Data Dimension

- Scalars = Numeric constants

- Vectors = $\underbrace{[1 \;\; 2 \;\; 3]}_{\text{Row vector}}$ $\qquad \left.\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\right\}$ column vector

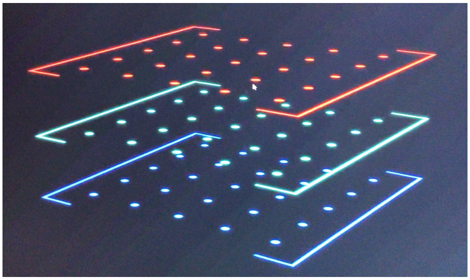
  ↓

  (1 Dimensional with here length = 3)

- Matrix = $\begin{array}{c|ccc} & C_1 & C_2 & C_3 \\ R_1 \rightarrow & 1 & 2 & 3 \\ R_2 \rightarrow & 4 & 5 & 6 \end{array}$ 2D grid (2×3) size.

- Tensors = 'n' dimensional collection of values

Scalar	Vector	Matrix	Tensor
(0 dimensional)	(1 dimensional tensor)	(2 dimensional tensor)	(2+)



3- Dimensional = Stack of '3' matrices

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

A

So, $a_{23} = 6$.

$$\begin{array}{c|ccc} & c_1 & c_2 & c_3 \\ \hline r_1 & & & \\ \vdots & \multicolumn{3}{c}{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}} \\ r_2 & & & \end{array}$$

actual notation

$\Updownarrow$ Programming

$$\begin{array}{c|ccc} & c_0 & c_1 & c_2 \\ \hline r_0 & & & \\ & \multicolumn{3}{c}{\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix}} \\ r_1 & & & \end{array}$$

Programming notation

Numpy : ndarray = 'n' dimensional array

1. Scalar (numpy) = [uint8, int8, uint16, int16]

(instead of using normal int of python)

- S = np.array(5)
- S.shape   O/P = ( )

means '0' dimension

- x = 8 + 3

numpy scalar   Python Scalar

- x   O/P = 8
- type(x)   O/P = numpy.int64
- x.shape()   O/P = ( )

↳ Zero dimension

- m = 8   ] Python Scalar
- m.shape   O/P = error : shape not defined for python 'int'

2. Vectors : pass a python list

- v = np.array([1, 2, 3]) ↵
- v.shape ↵   O/P = (3,)

Means its → a one dimensional item i.e. along one of dimension we have a length = 3.

> * Shape: O/P is a tuple with size of each of ndarray's dimension.
> - scalar = empty tuple
> - vector = a number and a comma

Tuple → '0' element
ⓐ empty = () ↵
len(empty) ↵ O/P = 0
ⓑ s = 'hello', ↵
len(s) ↵ O/P = 1
s ↵ O/P = ('hello',)

- x = v[1]
  x ↵   O/P = 2.
- v[1 :]   O/P = [2, 3]   } numpy slicing

Numpy indexing

* All indexes | array scalar obtained by slicing are views

1. $\boxed{\text{Basic indexing}} = (i : j : k)$

Start index (i), Stop index (j), steps (k)

eg: x = np.array([0,1,2,3,4,5,6,7,8,9])

x[1:7:2]   o/p array([1, 3, 5])

$\boxed{\text{note: stop index is not included}}$

2. $\boxed{\text{Negative 'i' and 'j'}}$:
$-i = n+i$
$-j = n+j$

'n' = no. of elements in corresponding dimension

-k = steps go towards smaller index

← or, ↑

eg. x[-2 : 10] = [10-2 : 10] = [8 : 10]

ie. element 8, 9

= array([8, 9])

- $x[-3:3:-1]$

$\Rightarrow x[10-3 : 3 : -1]$

$\Rightarrow x[7:3:-1] \Rightarrow x[7,6,5,4]$ elements

$\Rightarrow$ O/P = array([7,6,5,4,])

3. x = np.array([[[1],[2],[3]], [[4],[5],[6]]])

x.shape ↵ O/P= (2,3,1)

x[1:2]

O/p: array([[[4], [5], [6]]])

{Matrices}

- m = np.array([[1,2,3], [4,5,6]])
- m.shape O/P: (2,3)
- m[1][2] O/P: 6

Tensors

t = np.array([[[[1],[2]], [[3],[4]],
[[5],[6]]], [[[7],[8]],
[[9],[10]], [[11],[12]]],
[[[13],[14]],
[[15],[16]],
[[17],[17]]]])

t.shape O/P: (3, 3, 2, 1)

Changing Shapes

- v = np.array([1, 2, 3, 4])
- v.shape : O/P = (4,)

Convert to 1x4 ?

