Vectorization + Linear Algebra

Input Dimensions

- d^n
- # of features • # observation
- Each column is a datapoint
- d = dimensionality of input
- n = # of examples in dataset

We use nxd because it works w $W^T X$

Vectorization – get rid of explicit for loops w linear algebra

- For loops are slow, especially when we work w/ large datasets

$$z = \sum_{i=1}^{n_x} w_i x_i + b$$

$\hat{y} = \sigma(z)$ → Take as input, do sigmoid operation

two col vectors

$$\vec{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n_x} \end{bmatrix}$$

Non vectorized:

```
z = 0

for i in range n(x):
    z += w[i] * x[i]

z += b
```

→ this is slow!

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_x} \end{bmatrix}$$

Vectorized version:

```
z = np.dot(w, x) + b
```

$w^T x$

GPU/CPU have parallelization instructions to optimize this operations

- Numpy, tensorflow functions take advantage of parallelism

Linear Algebra

scalar: 24

vector: [3, 4, 7, -9] ← ordered array of #, can be row or column; has a single index

$$\text{matrix} = \begin{bmatrix} 2, 4, 7 \\ -8, -9, 18 \end{bmatrix}$$

2d array of # with 2 indexes
- First is the row, second is column

$$M = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m1} & M_{m2} & & M_{mn} \end{bmatrix}$$

