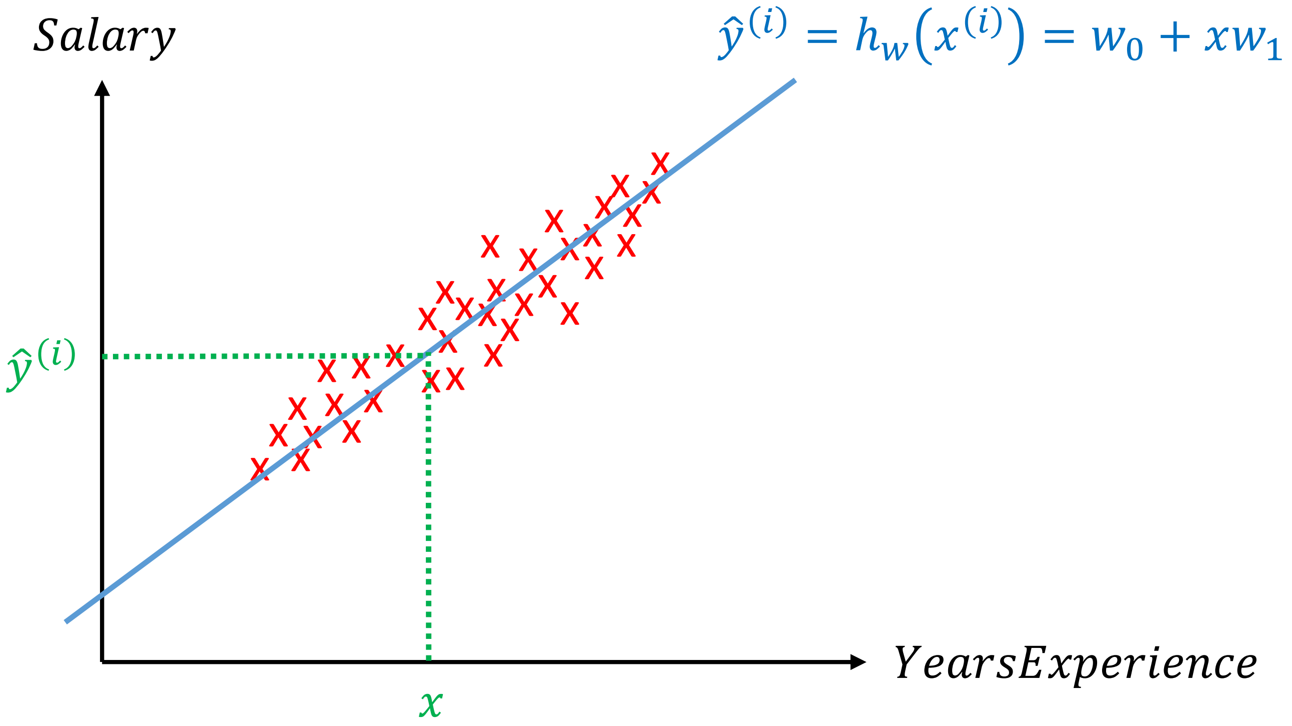
MAKİNE ÖĞRENMESİ

Tek Değişkenli Lineer Regresyon

1. Model Gösterimi



	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

$x^{(0)} = 1.1$

$x^{(1)} = 1.3$

$x^{(2)} = 1.5$

$y^{(1)} = 46205$

m : *Örnek Sayısı*

n : *Özellik Sayısı*

x : *girdi(input) değişkenleri*

y : *çıktı(output) etiketleri*

(x, y) : *toplam örnek sayısı*

$(x^{(i)}, y^{(i)})$: *i. örnek sayısı*

Eğitim Seti

↓

Öğrenme Algoritması

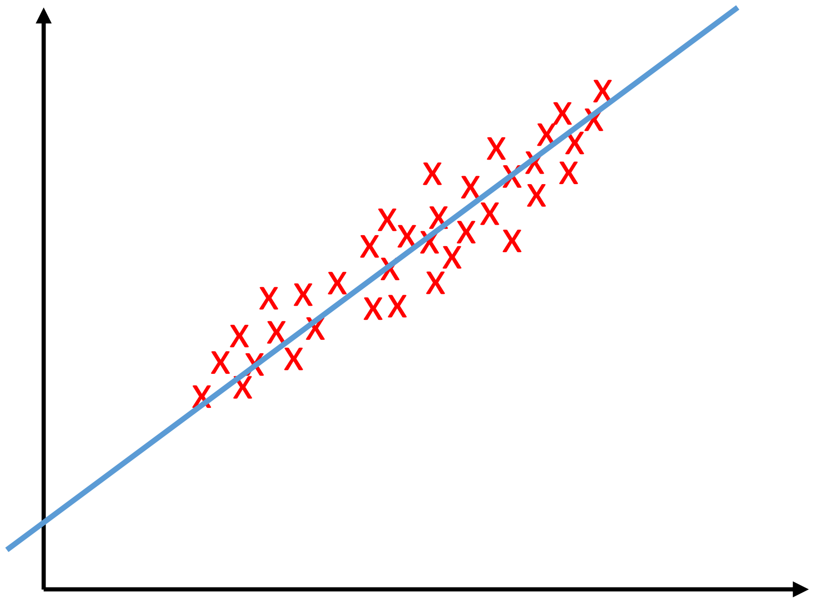
↓

Evin Büyüklüğü x → h *Hipotez* → *Evin Tahmini Değeri* $\hat{y}$

$h, x'ten\ y'ye\ haritalama\ yapar.$

$h(x)\ olarak\ da\ yazılabilir.$

$h_w(x) = w_0 + xw_1$



Tek değişkenli Doğrusal Regresyon

MAKİNE ÖĞRENMESİ

Tek Değişkenli Lineer Regresyon

2. Maliyet – Hata – Kayıp Fonksiyonu

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

Hipotez $h_w(x) = w_0 + xw_1$

Parametreler w_i : *Ağırlık Bileşenleri*

$$m = 30, n = 1$$

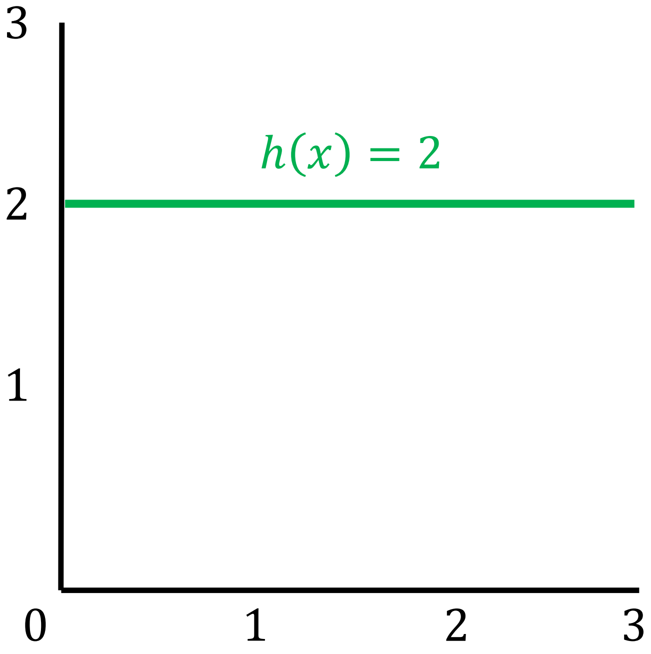
$$x_{m\times n} = x_{30\times 1}$$

$$y_{m\times 1} = y_{30\times 1}$$

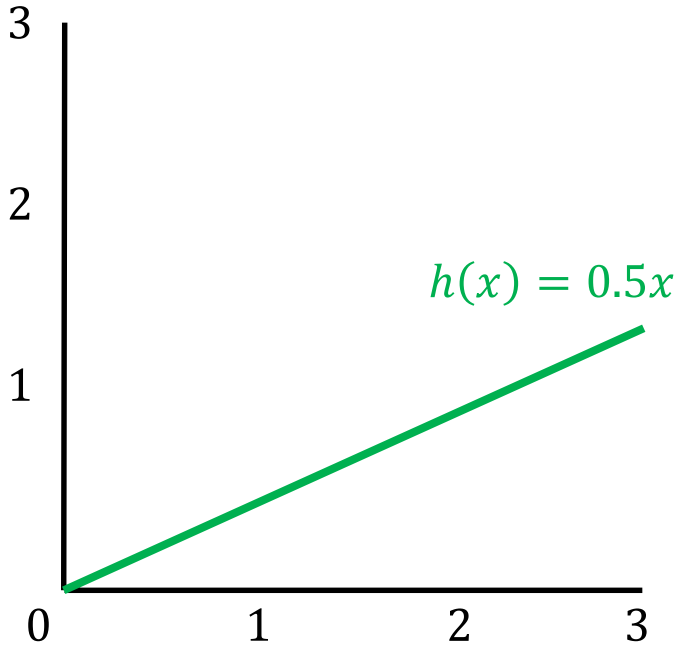
En uygun ağırlık bileşenlerini nasıl bulabiliriz ?

