An Intuitive Guide to Linear Algebra

Home › Math, Popular ›

Despite two linear algebra classes, my knowledge consisted of “Matrices, determinants, eigen something something”.

Why? Well, let’s try this course format:

- Name the course *Linear Algebra* but focus on things called matrices and vectors
- Teach concepts like Row/Column order with mnemonics instead of explaining the reasoning
- Favor abstract examples (2d vectors! 3d vectors!) and avoid real-world topics until the final week

The survivors are physicists, graphics programmers and other masochists. We missed the key insight:

Linear algebra gives you mini-spreadsheets for your math equations.

We can take a table of data (a matrix) and create updated tables from the original. It’s the power of a spreadsheet written as an equation.

Here’s the linear algebra introduction I wish I had, with a real-world stock market example.

What’s In A Name?

"Algebra" means, roughly, "relationships". Grade-school algebra explores the relationship between unknown numbers. Without knowing x and y, we can still work out that $(x+y)^2 = x^2 + 2xy + y^2$.

"Linear Algebra" means, roughly, "line-like relationships". Let's clarify a bit.

Straight lines are predictable. Imagine a rooftop: move forward 3 horizontal feet (relative to the ground) and you might rise 1 foot in elevation (The slope! Rise/run = 1/3). Move forward 6 feet, and you'd expect a rise of 2 feet. Contrast this with climbing a dome: each horizontal foot forward raises you a different amount.

Lines are nice and predictable:

- If 3 feet forward has a 1-foot rise, then going 10x as far should give a 10x rise (30 feet forward is a 10-foot rise)
- If 3 feet forward has a 1-foot rise, and 6 feet has a 2-foot rise, then (3 + 6) feet should have a (1 + 2) foot rise

In math terms, an operation F is linear if scaling inputs scales the output, and adding inputs adds the outputs:

$$F(ax) = a \cdot F(x)$$
$$F(x+y) = F(x) + F(y)$$

In our example, $F(x)$ calculates the rise when moving forward x feet, and the properties hold:

$$F(10 \cdot 3) = 10 \cdot F(3) = 10$$

$$F(3+6) = F(3) + F(6) = 3$$

Linear Operations

An operation is a calculation based on some inputs. Which operations are linear and predictable? Multiplication, it seems.

Exponents ($F(x) = x^2$) aren't predictable: 10^2 is 100, but 20^2 is 400. We doubled the input but quadrupled the output.

Surprisingly, regular addition isn't linear either. Consider the "add three" function $F(x) = x + 3$:

$$F(10) = 13$$
$$F(20) = 23$$

We doubled the input and did not double the output. (Yes, $F(x) = x + 3$ happens to be the equation for an *offset* line, but it's still not "linear" because $F(10) \neq 10 \cdot F(1)$. Fun.)

So, what types of functions are *actually* linear? Plain-old scaling by a constant, or functions that look like: $F(x) = ax$. In our roof example, $a = 1/3$.

But life isn't *too* boring. We can still combine multiple linear functions (A, B, C) into a larger one, G:

$$G(x, y, z) = A(x) + B(y) + C(z) = ax + by + cz$$

G is still linear, since doubling the input continues to double the output:

$$G(2x, 2y, 2z) = a(2x) + b(2y) + c(2z) = 2(ax + by + cz) = 2 \cdot G(x, y, z)$$

We have "mini arithmetic": multiply inputs by a constant, and add the results. It's actually useful because we can split inputs apart, analyze them individually, and combine the results:

$$G(x, y, z) = G(x, 0, 0) + G(0, y, 0) + G(0, 0, z)$$

If we allowed non-linear operations (like x^2) we couldn't split our work: $(a+b)^2 \neq a^2 + b^2$. Limiting ourselves to linear operations has its advantages.

Organizing Inputs And Operations

Most courses hit you in the face with the details of a matrix. "Ok kids, let's learn to speak. Select a subject, verb and object. Next, conjugate the verb. Then, add the prepositions…"

No! Grammar is not the focus. What's the key idea?

- We have a bunch of inputs to track
- We have predictable, linear operations to perform (our "mini-arithmetic")
- We generate a result, perhaps transforming it again

Ok. First, how should we track a bunch of inputs? How about a list:

```
x
y
z
```

Not bad. We could write it (x, y, z) too — hang onto that thought.