n = rows
m = cols

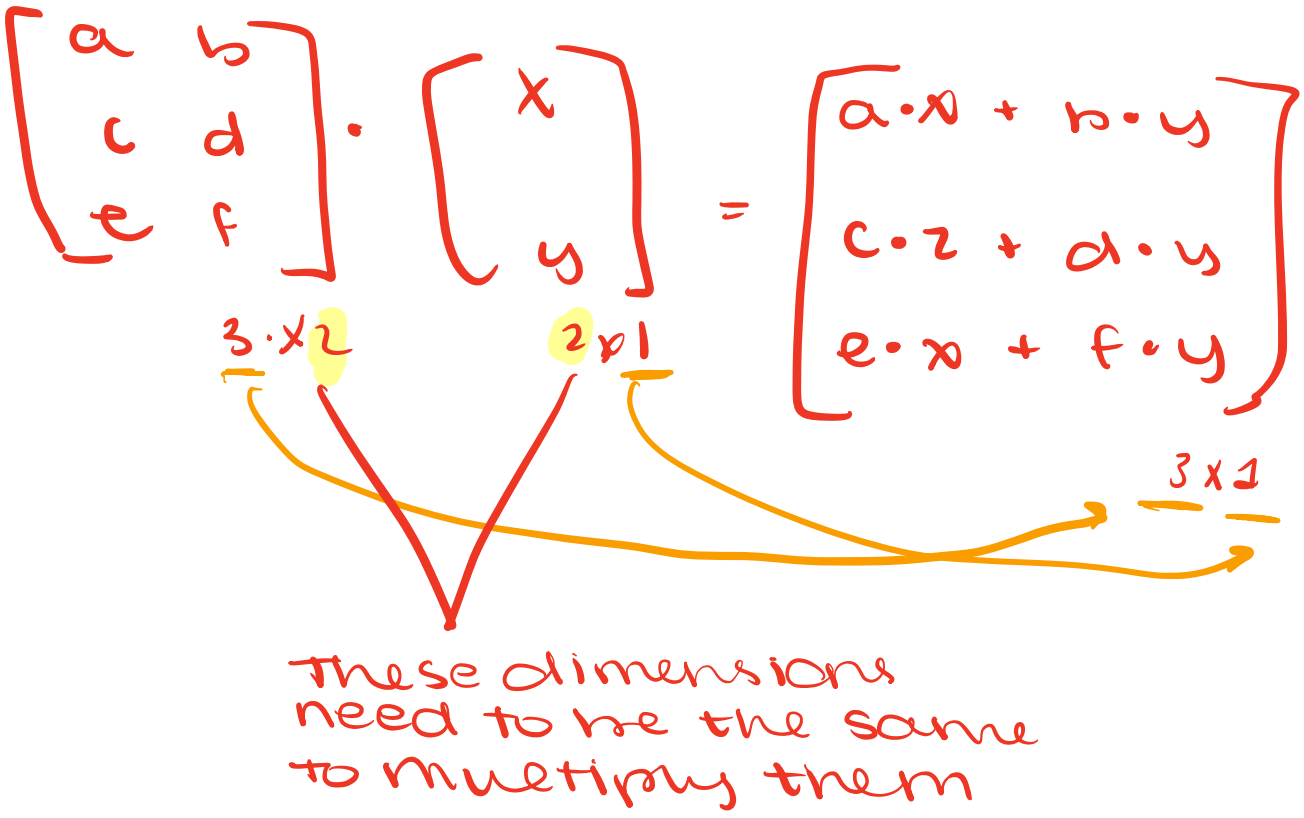
Index [3, 2] ⇒ 2nd element in 3rd row

Tensor = array of numbers in regular grid, variable # of axes

- 3 index: row, column, axes
- 1D tensor = vector
- 2D tensor = matrix

Computational Rules

- matrix scalar operations
- If you multiply/divide/subtract/etc a scalar, you do so to every element
- Multiply matrix by vector → multiply each row of matrix by the column of the vector
  - ↳ output is vector w/ the same # of rows as matrix



$W^T X$ = Matrix multiplication

① W^T

② X

} need to make output dimensions work

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

→ all entry wise

Matrix-Matrix multiplication

- Turn 2nd matrix into column vectors, multiply 1st matrix by each vector

Split

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 3 \\ 0 & 1 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} & \\ & \end{bmatrix}$$

$2 \cdot 3$ $\quad$ $3 \cdot 2$ $\quad$ $2 \cdot 2$

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

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \end{bmatrix}$$

$$= \begin{bmatrix} 11 & 10 \\ 9 & 14 \end{bmatrix}$$

$$\underset{A}{\begin{bmatrix} \quad \end{bmatrix}} \cdot \underset{B}{\begin{bmatrix} \quad \end{bmatrix}} = \underset{C}{\begin{bmatrix} \quad \end{bmatrix}}$$

$m \cdot n \quad \textcircled{=} \quad n \cdot l \qquad = m \cdot l$

Properties of Matrix-Matrix Multiplication

- Not commutive
- Associative
- Distributive
- Identity Matrix

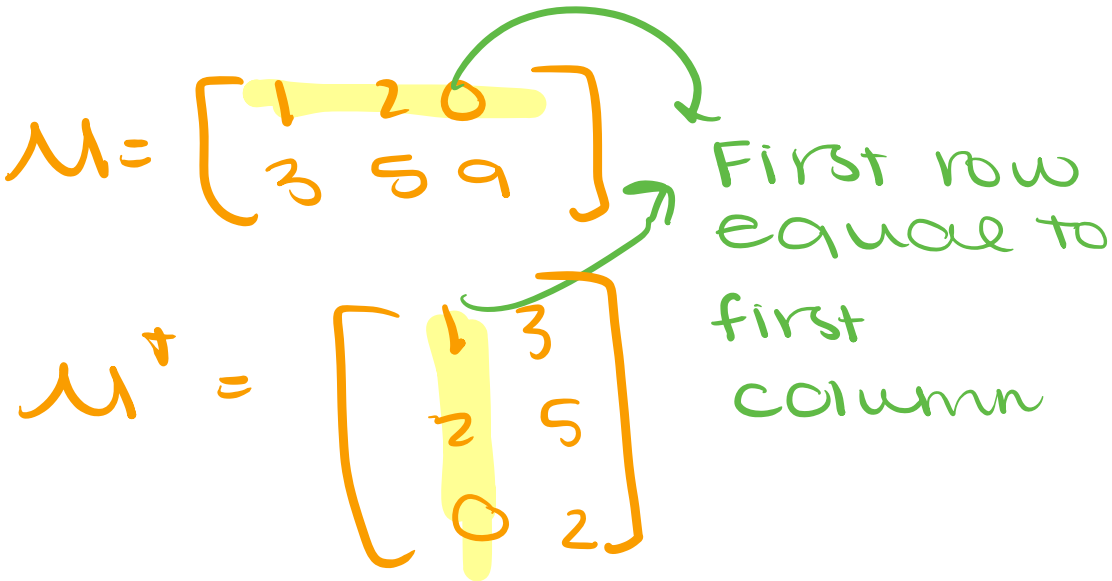
$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

