Numerical Computing & Images Tutorial

EECS 442

Fall 2023, University of Michigan

Colab Notebook Format

- Like jupyter notebooks on the web

- Execute cells of code

- Useful for immediate visualization (a lot of what we do!)

- Allows us to use GPUs remotely (important for ML)

Numpy Tutorial

Colab:
https://colab.research.google.com/drive/1f7nAcXVy7jgvt21LhP_dHSmM-ARqUXCD?usp=sharing

Image Loading & Manipulation

Colab:
https://colab.research.google.com/drive/1uNOh4DIBRb3gA3rH32g3sCyGrcn2crip

Problem Set 1

- You will work in Colab

- Examples from both of these notebooks will be helpful!

Linear Algebra Review

Vectors

- A vector $x \in R^n$ is a stack of *n* real values.

- $$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = [x_1\ x_2\ \ldots\ x_n]^T$$

- In the computer vision context, x can represent an image vector where each x_i is a pixel value in that image

Measuring Length - Norms

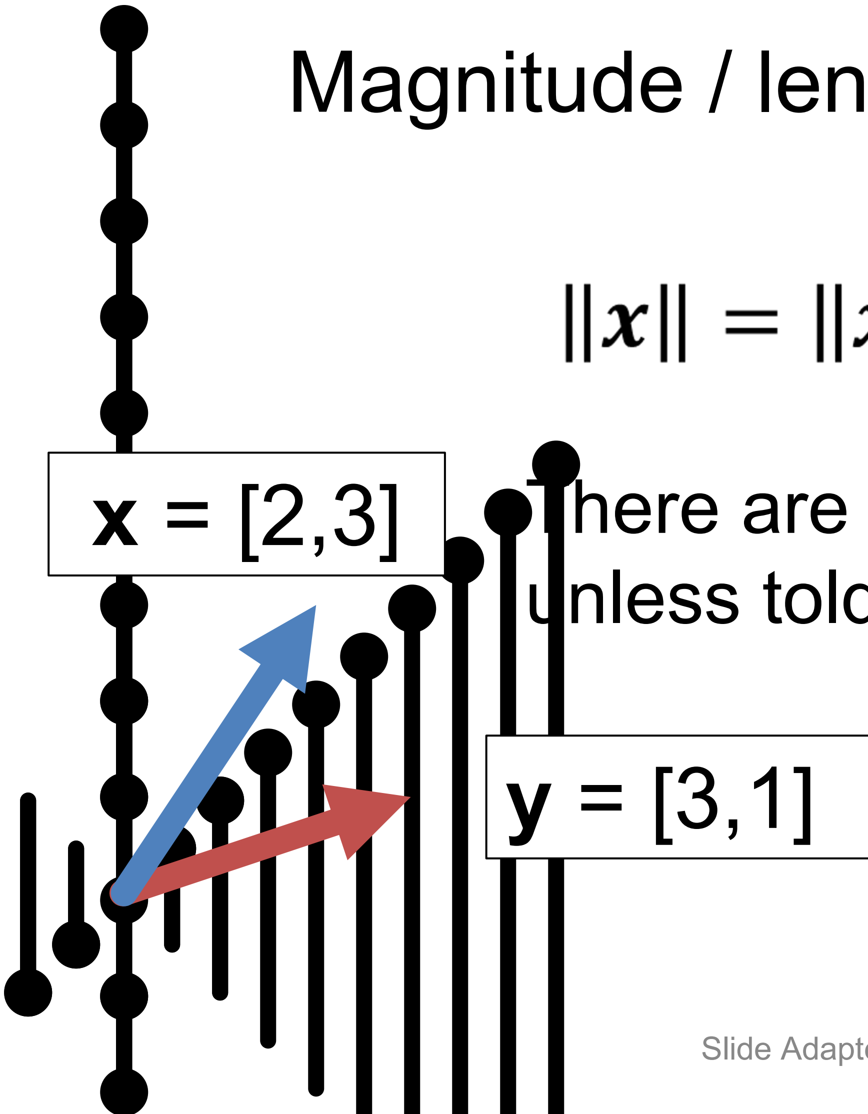
- Norms are a measure of the magnitude/ length of the vector
- Formally, a norm $\|\cdot\|: R^n \rightarrow [0, \infty)$ is a non-negative valued function.