Next, how should we track our operations? Remember, we only have "mini arithmetic": multiplications by a constant, with a final addition. If our operation F behaves like this:

$$F(x, y, z) = 3x + 4y + 5z$$

We could abbreviate the entire function as (3, 4, 5). We know to multiply the first input by the first value, the second input by the second value, etc., and add the result.

Only need the first input?

$$G(x, y, z) = 3x + 0y + 0z = (3, 0, 0)$$

Let's spice it up: how should we handle multiple sets of inputs? Let's say we want to run operation F on both (a, b, c) and (x, y, z). We could try this:

$$F(a, b, c, x, y, z) = ?$$

But it won't work: F expects 3 inputs, not 6. We should separate the inputs into groups:

```
1st Input  2nd Input
--------------------
a          x
b          y
c          z
```

Much neater.

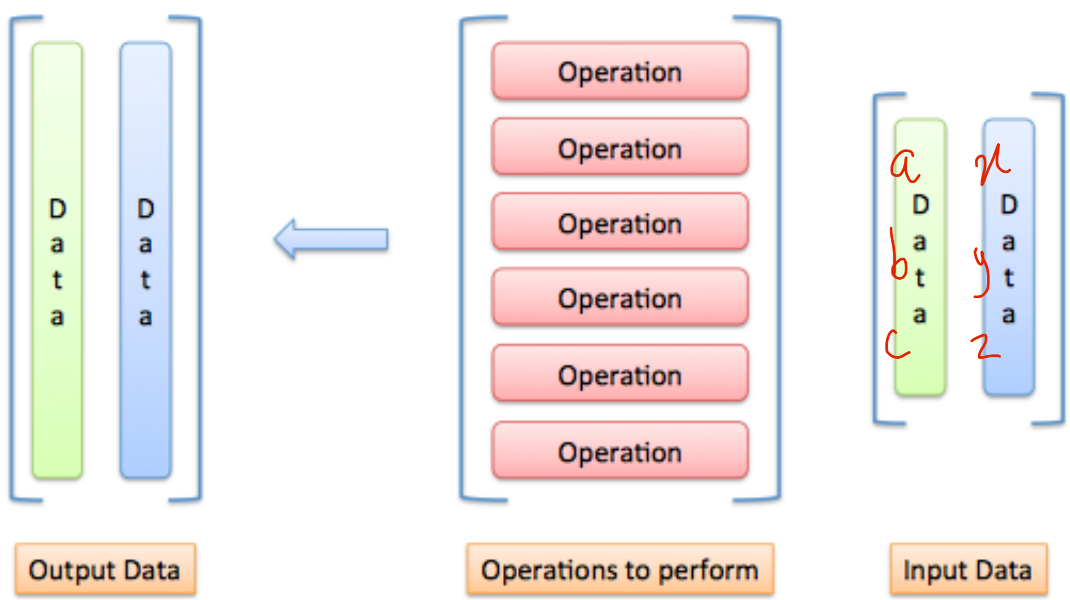
And how could we run the same input through several operations? Have a row for each operation:

```
F: 3 4 5
G: 3 0 0
```