All 1 on diagnol

$$A \cdot I = I \cdot A = A$$

MATRIX TRANSPOSE

- mirror image of matrix on 45° axis
- m·n matrix transformed to n·m



$$z = \sum w_i x_i + b = w^T x + b$$

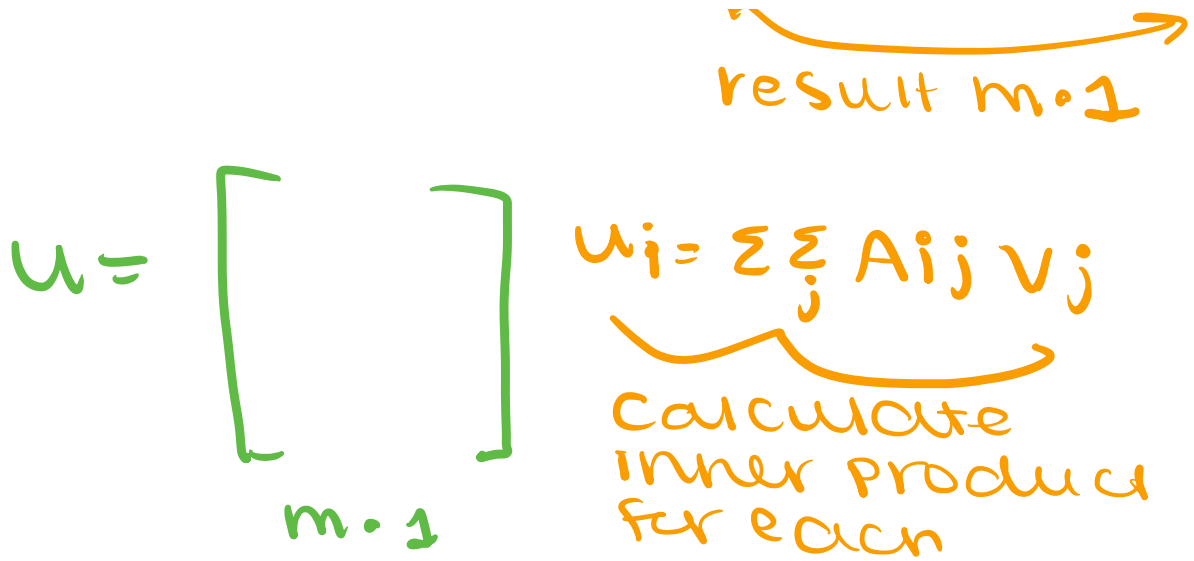
`np.dot(w, x)`

`tf.dot(w, x)`

whenever possible avoid for loops

$u = A \cdot v$ $\quad A = \begin{bmatrix} - & - \\ - & - \end{bmatrix}$ $\quad v = \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}$

A: $m \cdot d$ $\quad v$: $d \cdot 1$



NOT VECTORIZED

```
u = np.zeros((n,1))
for i ....
    for j ....
      u[i] += A[i][j] * v[j]
```

VECTORIZED

```
u = np.dot(A,v)
```

$$w = [w_1, w_2]$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \end{bmatrix}$$

$2 \times m$

Bias is scalar

Vectorizing Logistic Reg

$$X = \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \cdots & x^m \\ | & | & & | \end{bmatrix}$$

$n_x \cdot m$

$$w^T \cdot \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \cdots & x^m \\ | & | & & | \end{bmatrix}$$

Multiply w^T by each column

Absorb all calculations into one big matrix

$$Z = [z^1 \cdots z^m]$$

$$= w^T X + [b, b \cdots b]$$

$$Z = [w^T x^1 + b,\ w^T x^2 + b, \cdots \cdots w^T x^m + b]$$

$$A = [a^1, a^{21} \cdots a^m] = \sigma(z)$$

Apply same operation