

King of Dumbs

el rey de los tontos

Nivel: ♦ Intermedio

¿Cómo convierte un ordenador una foto en números?

Una foto son millones de números, y casi todo lo que la IA hace con ella es multiplicar y sumar. Vectores, producto escalar y matrices en tres niveles, con dos widgets.

Español Publicado 17 de octubre de 2026



Escanea para abrir la lección interactiva

<https://kingofdumbs.com/es/lecciones/vectores-y-matrices>

Qué vas a aprender

- Una foto es una lista de números: vectores y dimensión
- El producto escalar: multiplicar y sumar, y cuánto apuntan igual dos flechas
- Una matriz mueve el plano: sus columnas, el área y las dimensiones que quedan
- Multiplicar matrices: encadenar cambios, por qué importa el orden y qué hace una red neuronal

Símbolos que aparecen

E_a subíndice: la etiqueta de abajo (♠) · Σ sumatorio: súmalos todos (♠) · x_i índice i : el elemento número i (♠) · $\approx$ aproximadamente igual (♠) · $\leq \geq$ menor o igual / mayor o igual (♠) · $|x|$ valor absoluto: sin signo (♠) · punto: multiplicar (o producto escalar) (♠) · σ sigma: depende de la lección (♠) · θ theta: un ángulo (casi siempre) (♠) · λ lambda: un número que estira (♠) · $\|x\|$ norma: la longitud de la flecha (♠) · x^T traspuesta: tumbar la columna (♠) · $\in$ pertenece a (♠) · $\mathbb{R} \mathbb{Z}$ conjuntos de números (♠)

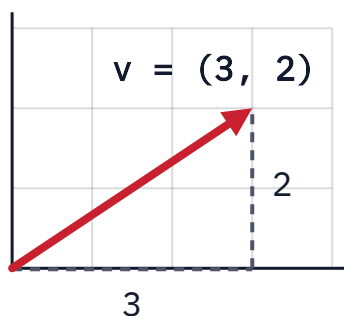
Haces una foto a tu gato y, si buscas “gato” en la galería del móvil, muchas veces la encuentra. Pero un ordenador no ve gatos: lo único que sabe hacer es sumar y multiplicar números, muy deprisa.

La pregunta tonta de esta lección es: **¿cómo convierte un ordenador una foto en números?** Y, ya puestos, ¿qué hace con ellos? La respuesta cabe en tres ideas que suenan a examen, vector, producto escalar y matriz, y en una sola cuenta: multiplicar y sumar.

Elige tu carta: ♣ si empiezas de cero, ♦ si no te asusta una fórmula sencilla, ♠ si quieres la notación. Puedes cambiarla en cualquier momento.

♣ ♦ ♠

Una foto es una lista de números



longitud $\sqrt{(3^2 + 2^2)} \approx 3,6$

2 números: el plano

3 números: el espacio

36 millones: no se dibuja, pero se calcula igual

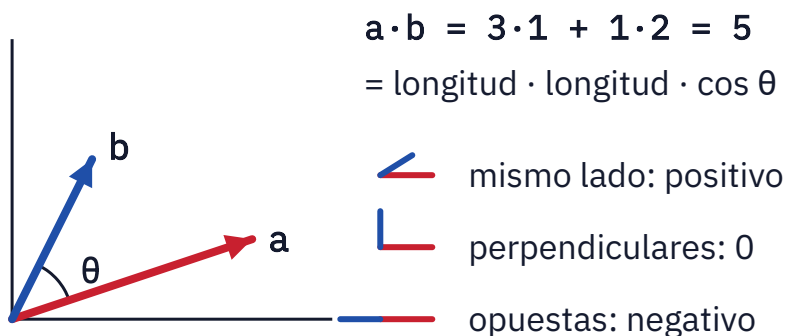
Un **vector** es una lista ordenada de números, sus **componentes**. Con dos componentes se dibuja como una flecha que sale del centro: $v = (3, 2)$ es “3 a la derecha y 2 arriba”. Su **longitud** sale por Pitágoras: $\sqrt{(3^2 + 2^2)} = \sqrt{13} \approx 3,6$ (OpenStax).