pandas

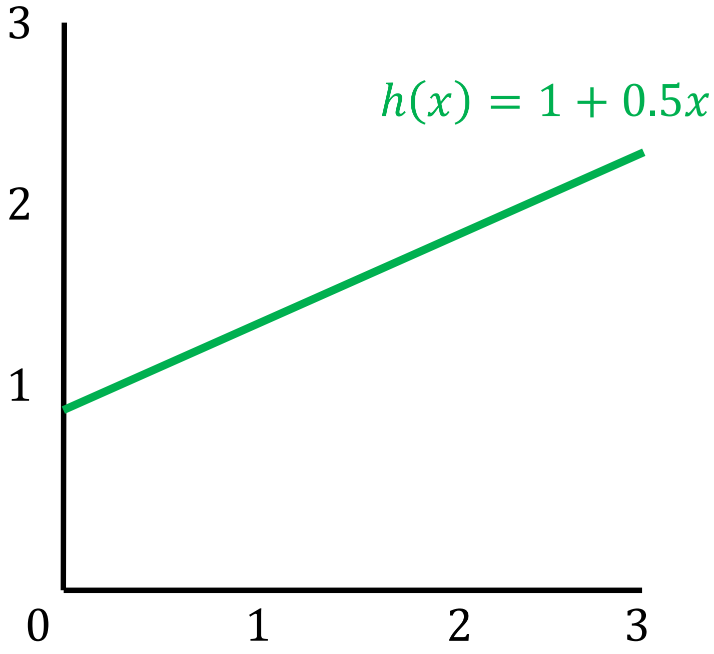
$$h_w(x) = w_0 + xw_1$$



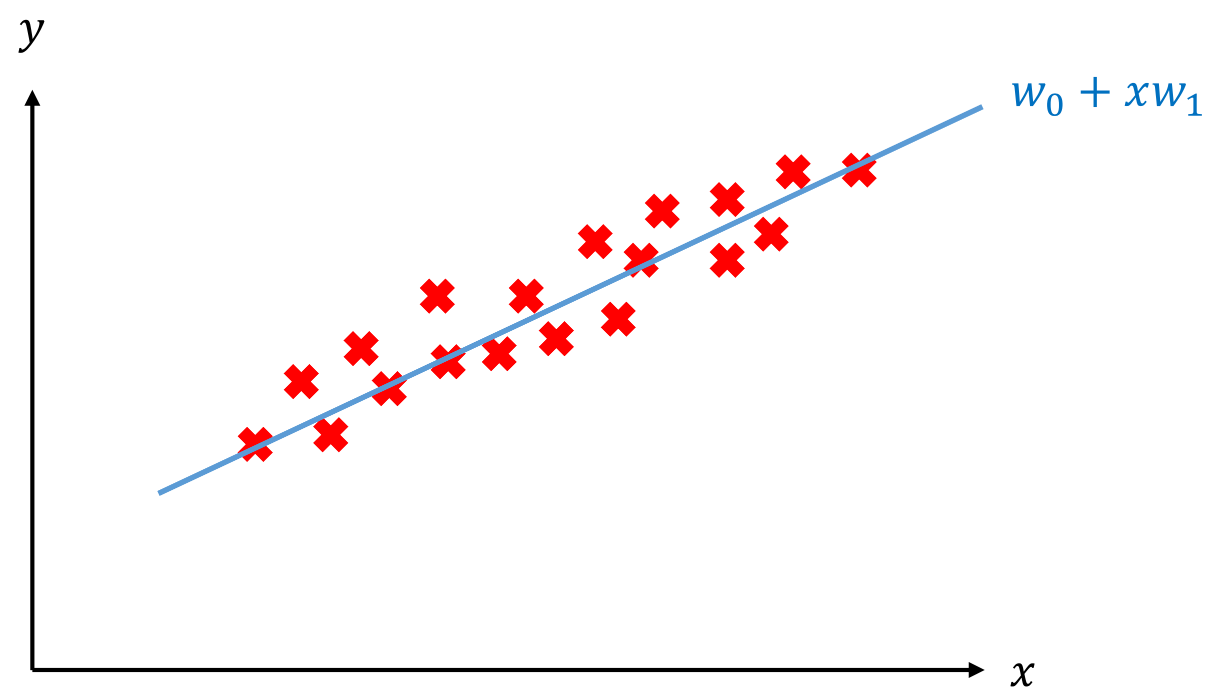
$w_0 = 2, w_1 = 0$



$w_0 = 0, w_1 = 0.5$



$w_0 = 1, w_1 = 0.5$



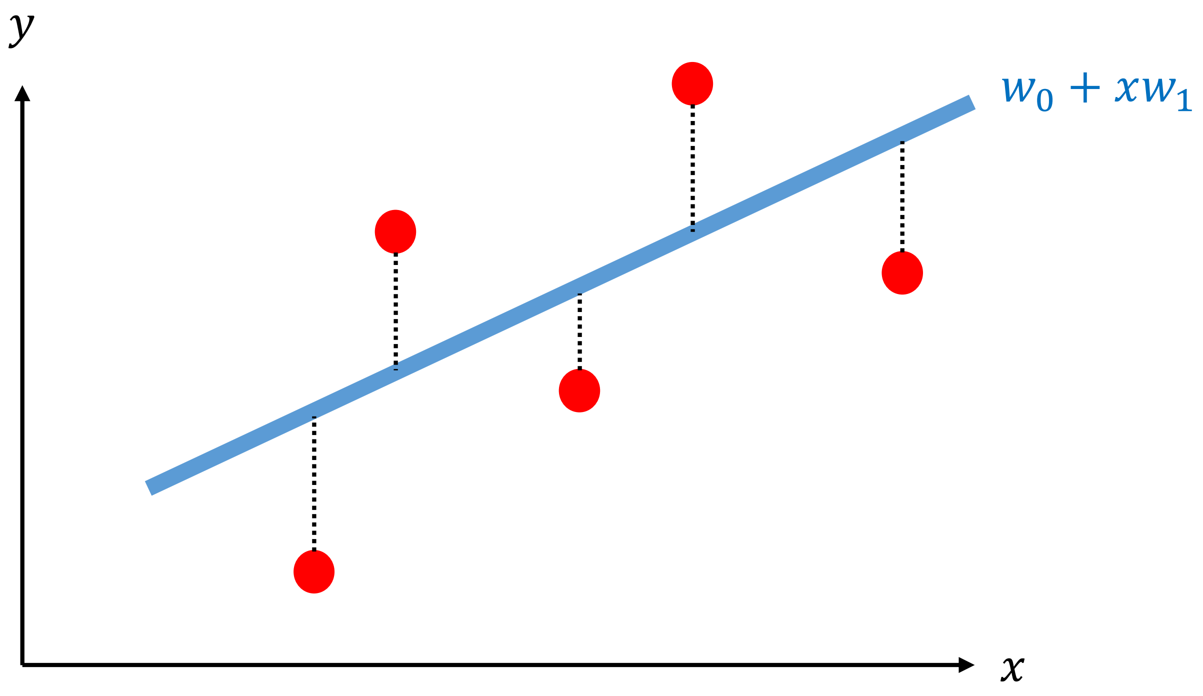
$$Kayıp(Cost):\ J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

$$Kayıp(Cost):\ J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$\hat{y}^{(i)} = h_w\left(x^{(i)}\right) = w_0 + xw_1$$

Amacımız kaybı minimize etmek

Öyle bir w_0 *ve* w_1 *seçki* $h_w(x)$*, y'ye çok yakın olsun.*



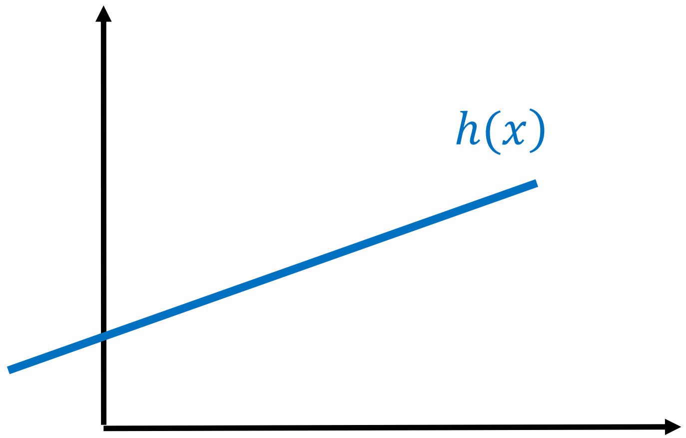
$$Kayıp(Cost):\ J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