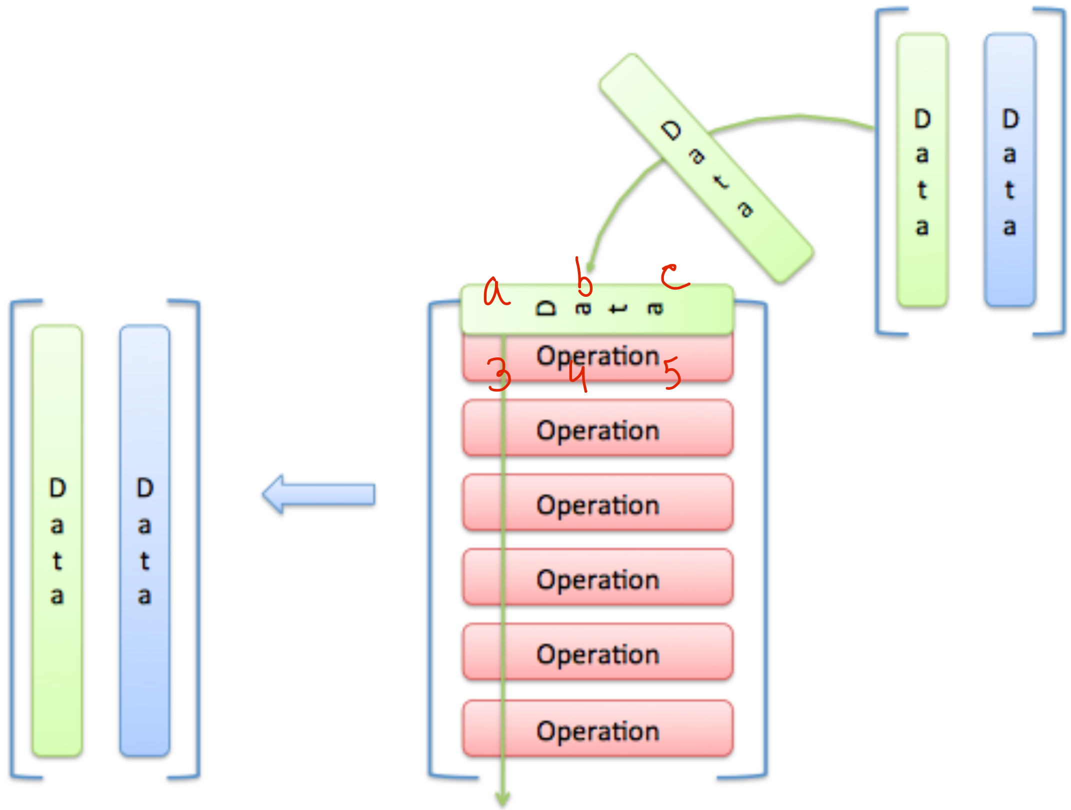
Neat. We're getting organized: inputs in vertical columns, operations in horizontal rows.

Visualizing The Matrix

Words aren't enough. Here's how I visualize inputs, operations, and outputs:



Imagine "pouring" each input through each operation:



As an input passes an operation, it creates an output item. In our example, the input (a, b, c) goes against operation F and outputs 3a + 4b + 5c. It goes against operation G and outputs 3a + 0 + 0.

Time for the red pill. A matrix is a shorthand for our diagrams:

$$\text{Inputs} = \mathrm{A} = \begin{bmatrix} \text{input1} & \text{input2} \end{bmatrix} = \begin{bmatrix} a & x \\ b & y \\ c & z \end{bmatrix}$$

$$\text{Operations} = \mathrm{M} = \begin{bmatrix} \text{operation1} \\ \text{operation2} \end{bmatrix} = \begin{bmatrix} 3 & 4 & 5 \\ 3 & 0 & 0 \end{bmatrix}$$

A matrix is a single variable representing a spreadsheet of inputs or operations.

Trickiness #1: The reading order

Instead of an input => matrix => output flow, we use function notation, like y = f(x) or f(x) = y. We usually write a matrix with a capital letter (F), and a single input column with lowercase (x). Because we have several inputs (A) and outputs (B), they're considered matrices too:

$$MA = B$$

$$\begin{bmatrix} 3 & 4 & 5 \\ 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} a & x \\ b & y \\ c & z \end{bmatrix} = \begin{bmatrix} 3a + 4b + 5c & 3x + 4y + 5z \\ 3a & 3x \end{bmatrix}$$

The output from operation one on data point one
the output from operation two on data point one
The output from operation one on data point two
the output from operation two on data point two

Trickiness #2: The numbering

Matrix size is measured as RxC: row count, then column count, and abbreviated "m x n" (I hear ya, "r x c" would be easier to remember). Items in the matrix are referenced the same way: a_{ij} is the ith row and jth column (I hear ya, "i" and "j" are easily confused on a chalkboard). Mnemonics are ok *with context*, and here's what I use:

- RC, like Roman Centurion or RC Cola
- Use an "L" shape. Count down the L, then across

Why does RC ordering make sense? Our operations matrix is 2×3 and our input matrix is 3×2. Writing them together:

```
[Operation Matrix] [Input Matrix]
[operation count x operation size] [input size x input count]
[m x n] [p x q] = [m x q]
[2 x 3] [3 x 2] = [2 x 2]
```

sizes should match up i.e. input sizes

Notice the matrices touch at the "size of operation" and "size of input" (n = p). They should match! If our inputs have 3 components, our operations should expect 3 items. In fact, we can *only* multiply matrices when n = p.

The output matrix has m operation rows for each input, and q inputs, giving a “m x q” matrix.

