Project 2: Matrix Transformations

Modern computer hardware optimized to do matrix multiplication.

$\Rightarrow$ we want to leverage this power to efficiently manipulate data.

In project 2, you will take this approach to

- move: translate
- stretch: scale
- rotate

data to uncover hidden patterns.

A matrix transformation is a matrix multiplication between a transformation matrix M and a data matrix A:

manipulated data

transformation matrix on left

Numpy matrix multiplication operator

data matrix on the right

$$A' = M @ A$$

Example: Temperature data taken in Maine, Florida, and Washington DC over different days:

$M = 3$ [3 variables]

col of 1s added

$$A = \begin{array}{c|cccc} & ME & FL & DC & \\ \text{day1} & 32 & 105 & 50 & 1 \\ \text{day2} & 6 & 85 & 40 & 1 \end{array}$$

we will see why this is here shortly

$$M = \begin{bmatrix} 1 & 0 & 0 & -32 \\ 0 & 1 & 0 & -80 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Let's do the matrix multiplication and see what happens:

1) check shape compatibility:

$M@A : (4, 4) \times (2, 4)$

mismatch! can't multiply

$\Rightarrow$ Need to transpose $\underline{A}$

$$A.T = \begin{bmatrix} 32 & 6 \\ 105 & 85 \\ 50 & 40 \\ 1 & 1 \end{bmatrix}$$

$M@A.T$: $(4, 4) \times (4, 2)$

Match! we're good

2) Get output shape: $(4, 4) \times (4, 2) = (4, 2)$

3) Do multiplication: $M@A.T$:

$$\begin{bmatrix} *1 & *2 \\ *3 & *4 \\ *5 & *6 \\ *7 & *8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -32 \\ 0 & 1 & 0 & -80 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 32 & 6 \\ 105 & 85 \\ 50 & 40 \\ 1 & 1 \end{bmatrix}$$

$*1 = 1 \cdot 32 + 0 \cdot 105 + 0 \cdot 50 - 32 \cdot 1 = 0$

$*2 = 1 \cdot 6 + 0 \cdot 85 + 0 \cdot 40 - 32 \cdot 1 = -26$

$*3 = 0 \cdot 32 + 1 \cdot 105 + 0 \cdot 50 - 80 \cdot 1 = 25$

$* 4 = 0 \cdot 6 + 1 \cdot 85 + 0 \cdot 40 - 80 \cdot 1 = 5$

$* 5 = 0.32 \cdot 0 \cdot 105 + 1.50 + 10 \cdot 1 = 60$

$* 6 = 0.6 + 0.85 + 1.40 + 10 \cdot 1 = 50$

$* 7 = 0 \cdot 32 + 0 \cdot 105 + 0.50 + 1 \cdot 1 = 1$

$* 8 = 0 \cdot 6 + 0 \cdot 85 + 0 \cdot 40 + 1 \cdot 1 = 1$

<u>Output</u>

$$= \begin{bmatrix} 0 & -26 \\ 25 & 5 \\ 60 & 50 \\ 1 & 1 \end{bmatrix} \qquad \text{shape} = (4, 2)$$

* this is a data transformation.

want shape(output) = shape(input)

[Not adding/dropping data]

original shape = (2, 4)

⇒ we need to transpose output:

$$A' = \begin{bmatrix} 0 & 25 & 60 & 1 \\ -26 & 5 & 50 & 1 \end{bmatrix}$$

Compare with original data matrix D:

$$A = \begin{array}{c} \\ \text{day1} \\ \text{day2} \end{array} \begin{array}{c} \begin{array}{cccc} \text{ME} & \text{FL} & \text{DC} & \end{array} \\ \begin{bmatrix} 32 & 105 & 50 & 1 \\ 6 & 85 & 40 & 1 \end{bmatrix} \end{array}$$

How are output ME values different from input? $\boxed{\text{32 less}}$

How are output FL values different from input? $\boxed{\text{80 less}}$

How are output DC values different from input? $\boxed{\text{10 more}}$

How did M do this?

our transformation matrix:

$$M = \begin{bmatrix} 1 & 0 & 0 & -32 \\ 0 & 1 & 0 & -80 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ME shift [var 1]

FL shift [var 2]

DC shift [var 3]

A transformation matrix that has the effect of shifting variables by some amount is called translation matrix [the transformation called translation]

and denoted T:

$$T = T_3(\vec{t}) = \overbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{4\times4 \text{ for 3D data}} \quad \text{where } \vec{t} = (t_x, t_y, t_z)$$

translates 3 variables

translation vector

amount to shift each variable

above $\vec{t} = (-32, -80, 10)$

- What would $\vec{t}$ be we wanted to subtract 6 from ME, but leave others alone?

$$\vec{t} = (-6, 0, 0)$$

- What if we wanted to add 100 to ME temps, 50 to FL temps, and subtract 20 from DC temps?

$$\vec{t} = (100, 50, -20)$$

Translation summary

To translate data variables by amounts $\vec{t}$, compute:

$$\boxed{A' = (T A.T).T}$$

Implementation tip: $T_3 = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Can be made by making a 4x4 identity matrix:

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

then replacing values in last column with $\vec{t}$

[with an extra 1 appended at the end]

$$\begin{bmatrix} 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \\ 0 & 0 & 0 & \end{bmatrix} \qquad \vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix}$$