LINEAR REGRESSION USING GRADIENT DESCENT

Multiple features

SINGLE FEATURE

Data table

size in feet^2	price in ₹1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870

MULTIPLE FEATURES

x_j = j^{th} feature

n = number of features

$\vec{x}^{(i)}$ = feature of i^{th} training example

$x_j^{(i)}$ = Value of j^{th} feature in i^{th} training example.

Size in feet² x_1	Number of bedrooms x_2	Number of floors x_3	Age of home in years x_4	Price (₹) in ₹ 1000's Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

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

$$f_{w,b}(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Parameters $\begin{cases} \vec{w} = [w_1 \; w_2 \; w_3 \cdots w_n] \\ b = \text{Single number} \end{cases}$

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

SIGNIFICANCE OF VECTORIZATION (PROGRAMMING PYTHON & NUMPY)

Declaration

```
w = np.array([2.0, 6.5, -3.2])
b = 3
x = np.array([15, 35, 24])
```

Without Vectorization

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

Without Vectorization

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

Vectorization

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

SIGNIFICANCE OF VECTORIZATION (PROGRAMMING PYTHON & NUMPY)

Without vectorization

```
for j in range(0,16):
    f = f + w[j] * x[j]
```

t_0
```
f + w[0] * x[0]
```
t_1
```
f + w[1] * x[1]
```
…

t_{15}
```
f + w[15] * x[15]
```

Vectorization

```
np.dot(w,x)
```