- Examples:
  - Euclidean (l_2) norm (Default choice): $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}$
  - Manhattan distance (l_1 norm): $\|\mathbf{x}\|_1 = |x_1| + |x_2| + \ldots + |x_n|$
  - In general l_p norm ($p \geq 1$): $\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{1/p}$
  - l_∞ norm: $\|\mathbf{x}\|_\infty := \lim_{p \to \infty} \|\mathbf{x}\|_p = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$

Measuring Length

Magnitude / length / (L2) norm of vector

$$\|\boldsymbol{x}\| = \|\boldsymbol{x}\|_2 = \left(\sum_i^n x_i^2\right)^{1/2}$$

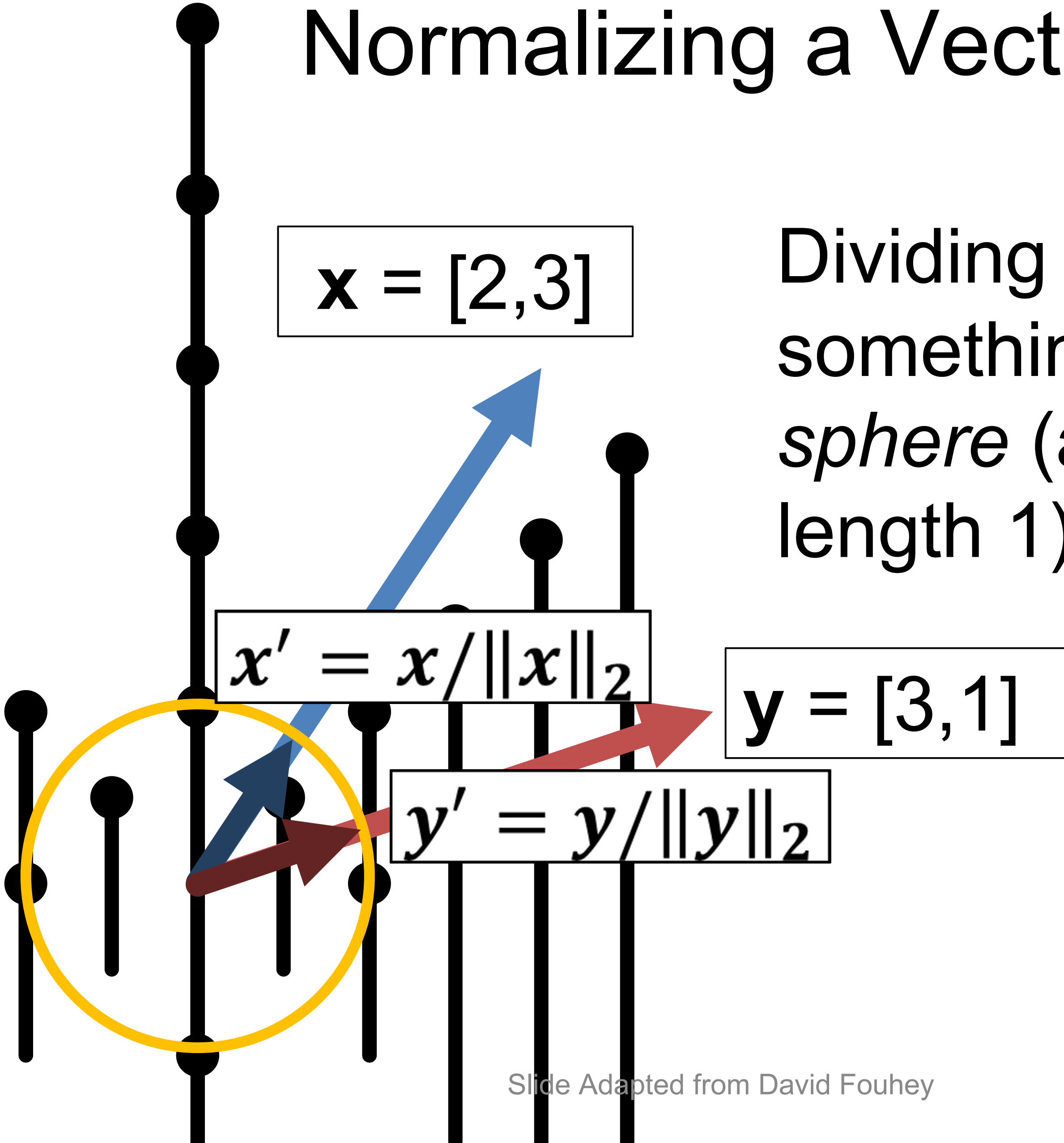


There are other norms; assume L2 unless told otherwise

$$\|\boldsymbol{x}\|_2 = \sqrt{13}$$
$$\|\boldsymbol{y}\|_2 = \sqrt{10}$$

Why?

Normalizing a Vector

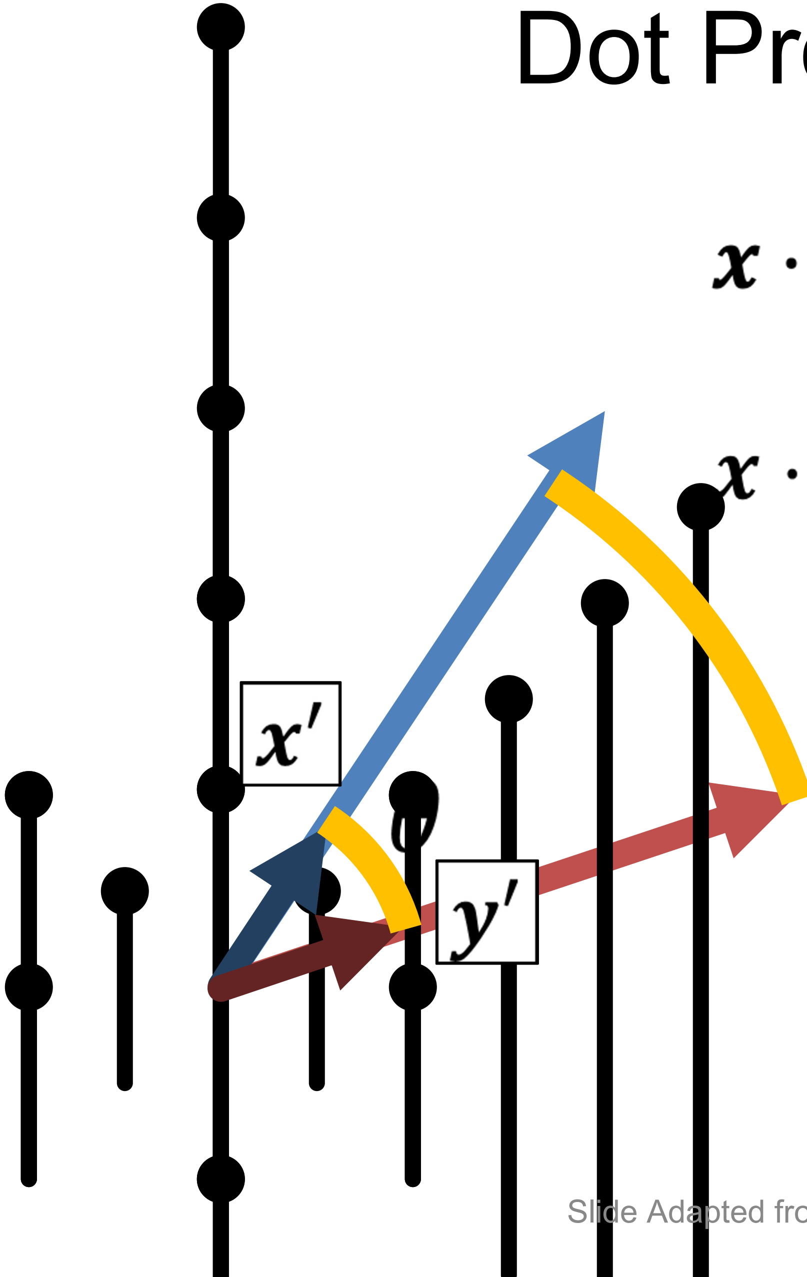


Dividing by norm gives something on the *unit sphere* (all vectors with length 1)