Fancier Operations

Let’s get comfortable with operations. Assuming 3 inputs, we can whip up a few 1-operation matrices:

- Adder: [1 1 1]
- Averager: [1/3 1/3 1/3]

The “Adder” is just a + b + c. The “Averager” is similar: (a + b + c)/3 = a/3 + b/3 + c/3.

Try these 1-liners:

- First-input only: [1 0 0]
- Second-input only: [0 1 0]
- Third-input only: [0 0 1]

And if we merge them into a single matrix:

```
[1 0 0]
[0 1 0]
[0 0 1]
```

Whoa — it’s the “identity matrix”, which copies 3 inputs to 3 outputs, unchanged. How about this guy?

```
[1 0 0]
[0 0 1]
[0 1 0]
```

He reorders the inputs: (x, y, z) becomes (x, z, y).

And this one?

```
[2 0 0]
[0 2 0]
[0 0 2]
```

He's an input doubler. We could rewrite him to `2*I` (the identity matrix) if we were so inclined.

And yes, when we decide to treat inputs as vector coordinates, the operations matrix will transform our vectors. Here's a few examples:

- Scale: make all inputs bigger/smaller
- Skew: make certain inputs bigger/smaller
- Flip: make inputs negative
- Rotate: make new coordinates based on old ones (East becomes North, North becomes West, etc.)

These are geometric interpretations of multiplication, and how to warp a vector space. Just remember that vectors are *examples* of data to modify.

A Non-Vector Example: Stock Market Portfolios

Let's practice linear algebra in the real world:

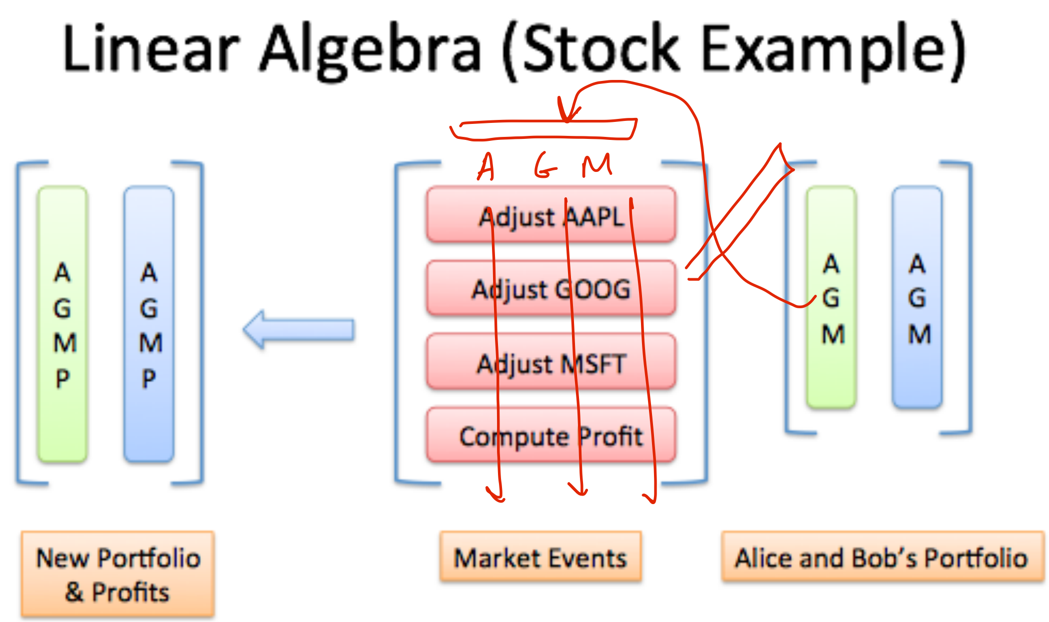
- Input data: stock portfolios with dollars in Apple, Google and Microsoft stock
- Operations: the changes in company values after a news event
- Output: updated portfolios

And a bonus output: let's make a new portfolio listing the net profit/loss from the event.

Normally, we'd track this in a spreadsheet. Let's learn to think with linear algebra:

- The input vector could be ($Apple, $Google, $Microsoft), showing the dollars in each stock. (Oh! These dollar values could come from *another* matrix that multiplied the number of shares by their price. Fancy that!)
- The 4 output operations should be: Update Apple value, Update Google value, Update Microsoft value, Compute Profit

Visualize the problem. Imagine running through each operation:



The key is understanding *why* we're setting up the matrix like this, not blindly crunching numbers.

Got it? Let's introduce the scenario.