Una foto en gris de dos píxeles sería una flecha así, en un plano; con tres píxeles, una flecha en el espacio. Con más ya no se puede dibujar, pero las cuentas son las mismas: la longitud sigue siendo la raíz de la suma de los cuadrados. La **dimensión** es cuántos números tiene el vector.

En una foto en color cada píxel lleva tres números de 0 a 255, uno por **canal** (rojo, verde y azul), porque cada uno ocupa 8 bits (Pillow, una librería de imágenes de Python). Así que una foto abierta es una tabla de **alto × ancho × 3** números. El fichero JPEG ocupa menos porque va comprimido, pero al abrirlo se descomprime en esa tabla. Para dársela a una red neuronal **se pone toda en fila**: $4000 \times 3000 \times 3 = 36$ millones de números.



El producto escalar: multiplicar y sumar



El **producto escalar** de $a = (a_1, a_2)$ y $b = (b_1, b_2)$ es $a \cdot b = a_1 b_1 + a_2 b_2$: se multiplican las componentes una a una y se suman (OpenStax). Con más componentes, igual. Con $a = (3, 1)$ y $b = (1, 2)$: $3 \cdot 1 + 1 \cdot 2 = 5$. Con dos fotos de 36 millones de números, son **36 millones de multiplicaciones**.

Con flechas tiene otra lectura: $a \cdot b = \text{longitud de } a \times \text{longitud de } b \times \text{coseno del ángulo}$ que forman (OpenStax). **El coseno vale 1 si apuntan igual, 0 si forman un ángulo recto y -1 si son opuestas**. Por eso dos flechas **perpendiculares** dan 0.

Si divides el producto escalar entre las dos longitudes queda solo el coseno, que mide cuánto se parecen las direcciones: es la **similitud coseno**. Con fotos tiene una gracia. Si multiplicas todos los números de una foto por 1,2, la ves más clara: **la flecha se alarga pero no gira**, y su similitud con la original sigue siendo 1 (cálculo propio, mientras ningún número pase de 255).

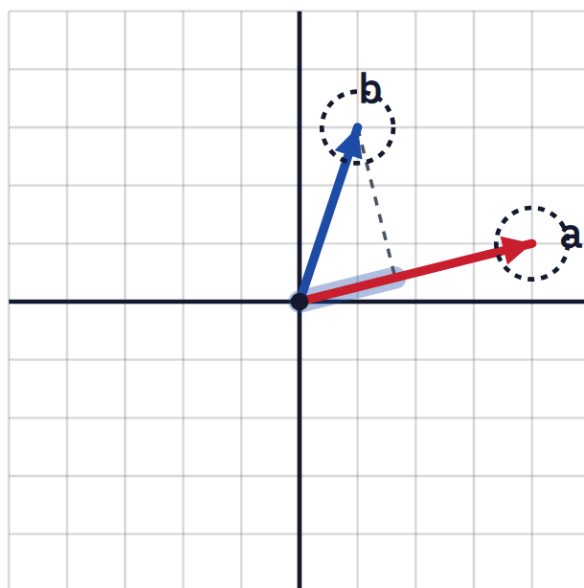
El Rey de los Tontos dice:

Multiplicar y sumar. No hay más. Con eso comparo precios, flechas y fotos, y aún me sobra tiempo para la siesta.



Prueba tú: dos flechas

Arrastra las puntas de las flechas, o muévelas con los deslizadores o con las flechas del teclado. Busca una posición en la que el producto escalar valga 0: las flechas quedarán en ángulo recto. La franja azul es la sombra de b sobre a : se alarga cuanto más apuntan hacia el mismo lado.



a, horizontal

4

a, vertical

1

b, horizontal

1

b, vertical

3

Producto escalar $a \cdot b$

$$4 \cdot 1 + 1 \cdot 3 = 7$$

Longitudes

$$\|a\| = 4,12 \cdot \|b\| = 3,16$$

Ángulo

$$57,5^\circ$$

Coseno (parecido de dirección) 0,54

$a = (4, 1)$, $b = (1, 3)$, $a \cdot b = 7$. Apuntan hacia el mismo lado: el producto escalar es positivo.