Dot Products

$$\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i = \boldsymbol{y}^T \boldsymbol{x}$$

$$\boldsymbol{x} \cdot \boldsymbol{y} = \cos(\theta) \|\boldsymbol{x}\| \, \|\boldsymbol{y}\|$$

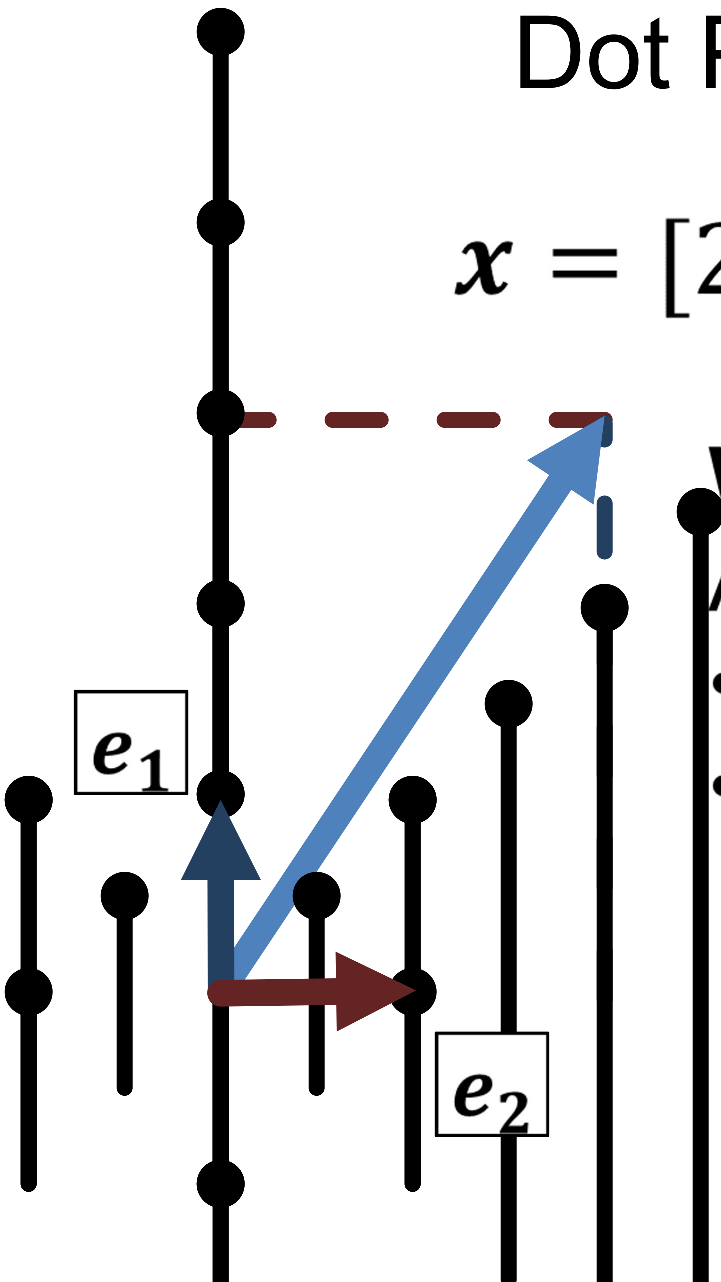


What happens with normalized / unit vectors?