Suppose a secret iDevice is launched: Apple jumps 20%, Google drops 5%, and Microsoft stays the same. We want to adjust each stock value, using something similar to the identity matrix:

```
New Apple     [1.2  0     0]
New Google    [0    0.95  0]
New Microsoft [0    0     1]
```

The new Apple value is the original, increased by 20% (Google = 5% decrease, Microsoft = no change).

Oh wait! We need the overall profit:

Total change = (.20 * Apple) + (-.05 * Google) + (0 * Microsoft)

Our final operations matrix:

```
New Apple        [1.2   0      0]
New Google       [0     0.95   0]
New Microsoft    [0     0      1]
Total Profit     [.20   -.05   0]
```

a Stack of operations

Making sense? Three inputs enter, four outputs leave. The first three operations are a "modified copy" and the last brings the changes together.

A G M A G M

Now let's feed in the portfolios for Alice $1000, $1000, $1000) and Bob $500, $2000, $500). We can crunch the numbers by hand, or use a Wolfram Alpha (calculation):

Input interpretation:

$$\begin{pmatrix} 1.2 & 0 & 0 \\ 0 & 0.95 & 0 \\ 0 & 0 & 1 \\ 0.2 & -0.05 & 0 \end{pmatrix} . \begin{pmatrix} 1000 & 1000 & 1000 \\ 500 & 2000 & 500 \end{pmatrix}^{\intercal}$$

transpose

4 x 3 2 x 3

you can't mutiply yet → you need to transpose the data

Result:

$$\begin{pmatrix} 1200. & 600. \\ 950. & 1900. \\ 1000. & 500. \\ 150. & 0. \end{pmatrix}$$

$$\begin{bmatrix} \text{operation 1} \\ \text{op 2} \\ \text{op 3} \\ \text{op 4} \end{bmatrix} \begin{bmatrix} A & A \\ G & G \\ M & M \end{bmatrix}$$

4 x 3 3 x 2

yay!

(Note: Inputs should be in columns, but it's easier to type rows. The Transpose operation, indicated by t (tau), converts rows to columns.)

The final numbers: Alice has $1200 in AAPL, $950 in GOOG, $1000 in MSFT, with a net profit of $150. Bob has $600 in AAPL, $1900 in GOOG, and $500 in MSFT, with a net profit of $0.

What's happening? **We're doing math with our own spreadsheet.** Linear algebra emerged in the 1800s yet spreadsheets were invented in the 1980s. I blame the gap on poor linear algebra education.

Historical Notes: Solving Simultaneous Equations

An *early* use of tables of numbers (not yet a "matrix") was bookkeeping for linear systems:

$$
\begin{aligned}
x + 2y + 3z &= 3 \\
2x + 3y + 1z &= -10 \\
5x + -y + 2z &= 14
\end{aligned}
$$

becomes

$$
\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 5 & -1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ -10 \\ 14 \end{bmatrix}
$$

We can avoid hand cramps by adding/subtracting rows in the matrix and output, vs. rewriting the full equations. As the matrix evolves into the identity matrix, the values of x, y and z are revealed on the output side.

This process, called Gauss-Jordan elimination, saves time. However, linear algebra is mainly about matrix transformations, not solving large sets of equations (it'd be like using Excel for your shopping list).

Terminology, Determinants, And Eigenstuff

Words have technical categories to describe their use (nouns, verbs, adjectives). Matrices can be similarly subdivided.

Descriptions like “upper-triangular”, “symmetric”, “diagonal” are the shape of the matrix, and influence their transformations.

The **determinant** is the “size” of the output transformation. If the input was a unit vector (representing area or volume of 1), the determinant is the size of the transformed area or volume. A determinant of 0 means matrix is “destructive” and cannot be reversed (similar to multiplying by zero: information was lost).

The **eigenvector** and **eigenvalue** represent the “axes” of the transformation.

Consider spinning a globe: every location faces a new direction, except the poles.

An “eigenvector” is an input that doesn’t change direction when it’s run through the matrix (it points “along the axis”). And although the direction doesn’t change, the size might. The eigenvalue is the amount the eigenvector is scaled up or down when going through the matrix.

(My intuition here is weak, and I’d like to explore more. Here’s a nice diagram and video.)

Matrices As Inputs

A funky thought: we can treat the operations matrix as inputs!

Think of a recipe as a list of commands (*Add 2 cups of sugar, 3 cups of flour…*).

