmenes
 - d. Porque el sonido se mide solo una vez por segundo

Respuestas

1. **a)** Saltarse una canción es una señal silenciosa e implícita: puede significar que no gusta, o solo que no encajaba con el momento.
2. **b)** La gente valora sobre todo aquello sobre lo que tiene una opinión fuerte, así que se ven muchos unos y cincos y pocos treses.
3. **c)** Sería absurdo tirar cada hora la historia del paciente y juzgar solo por las últimas lecturas.
4. **b)** A dieciséis mil mediciones por segundo, una palabra de medio segundo son ocho mil mediciones pero solo cinco o seis letras.

Ejemplos resueltos

Marcar los nombres En la frase «Tom has dinner in Washington with Sally», marque cada palabra como nombre (Ent) o no (-), como en un etiquetado sencillo de entidades nombradas. ¿Cuántas etiquetas hay, y cuántas son nombres?

El etiquetado está alineado: una etiqueta por palabra, en el mismo orden.

La frase tiene siete palabras, así que hay siete etiquetas.

Tom, Washington y Sally nombran a una persona, un lugar y una persona; las demás palabras no.

Las etiquetas quedan: Ent, -, -, -, Ent, -, Ent.

Respuesta: siete etiquetas, tres de ellas nombres: Tom, Washington y Sally

Ejercicios

1. En una línea telefónica el sonido se mide 8.000 veces por segundo. ¿Cuántas mediciones forman una palabra que se tarda un cuarto de segundo en decir, y por qué esto hace del reconocimiento de voz un problema de longitudes desiguales?

Pista: Un cuarto de segundo es 8.000 entre 4.

Respuesta: 8.000 por un cuarto son 2.000 mediciones, para una palabra de quizá cuatro o cinco letras. La entrada son miles de números y la salida solo unas pocas letras, así que no hay correspondencia uno a uno y la salida es mucho más corta que la entrada.

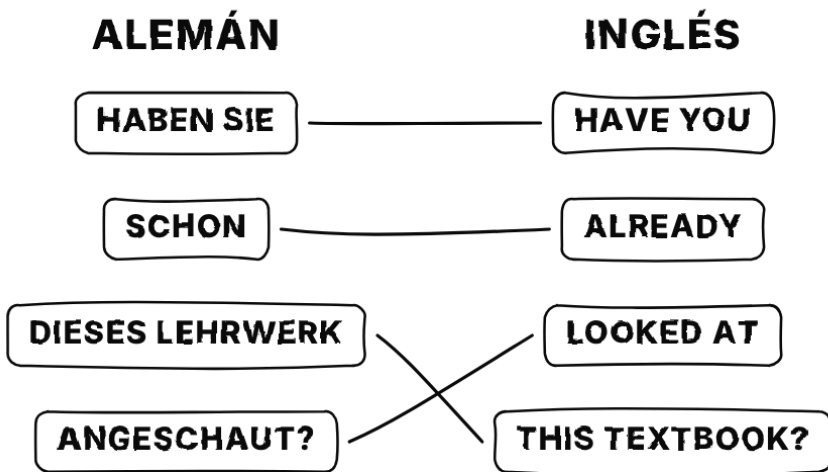
2. Un modelo espera que usted dé a tres películas 3,2, 4,5 y 2,8 estrellas. ¿En qué orden las muestra un sistema de recomendación sencillo, y cómo podría un bucle de retroalimentación hacer que la primera película parezca aún mejor de lo que es?

Pista: Se muestran primero las puntuaciones más altas; luego piense qué les pasa a las películas que se muestran más.

Respuesta: Muestra primero la película de 4,5 estrellas, luego la de 3,2 y luego la de 2,8. Al mostrarse primero, la película de arriba se ve o se compra más, esas compras la hacen parecer aún mejor al sistema, y se promociona todavía más: un bucle de retroalimentación.

Darle la vuelta a las cosas, y aprender sin respuestas

Escriba una frase corta en una aplicación de traducción, por ejemplo «Have you already looked at this excellent textbook?», y pulse el botón que lee el alemán en voz alta. Ocurren dos cosas que tiran en sentidos opuestos. Las palabras se reordenan, porque el alemán pone las cosas en otro orden. Y un puñado de letras se convierte en un largo flujo de sonido. Ambos son problemas de secuencias, y ambos son más difíciles de lo que parecen.



VERBO AL FINAL

La traducción cambia el orden

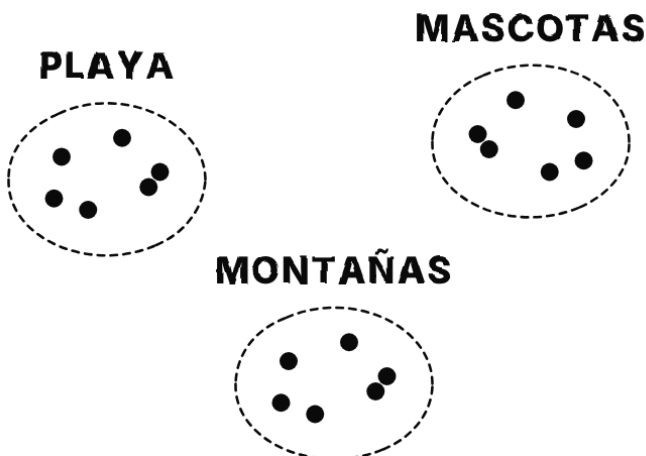
Leer un texto en voz alta se llama síntesis de voz, y es el reconocimiento de voz al revés. Ahora la entrada es texto y la salida una grabación de sonido, así que la salida es mucho más larga que la entrada. A dieciséis mil mediciones por segundo, una frase que se tarda dos segundos en decir sale como treinta y dos mil números, aunque entró como unas pocas docenas de letras. Cuando su asistente de voz le responde en voz alta, está haciendo exactamente esto.

La traducción automática añade una dificultad nueva. En el reconocimiento de voz, los sonidos y las letras al menos vienen en el mismo orden. En la traducción, la entrada y la salida pueden tener longitudes distintas, y las piezas correspondientes pueden venir en otro orden. El libro bromea con la costumbre alemana de poner el verbo al final. «Haben Sie sich schon dieses grossartige Lehrwerk angeschaut?» significa «¿Ya ha mirado usted este excelente libro de texto?». Mantenida en el orden alemán, palabra por palabra, se convierte en «¿Ha usted ya este excelente libro de texto mirado?», que ningún hablante diría. El libro añade que aparecen problemas parecidos en otros sitios: averiguar en qué orden lee alguien una página web es un rompecabezas en dos dimensiones, y mantener una conversación exige recurrir al conocimiento del mundo y a todo lo dicho mucho antes. El chatbot al que usted escribe tiene que manejar exactamente eso, teniendo a la vista toda la conversación para decidir qué decir a continuación.

Todo esto era aprendizaje supervisado: al modelo siempre se le daban las respuestas correctas. El libro lo compara con un trabajo con un jefe dictatorial que está detrás de usted y le dice exactamente qué hacer en cada situación. Aburrido, pero fácil de contentar: detectar rápido el patrón y copiar al jefe. El trabajo opuesto es un jefe que no tiene ni idea de lo que quiere, le entrega un enorme volcado de datos y

le dice: «¡Haga algo de ciencia de datos con esto!». Suena vago porque es vago. Aprender de datos sin ninguna etiqueta se llama aprendizaje no supervisado, y las preguntas que se pueden hacer solo tienen el límite de la imaginación.

AGRUPAMIENTO

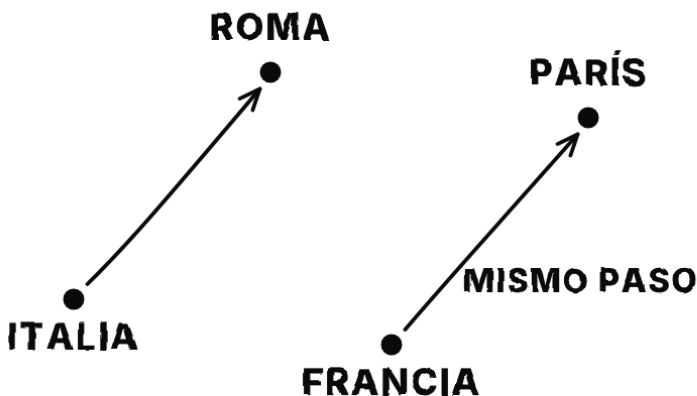


Grupos hallados sin respuestas

El libro ofrece una muestra. ¿Se pueden resumir los datos con unos pocos ejemplos típicos, de modo que las fotos se repartan en paisajes, perros, bebés, gatos y cumbres, o los compradores en grupos que navegan de forma parecida? Eso es el agrupamiento. ¿Pueden unos pocos números captar lo que importa? El vuelo de una pelota se describe bien con su velocidad, su tamaño y su masa, y los sastres ajustan la ropa con un pequeño conjunto de medidas del cuerpo; encontrar esos números se llama estimación de subespacios y, cuando

la relación es lineal, análisis de componentes principales. ¿Se pueden colocar las cosas como puntos en el espacio de modo que las relaciones se conviertan en direcciones? Entonces el paso de «Italia» a «Roma» es el mismo que el de «Francia» a «París»: tome Roma, reste Italia, sume Francia, y llegará a París. ¿Podemos encontrar las causas profundas, como la forma en que los precios de la vivienda, la contaminación, la delincuencia, la educación y los salarios dependen unos de otros, solo a partir de los datos? Ese es el campo de la causalidad y de los modelos gráficos probabilísticos.

MAPA DE PALABRAS

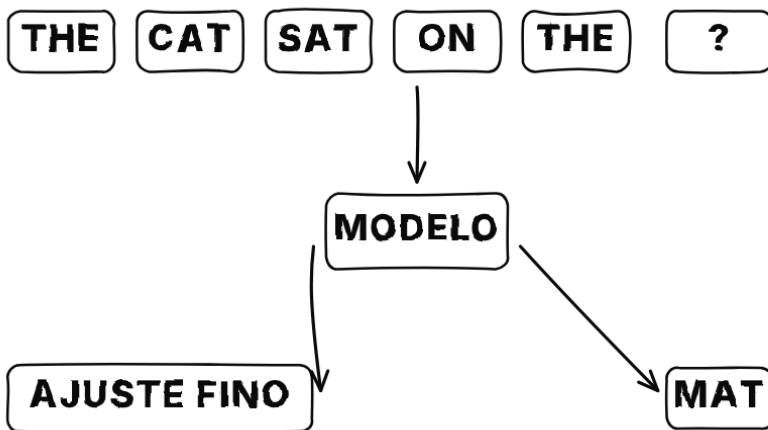


Palabras como puntos en un mapa

Una rama reciente y emocionante son los modelos generativos profundos. Aprenden cómo se reparten los datos tan bien que pueden puntuar lo probable que es un ejemplo, o producir ejemplos

completamente nuevos por su cuenta. Los primeros avances fueron los autocodificadores variacionales y luego las redes generativas antagónicas; más recientes son los flujos normalizadores y los modelos de difusión. Los generadores de imágenes que pintan un cuadro a partir de una frase que usted escribe pertenecen a esta familia.