Hata kare fonksiyonu
(squared error function)

pandas

Hipotez

$$h_w\left(x^{(i)}\right) = w_0 + x w_1$$



Parametreler

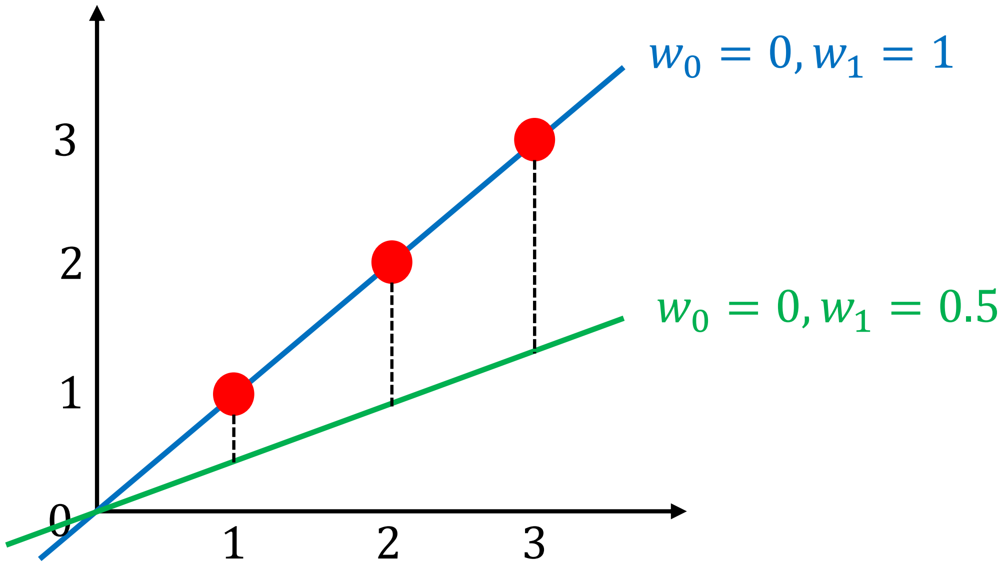
$$w_0, w_1$$

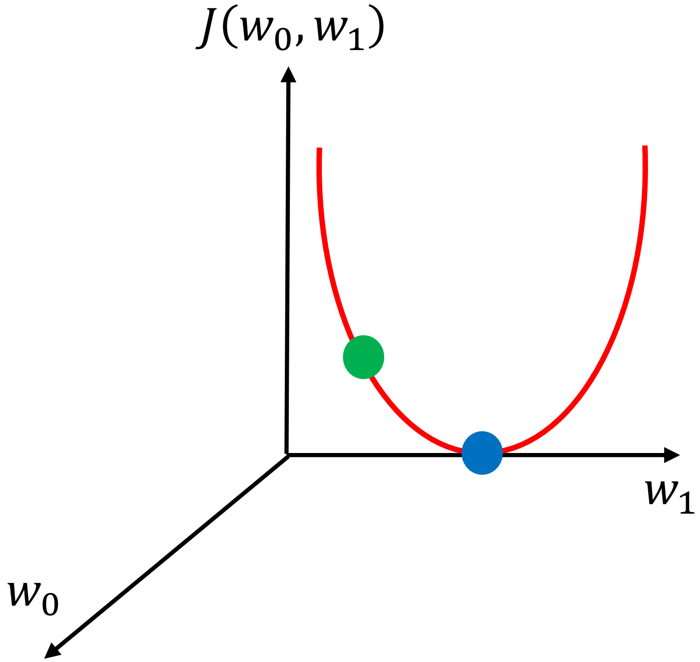
Kayıp (Maliyet) Fonksiyonu

$$Kayıp(Cost) : \ J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - h_w(x^{(i)})\right)^2$$

Amacımız kaybı minimize etmek

$$J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

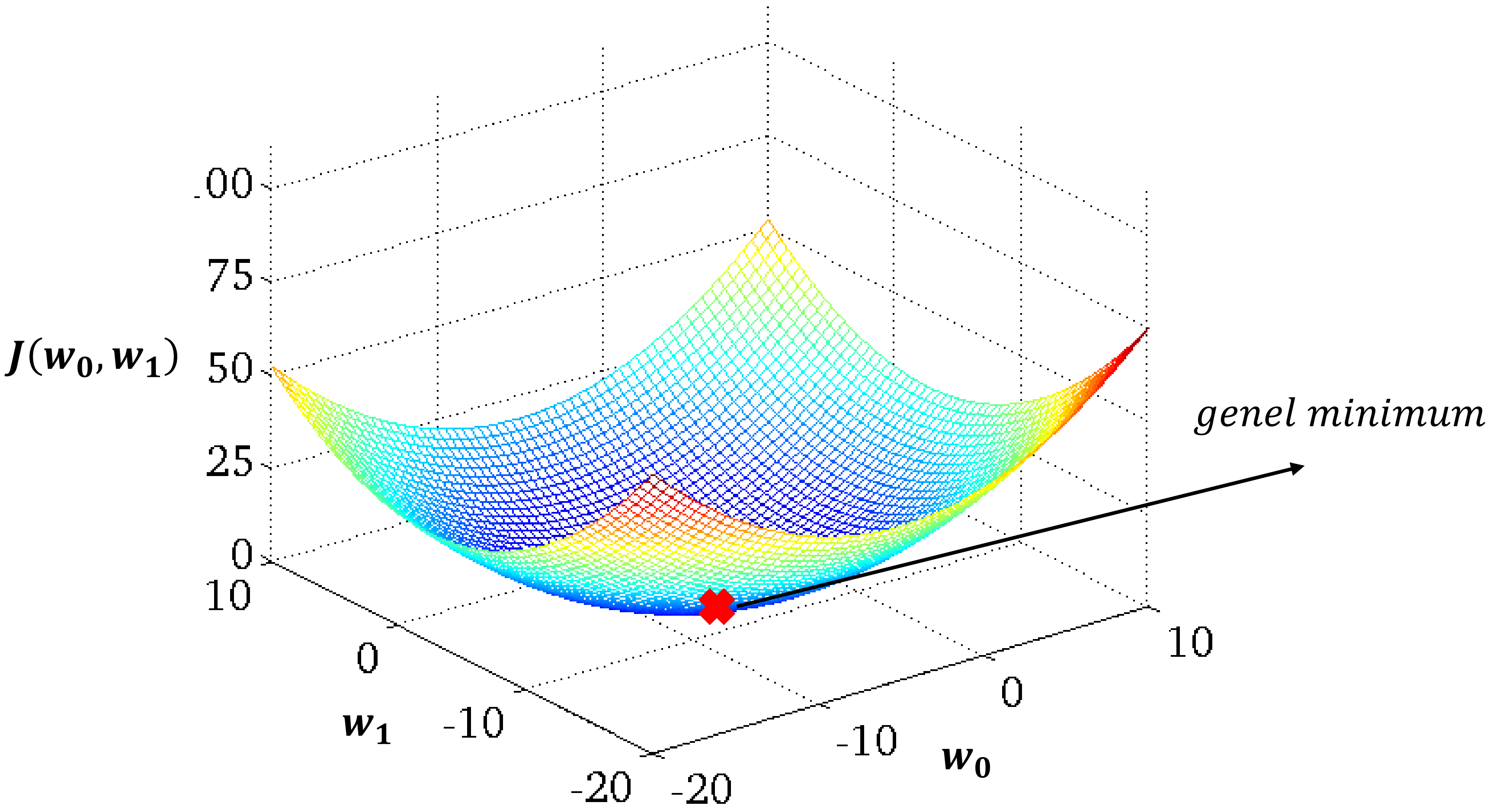




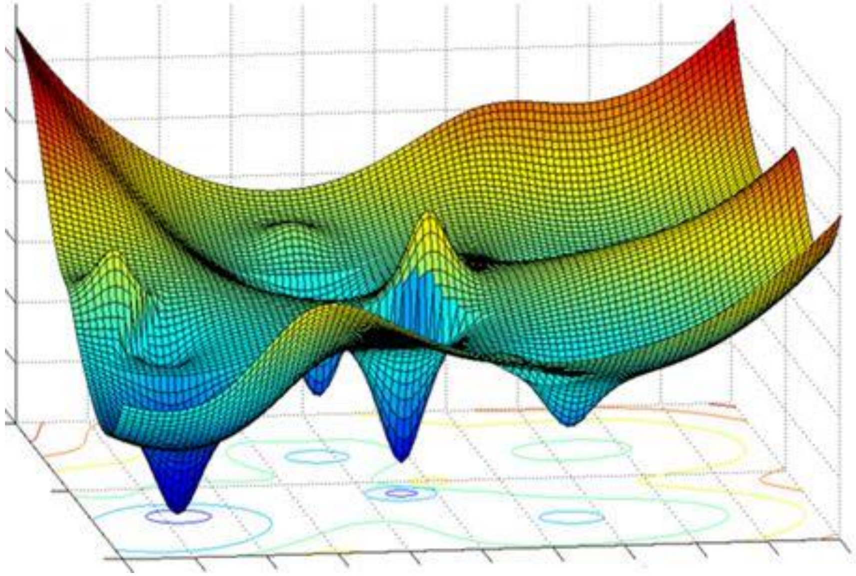
$$\boldsymbol{J(1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - x^{(i)}w_1\right)^2 = \frac{1}{2m}\left(0^2 + 0^2 + 0^2\right) = 0^2 = 0}$$

