Lecture Slides for 18.06 Linear Algebra Spring 2020

Alan Edelman

Syllabus link (serves as a TOC for these slides)

(MIT 18.06 students, comments, fixes, request for further clarifications, welcome. Please keep mathematical)

Lecture 1. A Modern (Personal View) of Linear Algebra

Lecture 1. A Modern (Personal View) of Linear Algebra

The many many applications of linear algebra to science, engineering, machine learning, statistics, ….

The abstractions that let you understand the very fabric of mathematics.

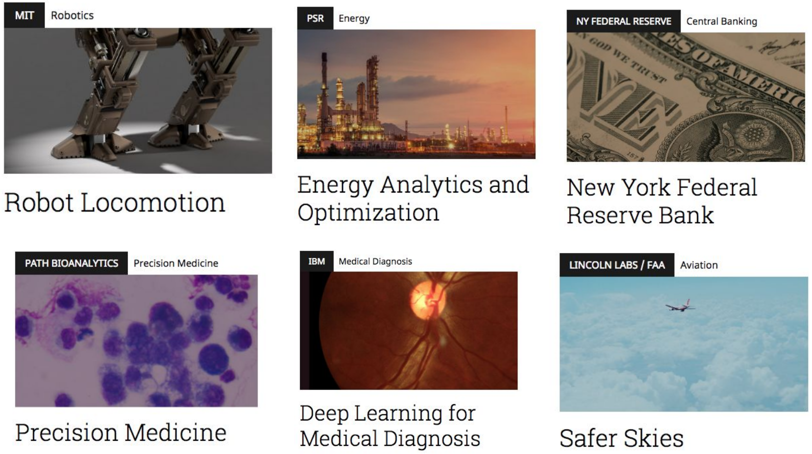
Lecture 1. A Modern (Personal View) of Linear Algebra

Increasingly used in practical Classes worldwide, machine Learning, big data: (See Julia.)

The many many applications of linear algebra to science, engineering, machine learning, statistics, ….

The abstractions that let you understand the very fabric of mathematics.

Lecture 1. A Modern (Personal View) of Linear Algebra



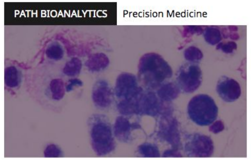
The abstractions that let you understand the very fabric of mathematics.

Lecture 1. A Modern (Personal View) of Linear Algebra

Robot Locomotion

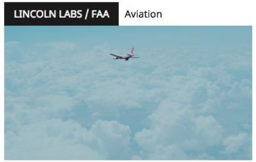
Energy Analytics and Optimization

New York Federal Reserve Bank



Precision Medicine

Deep Learning for Medical Diagnosis



Safer Skies

Vector Spaces, Linear Transformations. The relationship of the continuous to the discrete. Seeing things in high dimension spaces, like line fitting.

Lecture 1. A maybe silly analogy with, say, the number 7

Practicality:
The number 7 is useful. You can do things with it, in your head, by hand, or with a machine:

7 + 7 = 14
7 x 7 = 49
7 to the power 7 = I need a computer...

Abstraction:
The idea that “seven”, 7,7.0, vii, siete, sept, 七, سبعة, שבע, เจ็ด, Семь, सात, সাত, Yedi

Are all somehow the same....
Any 4 year old would tell you that the above are all different. Any 6 year old might say they are not!

What almost never happens on computers:

Matrix Inverses

Free variables and pivot variables

Echelon Forms

Determinants (though these are theoretically of immense value)

Representations of large matrices as tables of numbers

What does happen

Structured matrices (and of course sparse)

The SVD (and eigenvalues, but especially the SVD)

Lecture 1:

Linear Algebra lets you not miss the forest for the trees.

Sometimes a matrix IS just a table of numbers.

Sometimes a matrix should not be thought of as a table of numbers.

Sometimes one should step back and see a matrix as being its own entity.

Sometimes index notation A_{ij} is distracting.

Sometimes index notation A_{ij} is comforting.

… this may not make sense yet, but will over the course of the semester.

How 18.06 may differ this semester

- More emphasis on the conceptual and the practical.
- Hand computations: some, but not more than is needed.
- No pivot variable, free variable, echelon forms.
- Less emphasis on hand computation of inverses.
- Raise the importance of the Singular Value Decomposition (we will not think of it as much as an offshoot of the Eigendecomposition, but being its own creature)
- Matrix calculus: gradients with respect to vectors and matrices (Not in 18.02, not really anywhere in standard math courses)
- Applications to machine learning, statistics, and other areas
- Raise importance of linear transformations
- Less overlap with 18.03

1a. vectors

Examples of vectors:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad \text{or} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \text{such as} \quad x = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad \text{or} \quad y = \begin{bmatrix} 2.3 \\ 5.6 \\ 7.8 \end{bmatrix}$$

x is a 5 vector

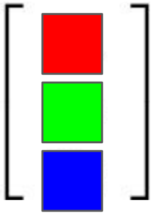
$x \in \mathbb{Z}^5$ ($\mathbb{Z}$ = the integers = $\{0,\pm 1,\pm 2,....\}$)

$y \in \mathbb{R}^3$ ($\mathbb{R}$ = the reals)

What can be elements of a vector?