Por último viene el aprendizaje autosupervisado, que usa los propios datos sin etiquetar para fabricar preguntas con respuestas conocidas. Con texto, se oculta una palabra al azar y se entrena al modelo para adivinarla a partir de las palabras de alrededor. En «the cat sat on the ...» («el gato se sentó en la ...»), la palabra que falta es casi seguro «mat», «alfombrilla», y la respuesta verdadera estuvo siempre en el texto, así que se pueden aprovechar enormes cantidades de texto sin ningún esfuerzo de etiquetado. Con imágenes, se puede pedir a un modelo dónde estaban, una respecto a otra, dos piezas recortadas de una imagen, que rellene una parte tapada a partir del resto, o que diga si dos imágenes alteradas vienen del mismo original. Lo que aprende un modelo así, su forma de describir los datos, se reutiliza a menudo: un poco más de entrenamiento, llamado ajuste fino, lo adapta a la tarea concreta que de verdad interesa. Los chatbots empezaron así: entrenados primero con cantidades enormes de texto para predecir palabras ocultas o siguientes, y solo después ajustados para responder preguntas.



Ocultar una palabra y adivinarla

Para recordar

La síntesis de voz convierte unas pocas letras en una larga grabación de sonido, lo contrario del reconocimiento de voz. La traducción es aún más difícil, ya que la entrada y la salida difieren en longitud y sus piezas pueden venir en otro orden, como el verbo alemán al final. El aprendizaje no supervisado trabaja con datos sin etiquetas: agrupamiento, encontrar unos pocos números reveladores, colocar las palabras de modo que Roma menos Italia más Francia caiga en París, y buscar causas. Los modelos generativos aprenden los datos lo bastante bien como para crear ejemplos nuevos, como hacen los modelos de difusión. El aprendizaje autosupervisado fabrica sus propias preguntas a partir de los datos, como palabras ocultas, y el resultado se ajusta luego finamente para la tarea concreta.

Preguntas

1. ¿Qué hace la traducción automática más difícil que el reconocimiento de voz, según la lección?
 - a. Su entrada es mucho más larga
 - b. Las piezas correspondientes de la entrada y la salida pueden venir en otro orden
 - c. No tiene etiquetas
 - d. Solo funciona con el alemán

2. ¿Por qué el aprendizaje autosupervisado puede usar enormes cantidades de texto sin esfuerzo de etiquetado?
 - a. Porque la respuesta verdadera de la palabra oculta ya estaba en el texto
 - b. Porque el texto no tiene estructura que aprender
 - c. Porque la gente etiqueta el texto gratis
 - d. Porque ignora las palabras alrededor del hueco
3. ¿Cuál de estas es una pregunta de agrupamiento?
 - a. ¿Qué pocos números describen el vuelo de una pelota?
 - b. ¿Causa la contaminación más delincuencia?
 - c. ¿Se pueden dividir estos compradores en grupos que navegan de forma parecida?
 - d. ¿Cuál es la siguiente palabra de esta frase?
4. ¿Qué es el ajuste fino?
 - a. Entrenar un modelo desde cero con datos etiquetados
 - b. Quitar las partes de un modelo que no se necesitan
 - c. Hacer que un modelo generativo produzca imágenes más nítidas
 - d. Un poco más de entrenamiento que adapta un modelo autosupervisado a la tarea que interesa

Respuestas

1. **b)** En la traducción, la entrada y la salida pueden tener longitudes distintas, y las piezas correspondientes pueden venir en otro orden.
2. **a)** En «the cat sat on the ...», la respuesta verdadera estuvo siempre en el texto, así que nadie tiene que etiquetar nada.

- 3. c)** Resumir los datos con unos pocos ejemplos típicos, como grupos de compradores que navegan de forma parecida, es agrupamiento.
- 4. d)** Lo que aprende un modelo autosupervisado se reutiliza a menudo: un poco más de entrenamiento, llamado ajuste fino, lo adapta a la tarea concreta.

Ejemplos resueltos

De las letras al sonido Un sistema de síntesis de voz produce sonido a 16.000 mediciones por segundo. Una frase tarda 2 segundos en decirse. ¿Cuántos números produce, y por qué la síntesis de voz es un problema en el que la salida es mucho más larga que la entrada?

Cada segundo de sonido son 16.000 mediciones.

Dos segundos dan 16.000 por 2, que son 32.000.

La frase entró como unas pocas docenas de letras.

Así que una entrada corta se convierte en una salida muy larga, lo contrario del reconocimiento de voz.

Respuesta: 32.000 números para una frase de unas pocas docenas de letras

Ejercicios

1. Usando la idea de que las relaciones se convierten en direcciones, complete: ¿París menos Francia más Italia da qué? ¿Y Roma menos Italia más Francia?

Pista: Quitar un país y añadir otro le lleva de una capital a la capital del otro.

Respuesta: París menos Francia más Italia cae en Roma, y Roma menos Italia más Francia cae en París: el paso de un país a su capital es la misma dirección en los dos casos.

2. Explique con los dos jefes del libro la diferencia entre el aprendizaje supervisado y el no supervisado.

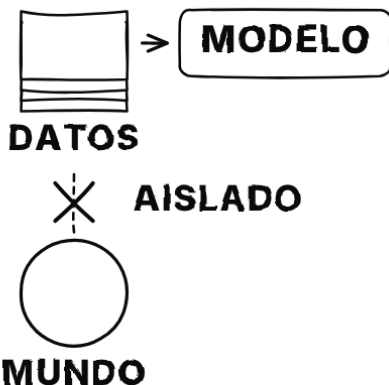
Pista: Un jefe se lo dice todo; el otro no le dice nada.

Respuesta: En el aprendizaje supervisado, el jefe está detrás de usted y le da la respuesta correcta para cada situación, así que solo tiene que copiar el patrón. En el no supervisado, el jefe se limita a entregarle un montón de datos sin respuestas y le pide que encuentre algo útil en ellos, así que tiene que descubrir la estructura usted mismo.

Cuando la máquina actúa

Observe a un niño pequeño construyendo una torre de bloques. Nadie le da una hoja con las respuestas correctas. Pone un bloque sobre otro, la torre se tambalea, se mantiene o se cae, y prueba algo distinto. Cada movimiento cambia lo que verá después, y el único veredicto sobre todas sus decisiones es si la torre sigue en pie al final.

FUERA DE LÍNEA



ACTUAR



Aprender aislado, o actuar

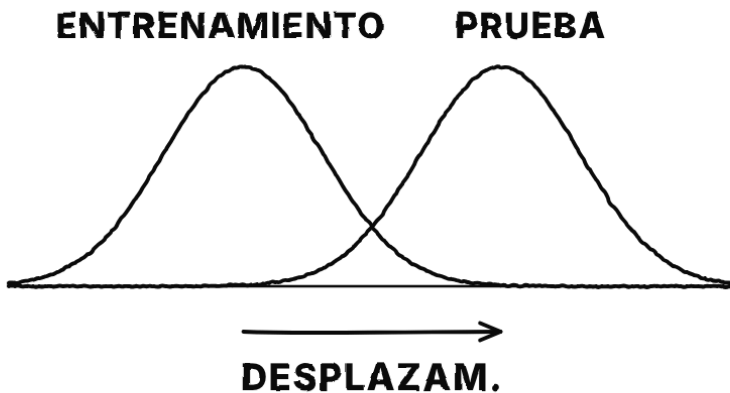
Hasta ahora, admite el libro, no se ha preguntado de dónde vienen los datos ni qué pasa cuando un modelo da su respuesta. El aprendizaje supervisado y el no supervisado funcionan de la misma forma

sencilla: coger de antemano un gran montón de datos y luego poner en marcha la máquina buscadora de patrones sin volver a tocar el mundo. Como todo el aprendizaje ocurre después de que la máquina queda aislada del mundo, esto se llama aprendizaje fuera de línea. El libro dibuja el aprendizaje supervisado como una imagen sencilla: el mundo aporta ejemplos, alguien los etiqueta y el modelo aprende del montón terminado.

El aprendizaje fuera de línea tiene su encanto, porque se puede estudiar el reconocimiento de patrones por sí solo, sin las complicaciones de un mundo que se mueve y responde. Pero es limitante. Los lectores que crecieron con las novelas de robots de Asimov imaginan máquinas inteligentes que no solo predicen, sino que actúan en el mundo. El libro quiere pensar en agentes, no solo en modelos predictivos, y en elegir acciones, no solo en hacer predicciones. Una predicción deja el mundo como estaba; una acción lo cambia. Para entrenar a un agente hay que tener en cuenta cómo sus propias acciones cambiarán lo que verá después, y para eso el aprendizaje fuera de línea es la herramienta equivocada.

En cuanto hay un intercambio con el mundo, se acumulan preguntas nuevas. ¿Recuerda el mundo lo que hicimos antes? ¿Quiere ayudar, como una persona que lee un texto despacio y con claridad a un reconocedor de voz? ¿Quiere ganarnos? Los spammers son el ejemplo del libro: no dejan de reescribir sus mensajes para colarse por su filtro de spam, así que el mundo del filtro contraataca. ¿Cambia el mundo con el tiempo, por sí solo o por nuestras propias herramientas, de modo que los datos futuros ya no se parecen a los pasados? Todo esto lleva al desplazamiento de distribución, cuando los datos en la prueba difieren de los del entrenamiento. El ejemplo del libro le resultará

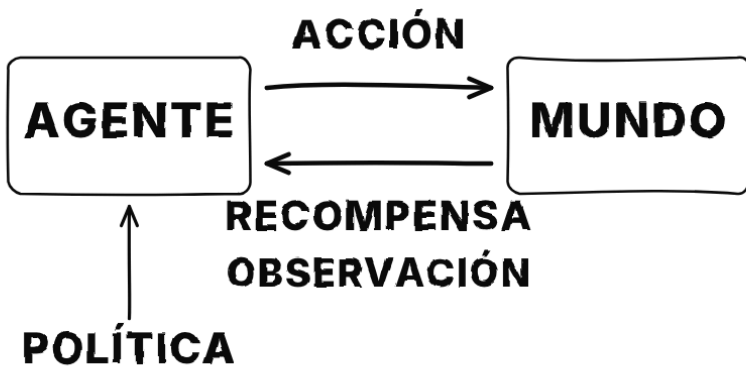
familiar a cualquier estudiante: deberes escritos por los ayudantes de cátedra, y luego un examen escrito por el profesor.



Los datos de prueba pueden desplazarse

El marco principal para aprender a actuar es el aprendizaje por refuerzo. Se usa en robótica, en sistemas que mantienen conversaciones y en los rivales controlados por el ordenador en los videojuegos. Combinado con el aprendizaje profundo, como aprendizaje por refuerzo profundo, ha llegado a los titulares. Un programa llamado red Q profunda aprendió a ganar a las personas en viejos juegos de recreativa de Atari sin ver nada más que la pantalla. Y AlphaGo derrotó al campeón del mundo del juego de tablero go.