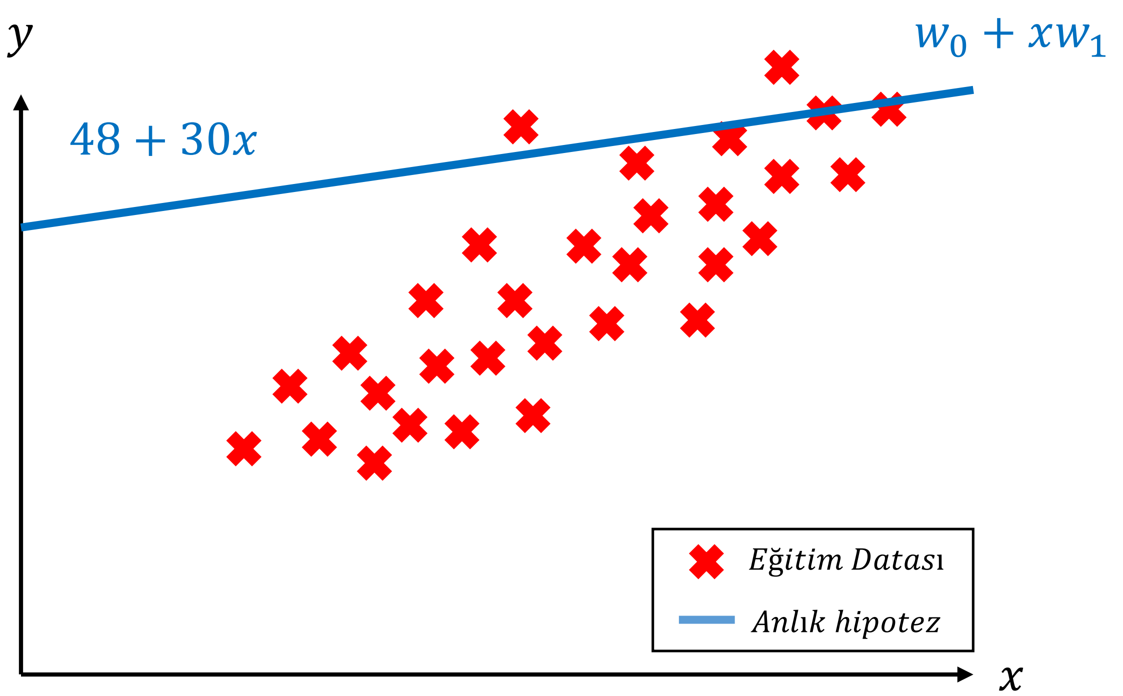
$$\boldsymbol{J(0.5) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - x^{(i)}w_1\right)^2 = \frac{1}{2m}\left((1 - 0.5)^2 + (2 - 1)^2 + (3 - 1.5)^2\right) = \frac{3.5}{6}}$$