- x = v.reshape(1,4)
- x.shape   O/P: (1,4)
- x = v.reshape(4,1)

  x.shape   O/P: (4,1)

another approach to reshape

- x = v[None, :] ⇒ (1, 4) ← O/P

  all

  x.shape

- x = v[:, None]

  x.shape   O/P: (4,1)

Element wise matrix Operation

Scalar add

- $2 + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2+1 & 2+2 \\ 2+3 & 2+4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix}$

- normalize (say red channel)

  $= \begin{bmatrix} 143 & \cdots & 29 \\ \vdots & & \\ 25 & \cdots & 40 \end{bmatrix} / 255 = \begin{bmatrix} 0.56 & \cdots & 0.11 \\ \vdots & & \vdots \\ 0.17 & & 0.18 \end{bmatrix}$

- Matrix to matrix add:

<u>rule</u>: size of both matrix should be same

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} + \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 1+2 & 3+4 \\ 5+6 & 7+8 \end{bmatrix}$$

(2x2) (2x2)

$$= \begin{bmatrix} 3 & 7 \\ 11 & 15 \end{bmatrix}$$

(2x2)

Matrix multiplication

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \times \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 16 & 49 \\ 4 & 25 & 64 \\ 9 & 36 & 81 \end{bmatrix}$$

$$(3 \times 3) \times (3 \times 3) = (3 \times 3)$$

Shape (ip) Shape o/p

- Element wise matrix multiplication has been done.

Matrix product

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 30 & 36 & 42 \\ 66 & 81 & 96 \\ 102 & 126 & 150 \end{bmatrix}$$

→ element wise matrix multiplication needs matrices of same shape

→ dot product works on!

$$\begin{bmatrix} 0 & 2 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 7 & 13 & 19 \end{bmatrix}$$

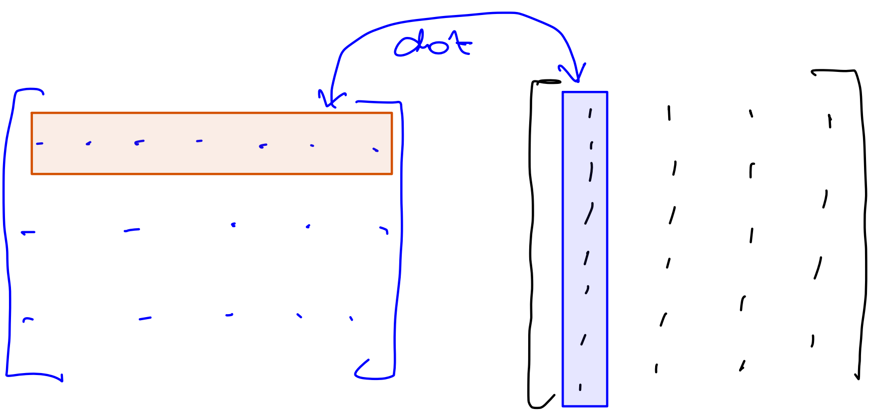
0×1

2×7

4×13

6×19

$$= 0 + 14 + 52 + 114 = 180$$



- Every row in left matrix multiplied to each column in right matrix

$$\begin{bmatrix} 0 & 2 & 4 & 6 \\ 8 & 10 & 12 & 14 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \\ 19 & 21 & 23 \end{bmatrix}$$

2x3   4 x 3

just 1 row & column

$$= \begin{bmatrix} 0 & 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ 7 \\ 13 \\ 19 \end{bmatrix} = 0\times1 + 2\times7 + 4\times13 + 6\times19 = 180$$

O/P

$$\begin{bmatrix} 180 & 204 & 228 \\ 500 & 588 & 676 \end{bmatrix}$$

$M_1 \cdot row_1$ dot $M_2\ col_1$ $=$ O/P M $row_1\ col_1$

- 3 columns in $M_2 \Rightarrow$ 3 dot products (for each row of M_1)
- 2 rows in $M_1 \Rightarrow$ 3 dot products for 1st row & 3 dot products for 2nd row

⇒ Total = 6 dot products

Here, to get 6 no.s we did 24 multiplication & 16 additions

- To do dot product, we need same size row and column vector from both matrices

$$\begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{bmatrix} \cdot \begin{bmatrix} 0 & 2 & 4 \\ 6 & 8 & 10 \end{bmatrix} =$$

let's see for Matrix M_1 row 1 × Matrix$_2$ col$_1$

$= (1 \times 0 + 3 \times 6 + 5 \times ?)$ and hence dot product is not possible

⇒ $\boxed{\text{No. of columns in left matrix} = \text{No. of rows in right matrix for Dot Product}}$

