Linear Regression with Multiple featurer (variables)

feet^2 X_1	bedroom X_2	floor X_3	age X_4	price
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

X_j = j^{th} featurer $j = 1, .. 4$

n = number of features → 4

$\vec{X}^{(i)}$ = features of i^{th} training exampler → $\vec{X}^{(2)} = [1416 \; 3 \; \underline{2} \; 40]$

$X_j^{(i)}$ = value of feature j in i^{th} example $X_3^{2} = 2$

$f_{w,b}(x) = wx + b$

$f_{w,b}(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$

$\vec{W} = [w_1 \; w_2 \; w_3 \ldots w_n]$ → vektör

$\vec{X} = [x_1 \; x_2 \; x_3 \ldots x_n]$

$\boxed{f_{\vec{w},b}(\vec{x}) = \vec{w}.\vec{x} + b}$ $= w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$

↳ Multiple Linear regression

Vectorization: kodu daha kısa ve daha hızlı yapar

code count from 0

```
w = np.array([1.0, 2.5, -3.3])
              w[0] w[1] w[2]
b = 4
x = np.array([10, 20, 30])
              x[0] x[1] x[2]
```

without vectorization :

$$f_{\vec{w}b}(\vec{x}) = \sum_{j=1}^{n} w_j x_j + b$$

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

range (0,n) → j=0,...,n-1

<u>vectorization</u>

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

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

→ çok daha hızlıdır.
→ çok daha kısa

* efficient → scale to large datasets

Gradient descent for multiple regression

$J(\vec{w}, b)$

$$w_j = w_j - a \frac{\partial}{\partial w_j} \underbrace{J(w_1, w_2, \ldots, w_n, b)}_{J(\vec{w}, b)}$$

$$b = b - a. \frac{\partial}{\partial b} J(w_1, w_2, \ldots, w_n, b)$$

$$w_1 = w_1 - a.\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_1^{(i)}$$

$\vdots$

$$w_n = w_n - a.\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_n^{(i)}$$

$$b = b - a.\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

simultaneously update

Feature Scalling

$\hat{price} = w_1 . x_1 + w_2 . x_2 + b$, x_1 = size, x_2 = bedroom

House : $x_1 = \underline{2000}$, $x_2 = \underline{5}$, price = 500k

- size of parameters w_1, w_2 ?

$w_1 = \underline{0.1}$, $w_2 = \underline{50}$, b = 50

$\hat{price} = 0.1 * 2000 + 50.5 + 50$ → more reasonable

Features

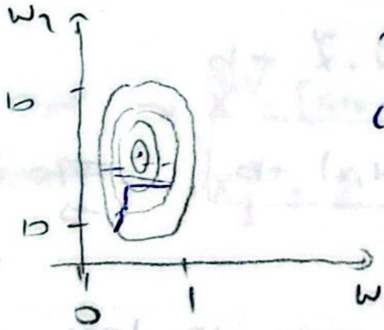
scatterplot

$x_1 \uparrow \rightarrow w_1 \downarrow$,
$x_2 \downarrow \rightarrow w_1 \uparrow$

Parameters

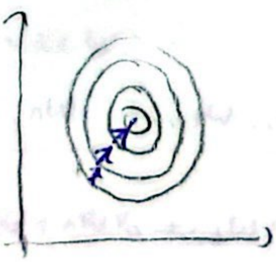
$J(\vec{w}, b)$

contourplot



↳ minimuma ulaşmak zaman alır

→



Böylece dereceli alçalma, global minimuma çok daha düzgün yoldan gider.
→ çok daha hızlı olur.

$300 \le x_1 \le 2000$

$x_{1\,scalled} = \frac{x_1}{2000}$

$0.15 \le x_{1\,scalled} \le 1$

$0 \le x_2 \le 5$

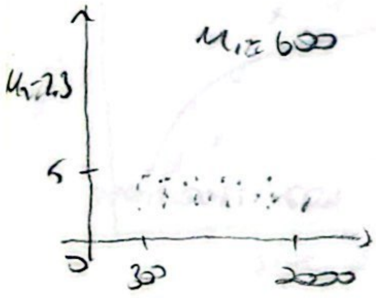
$x_{2\,scalled} = \frac{x_2}{5}$

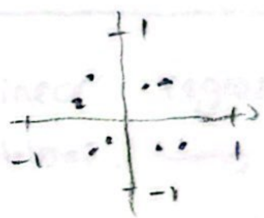
$0 \le x_{2\,scalled} \le 1$

Mean normalization

· orijinal özellikleri alıp hepsinin ort 0 olacak şekilde tekrar ölçeklendirme

· genelde -1 ile +1 aralında olur.