pandas

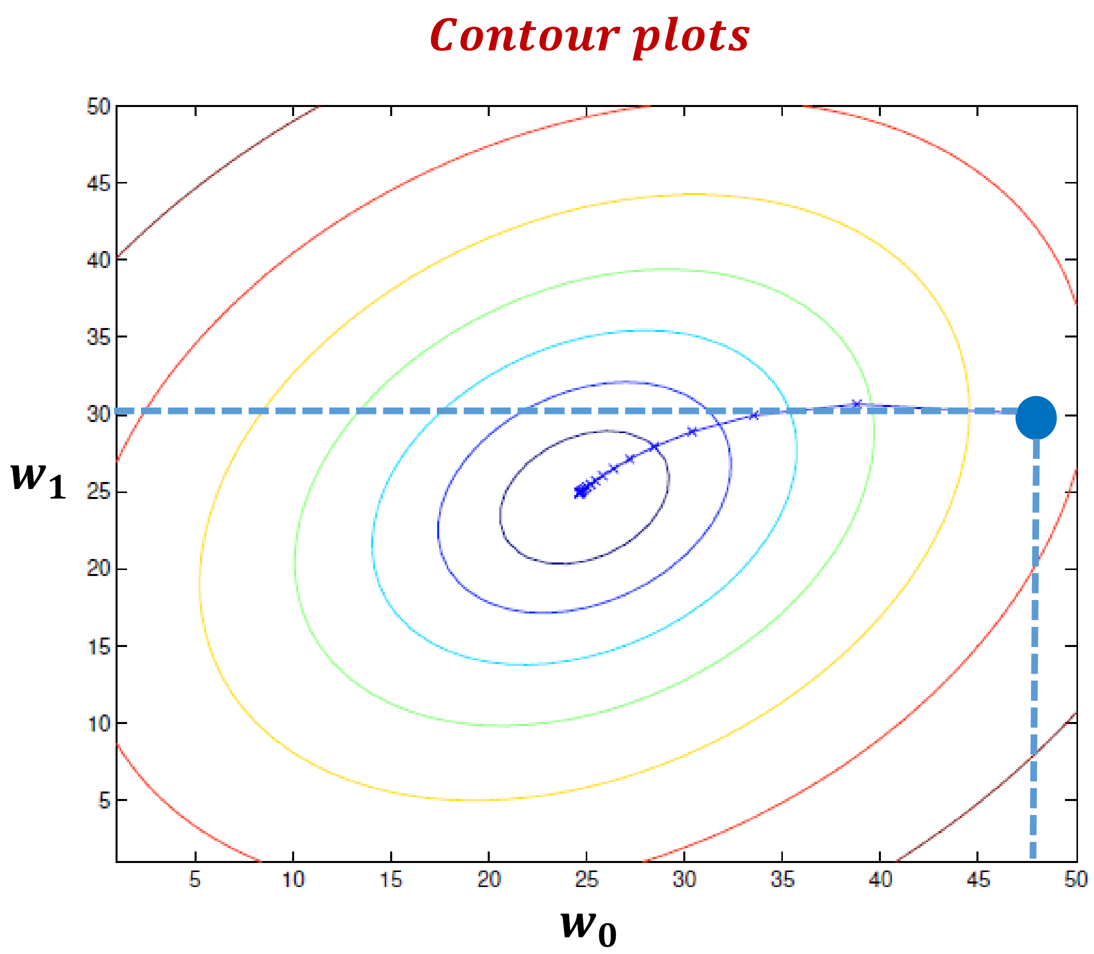


.00
75
50
25
0
$J(w_0, w_1)$
genel minimum
10
0
-10
-20
w_1
-20
-10
0
10
w_0

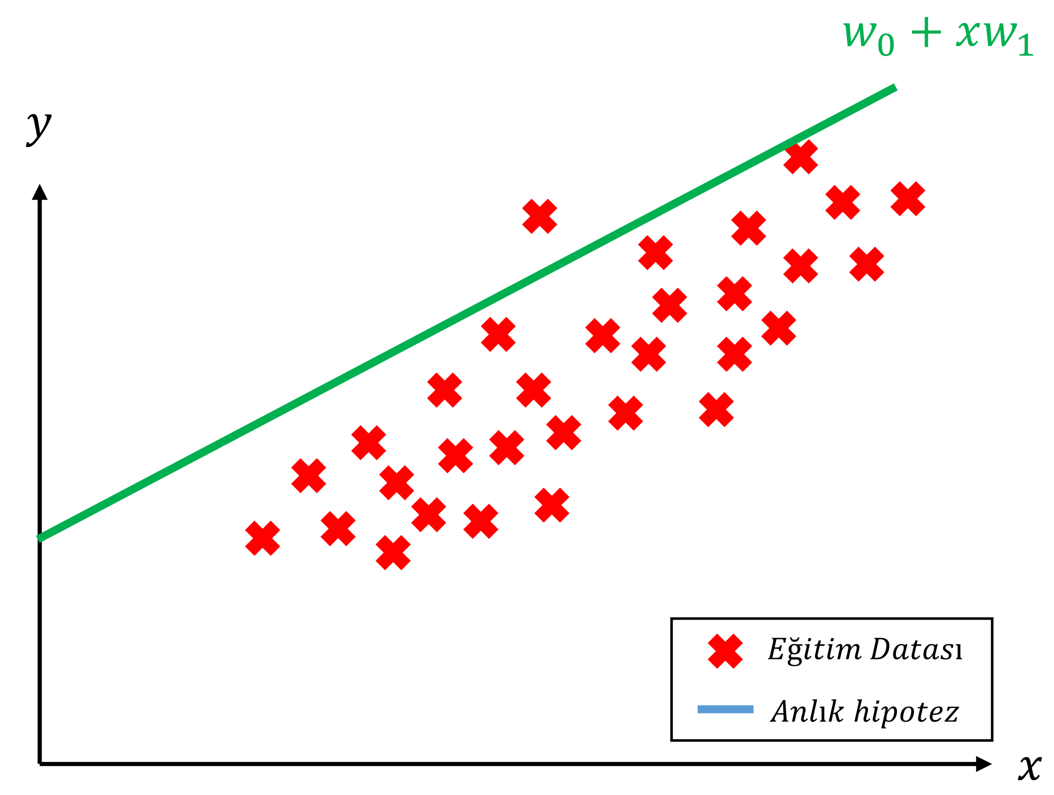




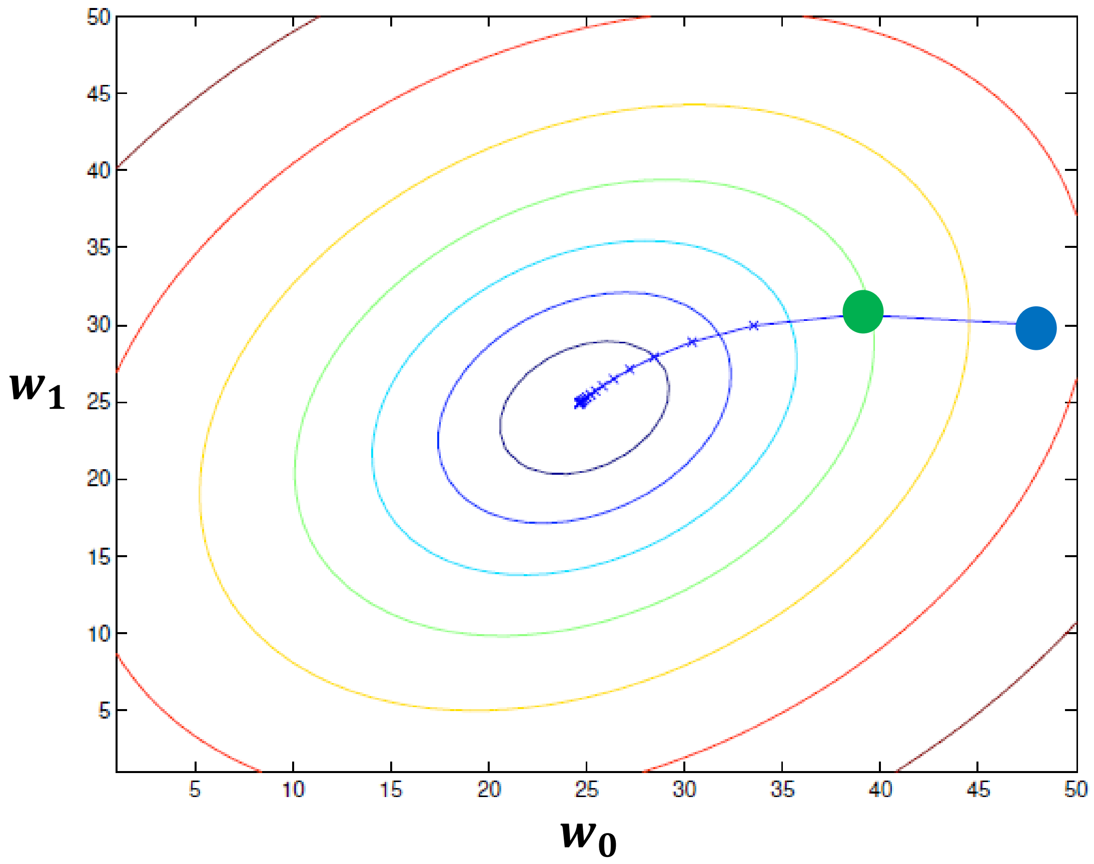
y
$w_0 + xw_1$
$48 + 30x$
x
Eğitim Datası
Anlık hipotez