- length of row vector = no. of columns
- length of column vector = no. of rows

so,

$$\begin{array}{ccc} M_1 & \cdot & M_2 & & O/P \\ 2\times\boxed{3 & \cdot & 3}\times 2 & = & 2\times 2 \end{array}$$

must be same

same

same

○ **Order matter in dot product**

$$\begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{bmatrix} \cdot \begin{bmatrix} 0 & 6 \\ 2 & 8 \\ 4 & 10 \end{bmatrix} \neq \begin{bmatrix} 0 & 6 \\ 2 & 8 \\ 4 & 10 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \end{bmatrix}$$

$(2\times 3) \cdot (3\times 2)$ $\quad$ $(3\times 2) \cdot (2\times 3)$

$O/P = (2\times 2)$ $\quad$ $O/P = 3\times 3$

$$\boxed{A \cdot B \neq B \cdot A}$$

Not Commutative

○ Data in the left matrix should be arranged as rows & right matrix in columns

Matrix Transposes

$$\begin{bmatrix} \boxed{1 \quad 2 \quad 3} \\ 4 \quad 5 \quad 6 \\ 7 \quad 8 \quad 9 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$\longleftrightarrow$
Transpose

- Only rows and columns are swapped

- $A_{(2,4)} \xrightarrow{\text{Transpose}} B_{(4,2)}$

eg: $\begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{bmatrix} \cdot \begin{bmatrix} 0 & 2 \\ 4 & 6 \\ 8 & 10 \\ 12 & 14 \end{bmatrix}$ O/P : not possible

(3×2) $\quad$ (4×2)

$\Updownarrow$ Let's transpose right matrix

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{bmatrix} \cdot \begin{bmatrix} 0 & 4 & 8 & 12 \\ 2 & 6 & 10 & 14 \end{bmatrix}$$ (Approach A)

$(3 \times 2) \cdot (2 \times 4) \Rightarrow O/P = (3 \times 4)$

Or,

$$\begin{bmatrix} 0 & 2 \\ 4 & 6 \\ 8 & 10 \\ 12 & 14 \end{bmatrix} \cdot \begin{bmatrix} 1 & 5 & 9 \\ 3 & 7 & 11 \end{bmatrix} \Rightarrow O/P = (4 \times 3)$$ (Approach B)

$(4 \times 2) \quad (2 \times 3)$

- Question is when to use Approach A vs B?

Ans: If both matrix contains original data in rows, then,

$B^T \rightarrow$ (row will be moved to column)

& outcome would be ok for both Approaches

$$\begin{array}{c}A\\B\\C\end{array}\begin{bmatrix}1 & 3\\5 & 7\\9 & 11\end{bmatrix} \cdot \begin{array}{c}D\\E\\F\\G\end{array}\begin{bmatrix}0 & 2\\4 & 6\\8 & 10\\12 & 14\end{bmatrix} \qquad \longleftrightarrow \qquad \begin{array}{c}A\\B\\C\end{array}\begin{bmatrix}1 & 3\\5 & 7\\9 & 11\end{bmatrix} \cdot \begin{array}{cccc}D & E & F & G\\ \end{array}\begin{bmatrix}0 & 4 & 8 & 12\\2 & 6 & 10 & 14\end{bmatrix}$$

$$3\times 2 \qquad 4\times 2 \qquad\qquad 3\times 2 \qquad 2\times 4$$

$$= \underset{3\times 2}{\begin{bmatrix}1 & 3\\5 & 7\\9 & 11\end{bmatrix}} \cdot \underset{2\times 4}{\begin{bmatrix}0 & 4 & 8 & 12\\2 & 6 & 10 & 14\end{bmatrix}} = \underset{3\times 4}{\begin{bmatrix}6 & 22 & 38 & 54\\14 & 62 & 110 & 158\\22 & 102 & 182 & 262\end{bmatrix}} \qquad \begin{bmatrix}A\times D & A\times E & \\B\times D & B\times E & \\C\times D & C\times E & \cdots\end{bmatrix}$$

- But, if original data was arranged in columns then, none of the Approach A or B works

$$\underset{(3\times 2)}{\begin{array}{cc}A & B\\ \end{array}\begin{bmatrix}1 & 3\\5 & 7\\9 & 11\end{bmatrix}} \cdot \underset{(4\times 2)}{\begin{array}{cc}C & D\\ \end{array}\begin{bmatrix}0 & 2\\4 & 6\\8 & 10\\12 & 14\end{bmatrix}}$$

Product ⇒

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 11 \end{bmatrix} \cdot \begin{bmatrix} 0 & 4 & 8 & 12 \\ 2 & 6 & 10 & 14 \end{bmatrix}$$

A $\quad$ θ

thus, all elements of A is multiplied to 1 element of item type c and 1 of θ which is wrong