El libro dibuja el aprendizaje por refuerzo como un ciclo. En cada paso el agente recibe una observación del mundo y elige una acción, que se transmite de vuelta al mundo mediante algún mecanismo, a veces llamado actuador. El mundo le da entonces al agente una recompensa y una nueva observación, y el agente vuelve a elegir, paso tras paso. El comportamiento del agente lo rige su política, simplemente una regla que asigna a lo que observa lo que hace. El objetivo del aprendizaje por refuerzo es encontrar buenas políticas.



El ciclo del aprendizaje por refuerzo

El marco es extraordinariamente general. Incluso la clasificación corriente se puede disfrazar de aprendizaje por refuerzo: se le da al agente una acción por cada clase, y el mundo devuelve una recompensa igual a la pérdida del problema original. Pero el

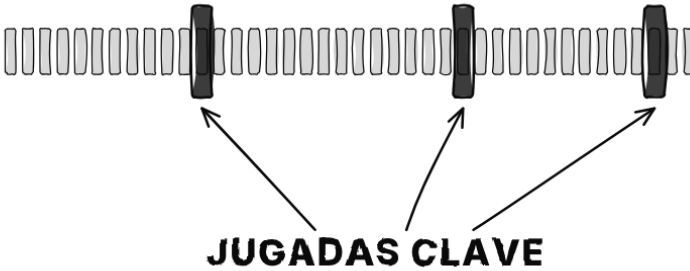
aprendizaje por refuerzo abarca también problemas que el supervisado no puede. En el aprendizaje supervisado, cada entrada de entrenamiento llega con la etiqueta correcta. En el aprendizaje por refuerzo, el mundo no dice cuál fue la mejor acción; solo reparte una recompensa, y a menudo ni siquiera dice qué acción se la ganó.

El ajedrez lo hace evidente. La única recompensa real llega al final: un punto por una victoria, digamos, y menos uno por una derrota. Una partida puede durar cuarenta movimientos de cada jugador. Si repartiéramos ingenuamente una victoria a partes iguales entre los cuarenta, cada movimiento recibiría un cuarentavo de punto, dos centésimas y media, tanto si fue la jugada brillante que ganó como una descuidada que casi pierde. Eso está claramente mal, y averiguar qué acciones merecen de verdad el mérito o la culpa se llama el problema de la asignación de mérito. Las personas también se enfrentan a él: alguien ascendido el once de octubre debe el ascenso a muchas buenas decisiones del último año, y para volver a ascender tiene que averiguar cuáles fueron.

ASIGNACIÓN DE MÉRITO

1/40 CADA UNO

+1



¿Quién se ganó la victoria?

Para recordar

El aprendizaje fuera de línea reúne primero los datos y aprende después, aislado del mundo; un agente que actúa cambia los mismos datos que verá. Cuando el mundo recuerda, ayuda, contraataca o cambia, los datos de prueba pueden diferir de los de entrenamiento, lo que se llama desplazamiento de distribución. El aprendizaje por refuerzo es un ciclo: observar, actuar, recibir una recompensa, volver a observar, con una política que decide las acciones. El aprendizaje por refuerzo profundo ha ganado a las personas en juegos de Atari y en el go. Las recompensas a menudo llegan tarde y no dicen qué acciones se las ganaron, y repartir mérito y culpa es el problema de la asignación de mérito.

Preguntas

1. ¿Por qué el aprendizaje fuera de línea es la herramienta equivocada para entrenar a un agente?
 - a. Porque las acciones de un agente cambian el mundo y, con él, los datos que verá después
 - b. Porque los agentes necesitan menos datos
 - c. Porque el aprendizaje fuera de línea no puede reconocer patrones
 - d. Porque los agentes solo trabajan con texto

2. ¿En qué se diferencia principalmente el aprendizaje por refuerzo del supervisado en lo que se le dice?
 - a. Se le dice la mejor acción en cada paso
 - b. Solo recibe una recompensa, a menudo sin saber qué acción se la ganó
 - c. No recibe ninguna retroalimentación
 - d. Se le dan ejemplos etiquetados de antemano
3. ¿Por qué repartir una victoria de ajedrez a partes iguales entre los cuarenta movimientos es una mala forma de asignar el mérito?
 - a. Porque cuarenta movimientos son muy pocos
 - b. Porque la recompensa de una victoria es cero
 - c. Porque da el mismo mérito a una jugada brillante que ganó y a una descuidada
 - d. Porque el ajedrez no tiene recompensas
4. Los spammers reescriben sus correos para burlar un filtro de spam. ¿A qué pregunta sobre el mundo responde esto que sí?
 - a. ¿Quiere el mundo ayudarnos?
 - b. ¿Carece el mundo de memoria?
 - c. ¿Se observa el mundo por completo?
 - d. ¿Quiere el mundo ganarnos?

Respuestas

1. a) Para entrenar a un agente hay que tener en cuenta cómo sus propias acciones cambiarán lo que verá después, y el aprendizaje fuera de línea no puede.

- 2. b)** En el aprendizaje por refuerzo el mundo no dice cuál fue la mejor acción; solo reparte una recompensa, a menudo sin decir qué acción se la ganó.
- 3. c)** Repartida por igual, cada jugada recibe un cuarentavo de punto, fuera brillante o descuidada, lo que está claramente mal.
- 4. d)** Los spammers no dejan de reescribir sus mensajes para colarse por el filtro: el mundo del filtro contraataca.

Ejemplos resueltos

Repartir ingenuamente una victoria Una partida de ajedrez da una recompensa de 1 por una victoria. El ganador hizo 40 movimientos. Si la recompensa se repartiera a partes iguales entre todos los movimientos, ¿cuánto mérito recibiría cada uno, y por qué es un problema?

La única recompensa, 1, llega al final de la partida.

Repartida por igual entre 40 movimientos, cada uno recibe $\frac{1}{40}$.

Cada movimiento recibe los mismos $\frac{1}{40}$, tanto si ganó la partida como si casi la pierde, así que el reparto igual no distingue las buenas jugadas de las malas: este es el problema de la asignación de mérito.

Respuesta: $\frac{1}{40}$ por movimiento, igual para los buenos y los malos

Ejercicios

1. Describa el ciclo del aprendizaje por refuerzo para un rival controlado por el ordenador en un videojuego: ¿qué observa, cuáles son sus acciones, cuál podría ser su recompensa y cuál es su política?

Pista: Observar, actuar, recibir una recompensa, volver a observar.

Respuesta: Observa la pantalla o el estado del juego, sus acciones son los movimientos o pulsaciones de botones que puede hacer, y su recompensa podría ser los puntos conseguidos o superar el nivel. Su política es la regla que elige una acción a partir de lo que observa en ese momento, y el aprendizaje busca que esa regla sea buena.

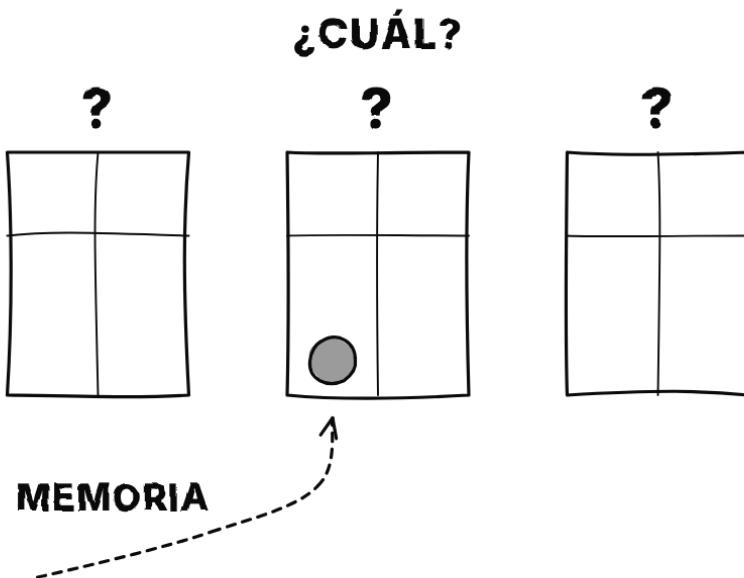
2. Explique cómo la clasificación puede replantearse como aprendizaje por refuerzo, tal como lo describe el libro.

Pista: ¿Cuáles serían las acciones y la recompensa?

Respuesta: Se le da al agente una acción por cada clase, de modo que elegir una acción es elegir una clase, y el mundo devuelve una recompensa igual a la pérdida del problema de clasificación original. El agente aprende entonces a elegir las clases que dan la mejor recompensa, que es lo mismo que clasificar bien.

Puntos ciegos, apuestas y raíces antiguas

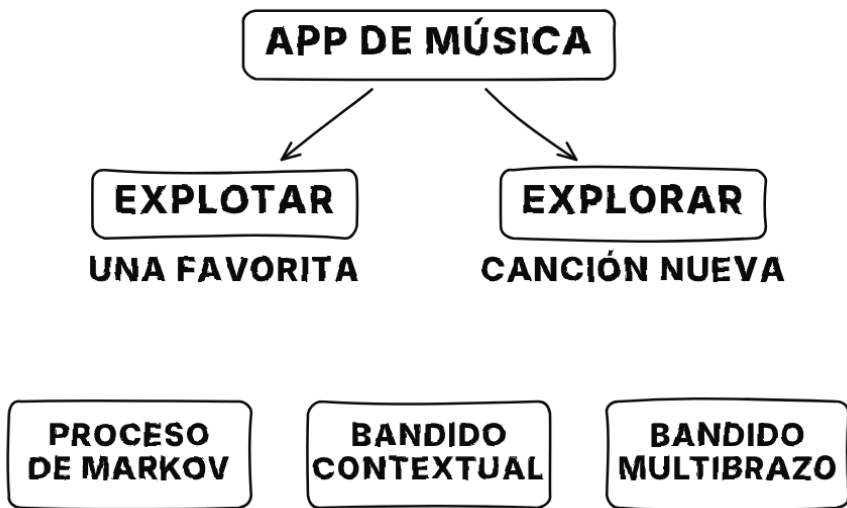
Imagine que su robot aspirador se ha metido en uno de varios armarios idénticos de su casa y se ha quedado atascado. Por dentro, todos los armarios se ven exactamente iguales: cuatro paredes, una puerta, un estante. Lo que ve ahora mismo no puede decirle en qué armario está. Solo recordando lo que vio antes de entrar puede él, o usted, averiguar dónde está.



Armarios idénticos

Este es el problema de la observabilidad parcial. En el aprendizaje por refuerzo, la observación actual del agente puede no decirle todo sobre el estado en que se encuentra. Para actuar bien, puede necesitar reconstruir el presente a partir de lo que ha visto antes.

Hay otra dificultad. En cualquier momento el agente puede conocer una estrategia que funciona razonablemente bien, mientras que puede haber muchas mejores que nunca ha probado. Tiene que elegir continuamente entre explotar la mejor estrategia que conoce y explorar otras, quizá renunciando a algo de recompensa ahora a cambio de conocimiento. Su aplicación de música se enfrenta a esto cada día: poner una canción que está segura de que le gusta, o arriesgarse con una nueva que podría encantarle o que podría saltarse.

