

Parameters and features

$$\vec{w} = [w_1 \ w_2 \ w_3] \quad n=3$$

b is a number

$$\vec{x} = [x_1 \ x_2 \ x_3]$$

linear algebra: count from 1

NumPy

```
w = np.array([1.0, 2.5, -3.3])
```

```
b = 4
```

```
x = np.array([10, 20, 30])
```

code: count from 0

Without vectorization $n=100,000$

$$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + w_3x_3 + b$$

```
f = w[0] * x[0] +  
     w[1] * x[1] +  
     w[2] * x[2] + b
```



Without vectorization

$$f_{\vec{w},b}(\vec{x}) = \left(\sum_{j=1}^n w_j x_j \right) + b \quad \sum_{j=1}^n \rightarrow j=1 \dots n \quad 1, 2, 3$$

$\text{range}(0, n) \rightarrow j=0 \dots n-1$

```
f = 0  
for j in range(0, n):  
    f = f + w[j] * x[j]  
f = f + b
```



Vectorization

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

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



→ We have seen the multitasked linear regression where we have multiple feature x with multiple w for each x .

→ So, we can compute the calculation manually or using loop.

→ But it would not be efficient.

→ That's why we will use Numpy dot function. It will compute the dot function with vectorization.