

King of Dumbs

Level: ♣ Newbie

How does a computer turn a photo into numbers?

A photo is millions of numbers, and almost everything AI does with it is multiply and add. Vectors, the dot product and matrices at three levels, with two widgets.

English Published October 17, 2026



Scan to open the interactive lesson

<https://kingofdumbs.com/en/lessons/vectors-and-matrices>

What you'll learn

- A photo is a list of numbers: vectors and dimension
- The dot product: multiply and add, and how much two arrows point the same way
- A matrix moves the plane: its columns, the area and the dimensions that are left
- Multiplying matrices: chaining changes, why order matters and what a neural network does

Symbols you'll meet

E_a subscript: the label underneath (♠) · Σ sum: add them all up (♠) · x_i index i: item number i (♠) · $\approx$ approximately equal (♠) · $\leq \geq$ less or equal / greater or equal (♠) · $|x|$ absolute value: drop the sign (♠) · $\cdot$ dot: multiply (or dot product) (♠) · σ sigma: depends on the lesson (♠) · θ theta: an angle (almost always) (♠) · λ lambda: a stretching number (♠) · $\|x\|$ norm: the length of the arrow (♠) · x^T transpose: lay the column down (♠) · $\in$ belongs to (♠) · $\mathbb{R} \mathbb{Z}$ sets of numbers (♠)

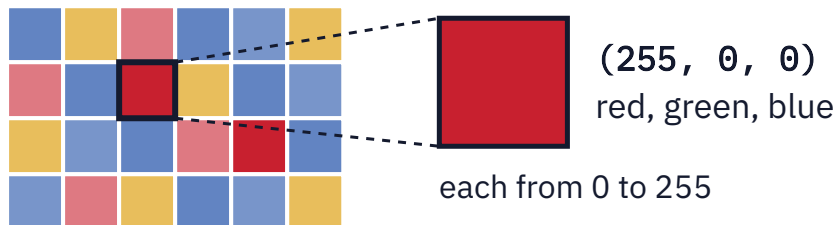
You take a photo of your cat and, if you search for “cat” in your phone’s gallery, it often finds it. But a computer doesn’t see cats: all it knows how to do is add and multiply numbers, very fast.

This lesson’s dumb question is: **how does a computer turn a photo into numbers?** And while we’re at it, what does it do with them? The answer fits into three ideas that sound like an exam, vector, dot product and matrix, and a single sum: multiply and add.

Pick your card: ♣ if you’re starting from zero, ♦ if a simple formula doesn’t scare you, ♠ if you want the notation. You can switch at any time.



A photo is a list of numbers



$4,000 \times 3,000 \times 3 = 36$ million numbers
in a row: a vector of dimension 36 million

A digital photo is like a mosaic: it's made of tiny coloured squares, the **pixels**, too small to see with the naked eye. When the computer opens a photo, it stores each square as three numbers: how much red, how much green and how much blue it has, each from 0 to 255. Pure red is (255, 0, 0); white, (255, 255, 255); and black, (0, 0, 0).

A photo 4,000 squares wide and 3,000 high has 12 million pixels. With three numbers each, that's 36 million numbers in a row.

A list of numbers in a row is called a **vector**, and how many numbers it has is its **dimension**. To the computer, your photo is a vector of dimension 36 million. Everything it does with it is sums with that list.



The dot product: multiply and add

	30 apples × \$0.50	= 15
	12 bananas × \$0.25	= 3
	18 oranges × \$1	= 18
		15 + 3 + 18 = \$36

Multiply one by one and add: the dot product.

A fruit seller sells 30 apples, 12 bananas and 18 oranges in a day, at \$0.50, \$0.25 and \$1 each. How much did they take? You multiply each quantity by its price and add: $15 + 3 + 18 = \$36$. That “multiply one by one and add” of two lists is the **dot product**, and it's the sum a computer does most with a photo. The example comes from a maths textbook, OpenStax.

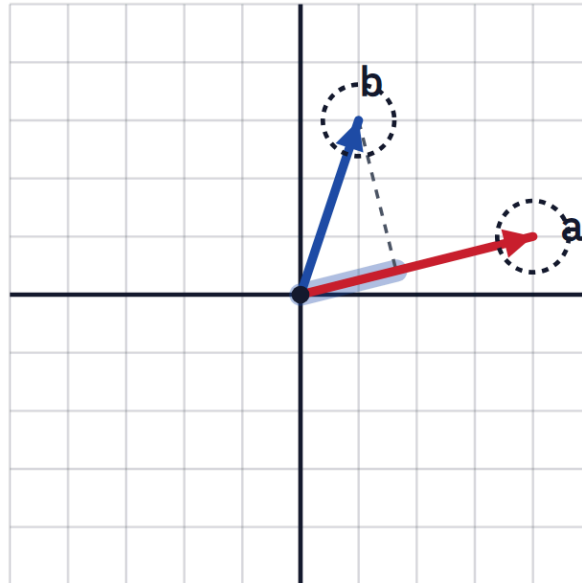
With arrows, the dot product tells you something else: how much they point the same way. If you push a trolley at an angle, only the part of the push that goes where it's moving counts. If two arrows point the same way, the dot product comes out positive; if they cross at a right angle, zero; and if they point in opposite directions, negative.