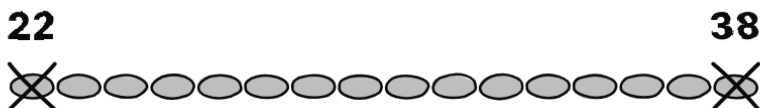


Explotar o explorar; casos más sencillos

Juntándolo todo, el problema general es muy amplio: las acciones cambian lo que el agente ve después, las recompensas solo se ven para las acciones que de verdad se realizan, y el mundo puede verse entero o solo en parte. Abordar todo eso a la vez quizá sea pedir demasiado, y muchos problemas reales no lo tienen todo, así que los investigadores estudian casos especiales. Cuando el agente ve el estado completo del mundo, el problema se llama proceso de decisión de Markov. Cuando el estado no depende de las acciones anteriores del agente, es un problema de bandido contextual. Y cuando no hay estado alguno, solo un conjunto de acciones posibles cuyas recompensas son al principio desconocidas, es el clásico problema del bandido multibrazo, llamado así por una fila de máquinas tragaperras, los bandidos de un brazo.

Terminado este recorrido por los problemas, el libro se vuelve hacia la historia. Muchos métodos de aprendizaje profundo son recientes, pero las ideas detrás de aprender de los datos se estudian desde hace siglos. El deseo de analizar datos y prever lo que viene después está en la raíz de buena parte de la ciencia y las matemáticas. Dos patrones del azar llevan nombres de grandes matemáticos: la distribución de Bernoulli, por Jacob Bernoulli, que murió en 1705, y la distribución gaussiana, por Carl Friedrich Gauss. Gauss ideó además el método de mínimos cuadrados, que ajusta una regla a las mediciones haciendo los fallos al cuadrado lo más pequeños posible, y que todavía se usa desde los seguros hasta el diagnóstico médico. Esas herramientas reforzaron las ciencias experimentales; la ley de Ohm, que relaciona la corriente que atraviesa una resistencia con el voltaje entre sus extremos, se describe perfectamente con un modelo lineal.

UN PIE TÍPICO



$$480 \div 16 = 30$$

$$420 \div 14 = 30$$

MEDIA RECORTADA

Dieciséis pies, una media recortada

Incluso antes, la gente tenía buen olfato para las estimaciones. Un libro de geometría del siglo dieciséis de Jacob Köbel muestra cómo hallar la longitud de un pie típico. Cuando la gente salía de la iglesia, dieciséis hombres adultos se ponían en fila talón con punta, se medía la longitud total de sus pies y se dividía entre dieciséis. Digamos que los dieciséis pies sumaban cuatrocientos ochenta centímetros: entre dieciséis, da treinta centímetros. Más tarde el método se mejoró para lidiar con pies raros: se apartaba al hombre de pies más cortos y al de pies más largos, y solo se promediaba a los otros catorce. Si esos dos medían veintidós y treinta y ocho centímetros, los catorce restantes suman cuatrocientos veinte, y cuatrocientos veinte entre catorce vuelve a ser treinta. Es uno de los primeros ejemplos de media

recortada, una media que descarta los extremos para que un valor raro no pueda estropearla.

La estadística despegó de verdad cuando se empezaron a recoger datos en cantidad. Ronald Fisher, uno de sus pioneros, dio forma tanto a su teoría como a su uso en genética, y algunos de sus métodos e ideas siguen en la base de la estadística moderna. Hasta sus datos perduraron: las medidas de flores de iris que publicó en 1936 todavía se usan a veces para probar métodos de aprendizaje. Fisher fue también un promotor de la eugenesia, un recordatorio de que el uso moralmente dudoso de la ciencia de datos tiene una historia tan larga como el útil.

Otras influencias llegaron de la teoría de la información de Claude Shannon y de la teoría de la computación de Alan Turing. En un artículo famoso, Turing preguntó: «¿Pueden pensar las máquinas?», y propuso lo que hoy se llama la prueba de Turing: una máquina cuenta como inteligente si a un juez, que intercambia con ella solo mensajes escritos, le cuesta distinguir sus respuestas de las de una persona. Al charlar con un chatbot moderno, mucha gente ha descubierto que esa prueba es menos hipotética de lo que parecía. La neurociencia y la psicología aportaron más, ya que los humanos se comportan claramente de forma inteligente y cabe esperar entender y copiar cómo. Donald Hebb propuso en su libro sobre la organización del comportamiento que las células nerviosas aprenden por refuerzo positivo, una idea que hoy se llama la regla de aprendizaje de Hebb. Inspiró el perceptrón de Rosenblatt y sentó las bases de los métodos de gradiente que se usan hoy en el aprendizaje profundo: reforzar lo que lleva a un buen comportamiento, debilitar lo que lleva a uno malo, y así llegar a buenas posiciones de las perillas. De esa inspiración biológica viene el nombre de las redes neuronales;

durante más de un siglo, desde los modelos de Bain y Sherrington, los investigadores han intentado construir circuitos que se parezcan a las redes de células nerviosas.

Para recordar

Un agente puede ver solo parte de su situación, como un robot atascado en uno de muchos armarios idénticos, y debe usar lo que vio antes. También debe equilibrar la explotación de la mejor estrategia conocida con la exploración de otras que podrían ser mejores. Casos especiales del aprendizaje por refuerzo son el proceso de decisión de Markov, el bandido contextual y el bandido multibrazo. Aprender de los datos tiene siglos, desde Bernoulli y los mínimos cuadrados de Gauss hasta los pies de dieciséis hombres promediados a la puerta de una iglesia, más tarde como media recortada. La estadística, la teoría de la información, la teoría de la computación y el estudio del cerebro alimentaron el aprendizaje automático, y las redes neuronales toman su nombre de las células nerviosas.

Preguntas

1. ¿Por qué un robot atascado en uno de varios armarios idénticos no puede saber dónde está por su vista actual?
 - a. Porque sus sensores están rotos
 - b. Porque todos los armarios se ven iguales, así que la observación actual no revela toda su situación

- c. Porque no tiene política
 - d. Porque está explorando
2. ¿Qué caso especial del aprendizaje por refuerzo no tiene ningún estado, solo acciones con recompensas desconocidas?
- a. El proceso de decisión de Markov
 - b. El problema del bandido contextual
 - c. El problema del bandido multibrazo
 - d. El aprendizaje fuera de línea
3. ¿Por qué los medidores de pies posteriores apartaban a los hombres de pies más cortos y más largos?
- a. Para que un valor raro no pudiera estropear la media
 - b. Para que el pie saliera más largo
 - c. Porque dieciséis hombres eran demasiados para medir
 - d. Para seguir los mínimos cuadrados de Gauss
4. ¿Qué principio de la idea de Hebb sigue guiando cómo el aprendizaje profundo ajusta sus perillas?
- a. Medir directamente cada célula nerviosa
 - b. Copiar exactamente el cableado del cerebro
 - c. Dejar fijas todas las perillas tras el primer intento
 - d. Reforzar lo que lleva a un buen comportamiento y debilitar lo que lleva a uno malo

Respuestas

1. **b)** Esto es observabilidad parcial: la observación actual del agente puede no decirle todo sobre el estado en que se encuentra.

- 2. c)** Cuando no hay estado, solo un conjunto de acciones cuyas recompensas son al principio desconocidas, es el clásico problema del bandido multibrazo.
- 3. a)** Una media recortada descarta los extremos para que un valor raro no pueda estropearla.
- 4. d)** La idea de Hebb sentó las bases de los métodos de gradiente actuales: reforzar lo que lleva a un buen comportamiento, debilitar lo que lleva a uno malo.

Ejemplos resueltos

El pie de Köbel, y su versión recortada Los pies de dieciséis hombres, en fila talón con punta, miden 480 centímetros en total. El pie más corto mide 22 centímetros y el más largo 38. Halle la media simple y la media recortada tras apartar a esos dos hombres.

La media simple divide el total entre el número de hombres: 480 entre 16 son 30 centímetros.

Para la media recortada, quite el pie más corto y el más largo: 480 menos 22 menos 38 son 420 centímetros.

Quedan catorce hombres, así que se divide entre 14.

420 entre 14 son 30 centímetros; aquí los dos extremos se compensaron, pero un solo pie muy raro ya no arrastraría la respuesta.

Respuesta: 30 centímetros de las dos maneras

Ejercicios

- 1.** Los pies de diez hombres suman 300 centímetros, pero uno de ellos, que midió 50 centímetros, llevaba por error unas botas enormes, y el pie más corto mide 25 centímetros. Compare la media simple con la media recortada que quita el más corto y el más largo.

Pista: Para la media recortada, quite los dos extremos y divida entre ocho.

Respuesta: La media simple es 300 entre 10, que son 30 centímetros. Quitando 25 y 50 quedan 225 centímetros entre 8 hombres, unos 28,1 centímetros. La bota tiró de la media simple hacia arriba; la media recortada no se deja engañar.

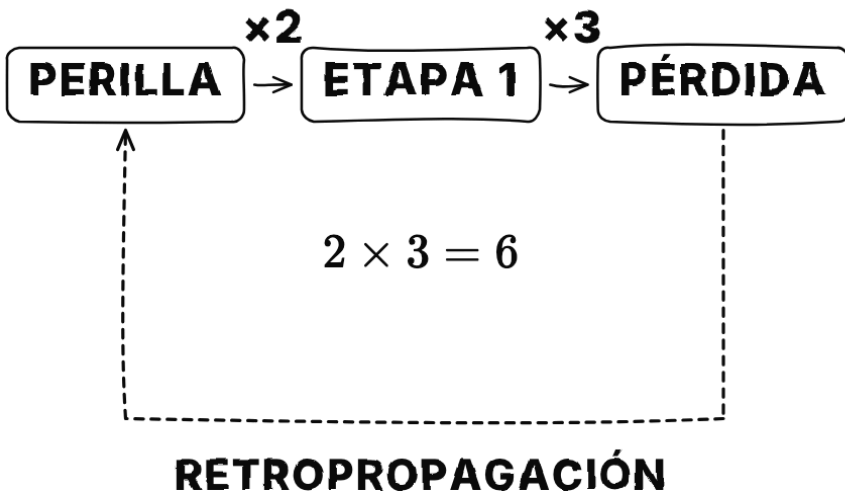
2. Explique con un ejemplo propio la elección entre explorar y explotar.

Pista: Piense en un restaurante, un camino al trabajo o una lista de canciones.

Respuesta: Ir a su restaurante favorito es explotar: sabe que comerá bien. Probar uno nuevo es explorar: puede ser peor, pero podría ser mejor, y solo se sabe probando. Un aprendiz que nunca explora puede quedarse atascado en una opción simplemente buena; uno que siempre explora nunca disfruta de lo que ha aprendido.