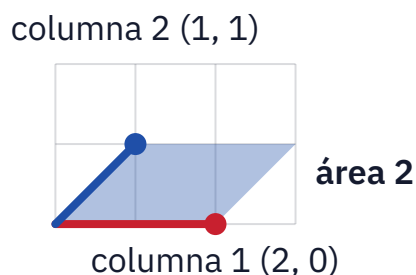
Widget interactivo. Pruébalo en la web: <https://kingofdumbs.com/es/lecciones/vectores-y-matrices>

Una matriz mueve el plano

$$\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

fila 1: $2 \cdot 1 + 1 \cdot 2 = 4$

fila 2: $0 \cdot 1 + 1 \cdot 2 = 2$



Una **matriz** es una tabla rectangular de números; una de m filas y n columnas es de $m \times n$ (OpenStax). Multiplicar una matriz por un vector es hacer el producto escalar de **cada fila** con el vector. Con $A = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$ y $x = (1, 2)$, la primera fila da $2 \cdot 1 + 1 \cdot 2 = 4$ y la segunda, $0 \cdot 1 + 1 \cdot 2 = 2$, así que $Ax = (4, 2)$.

Hay un truco para leerla: **la primera columna es adónde va (1, 0) y la segunda, adónde va (0, 1)** (Austin). En el ejemplo, (1, 0) va a (2, 0) y (0, 1) va a (1, 1): la matriz estira en horizontal y **tuerce**. Con eso sabes adónde va todo, porque **cualquier punto es una mezcla de esos dos**.

El cuadradito de lado 1 se convierte en un paralelogramo. Con la matriz $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, su área es **$ad - bc$** : en el ejemplo, $2 \cdot 1 - 1 \cdot 0 = 2$, el doble (Austin). Si da 0, la matriz **aplasta** el plano entero en una recta: se pierde una dimensión y no hay vuelta atrás. $\begin{pmatrix} 1 & -1 \\ 2 & -2 \end{pmatrix}$ lo hace. Si da negativo, el dibujo sale **como en un espejo**.

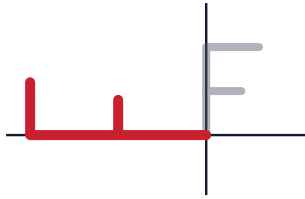
Pillow lo usa así para **girar** o torcer una foto: para cada píxel nuevo, calcula con una matriz **de qué punto de la original sale** (más un desplazamiento).

Prueba tú: deforma la cuadrícula

Mueve los cuatro números de la matriz o prueba los botones. La flecha roja es la primera columna, adónde va (1, 0); la azul, la segunda. Busca una matriz que aplaste la cuadrícula en una recta: el área se hará 0. Y mira qué le pasa a la F con el espejo.

Multiplicar matrices: el orden importa

girar 90° y luego estirar



$$S \cdot R = \begin{pmatrix} 0 & -2 \\ 1 & 0 \end{pmatrix}$$

estirar y luego girar 90°



$$R \cdot S = \begin{pmatrix} 0 & -1 \\ 2 & 0 \end{pmatrix}$$

R: girar 90° · S: estirar ×2 a lo ancho

Para **multiplicar dos matrices**, cada número del resultado es el producto escalar de **una fila de la primera por una columna de la segunda** (OpenStax). Solo se puede si la primera tiene tantas columnas como filas la segunda: **una de $m \times k$ por una de $k \times n$ da una de $m \times n$** .

Aplicar primero B y luego A a un punto da lo mismo que aplicar de una vez la matriz **AB**. Ojo: se escribe al revés de como se hace (Austin). Y **AB no suele ser igual que BA** (OpenStax). Con R, girar 90°, y S, estirar al doble a lo ancho: girar y luego estirar es $S \cdot R = \begin{pmatrix} 0 & -2 \\ 1 & 0 \end{pmatrix}$, y estirar y luego girar es $R \cdot S = \begin{pmatrix} 0 & -1 \\ 2 & 0 \end{pmatrix}$.