What if we want the metric version? Take the instructions, treat them like text, and convert the units. The recipe is “input” to modify. When we’re done, we can follow the instructions again.

An operations matrix is similar: commands to modify. Applying one operations matrix to another gives a new operations matrix that applies *both* transformations, in order.

If N is “adjust for portfolio for news” and T is “adjust portfolio for taxes” then applying both:

TN = X

means “Create matrix X, which first adjusts for news, and then adjusts for taxes”. Whoa! We didn’t need an input portfolio, we applied one matrix directly to the other.

The beauty of linear algebra is representing an entire spreadsheet calculation with a single letter. Want to apply the same transformation a few times? Use N^2 or N^3.

Can We Use Regular Addition, Please?

Yes, because you asked nicely. Our “mini arithmetic” seems limiting: multiplications, but no addition? Time to expand our brains.

Imagine adding a dummy entry of 1 to our input: (x, y, z) becomes (x, y, z, 1).

Now our operations matrix has an extra, known value to play with! If we want `x + 1` we can write:

```
[1 0 0 1]
```

And `x + y - 3` would be:

```
[1 1 0 -3]
```

Huzzah!

Want the geeky explanation? We’re pretending our input exists in a 1-higher dimension, and put a “1” in that dimension. We *skew* that higher dimension, which looks like a *slide* in the current one. For example: take input (x, y, z, 1) and run it through:

```
[1 0 0 1]
[0 1 0 1]
[0 0 1 1]
[0 0 0 1]
```

The result is (x + 1, y + 1, z + 1, 1). Ignoring the 4th dimension, every input got a +1. We keep the dummy entry, and can do more slides later.

Mini-arithmetic isn't so limited after all.

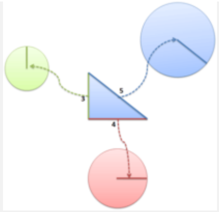
Onward

I've overlooked some linear algebra subtleties, and I'm not too concerned. Why?

These metaphors are helping me *think* with matrices, more than the classes I "aced". I can finally respond to "Why is linear algebra useful?" with "Why are spreadsheets useful?"

They're not, unless you want a tool used to attack nearly every real-world problem. Ask a businessman if they'd rather donate a kidney or be banned from Excel forever. That's the impact of linear algebra we've overlooked: efficient notation to bring spreadsheets into our math equations.

Happy math.



Enjoy the article? There's plenty more to help you build a lasting, intuitive understanding of math. Join the newsletter for bonus content and the latest updates.

Email Address

Join Newsletter

Facebook Twitter LinkedIn Email Print

Other Posts In This Series

78 Comments BetterExplained Discussions 1 Login

Recommend 4 Tweet Share Sort by Best

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Bart
2 years ago

You lost me at "Exponents...aren't predictable."

2 | Reply

ben → Bart
2 months ago

By predictable, he means there is a consistent linear growth with simple multiplication. For example if your function is x/2 (or x * 1/2), there is a .5 growth for each unit increase in x, f(x) - this will give you a straight line on the xy plane. However, if your function is x^2, then each unit increase in x will not give you consistent linear growth, but a curve that looks like the right half of a 'U' (exponential growth) which is not a straight line.

Reply

ben
2 months ago edited

'G is still linear, since doubling the input continues to double the output:

G(2x, 2y, 2z) = a(2x) + b(2y) + c(2z) = 2(ax + by + cz) = 2 * G(x, y, z)'

Should you not show also that adding an input is equal to adding them individually before you can say that it is linear? I thought it has to satisfy both:

'In math terms, an operation F is linear if scaling inputs scales the output, and adding inputs adds the outputs:'

Or do you only have to satisfy one?

Reply

Mukesh KG
8 months ago

Matrix ? Got that!
Where does VECTOR fit in ?

Reply

ミハイ
9 months ago

I can solve all of this when coding but the style in which it gets represented on the blackboard is really bad at underlining the operations that take place, maybe it is the syntax or I'm just not used to this. Your way of explaining was a bit closer, but also fell short for me.

Reply

Jeff Connors
a year ago

Such a great explanation,, can't tell you how thankful I am :-) THANK YOU so much for the beautiful explanation,,

Reply

Jae Duk Seo
a year ago

amazing and very veyr good

Reply

Aleix
3 years ago

Hello

ɔpration was the same for all inputs, like in

ɔuld be like {F(x, y, z) = A(x) +B(y)+ C(z)}

F(x), but how this one respect the rule "adding inputs adds the outputs"?