De un callejón sin salida a una explosión

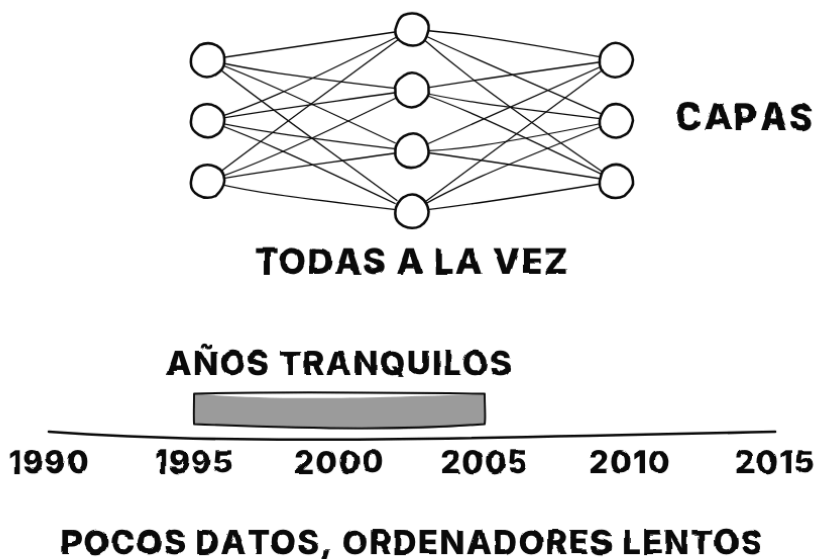
Piense en el primer videojuego al que jugó, y luego en un juego de un teléfono de hoy. El salto en lo que una máquina pequeña puede dibujar, y en lo rápido que lo hace, es enorme. Y aquí está el giro: los chips contruidos para que los juegos se vieran bien resultaron ser justo lo que las redes neuronales estaban esperando. Para ver por qué, necesitamos las dos ideas que están en el corazón de estas redes, y la historia de por qué un día se estancaron.



La regla de la cadena, con números

Casi toda red neuronal actual se basa en dos principios. El primero son las capas: etapas de procesamiento que alternan combinaciones lineales sencillas de números y curvaturas sencillas, de modo que cada etapa transforma un poco lo que recibe antes de pasarlo adelante. El segundo es una forma de ajustar a la vez las perillas de toda la red usando la regla de la cadena del cálculo, un método llamado retropropagación. La regla de la cadena dice, con palabras, que si una etapa alimenta a la siguiente, el efecto de una perilla temprana sobre la pérdida final es el producto de los efectos a lo largo del camino. Por ejemplo, si girar un poco una perilla sube la salida de la primera etapa dos unidades por unidad de giro, y cada unidad de esa salida sube la pérdida tres, entonces la perilla sube la pérdida dos por tres, seis unidades por unidad de giro. La retropropagación aplica esto desde el final de la red hacia el principio, y así sabe para cada perilla, en una sola pasada, hacia dónde girarla.

Tras un rápido progreso inicial, la investigación en redes neuronales languideció aproximadamente entre 1995 y 2005, por dos razones. Entrenar una red exige muchísimo cálculo, y aunque a finales de siglo la memoria abundaba, la potencia de cálculo escaseaba. Y los conjuntos de datos eran pequeños: las flores de iris de Fisher de 1936 seguían siendo una prueba popular, y una colección de sesenta mil dígitos escritos a mano se consideraba enorme. Con tan pocos datos y tan poco cálculo, otras potentes herramientas estadísticas funcionaban mejor en la práctica, entre ellas los métodos de núcleo, los árboles de decisión y los modelos gráficos. A diferencia de las redes neuronales, no necesitaban semanas de entrenamiento, y daban resultados predecibles con sólidas garantías matemáticas.



Capas, y los años tranquilos

Lo que cambió fueron los datos. La web, empresas que atienden en línea a cientos de millones de personas, sensores baratos y buenos por todas partes y un almacenamiento cada vez más barato produjeron juntos datos a una escala enorme. El cálculo también siguió abaratándose y, sobre todo, los chips gráficos diseñados para los videojuegos, las GPU, transformaron lo que el aprendizaje profundo podía calcular. De repente, algoritmos y modelos que habían parecido imposibles estaban al alcance.

El libro lo expone en una tabla, década a década. En 1970 un conjunto de datos típico eran las flores de iris, unos cien ejemplos; un ordenador tenía un kilobyte de memoria y hacía unos cien mil cálculos por segundo. En 1990 el conjunto típico eran diez mil

caracteres escaneados, con diez megabytes de memoria y diez millones de cálculos por segundo. En 2020 un conjunto de datos podía ser una red social entera, alrededor de un billón de elementos; la memoria había llegado a cien gigabytes, y una sola máquina repleta de chips gráficos hacía unos mil billones de cálculos por segundo. Compare el crecimiento en esos cincuenta años. Los datos crecieron de cien a un billón, diez mil millones de veces; la memoria creció de un kilobyte a cien gigabytes, cien millones de veces. Así que la memoria se quedó cien veces por detrás de los datos. La potencia de cálculo, dice el libro, superó incluso a los datos. Eso empujó los mejores métodos hacia modelos ahorradores de memoria pero capaces de gastar un cálculo generoso en ajustar sus perillas, y el punto óptimo pasó de los modelos lineales y los métodos de núcleo a las redes neuronales profundas. También explica por qué muchos de los pilares del aprendizaje profundo, como los perceptrones apilados, las redes convolucionales, las redes de memoria a corto y largo plazo y el aprendizaje Q, se redescubrieron en esencia en la última década, tras años dormidos.

CRECIMIENTO DESDE 1970

DATOS ×10 MIL MILL.



CÁLCULO: AÚN MÁS



MEMORIA ×100 MILL.



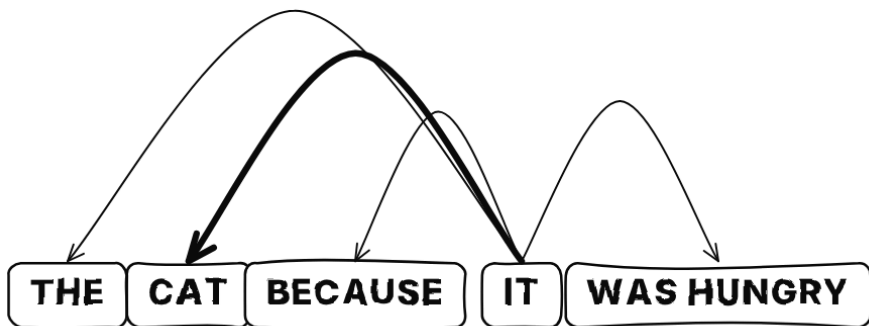
| | | | | | | | | |

CADA MARCA: ×10

Qué creció, y cuánto

Aun así, el progreso fue más que recetas viejas en máquinas más rápidas. El libro lo compara con la explosión cámbrica, el estallido en que la vida animal se diversificó de repente, y enumera algunas de las ideas nuevas. El dropout combate el sobreajuste inyectando ruido aleatorio en toda la red durante el entrenamiento. La atención resolvió un viejo problema: cómo dar a un sistema más memoria y complejidad sin añadir más perillas. En lugar de comprimir una frase entera en un resumen de tamaño fijo antes de traducirla, un modelo con atención mantiene una especie de puntero aprendible hacia la parte que necesita, y las frases largas se volvieron mucho más precisas.

ATENCIÓN



TRANSFORMER: SOLO ATENCIÓN

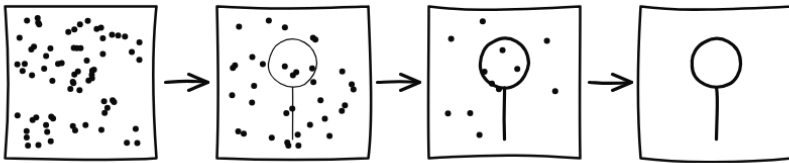
La atención vuelve la vista

El Transformer, construido enteramente sobre la atención, no deja de mejorar a medida que crecen los datos, el modelo y el cálculo. Ha triunfado en lenguaje, visión, habla, aprendizaje por refuerzo y más; un único Transformer entrenado con texto, imágenes, movimientos de las articulaciones de un robot y pulsaciones de botones puede jugar a Atari, poner pies de foto, conversar y controlar un robot. Los modelos de lenguaje predicen texto a partir de otro texto, y a medida que se hicieron más grandes ganaron cada vez más capacidades. El chatbot al que usted escribe es el propio ejemplo del libro: ChatGPT, un modelo de lenguaje alineado con lo que quiere la gente, que puede ayudar a depurar código informático o escribir de forma creativa en una conversación. Los diseños de varias etapas, como las redes de memoria, permiten a una red cambiar su estado interno paso a paso,

llevando a cabo una cadena de razonamiento casi como un procesador actualiza su memoria durante un cálculo.

Los modelos generativos también dieron saltos. Las redes generativas antagónicas enfrentan a un generador, que fabrica ejemplos falsos, con un discriminador, que intenta distinguir lo falso de lo real; las perillas del generador se ajustan hasta que el discriminador no puede distinguirlos. Entre los resultados hay cebras al galope hechas a partir de caballos y caras falsas convincentes de famosos, e incluso dibujantes aficionados pueden convertir un boceto tosco en una imagen casi fotográfica. Más recientemente, los modelos de difusión aprenden a invertir la adición gradual de ruido, construyendo una imagen paso a paso a partir de interferencias, y están detrás de sistemas de texto a imagen como DALL-E 2 e Imagen. Un generador de imágenes que pinta lo que usted describa es esta idea en acción.

DIFUSIÓN



QUITAR RUIDO

Una imagen a partir del ruido

La ingeniería también importó. El caballo de batalla de la optimización, el descenso de gradiente estocástico, trabaja con pequeños lotes de ejemplos cada vez, pero los lotes pequeños desaprovechan los chips gráficos. Reparta el entrenamiento entre mil veinticuatro chips, cada uno con treinta y dos imágenes, y cada paso maneja treinta y dos mil setecientas sesenta y ocho imágenes, unas treinta y dos mil. Trabajos posteriores lo llevaron a sesenta y cuatro mil y redujeron el entrenamiento de una red estándar de reconocimiento de imágenes, ResNet-50 con la colección ImageNet, a menos de siete minutos, cuando al principio llevaba días. El cálculo en paralelo también ayudó al aprendizaje por refuerzo a alcanzar un juego sobrehumano en el go, en los juegos de Atari y en StarCraft,

siempre que un simulador pueda aportar muchos estados, acciones y recompensas de los que aprender.

Por último, los marcos, herramientas gratuitas para construir redes, difundieron las ideas. La primera generación, Caffe, Torch y Theano, sostuvo muchos artículos emblemáticos. Luego llegó TensorFlow, a menudo usado a través de Keras, junto con CNTK, Caffe 2 y MXNet. La tercera generación es imperativa: se escribe un modelo como se escribe Python numérico corriente, una idea que lanzó Chainer y que adoptaron PyTorch, Gluon de MXNet y JAX.