Una **capa** de una red neuronal multiplica su entrada por una matriz de **pesos** (PyTorch). Una foto en gris de 100×100 son 10 000 números; una capa con 1000 salidas usa una matriz de $1000 \times 10\,000$ y hace **10 millones de multiplicaciones** por foto (cálculo propio). Entre capa y capa hay **un paso que no es una multiplicación**; la lección «¿Regresión logística o red neuronal?» cuenta qué es y qué gana una red con él.

Las tarjetas gráficas meten **muchas fotos a la vez**, una por columna, y así todo se vuelve una sola multiplicación de matrices enorme (NVIDIA).

El Rey de los Tontos dice:

Yo no veo gatos: veo 36 millones de números. Lo difícil es saber por qué matrices multiplicarlos.



Reto final

1. Una foto en color de 100×100 píxeles. ¿Cuántos números guarda el ordenador al abrirla?

- a. 100
- b. 10 000, uno por píxel
- c. 30 000: tres por píxel, rojo, verde y azul

2. Dos flechas forman un ángulo recto. ¿Cuánto vale su producto escalar?

- a. 1
- b. 0
- c. Depende de lo largas que sean

3. Giras una foto 90° y luego la estiras a lo ancho. ¿Sale lo mismo que si la estiras primero y la giras después?

- a. Sí: multiplicar da lo mismo en cualquier orden
- b. No: con matrices, el orden importa
- c. Solo si la foto es cuadrada



En tres frases

1. Para un ordenador, una foto es una lista de números (tres por píxel, de 0 a 255): un vector, cuya dimensión es cuántos números tiene.
2. El producto escalar multiplica dos listas término a término y suma: da un total, como en la cuenta del frutero, y dice cuánto apuntan hacia el mismo lado dos flechas (0 si son perpendiculares).

3. Una matriz es una tabla que mueve cada punto del plano, y multiplicar matrices encadena esos cambios, con el orden importando: es la cuenta que una red neuronal hace millones de veces con cada foto.



Fuentes

- Vectores, longitud y producto escalar: OpenStax, *Calculus Volume 3*, [2.1](#) y [2.3](#) (el frutero, teoremas 2.4 y 2.5). Matrices y su producto: OpenStax, *College Algebra 2e*, [7.5](#).
- Transformaciones, determinante y dimensión: D. Austin, *Understanding Linear Algebra*, [2.2](#), [2.5](#), [2.6](#), [3.4](#) y [3.5](#).
- Píxeles y transformaciones de imágenes: Pillow 13.0, [Concepts](#) y [ImageTransform](#). De la foto al tensor: torchvision 0.29, [ToTensor](#).
- Redes neuronales: PyTorch 2.14, [nn.Linear](#) y [cosine_similarity](#). NVIDIA, [Matrix Multiplication Background](#) y [Linear/Fully-Connected Layers](#).

Cómo está hecha esta lección: investigada y redactada con ayuda de Claude a partir de las fuentes de arriba, consultadas el 17 de octubre de 2026. Las cifras sin enlace propio son cálculos a partir de sus definiciones. Los ejemplos (la foto de 4000×3000 , las dos tiendas, la capa de 1000 salidas) son inventados. Los dibujos son propios.

Soluciones

1. **c)** 30 000: tres por píxel, rojo, verde y azul. Eso es. $100 \times 100 \times 3$: un vector de dimensión 30 000.
2. **b)** 0. Bien. Perpendiculares, producto escalar 0, sean como sean de largas.
3. **b)** No: con matrices, el orden importa. Exacto. Como el calcetín y el zapato.

Lección de King of Dumbs, kingofdumbs.com. Versión para leer sin conexión; la original, con widgets que se tocan, está en: <https://kingofdumbs.com/es/lecciones/vectores-y-matrices>

Actualizada el 17 de octubre de 2026.