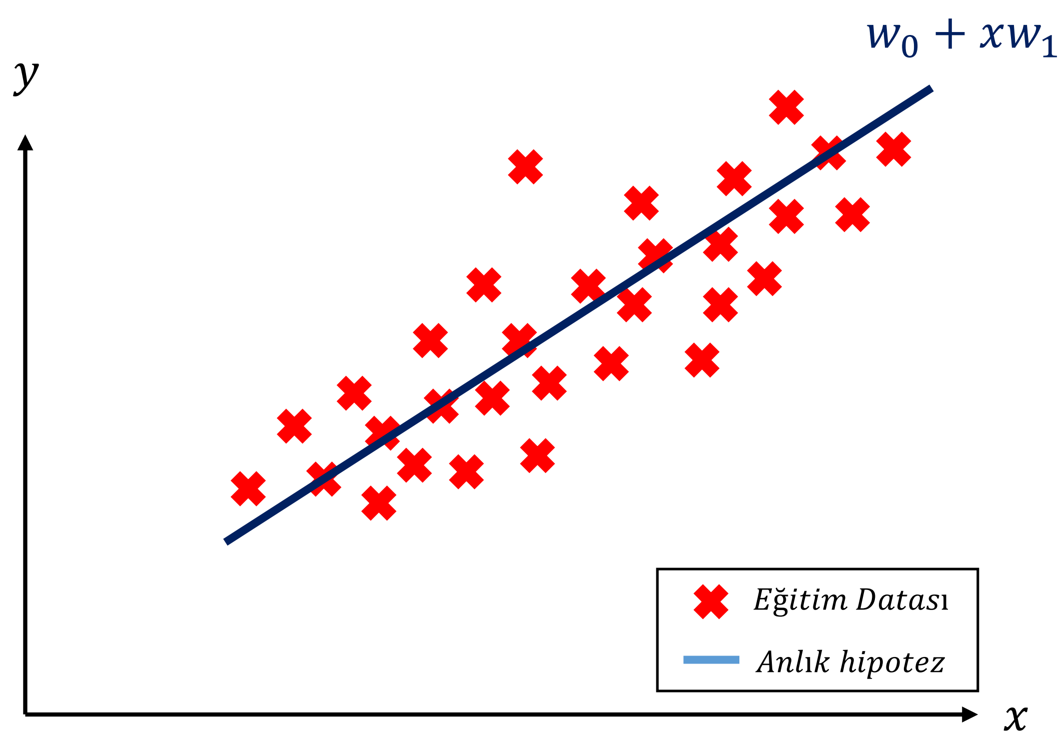


Contour plots
w_1
w_0
50 45 40 35 30 25 20 15 10 5
5 10 15 20 25 30 35 40 45 50

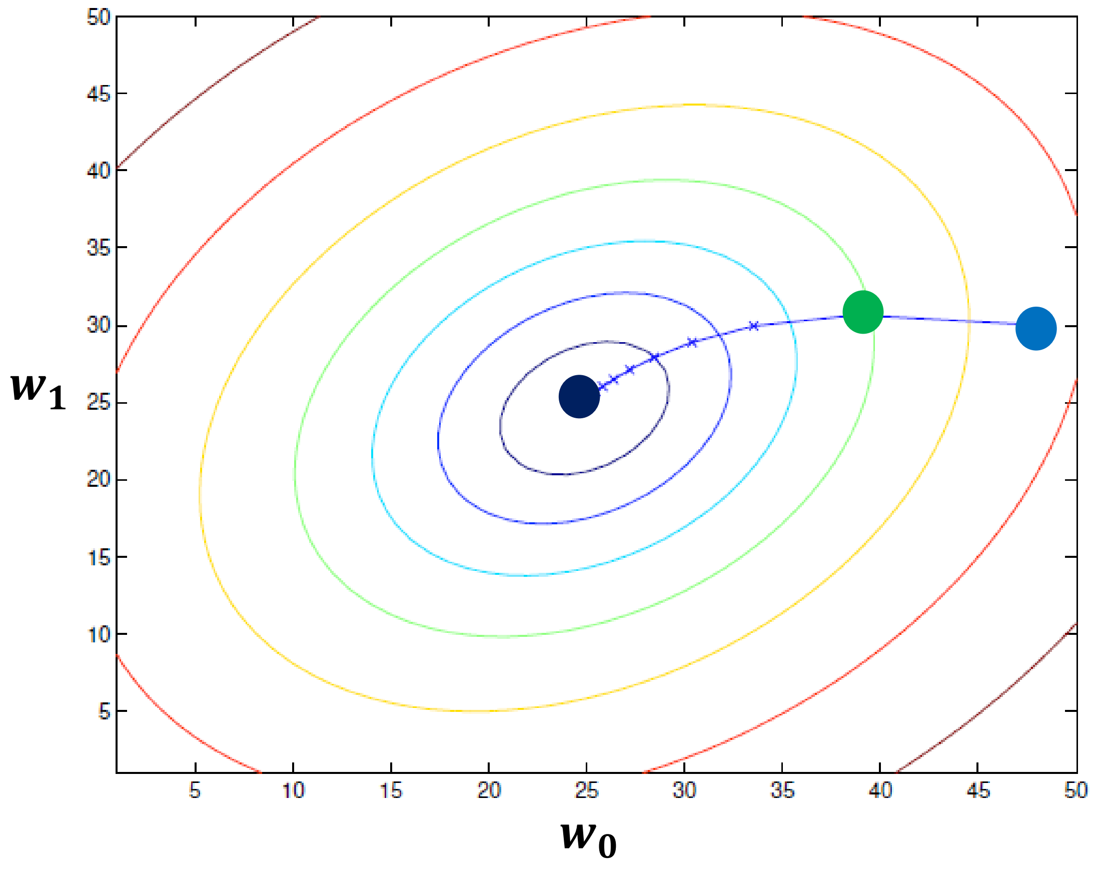


Contour plots





Contour plots



MAKİNE ÖĞRENMESİ

Tek Değişkenli Lineer Regresyon

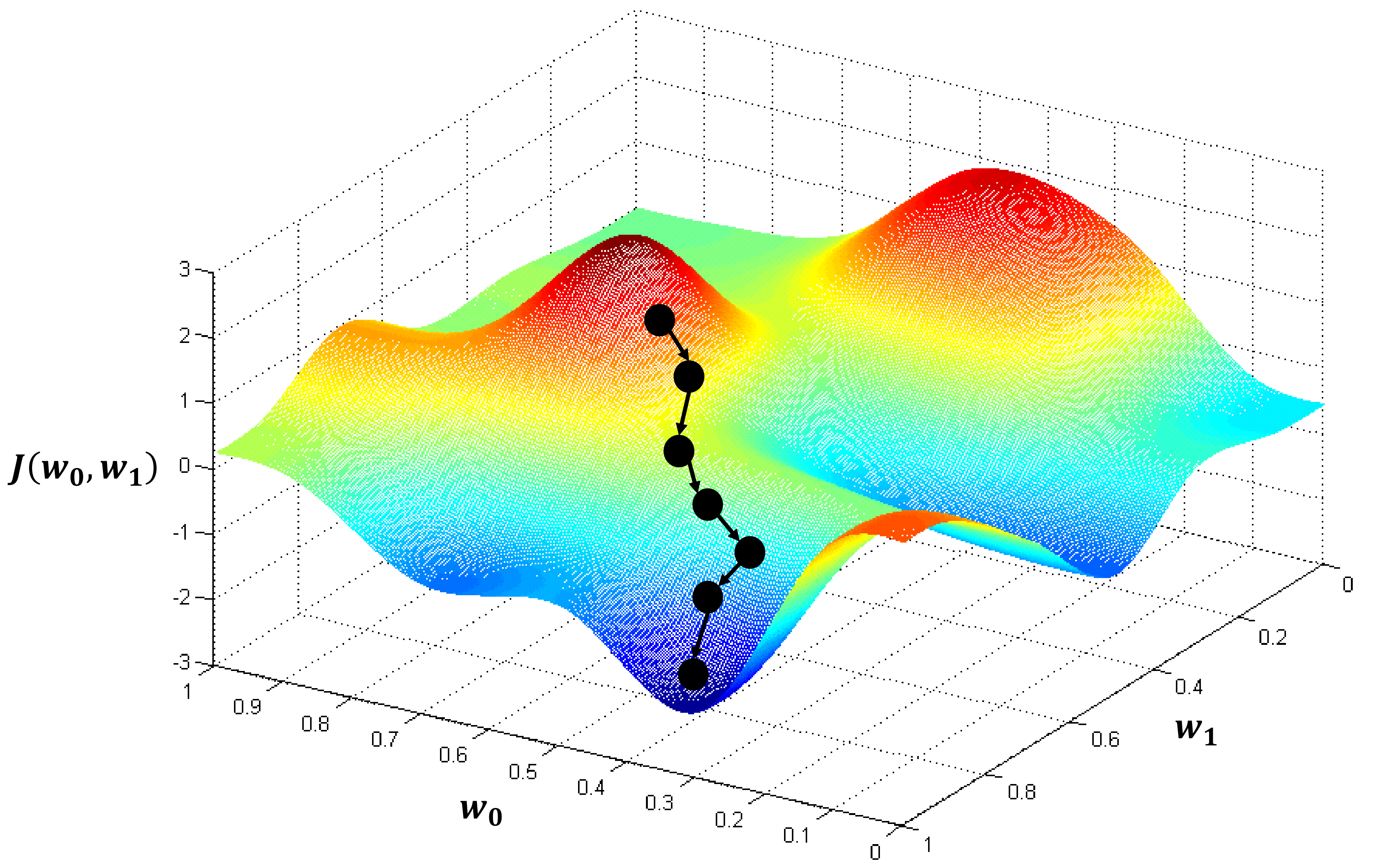
3. Gradyenik Alçalma
(Gradient Descent)

Elimizde bir $J(w_0, w_1)$ fonksiyonu var.

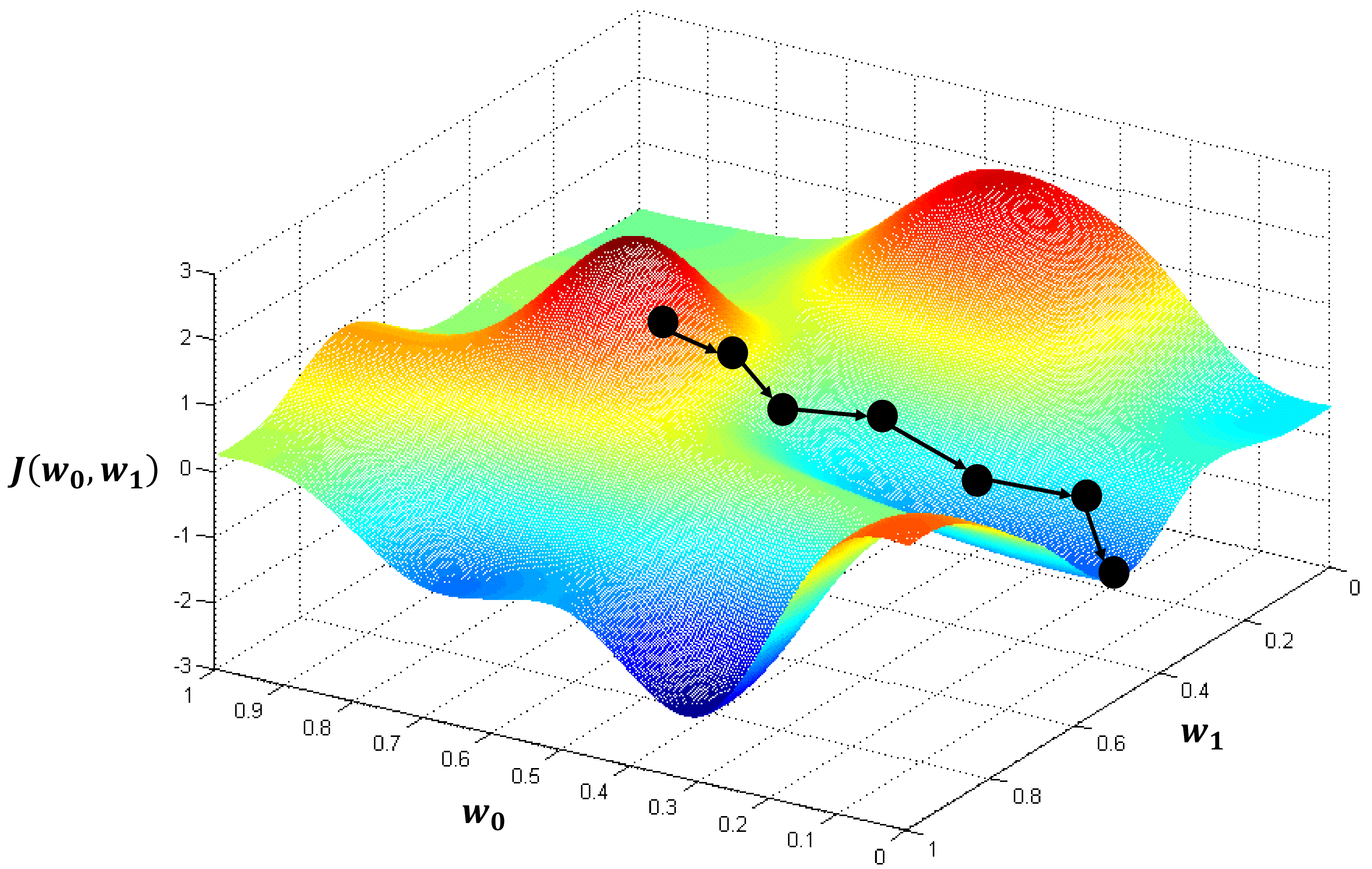
Amacımız bu fonksiyonu minimize etmek.