Para recordar

Las redes neuronales alternan capas lineales y capas que curvan, y la retropropagación usa la regla de la cadena, multiplicando los efectos a lo largo del camino, para ajustar todas las perillas a la vez. Se estancaron aproximadamente entre 1995 y 2005 porque el cálculo escaseaba y los conjuntos de datos eran pequeños, así que ganaban los métodos de núcleo y los árboles de decisión. De 1970 a 2020 los datos crecieron diez mil millones de veces mientras la memoria creció cien millones de veces, y el cálculo creció aún más deprisa, lo que favoreció a las redes profundas. El dropout, la atención, el Transformer, los modelos de lenguaje, las redes antagónicas y los modelos de difusión impulsaron un estallido de progreso. El entrenamiento repartido entre miles de chips, y marcos como PyTorch, lo hicieron rápido y ampliamente accesible.

Preguntas

1. ¿Qué usa la retropropagación para averiguar, para cada perilla a la vez, hacia dónde girarla?
 - a. Pruebas al azar de cada perilla
 - b. La regla de la cadena, multiplicando los efectos a lo largo de las etapas desde el final hasta el principio
 - c. Una tabla de las mejores posiciones halladas antes
 - d. El juicio humano sobre cada capa
2. ¿Por qué los métodos de núcleo y los árboles de decisión ganaban a las redes neuronales hacia 1995-2005?
 - a. Porque se había demostrado que las redes neuronales estaban equivocadas
 - b. Porque la memoria era demasiado escasa para ellas
 - c. Porque el cálculo escaseaba y los conjuntos de datos eran pequeños, y esos métodos se entrenaban más deprisa y con garantías sólidas
 - d. Porque usaban chips gráficos
3. De 1970 a 2020, los datos crecieron mucho más que la memoria. ¿Hacia dónde empujó eso a los mejores métodos?
 - a. Hacia modelos ahorradores de memoria que gastan mucho cálculo en ajustar sus perillas
 - b. Hacia modelos que guardan cada ejemplo
 - c. Hacia modelos sin ninguna perilla
 - d. Hacia conjuntos de datos más pequeños

4. ¿Qué problema resolvió la atención en la traducción?
- a. Hizo innecesarios los datos de entrenamiento
 - b. Quitó el ruido de las imágenes
 - c. Sustituyó al chip gráfico
 - d. Permitió al modelo apuntar a la parte de la frase que necesita en lugar de comprimir toda la frase en un resumen fijo

Respuestas

1. **b)** La retropropagación aplica la regla de la cadena desde el final de la red hacia el principio y así sabe, para cada perilla en una sola pasada, hacia dónde girarla.
2. **c)** Con tan pocos datos y tan poco cálculo, otras herramientas funcionaban mejor; no necesitaban semanas de entrenamiento y daban resultados predecibles con garantías.
3. **a)** Eso empujó los mejores métodos hacia modelos ahorradores de memoria pero capaces de gastar un cálculo generoso en ajustar sus perillas.
4. **d)** En lugar de comprimir una frase entera en un resumen de tamaño fijo, un modelo con atención mantiene un puntero aprendible hacia la parte que necesita.

Ejemplos resueltos

Un paso enorme entre muchos chips El entrenamiento se reparte entre 1.024 chips gráficos, y cada chip toma un lote de 32 imágenes en cada paso. ¿Cuántas imágenes maneja un paso en total?

Cada chip maneja 32 imágenes por paso.

Los 1.024 chips trabajan a la vez, así que se multiplica.

1.024 por 32 son 32.768.

Son unas 32.000 imágenes por paso, lo que mantiene ocupados a todos los chips.

Respuesta: 32.768 imágenes, unas 32.000

Ejercicios

1. Usando la regla de la cadena tal como la cuenta la lección: girar una perilla sube la salida de una capa 4 unidades por unidad de giro, y cada unidad de esa salida baja la pérdida 0,5. ¿Cuál es el efecto de la perilla sobre la pérdida, y hacia dónde debería girarla el descenso de gradiente?

Pista: Multiplique los efectos a lo largo del camino, fijándose en las subidas y las bajadas.

Respuesta: 4 por menos 0,5 da menos 2: cada unidad de giro baja la pérdida 2. Como girar la perilla hacia arriba baja la pérdida, el descenso de gradiente la gira hacia arriba.

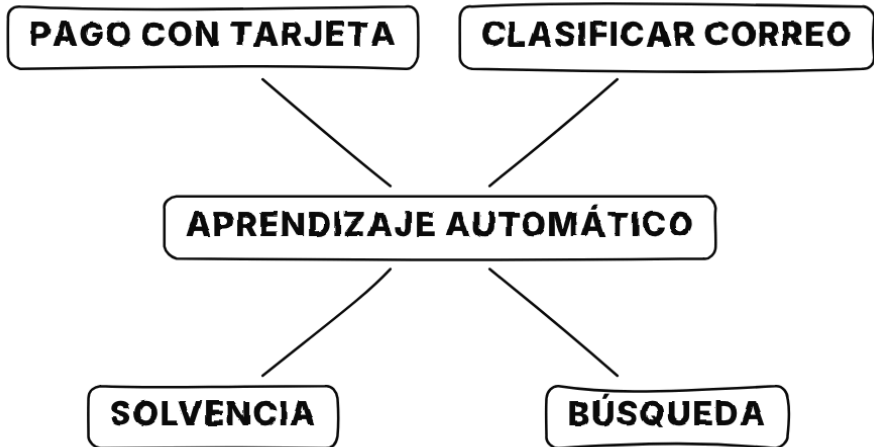
2. La memoria creció de 1 kilobyte en 1970 a 100 gigabytes en 2020, mientras los conjuntos de datos crecieron de unos 100 ejemplos a alrededor de un billón. ¿En qué factor creció cada uno, y qué dice esto de guardar conjuntos de datos enteros en memoria?

Pista: Un gigabyte es un millón de kilobytes.

Respuesta: 100 gigabytes son 100 millones de kilobytes, así que la memoria creció 100 millones de veces; los datos crecieron de 100 a un billón, 10 mil millones de veces. Los datos crecieron cien veces más deprisa, así que los conjuntos de datos enteros caben cada vez menos en memoria, y los métodos tuvieron que volverse ahorradores de memoria.

Éxitos silenciosos y preocupaciones reales

Pague algo en internet con una tarjeta y, en el instante antes de que se apruebe el pago, un programa ya ha juzgado si parece un fraude. Envíe una carta, y muy probablemente una máquina habrá leído la dirección escrita a mano. Ninguna de las dos cosas parece inteligencia artificial. Pero ambas son aprendizaje automático, y ambas llevan años funcionando, casi siempre sin que se vean.



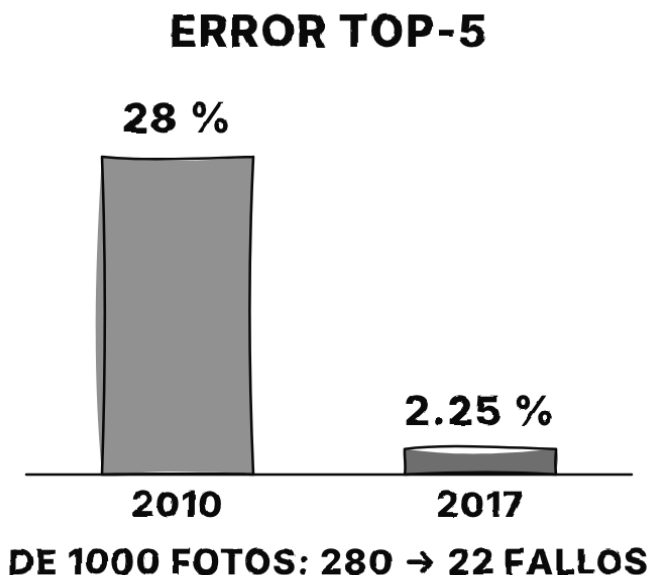
Aprendizaje automático, fuera de la vista

Una de las razones de que el campo haya avanzado tan deprisa es la división del trabajo. Unos investigadores construyen mejores herramientas, otros mejores modelos, y cada lado se beneficia del otro. El libro da un ejemplo revelador. En 2014, entrenar una regresión logística lineal, uno de los clasificadores más sencillos, todavía era un problema de deberes que merecía la pena poner a los nuevos doctorandos de aprendizaje automático de la Universidad Carnegie Mellon. Hoy lleva menos de diez líneas de código, al alcance de cualquier programador.

Algunos éxitos son antiguos. Las máquinas clasificadoras de correo que leen direcciones escritas a mano se usan desde los años noventa, y de ahí salió la famosa colección de dígitos escritos a mano. Las máquinas leen cheques para ingresos bancarios y puntúan si quienes piden un préstamo son solventes. Los pagos se revisan automáticamente en busca de fraude, y eso forma la columna vertebral de sistemas de pago como PayPal, Stripe, Visa y Mastercard. Los programas de ajedrez son competitivos desde hace décadas. Y el aprendizaje automático alimenta la búsqueda, la recomendación, la personalización y la ordenación en todo internet. El orden de los vídeos de su sección de recomendaciones es uno de esos resultados silenciosos. En resumen, el aprendizaje automático está en todas partes, aunque a menudo oculto.

La inteligencia artificial ha saltado a primer plano solo hace poco, porque empezó a resolver problemas que antes se consideraban inabordables, problemas con los que los consumidores corrientes se encuentran directamente, y muchos de esos avances se deben al aprendizaje profundo. Los asistentes de voz como Siri, Alexa y el asistente de Google responden a peticiones habladas con una precisión razonable, desde encender una luz hasta pedir cita en la

barbería, y eso es probablemente la señal más notable de la inteligencia artificial en la vida diaria. Su ingrediente clave, el reconocimiento de voz, ha mejorado hasta igualar a las personas en ciertos usos.



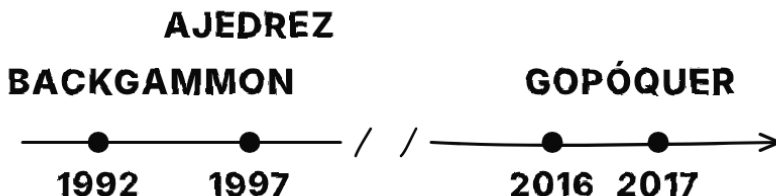
Errores en ImageNet, de 2010 a 2017

El reconocimiento de objetos ha avanzado otro tanto. En la prueba de ImageNet, un modelo cuenta como equivocado en una foto si la respuesta correcta no está entre sus cinco mejores estimaciones, y la proporción de esos fallos es su tasa de error top-5. En 2010 un equipo de NEC Labs y la Universidad de Illinois alcanzó una tasa de error top-5 del veintiocho por ciento. En 2017 había caído al dos coma veinticinco por ciento. De cada mil fotos, eso es una caída de doscientos ochenta fallos a unos veintidós, más de doce veces menos.

Ha habido resultados igual de llamativos en la identificación de pájaros por su canto y en el diagnóstico del cáncer de piel.