$\boldsymbol{e_1} = [1,0]$
$\boldsymbol{e_2} = [0,1]$

Dot Products

$$\boldsymbol{x} = [2,3] \qquad \boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i}^{n} x_i y_i$$

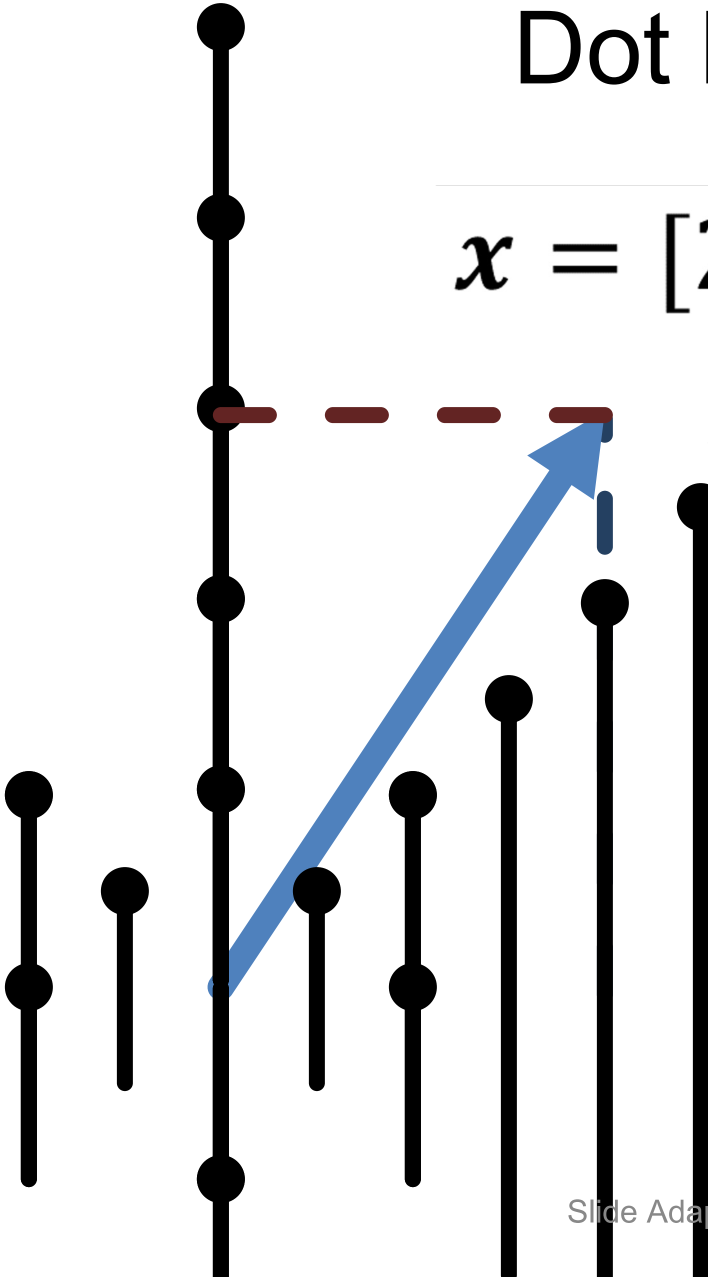


What's $\boldsymbol{x} \cdot \boldsymbol{e_1}$, $\boldsymbol{x} \cdot \boldsymbol{e_2}$?

Ans: $\boldsymbol{x} \cdot \boldsymbol{e_1} = 2$; $\boldsymbol{x} \cdot \boldsymbol{e_2} = 3$

- Dot product is projection
- Amount of **x** that's also pointing in direction of **y**

Dot Products



$$\boldsymbol{x} = [2,3] \qquad \boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i}^{n} x_i y_i$$

What's $\boldsymbol{x} \cdot \boldsymbol{x}$?

Ans: $\boldsymbol{x} \cdot \boldsymbol{x} = \sum x_i x_i = \|\boldsymbol{x}\|_2^2$

Matrices

Horizontally concatenate n, m-dim column vectors and you get a m x n matrix A (here 2x3)

$$\boldsymbol{A} = \begin{bmatrix} | & & | \\ \boldsymbol{v_1} & \cdots & \boldsymbol{v_n} \\ | & & | \end{bmatrix} = \begin{bmatrix} \boxed{\begin{matrix} v_{1_1} \\ v_{1_2} \end{matrix}} & \boxed{\begin{matrix} v_{2_1} \\ v_{2_2} \end{matrix}} & \boxed{\begin{matrix} v_{3_1} \\ v_{3_2} \end{matrix}} \end{bmatrix}$$