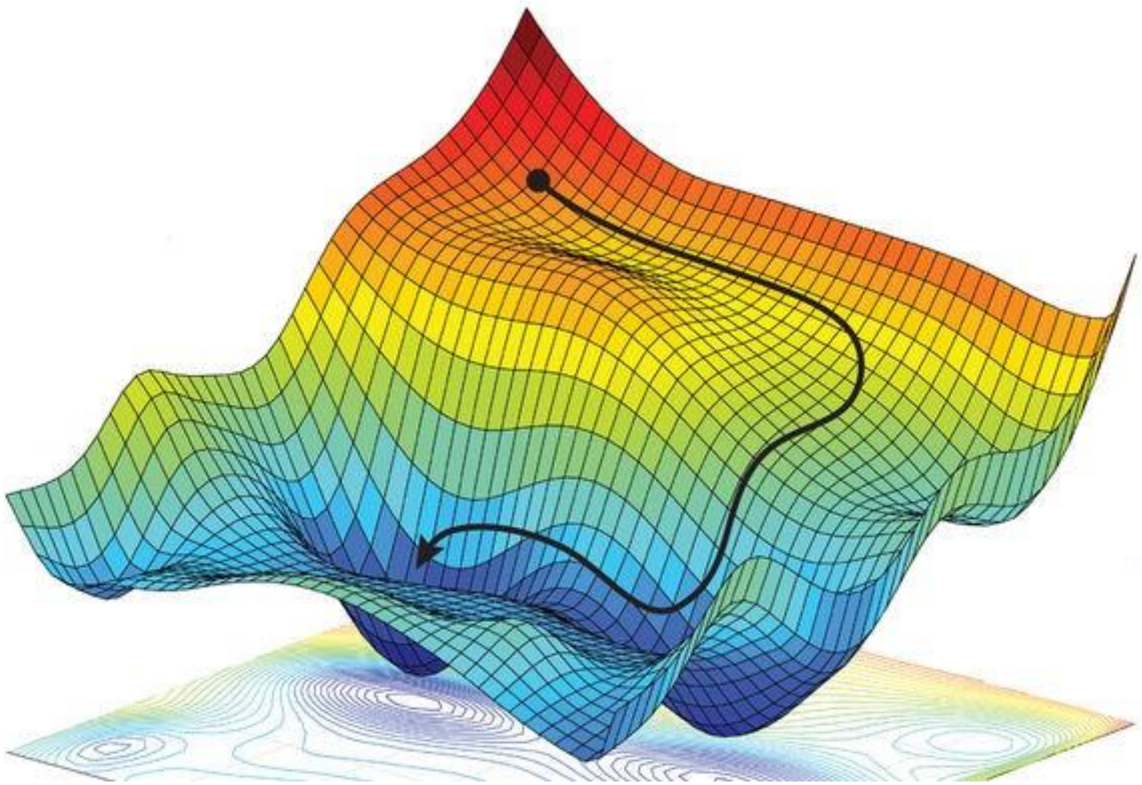
w_0, w_1 bileşenleri ile başlıyalım. (0 ile 1 arası ranstgele değerler atayalım.)

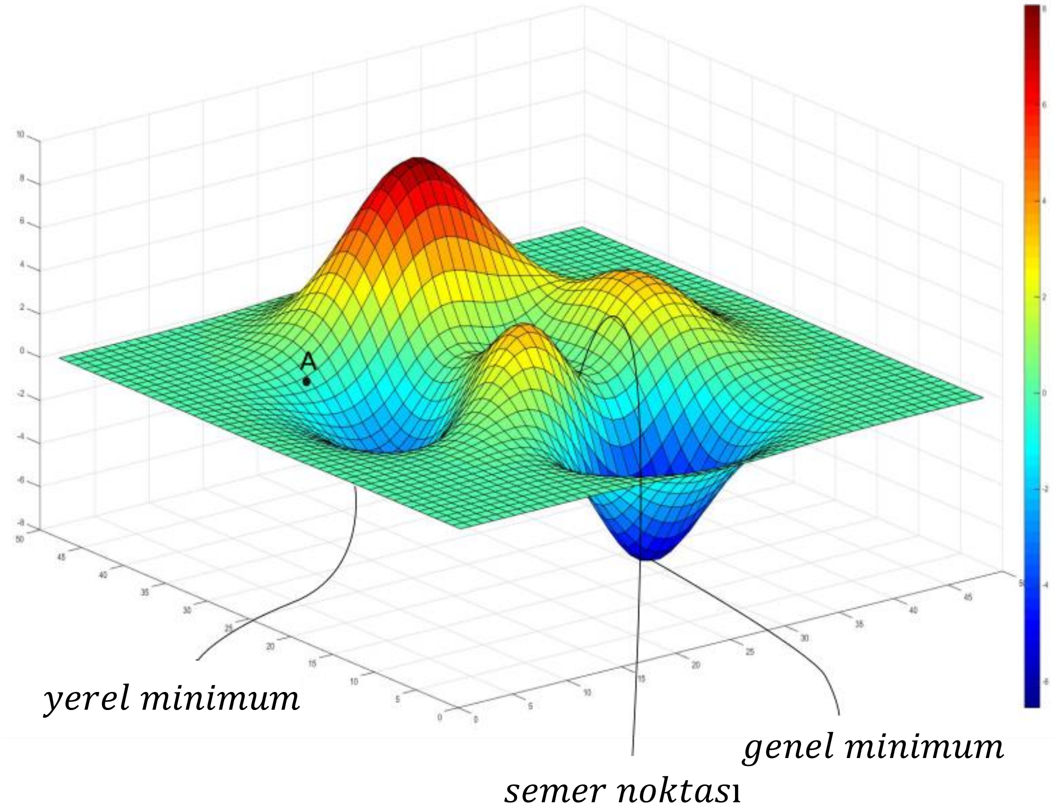
$J(w_0, w_1)$ fonksiyonunu azaltana kadar(minimum'u bulana kadar) bu bileşenleri değiştirelim.



$J(w_0, w_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
w_0
0
0.2
0.4
0.6
0.8
1
w_1


J(w0, w1)
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
w0
0
0.2
0.4
0.6
0.8
1
w1




A
yerel minimum
semer noktası
genel minimum

Gradyenik Alçalma(Gradient Descent) algoritması

$$İterasyon\ \{$$

$$w_j := w_j - \alpha \frac{\partial J(w_j)}{\partial w_j} \quad (j = 0\ ve\ j = 1\ için)$$

$Eş\ zamanlı\ güncelleme\ yapılmalı\,!$

$$\}$$

$Öğrenme\ oranı(Learning\ rate)$

$$temp0 := w_0 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_0}$$

$$temp1 := w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

$$w_0 := temp0$$

$$w_1 := temp1$$

DOĞRU !

$$temp0 := w_0 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_0}$$

$$w_0 := temp0$$

$$temp1 := w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

$$w_1 := temp1$$

YANLIŞ !

Gradyenik Alçalma(Gradient Descent) algoritması

$$\text{İterasyon}\ \{$$

$$w_j := w_j - \alpha \frac{\partial J(w_j)}{\partial w_j}, \qquad (j = 0\ ve\ j = 1\ için)$$

$$\}$$

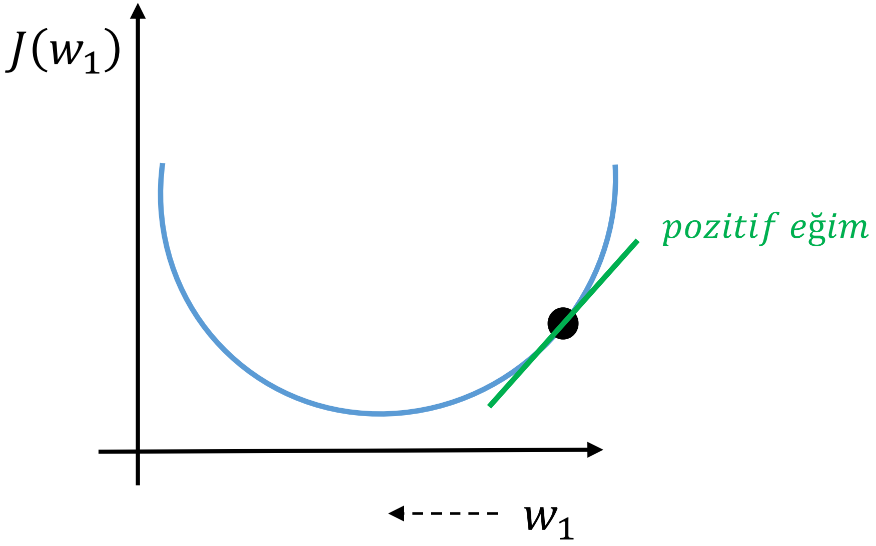
$$\text{İterasyon}\ \{$$

$$w_j := w_j - \alpha \nabla J(w_j), \qquad (j = 0\ ve\ j = 1\ için)$$

$$\}$$

Eş zamanlı güncelleme yapılmalı !