Los juegos fueron en su día una vara de medir la capacidad humana, y las máquinas la han escalado. TD-Gammon aprendió backgammon con una forma de aprendizaje por refuerzo. El ajedrez tiene muchas más posiciones y movimientos, y Deep Blue venció a Garri Kaspárov con cálculo masivamente paralelo, hardware especial y una búsqueda eficiente en el árbol de partidas posibles. El go es más difícil todavía, con un número de posiciones aún más inmenso, y AlphaGo alcanzó el nivel humano en 2015 combinando el aprendizaje profundo con una forma de muestrear continuaciones probables de la partida, llamada muestreo de árbol de Monte Carlo. El póquer añade una dificultad distinta: el estado es grande y solo parcialmente visible, porque no se ven las cartas de los otros jugadores, y aun así un programa llamado Libratus superó a jugadores humanos con estrategias cuidadosamente estructuradas.

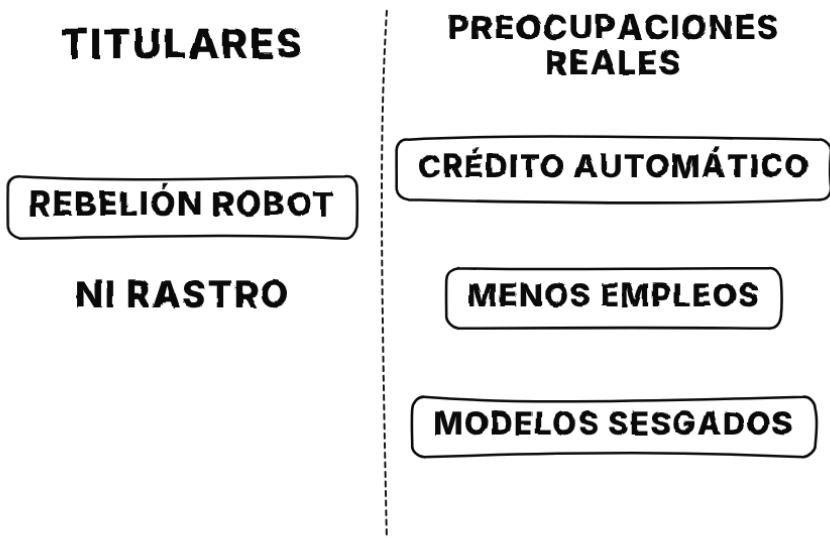
JUEGOS GANADOS POR MÁQUINAS



Juegos que las máquinas aprendieron a ganar

Los vehículos autónomos son otra señal de progreso. La autonomía total aún no está al alcance, pero empresas como Tesla, NVIDIA y Waymo venden productos que conducen solos parte del tiempo. La autonomía total es difícil porque conducir bien exige percibir, razonar y seguir normas; por ahora el aprendizaje profundo se ocupa sobre todo de percibir, y los ingenieros ajustan el resto a mano. Además, la robótica, la logística, la biología computacional, la física de partículas y la astronomía deben algunos de sus avances recientes más impresionantes, al menos en parte, al aprendizaje automático, que se está convirtiendo en una herramienta cotidiana de ingenieros y científicos.

Los artículos de divulgación suelen avivar el miedo a un apocalipsis de la inteligencia artificial, la llamada singularidad, en la que las máquinas se vuelven conscientes y deciden nuestro destino sin sus programadores. En cierta medida la inteligencia artificial ya influye directamente en la vida de las personas: la solvencia se juzga de forma automática, los pilotos automáticos se encargan de la mayor parte de la navegación de muchos vehículos, y las decisiones sobre conceder la libertad bajo fianza usan datos estadísticos. En un plano más frívolo, podemos pedirle a un asistente de voz que encienda la cafetera.



Miedos de titular, preocupaciones reales

Los autores tranquilizan en cuanto al apocalipsis. Primero, los sistemas actuales se diseñan, entrenan y despliegan para objetivos concretos; por general que parezca su comportamiento, por debajo

hay una combinación de reglas, reglas prácticas y modelos estadísticos. Segundo, sencillamente no hay hoy herramientas para una inteligencia artificial general que pudiera mejorarse a sí misma, razonar sobre sí misma y rediseñar su propia estructura mientras resuelve tareas generales.

La preocupación urgente, dicen, es cómo se usa la inteligencia artificial en la vida diaria. Muchas tareas rutinarias que hoy hacen las personas pueden automatizarse y se automatizarán. Los robots agrícolas probablemente reducirán costes a los agricultores ecológicos, pero también automatizarán la cosecha, y como ese trabajo manual da buena parte del empleo en muchos países, esta etapa de la revolución industrial podría tener consecuencias profundas para amplios sectores de la sociedad. Y los modelos estadísticos aplicados sin cuidado pueden tener sesgos de raza, género o edad, lo que plantea preocupaciones reales de justicia cuando guían decisiones con consecuencias graves. Una decisión sobre un préstamo o un empleo tomada por un modelo entrenado con las decisiones injustas de ayer puede repetirlas automáticamente, como vimos con el ejemplo de la contratación. Estos algoritmos deben usarse con cuidado, y los autores lo consideran mucho más urgente que cualquier superinteligencia malévola.

Para recordar

Las mejores herramientas convirtieron lo que en 2014 era un problema de deberes de doctorado en menos de diez líneas de código. El aprendizaje automático lleva mucho tiempo trabajando en silencio en la clasificación del correo, la lectura de cheques, la detección de fraudes y la ordenación en internet. El aprendizaje profundo trajo saltos visibles: asistentes de voz, reconocimiento de voz, reconocimiento de imágenes con errores reducidos del veintiocho por ciento a cerca del dos, y victorias en backgammon, ajedrez, go y póquer. Los coches autónomos y muchas ciencias avanzan con él, aunque la autonomía total todavía no ha llegado. Las preocupaciones reales no son una máquina consciente, sino los empleos perdidos y las decisiones automáticas sesgadas, que exigen un uso cuidadoso.

Preguntas

1. ¿Qué significa una tasa de error top-5 del veintiocho por ciento?
 - a. La primera estimación del modelo falla en 28 de cada 100 fotos
 - b. En 28 de cada 100 fotos, la respuesta correcta no está entre las cinco mejores estimaciones del modelo
 - c. El modelo necesita cinco intentos para llegar al 28 por ciento de acierto
 - d. El 28 por ciento de las fotos tiene cinco objetos

2. ¿Por qué la conducción totalmente autónoma sigue fuera de alcance, según la lección?
 - a. Porque conducir exige percibir, razonar y seguir normas, y el aprendizaje profundo se ocupa sobre todo solo de percibir
 - b. Porque los coches no pueden llevar chips gráficos
 - c. Porque no hay datos sobre las carreteras
 - d. Porque las leyes lo prohíben en todas partes
3. ¿Por qué a los autores no les preocupa hoy una IA consciente y maquinadora?
 - a. Porque los sistemas de IA son demasiado lentos
 - b. Porque la IA tiene prohibido intervenir en decisiones importantes
 - c. Porque los sistemas actuales se construyen para objetivos concretos y no hay herramientas para una inteligencia general que se mejore a sí misma
 - d. Porque la IA no sabe jugar
4. ¿Qué preocupación consideran los autores más urgente?
 - a. La singularidad
 - b. Que los ordenadores se queden sin memoria
 - c. Que los robots jueguen al ajedrez demasiado bien
 - d. La automatización de trabajos rutinarios y las decisiones sesgadas de modelos usados sin cuidado

Respuestas

1. b) Un modelo cuenta como equivocado en una foto si la respuesta correcta no está entre sus cinco mejores estimaciones, y la proporción de esos fallos es la tasa de error top-5.

2. a) Conducir bien exige percibir, razonar y seguir normas; el aprendizaje profundo se ocupa sobre todo de percibir, y los ingenieros ajustan el resto a mano.

3. c) Los sistemas se diseñan para objetivos concretos, y sencillamente no hay hoy herramientas para una inteligencia artificial general que pudiera mejorarse y rediseñarse a sí misma.

4. d) La preocupación urgente es cómo se usa la IA en la vida diaria: trabajos rutinarios automatizados, y modelos aplicados sin cuidado con sesgos de raza, género o edad.

Ejemplos resueltos

Cuánto mejoró el reconocimiento de imágenes En la prueba de ImageNet, la tasa de error top-5 era del 28 por ciento en 2010 y del 2,25 por ciento en 2017. ¿Cuántos fallos supone eso de cada 1.000 fotos en cada año, y aproximadamente cuántas veces menos?

El 28 por ciento de 1.000 fotos son 280 fallos.

El 2,25 por ciento de 1.000 fotos son 22,5, unos 22 fallos.

Divida la cifra antigua entre la nueva: 280 entre 22,5 es unos 12,4.

Así que los fallos bajaron a aproximadamente una doceava parte.

Respuesta: 280 fallos frente a unos 22; más de doce veces menos

Ejercicios

1. Un modelo tiene una tasa de error top-5 del 5 por ciento en 2.000 fotos de prueba. ¿En cuántas fotos falta la respuesta correcta entre sus cinco mejores estimaciones?

Pista: Calcule el 5 por ciento de 2.000.

Respuesta: El 5 por ciento de 2.000 son 100 fotos.

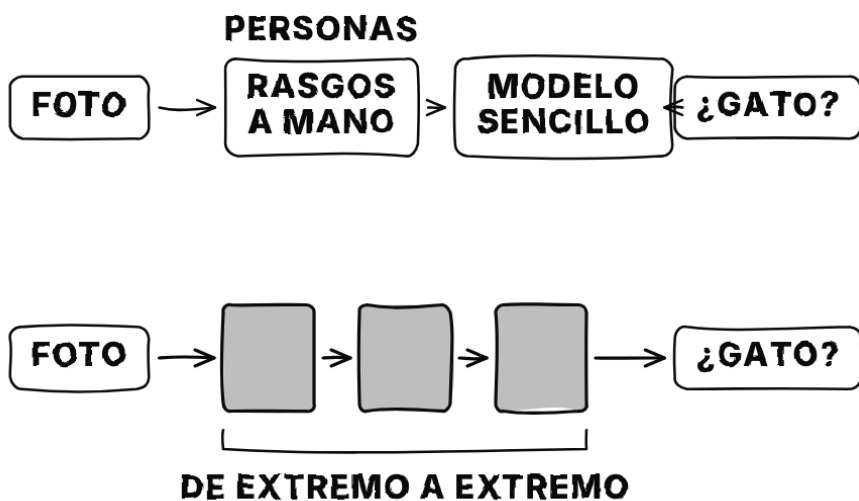
2. Explique cómo cambiaron las mejores herramientas quién puede usar el aprendizaje automático, con el ejemplo del libro de la regresión logística.

Pista: Compare 2014 con hoy.

Respuesta: En 2014, entrenar una regresión logística era un problema de deberes digno de nuevos doctorandos; hoy, gracias a herramientas construidas por investigadores de sistemas, lleva menos de diez líneas de código, así que cualquier programador puede hacerlo. La división del trabajo entre quienes construyen herramientas y quienes construyen modelos abrió el campo a muchísimas más personas.

Qué lo hace profundo

Imagine enseñar a un ordenador a detectar gatos como se hacía hace quince años. Primero, unas personas listas escriben un programa que encuentra bordes, esquinas y pequeñas zonas con textura en una foto. Sus hallazgos, una lista de números, van a un modelo de aprendizaje sencillo que solo tiene que decidir «gato» o «no gato». Si el primer paso hecho a mano pasa por alto algo que importa, nada de lo que el modelo aprenda después podrá recuperarlo. El aprendizaje profundo elimina esa división entre una parte delantera hecha a mano y una parte trasera aprendida, y a eso, al final, es a lo que conducía toda esta introducción.