Computers use vectors to contain all kinds of objects. Mathematicians usually insist that vector elements come from something (beyond the scope of this class) called a field. The real and the complex numbers are the most typical elements.

Going with the more permissive meaning of vector we can imagine:

$y =$  $\in (\text{colors})^3$ or nested vectors such as $\begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\ \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \\ \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \\ \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix} \end{bmatrix} \in (\mathbb{Z}^3)^4$

1b. Matrices

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \end{bmatrix}$$

Example

$$A= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

A is 3 x 5
$A \in \mathbb{Z}^{3,5}$
Julia: A has type Array(Int64,2)

Int64 means a 64 bit element type and 2 means a 2 dimensional array: a matrix.

1c. Matrix times vector

This you can do in your head:

$$\begin{bmatrix} 12 \\ 32 \\ 52 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

1c. Matrix times vector:

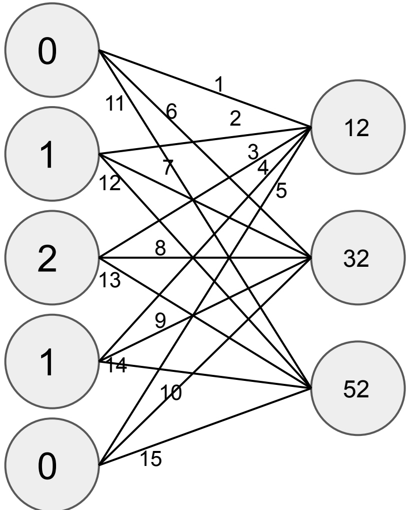
This maybe better done on a machine:

$$\begin{bmatrix} 276 \\ 640 \\ 476 \\ 612 \\ 800 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 4 \\ 1 & 5 & 9 \\ 2 & 6 & 5 \\ 3 & 5 & 8 \\ 9 & 7 & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

Neural Network Notation for matrix-vector

Popular right now but perhaps cumbersome

$$\begin{bmatrix} 12 \\ 32 \\ 52 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

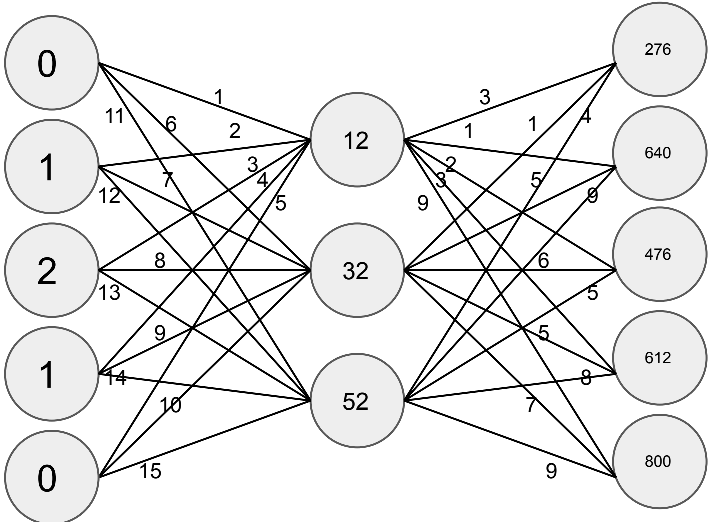


This is called a “linear neural network”
Matrix elements are called weights.

While a bit cumbersome to draw, it does illustrate nicely how every element of the output depends on every element of the input. Sometimes this picture is called a “complete bipartite graph”

Matrix times Matrix times vector denoted with a neural net

$$\begin{bmatrix} 276 \\ 640 \\ 476 \\ 612 \\ 800 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 4 \\ 1 & 5 & 9 \\ 2 & 6 & 5 \\ 3 & 5 & 8 \\ 9 & 7 & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$



Where do the rows of A connect?
Where do the columns of A connect?
Same questions for B?

Linear Transformations in 2d

1. See the corgi demo.
2. 3blue1brown's video on the topic.
3. A few words on linear transformations of photos:

Think of a photo as a rectangle in the plane with a color at every point $x = (x_1,x_2)$ in the rectangle.

Fix a 2x2 matrix A, and compute the point $y = Ax$ which has two coordinates $y = (y_1,y_2)$. In the new image, "paint" the point (y_1,y_2) with the color that was at (x_1,x_2), and do this for all points. This is what the corgi demo is doing. You may notice that generally squares become parallelograms, circles become ellipses, and parallel lines remain parallel, but orthogonal lines may not remain orthogonal.