The King of Dumbs says:

Multiply and add. That's all there is. With that I compare prices, arrows and photos, and still have time left for a nap.

Try it: two arrows

Drag the tips of the arrows, or move them with the sliders or the arrow keys. Find a position where the dot product is 0: the arrows will end up at a right angle. The blue band is the shadow of b on a: it gets longer the more they point the same way.



a, across	<input type="range" value="4"/>	4	a, up	<input type="range" value="1"/>	1
b, across	<input type="range" value="1"/>	1	b, up	<input type="range" value="3"/>	3

Dot product $a \cdot b$ $4 \cdot 1 + 1 \cdot 3 = 7$

Lengths $\|a\| = 4.12 \cdot \|b\| = 3.16$

Angle 57.5°

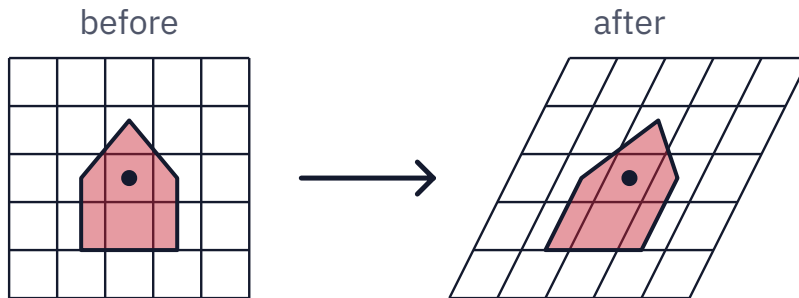
Cosine (how alike the directions are) 0.54

$a = (4, 1)$, $b = (1, 3)$, $a \cdot b = 7$. They point the same way: the dot product is positive.



Interactive widget. Try it on the web: <https://kingofdumbs.com/en/lessons/vectors-and-matrices>

A matrix moves the plane



The matrix says where each point goes.
Straight lines stay straight and the centre stays put.

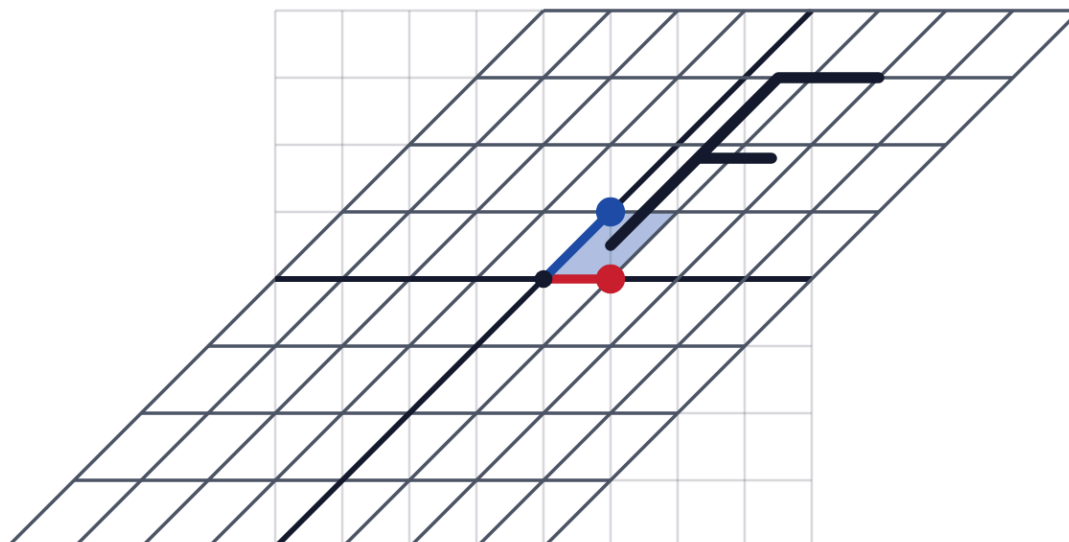
Now the fruit seller compares two shops, each with its own prices. They write the prices in a table: one row per shop and one column per fruit. That table is a **matrix**. To find out how much they'd pay in each shop, they multiply their shopping list by each row and add: one dot product per shop, all at once.

A matrix can also move drawings. Picture a grid painted on a rubber sheet: the matrix says where each point goes, as if you stretched or slanted the sheet, but with two rules: straight lines stay straight and the centre stays put.

Programs that rotate or stretch a photo do sums like these to decide where each pixel goes. And some matrices squash the whole drawing onto a line: after that, there's no way to get it back.