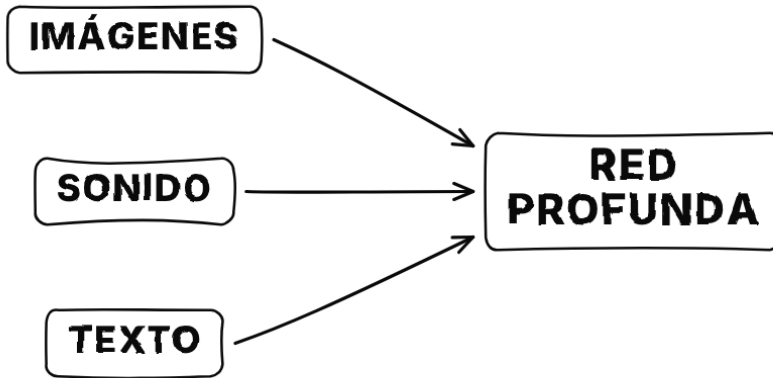


Parte delantera artesanal, o de extremo a extremo

El aprendizaje profundo es la parte del aprendizaje automático que se ocupa de modelos construidos con redes neuronales de muchas capas. Es profundo justo en este sentido: sus modelos aprenden muchas capas de transformaciones, cada una de las cuales remodela lo que recibe y lo pasa adelante. Puede sonar a definición estrecha, pero ha producido una variedad vertiginosa de modelos, técnicas, tipos de problemas y usos. En cierto modo, todo aprendizaje automático tiene varias capas de cálculo, la primera de las cuales es algún procesamiento de las características. Lo que distingue al aprendizaje profundo es que las operaciones de cada una de sus muchas capas de representación se aprenden conjuntamente, todas a la vez, a partir de los datos.

Por eso el aprendizaje profundo sobresale justo donde los métodos tradicionales fallaban: aprender del sonido en bruto, de los valores de píxel en bruto de las imágenes, o traducir frases de cualquier longitud a otro idioma. Estos modelos de muchas capas pueden manejar datos perceptivos de bajo nivel de una forma que las herramientas anteriores no podían. Lo más importante que comparten los métodos de aprendizaje profundo es el entrenamiento de extremo a extremo: en lugar de montar un sistema con piezas que se ajustan cada una por su cuenta, se construye el sistema entero y se ajusta todo a la vez.

La visión por computadora muestra el cambio. Durante años, los científicos mantuvieron la ingeniería de características, el diseño a mano de formas de convertir imágenes en números, separada de la construcción del modelo de aprendizaje. Dos métodos diseñados a mano, el detector de bordes de Canny y el extractor de características SIFT de Lowe, reinaron durante más de una década como las formas estándar de convertir una imagen en un vector de características. En aquellos tiempos, la parte crucial de aplicar el aprendizaje automático era inventar ingeniosas transformaciones hechas a mano que un modelo poco profundo pudiera manejar. Pero el ingenio humano solo llega hasta cierto punto frente a un algoritmo que evalúa sistemáticamente millones de opciones de forma automática. Cuando el aprendizaje profundo tomó el relevo, los extractores de características hechos a mano se sustituyeron por filtros ajustados automáticamente, y la precisión subió.



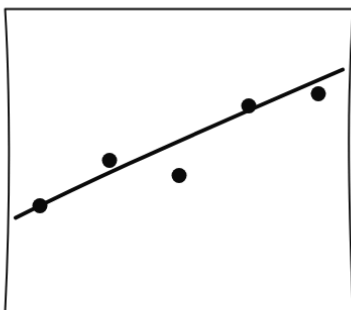
Un tipo de red para todos los campos

Así que el aprendizaje profundo sustituye no solo el modelo poco profundo del final de la vieja cadena, sino también la laboriosa ingeniería de características previa. Y como gran parte de ese preprocesamiento era propio de un solo campo, el aprendizaje profundo ha eliminado muchas de las fronteras entre la visión por computadora, el reconocimiento de voz, el procesamiento del lenguaje, la informática médica y otras áreas, ofreciendo un conjunto compartido de herramientas para problemas muy distintos. Por eso el asistente de voz que le oye y el chatbot que le responde por escrito pueden construirse con la misma familia de modelos.

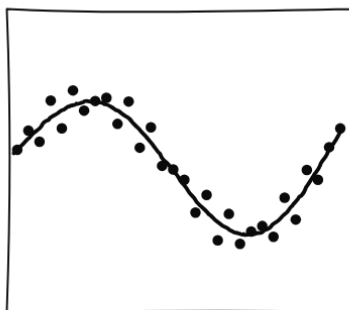
Hay otro cambio más, de los modelos paramétricos a los totalmente no paramétricos. Cuando los datos escasean, un modelo útil tiene que

apoyarse en suposiciones simplificadoras sobre la realidad, unos pocos parámetros que describen una forma fija. Cuando los datos abundan, esas suposiciones pueden dar paso a modelos que siguen los datos con más libertad. El libro lo compara con la física de mediados del siglo pasado: con la llegada de los ordenadores, los físicos pudieron simular numéricamente las ecuaciones del comportamiento de los electrones en lugar de resolver a mano aproximaciones simplificadas. Los modelos se volvieron mucho más precisos, aunque a menudo más difíciles de interpretar.

POCOS: SUPONER



MUCHOS: SEGUIR



Suponer menos cuando hay muchos datos

El aprendizaje profundo se diferencia también de los trabajos anteriores en su actitud. Acepta soluciones buenas en lugar de demostrablemente óptimas, aborda problemas de optimización con

muchas colinas y valles, y está dispuesto a probar las cosas antes de demostrarlas. Este empirismo, junto con una rápida llegada de talento, ha traído un progreso rápido en métodos prácticos, aunque a veces a costa de reinventar herramientas que existían desde hacía décadas. Y la comunidad se enorgullece de compartir herramientas entre universidades y empresas, publicando libremente bibliotecas, modelos y redes entrenadas; con el mismo espíritu, los cuadernos del propio libro son libres de usar y compartir.

El resumen del libro une los hilos. El aprendizaje automático estudia cómo los sistemas informáticos usan la experiencia, normalmente datos, para mejorar en tareas concretas, combinando ideas de la estadística, la minería de datos y la optimización, y a menudo es el medio para construir inteligencia artificial. El aprendizaje de representaciones, un tipo de aprendizaje automático, trata de encontrar automáticamente la forma adecuada de representar los datos, y el aprendizaje profundo es aprendizaje de representaciones en muchos niveles, a través de muchas capas aprendidas. Su progreso reciente lo desencadenaron la abundancia de datos de sensores baratos y aplicaciones a escala de internet, grandes avances en el cálculo, sobre todo gracias a los chips gráficos, y marcos eficientes que facilitan mucho diseñar y optimizar sistemas completos. Con eso termina la introducción: de programas que siguen reglas fijas a sistemas que aprenden sus propias capas de descripción a partir de los datos.

Para recordar

El aprendizaje profundo es aprendizaje automático con redes neuronales de muchas capas, en las que cada capa de representación se aprende conjuntamente a partir de los datos. Su rasgo distintivo es el entrenamiento de extremo a extremo: todo el sistema se ajusta a la vez en lugar de por piezas ajustadas por separado. Sustituyó tanto los modelos poco profundos como la ingeniería de características hecha a mano, como los detectores de bordes, y reunió la visión, el habla y el lenguaje bajo un mismo conjunto de herramientas. Con abundancia de datos puede prescindir de suposiciones simplificadoras, y prefiere probar las cosas en la práctica antes que demostrarlas primero. Su auge se apoya en datos abundantes, chips gráficos rápidos y marcos gratuitos y eficientes.

Preguntas

1. Muchos sistemas de aprendizaje automático tienen varias etapas. ¿Qué hace distinto al aprendizaje profundo?
 - a. Sus etapas funcionan en chips gráficos
 - b. Las operaciones de todas sus muchas capas se aprenden conjuntamente a partir de los datos
 - c. Tiene exactamente tres capas
 - d. No necesita ningún tipo de preparación de datos

2. ¿Por qué los filtros ajustados automáticamente sustituyeron a extractores diseñados a mano como SIFT?
 - a. Porque los extractores diseñados a mano eran demasiado lentos
 - b. Porque un algoritmo que evalúa millones de opciones puede superar al ingenio humano, y la precisión subió
 - c. Porque SIFT dejó de funcionar con las cámaras nuevas
 - d. Porque los filtros no necesitan entrenamiento
3. ¿Cuándo puede un modelo prescindir de suposiciones simplificadoras y seguir los datos con más libertad?
 - a. Cuando los datos escasean
 - b. Cuando el modelo es pequeño
 - c. Cuando los datos abundan
 - d. Cuando el problema trata de física
4. ¿Qué ganó y qué perdió el paso a la simulación por ordenador en física, según la comparación del libro?
 - a. Ganó velocidad y perdió precisión
 - b. Ganó precisión pero a menudo perdió facilidad de interpretación
 - c. Ganó interpretación pero perdió precisión
 - d. Perdió tanto precisión como interpretación

Respuestas

1. b) Lo que distingue al aprendizaje profundo es que las operaciones de cada una de sus muchas capas de representación se aprenden conjuntamente, todas a la vez, a partir de los datos.

- 2. b)** El ingenio humano solo llega hasta cierto punto frente a un algoritmo que evalúa millones de opciones automáticamente; los filtros ajustados dieron mejor precisión.
- 3. c)** Cuando los datos escasean, los modelos se apoyan en suposiciones simplificadoras; cuando abundan, esas suposiciones pueden dar paso a modelos que siguen los datos.
- 4. b)** Los modelos se volvieron mucho más precisos, aunque a menudo más difíciles de interpretar.

Ejercicios

- 1.** Explique con sus palabras qué es el entrenamiento de extremo a extremo y por qué puede superar a una cadena de piezas ajustadas por separado.

Pista: Piense en el detector de gatos con un buscador de bordes hecho a mano delante.

Respuesta: El entrenamiento de extremo a extremo construye el sistema completo y lo ajusta todo a la vez con los datos, en lugar de ajustar cada pieza por su cuenta. En una cadena, un primer paso hecho a mano puede desechar algo que importa, y ningún aprendizaje posterior puede recuperarlo; cuando todas las capas se aprenden juntas, las primeras aprenden a conservar justo lo que necesitan las siguientes.

- 2.** Resuma en tres frases las tres condiciones a las que el libro atribuye el progreso reciente del aprendizaje profundo.

Pista: Datos, hardware y software.

Respuesta: Los datos abundantes de sensores baratos y aplicaciones a escala de internet dieron a los modelos suficientes ejemplos. Los avances en el cálculo, sobre todo los chips gráficos, hicieron viable entrenar redes grandes. Los marcos eficientes facilitaron mucho diseñar, implementar y optimizar sistemas completos.