Matrices

Transpose: flip rows / columns $\begin{bmatrix} a \\ b \\ c \end{bmatrix}^T = \begin{bmatrix} a & b & c \end{bmatrix}$ $(3x1)^T = 1x3$

Vertically concatenate m, n-dim row vectors and you get a m x n matrix A (here 2x3)

$$A = \begin{bmatrix} - & \boldsymbol{u_1^T} & - \\ & \vdots & \\ - & \boldsymbol{u_m^T} & - \end{bmatrix} = \begin{bmatrix} \boxed{u_{1_1} \quad u_{1_2} \quad u_{1_3}} \\ \boxed{u_{2_1} \quad u_{2_2} \quad u_{2_3}} \end{bmatrix}$$

Matrix-Vector Product

$$\boldsymbol{y}_{2x1} = \boldsymbol{A}_{2x3}\boldsymbol{x}_{3x1}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} | & | & | \\ \boldsymbol{v_1} & \boldsymbol{v_2} & \boldsymbol{v_3} \\ | & | & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\boldsymbol{y} = x_1\boldsymbol{v_1} + x_2\boldsymbol{v_2} + x_3\boldsymbol{v_3}$$

*Linear combination of columns of **A***

Matrix-Vector Product

$$\boldsymbol{y}_{2x1} = \boldsymbol{A}_{2x3}\boldsymbol{x}_{3x1}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$y_1 = \boldsymbol{u}_1^T\boldsymbol{x} \qquad y_2 = \boldsymbol{u}_2^T\boldsymbol{x}$$

Dot product between rows of ***A*** *and* ***x***

Matrix Multiplication

Generally: $\boldsymbol{A}_{mn}$ *and* $\boldsymbol{B}_{np}$ *yield product* $(\boldsymbol{AB})_{mp}$

$$\boldsymbol{AB} = \begin{bmatrix} - & \boldsymbol{a}_1^T & - \\ & \vdots & \\ - & \boldsymbol{a}_m^T & - \end{bmatrix} \begin{bmatrix} | & & | \\ \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_p \\ | & & | \end{bmatrix}$$

Yes – in **A**, I'm referring to the rows, and in **B**, I'm referring to the columns

Matrix Multiplication

Generally: $\boldsymbol{A}_{mn}$ *and* $\boldsymbol{B}_{np}$ *yield product* $(\boldsymbol{AB})_{mp}$

$$\boldsymbol{AB} = \begin{bmatrix} - & \boldsymbol{a}_1^T & - \\ & \vdots & \\ - & \boldsymbol{a}_m^T & - \end{bmatrix} \begin{bmatrix} | & & | \\ \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_p \\ | & & | \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{b}_1 & \cdots & \boldsymbol{a}_1^T\boldsymbol{b}_p \\ \vdots & \ddots & \vdots \\ \boldsymbol{a}_m^T\boldsymbol{b}_1 & \cdots & \boldsymbol{a}_m^T\boldsymbol{b}_p \end{bmatrix}$$

$$\boldsymbol{AB}_{ij} = \boldsymbol{a}_i^T\boldsymbol{b}_j$$

Matrix Multiplication

- Inner Dimensions must match
- Product gets the outer dimension
- (Yes, it's associative): $ABx = (A)(Bx) = (AB)x$
- (*No it's not commutative*): $ABx \neq (BA)x \neq (BxA)$

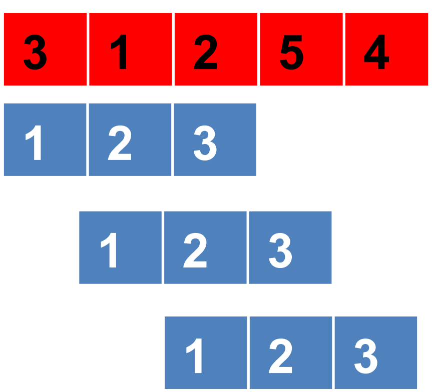
Cross-correlation

Consider 1D case for simplicity

- Correlation $c[m] = h * g = \sum_k h[m+k]g[k]$
- Convolution $f[m] = h \circ g = \sum_k h[m-k]g[k]$