Thanks a lot!

Reply

Dr CL Verma
3 years ago

I want to know how linear algebra is used in Applies Statistics. I did only normal classical algebra.

Reply

Dorothy
3 years ago

wow, I like it

Reply

Ben
3 years ago

Is there a book on this intuitive linear algebra and where can i buy it?

Reply

Tushar_Kadam → Ben
3 months ago

You can watch the videos related to Linear Algebra on Youtube channel named 3Blue1Brown. There you will get the best intuition.

Reply

Derek
3 years ago

A few more ideas, mostly for those who code:

Not only are square matrices useful for storing directed graphs, the relationship between matrix vector multiplication and Breadth First Search on a graph is a very powerful one. Put simply, for any given state on your graph, you can iterate each 'marker' one step forward via a matrix vector multiplication. In this sense, your matrix quite literally is a map from one allowed point to the next.

With respect to eigenvalues and eigenvectors: High levlel, I wouldn't worry about nits like Jordan blocks. The gist is: if you want to, you can diagonalize a matrix. You can always do so when you have unique eigenvalues. If eigenvalues are not unique, you still can sometimes -- but even then if you have linearly dependent eigenvectors associated with a matrix A, you can still diagonalize an approximation of A if you perturb the diagonal of A slightly. In such a case you have an approximation of A (of course some care is needed here in how/why you approximate A) in a very interpretable form and that can be very useful.

One of the things Linear Algebra really hits home on is how a change in representation (via factorization) can make certain problem a lot easier to understand. Indeed if you have diagonalized A such that $A = PDP^{-1}$ you now can easily see what is going on under the hood of A when you multiply AA, AAA and AAAA, and so on. Something that may have been a bit confusing or opaque, now has been transformed to simply using exponents on scalars (the eigenvalues along the diagonal matrix D) . And if A is not invertible, it becomes even more obvious why -- D is a diagonal matrix with one or more zeros on its diagonal... so to invert A we'd need to (among other things) get the reciprocoals of the diagonal of D... which turns this into a classic problem of trying to divide a number by zero which prevents you from completing the inversion of A.

see more

Reply

JJJ
3 years ago

I LOVE you! You're amazing! I didn't know that reading about linear algebra could be so entertaining! <3333 Many thanks ^^

Reply

AndalusOpenSource
3 years ago

Kalman Filter

http://www.bzarg.com/p/how-...

why don't you make a tutorial about this topic
i'm looking forward to see you talking about Kalman Filter

i gave you a good tutorial about Kalman filter

but i'm sure you will do some thing with more insight and betterunderstanding
as you always do

best of luck
and many thanks for your great effort

Reply

G ➦ AndalusOpenSource
9 months ago

I agree, I would also really like a tutorial on the Kalman filter. I think I need to understand it for a dynamical systems model that I'm currently struggling with and this might be helpful!

Reply

Arwa
4 years ago

خالد
!Some how when I am progressing in the 'learning scheme', I find your words describe exactly my thoughts

Along the way you ans question I have had on my mind & couldn't visualise it, like det of matrix = 0

!thanks a lot and keep doing what U do

Reply

Ramesh
4 years ago

Beautifully explained . Having always been baffled by matrices and determinants , I have to say that this is the best lesson that I have had

Reply

Prashant
4 years ago

I am getting a headache on tensors. Please post a lesson about it.

Reply

TI
4 years ago

@Fation
Economists use linear algebra and specifically matrix inversion all of the time. One classic example would be input - output analysis. The form would be something like Q = AQ + B. Q would be the overall quantity demanded of a good that is both an input in making other goods (The AQ portion) and is also sold to an end user on its own (the B portion). A is a matrix of "technical coefficients" that describes how much Q goes in to

make the other goods. In linear algebra because of the nature of matrices you can't simply divide one by the other. Inversion takes the place of division (it is even denoted A^-1 which is another way of saying it's a divisor). Multiplying by an inverse is = to division. So in this example Q-AQ=B --> (I-A)Q = B --> Q = (I-A)^-1 * B. "I" being the identity matrix. By multiplying the inverse of (I-A) by B we get Q. So through inversion we discovered an independent expression for Q which is very useful for figuring out some stuff. This is just a simple example but it is used in practice for modeling in the real world. This link provides more detail if you're interested http://www.math.unt.edu/~tu...

^ | ˅ Reply

Load more comments