$300 \le x_1 \le 2000$

$x_1 = \frac{x_1 - \mu_1}{2000-300}$

$-0.18 \le x_1 \le 0.82$

$0 \le x_2 \le 5$

$x_2 = \frac{x_2 - \mu_2}{5-0}$

$-0.46 \le x_2 \le 0.54$

Z-score normalization

$300 \le x_1 \le 2000$

$x_1 = \frac{x_1 - \mu_1}{\sigma_1}$

$-0.67 \le x_1 \le 3.1$

$0 \le x_2 \le 5$

$x_2 = \frac{x_2 - \mu_2}{\sigma_2}$

$-1.6 \le x_2 \le 1.9$

* Feature scaling

aim for about $-1 \le x_j \le 1$ for each feature x_j

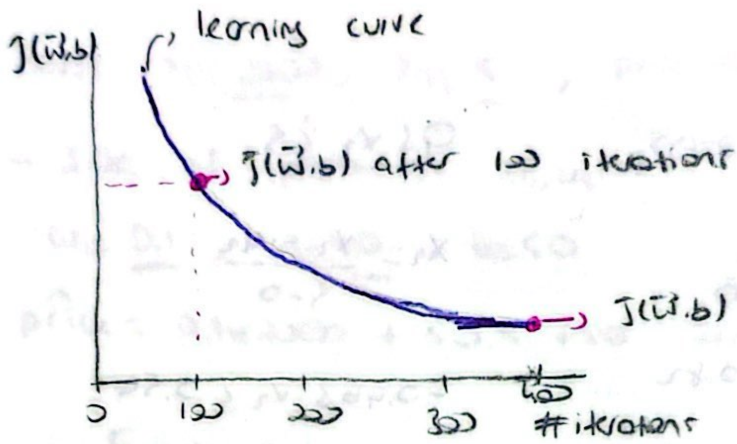
$-3 \le x_j \le 3$ ✓

$0 \le x_j \le 3$

Gradient descent for convergence

→ $\boxed{\min_{\vec{w},b} J(\vec{w},b)}$

✱ $J(\vec{w},b)$ should <u>decrease</u> after every iteration



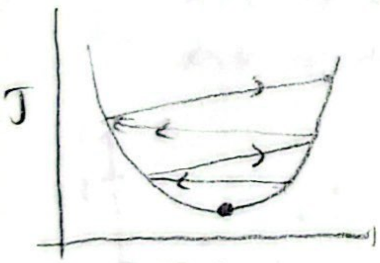
Automatic convergence test:

- Modeli eğitmenin ne zaman tamamlanacağını gösterir.

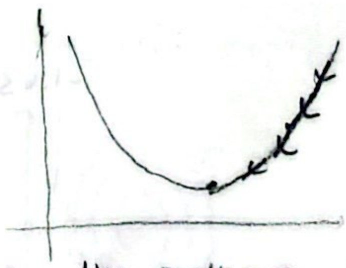
Let $\varepsilon = 10^{-3}$

→ If $J(\vec{w},b)$ decreases by $\leq \varepsilon$ in one iteration, declare convergence.

Choose learning rate



a is too big

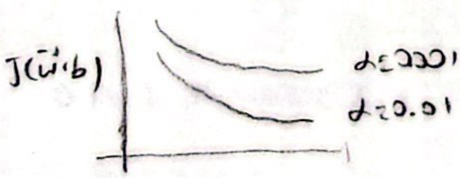


Use smaller a

→ a is too small, gradient descent takes a lot more iterations to converge

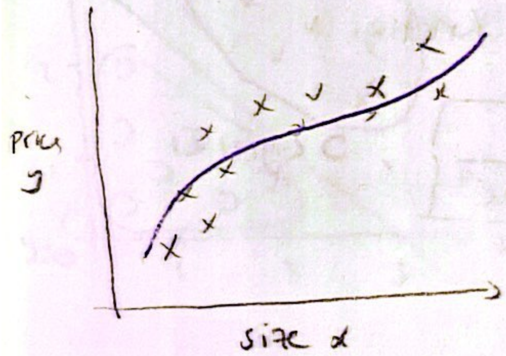
Values of α (Alpha) to try:

0.001 → 0.01 → 0.1 → 1



Scanned with CamScanner

* <u>Feature engineering</u> : Using intuition to design new features, by transforming or combining original features.



$\rightarrow$ $\cdot\ f_{\vec{w},b}(x) = w_1 x + w_2 x^2 + w_3 x^3 + b$

$\downarrow$ size, $1-10^3$

$\downarrow$ $size^2$, $1-10^6$

$\downarrow$ $size^3$, $1-10^9$

$\cdot\ f_{\vec{w},b} = w_1 x + w_2 \sqrt{x} + b$

Scanned with CamScanner