Let h= [3,1,2,5,4], g = [1,2,3], then c = [11, 20, 24]:

$$c[0] = \sum_k h[0+k]g[k] = h[0]g[0] + h[1]g[1] + h[2]g[2] = \begin{bmatrix}3\\1\\2\end{bmatrix} \cdot \begin{bmatrix}1\\2\\3\end{bmatrix}$$



Each output element is from a dot product!

$$c[0] = \begin{bmatrix}3\\1\\2\end{bmatrix} \cdot \begin{bmatrix}1\\2\\3\end{bmatrix} = 3 + 2 + 6 = 11$$

$$c[1] = \begin{bmatrix}1\\2\\5\end{bmatrix} \cdot \begin{bmatrix}1\\2\\3\end{bmatrix} = 1 + 4 + 15 = 20$$

$$c[2] = \begin{bmatrix}2\\5\\4\end{bmatrix} \cdot \begin{bmatrix}1\\2\\3\end{bmatrix} = 2 + 10 + 12 = 24$$

Slide Adapted from David Fouhey

Convolution

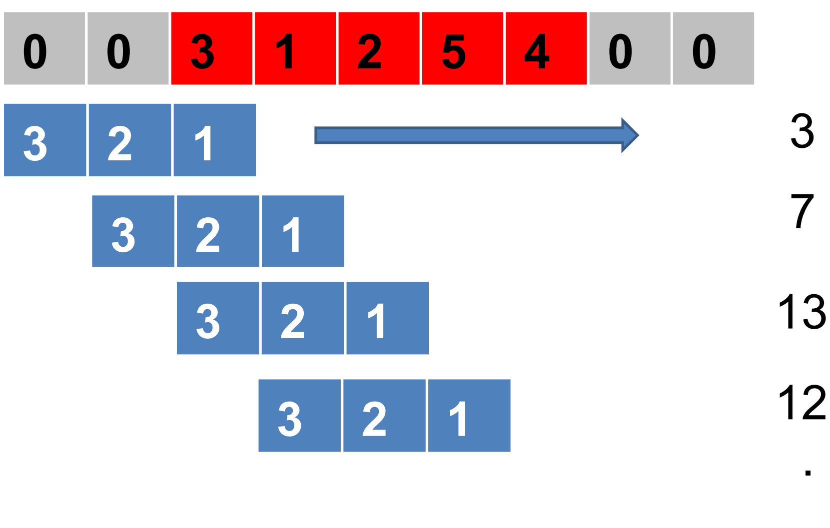
Consider 1D case for simplicity

- Correlation $c[m] = h * g = \sum_k h[m+k]g[k]$
- Convolution $f[m] = h \circ g = \sum_k h[m-k]g[k]$

Let h= [3,1,2,5,4], g = [1,2,3], then f = [3, 7, 13, 12, 20, 23, 12]:

$$f[0] = \sum_k h[0-k]g[k] = h[0]g[0] + h[-1]g[1] + h[-2]g[2] = \begin{bmatrix}0\\0\\3\end{bmatrix} \cdot \begin{bmatrix}3\\2\\1\end{bmatrix}$$

Each output element is from a dot product!



Operations They Don't Teach

You Probably Saw Matrix Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

What is this? FYI: e is a scalar

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + e = \qquad \begin{bmatrix} a+e & b+e \\ c+e & d+e \end{bmatrix}$$

Broadcasting

If you want to be pedantic and proper, you expand e by multiplying a matrix of 1s (denoted **1**)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + e = \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \mathbf{1}_{2x2}e$$

$$= \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & e \\ e & e \end{bmatrix}$$

Many smart matrix libraries do this automatically. This is the source of many bugs.

Broadcasting Example

Given: a nx2 matrix **P** and a 2D column vector **v**,
Want: nx2 difference matrix

$$\boldsymbol{P} = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\boldsymbol{P} - \boldsymbol{v}^T = \begin{bmatrix} x_1 & y_1 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} - \begin{matrix} [a & b] \\ & \vdots & \\ [a & b] \end{matrix} = \begin{bmatrix} x_1 - a & y_1 - b \\ \vdots & \vdots \\ x_n - a & y_n - b \end{bmatrix}$$