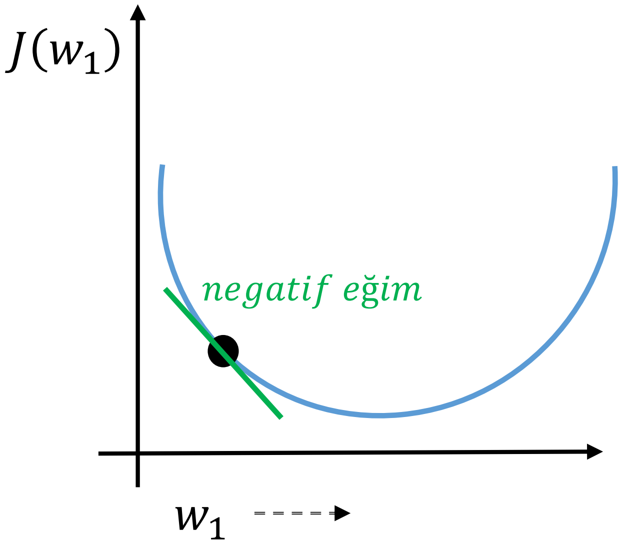
$$w_j := w_j - \alpha \frac{1}{2m} \nabla \left(\sum_{i=1}^{m} \left(y^{(i)} - h_w\left(x^{(i)}\right) \right)^2 \right)$$



$$w_1 := w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

$$\frac{\partial J(w_0, w_1)}{\partial w_1} > 0$$

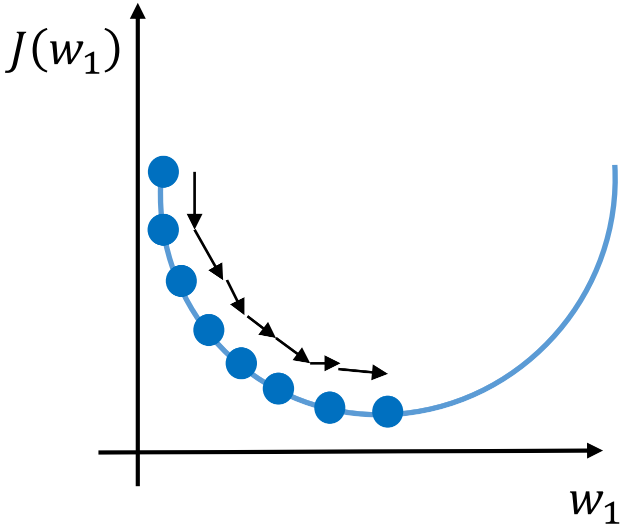
$$w_1 := w_1 - \alpha(pozitif\ sayı)$$



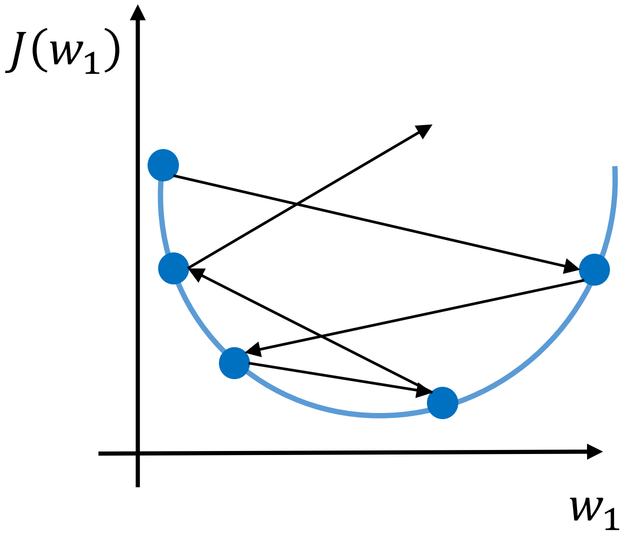
$$w_1 := w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

$$\frac{\partial J(w_0, w_1)}{\partial w_1} < 0$$

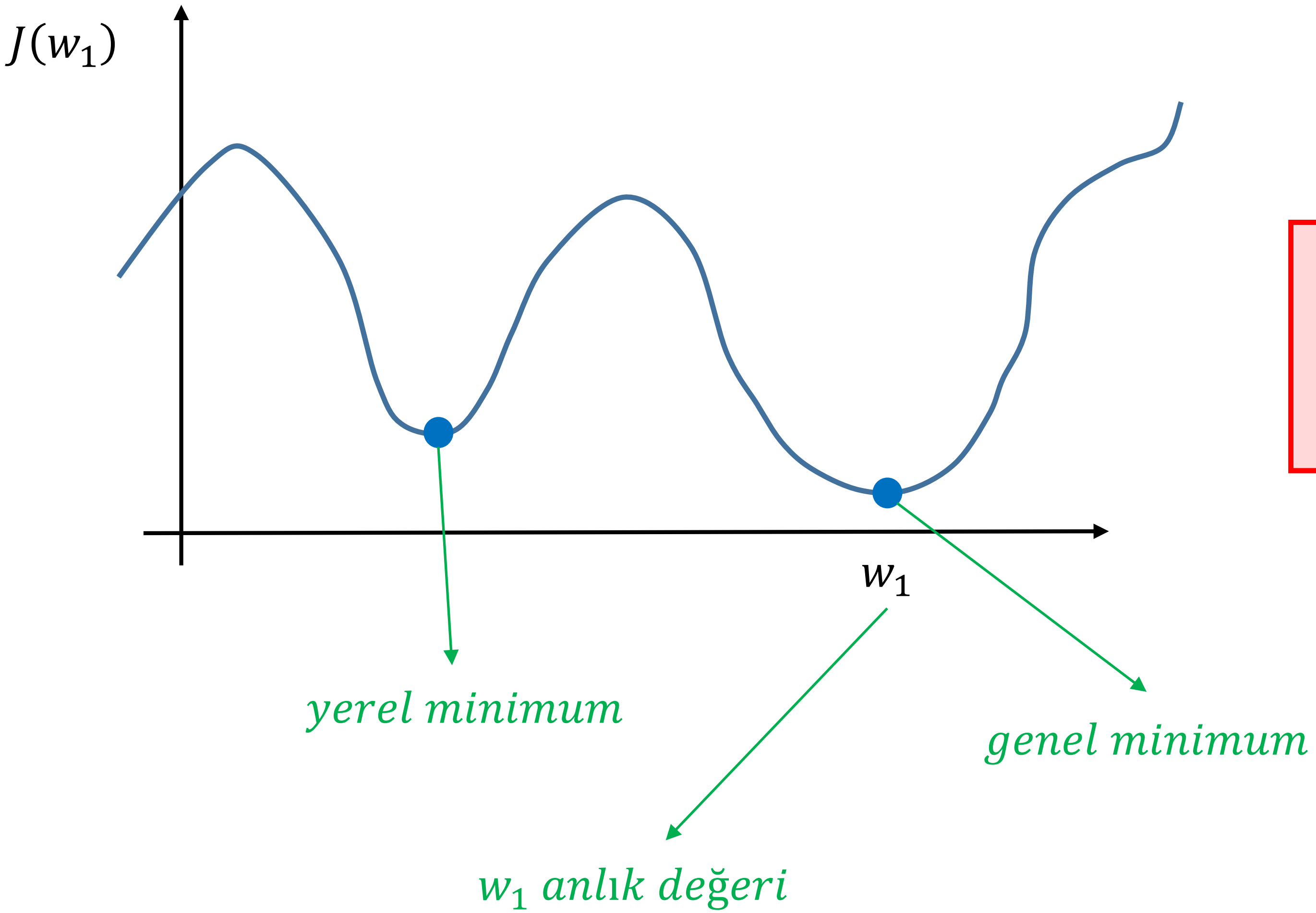
$$w_1 := w_1 - \alpha(negatif\ sayı)$$



$eğer\ \alpha\ çok\ küçük\ olursa, gradyenik\ alçalma\ çok\ yavaş\ olabilir.$



$eğer\ \alpha\ çok\ büyük\ olursa, gradyenik\ alçalma\ minimum\ değer\ bölgesini\ aşırabilir.$



$$w_1 := w_1 - \alpha \frac{\partial J(w_0, w_1)}{\partial w_1}$$

Lineer Regresyon Modeli

$$h_w(x) = w_0 + w_1 x$$

$$J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

$$\frac{\partial}{\partial w_j}J(w_0, w_1) = \frac{\partial}{\partial w_j}\frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

$$\frac{\partial}{\partial w_j}J(w_0, w_1) = \frac{\partial}{\partial w_j}\frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2$$

$$j = 0: \ \frac{\partial}{\partial w_0}J(w_0, w_1) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h_w\left(x^{(i)}\right)\right)$$

$$j = 1: \ \frac{\partial}{\partial w_1}J(w_0, w_1) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - h_w\left(x^{(i)}\right)\right)x^{(i)}$$

$$\text{İterasyon}\ \{$$

$$w_0 \coloneqq w_0 + \alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)$$

$$w_1 \coloneqq w_1 + \alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)x^{(i)}$$

$$\}$$

Lineer Regresyon Modeli

$$J(w_0, w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - h_w(x^{(i)})\right)^2$$

$$[1]_{m\times 1}$$

$$w_0 \coloneqq w_0 + \alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)x_0^{(i)}$$

$$w_1 \coloneqq w_1 + \alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)x^{(i)}$$