Try it: bend the grid

Move the four numbers of the matrix or try the buttons. The red arrow is the first column, where $(1, 0)$ goes; the blue one, the second. Find a matrix that squashes the grid onto a line: the area will become 0. And see what happens to the F with the mirror.



Try one

Do nothing

Stretch

Turn 90°

Slant

Mirror

Squash

a (top left)
1

b (top right)
1

c (bottom left)
0

d (bottom right)
1

(1, 0) goes to (1, 0)

(0, 1) goes to (1, 1)

The area is multiplied by (ad - bc) **1**

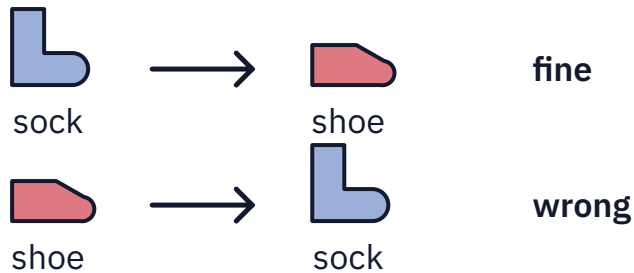
Dimensions left 2: the plane is still a plane

Matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$: the area is multiplied by 1 and 2 dimensions are left.



Interactive widget. Try it on the web: <https://kingofdumbs.com/en/lessons/vectors-and-matrices>

Multiplying matrices: order matters



**Chaining two matrices means multiplying them,
and order matters.**

To get dressed, first the sock and then the shoe. The other way round doesn't come out the same. Matrices are the same: making one change and then another is equivalent to a single matrix, which you get by **multiplying** the two, and order matters. Rotating a photo and then stretching it sideways isn't the same as stretching it and then rotating it.

And here's the answer to the question at the start. A neural network that recognises cats receives the photo as a huge vector and multiplies it by matrices, one after another, with a few extra steps in between. Each row of those matrices is like a price list: its dot product with the photo gives one number. For each photo, that's millions of multiply-and-add sums.

That's why graphics cards suit AI so well: they're built to do a huge number of multiplications at once.

The King of Dumbs says:

I don't see cats: I see 36 million numbers. The hard part is knowing which matrices to multiply them by.



Final challenge

1. A colour photo of 100×100 pixels. How many numbers does the computer store when it opens it?

- a. 100
- b. 10,000, one per pixel
- c. 30,000: three per pixel, red, green and blue

2. Two arrows make a right angle. What's their dot product?

- a. 1
- b. 0
- c. It depends on how long they are

3. You rotate a photo by 90° and then stretch it sideways. Is that the same as stretching it first and rotating it afterwards?

- a. Yes: multiplying gives the same in any order
- b. No: with matrices, order matters
- c. Only if the photo is square



In three sentences

1. To a computer, a photo is a list of numbers (three per pixel, from 0 to 255): a vector, whose dimension is how many numbers it has.
2. The dot product multiplies two lists term by term and adds: it gives a total, as in the fruit seller's sum, and says how much two arrows point the same way (0 if they're perpendicular).
3. A matrix is a table that moves every point of the plane, and multiplying matrices chains those changes, with the order mattering: it's the sum a neural network does millions of

times with each photo.



Sources

- Vectors, length and the dot product: OpenStax, *Calculus Volume 3*, [2.1](#) and [2.3](#) (the fruit seller, theorems 2.4 and 2.5). Matrices and their product: OpenStax, *College Algebra 2e*, [7.5](#).
- Transformations, determinant and dimension: D. Austin, *Understanding Linear Algebra*, [2.2](#), [2.5](#), [2.6](#), [3.4](#) and [3.5](#).
- Pixels and image transforms: Pillow 13.0, [Concepts](#) and [ImageTransform](#). From photo to tensor: torchvision 0.29, [ToTensor](#).
- Neural networks: PyTorch 2.14, [nn.Linear](#) and [cosine_similarity](#). NVIDIA, [Matrix Multiplication Background](#) and [Linear/Fully-Connected Layers](#).

How this lesson was made: researched and written with help from Claude from the sources above, consulted on 17 October 2026. Figures without their own link are calculations from their definitions. The examples (the 4,000 × 3,000 photo, the two shops, the layer with 1,000 outputs) are made up. The drawings are our own.

Answers

1. **c)** 30,000: three per pixel, red, green and blue. That's it. $100 \times 100 \times 3$: a vector of dimension 30,000.
2. **b)** 0. Right. Perpendicular, dot product 0, however long they are.
3. **b)** No: with matrices, order matters. Exactly. Like the sock and the shoe.

A King of Dumbs lesson, kingofdumbs.com. Offline version; the original, with widgets you can play with, is at: <https://kingofdumbs.com/en/lessons/vectors-and-matrices>

Updated on October 17, 2026.