Doğrusal Sınıflandırıcılar ve Perceptron

Perceptron Algoritması

- 1957 yılında Frank Rosenblatt tarafından Cornell Havacılık Laboratuvarında icat edilmiştir. (Cornell Aeronautical Laboratory)
- Doğrusal sınıflandırma algoritmasıdır.

Frank Rosenblatt

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

Düzlem(Plane) için $\mathrm{d} = 2$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \ \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

Eğitim için örnek sayısı

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da -1$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \left\{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\right\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^+ = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^- = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

Uzayı (düzlemi) ikiye bölmesidir.

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \left\{(x^{(i)}, y^{(i)}), i = 1, \dots, n\right\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

$x^{+} = \{x \in \mathcal{R}^d : h(x) = 1\}$

- Sınıflandırıcı : $h : \mathcal{R}^d \to \{-1, 1\}, \quad h(x) = 1 \; ya \; da -1$

$x^{-} = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası (Train Error) : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![h(x^{(i)}) \neq y^{(i)}]\!]$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^+ = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^- = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası (Train Error) : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![\underbrace{h(x^{(i)}) \neq y^{(i)}}]\!]$

1, eşit değil ise
0, eşitse

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \ldots, n\right\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^+ = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^- = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n}\left[\!\left[\underbrace{h(x^{(i)}) \neq y^{(i)}}\right]\!\right]$

'n' burada hatamızı hesaplamak için kullancağımız örnek sayıdır.

1, eşit değil ise
0, eşitse

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^+ = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^- = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![\underbrace{h(x^{(i)}) \neq y^{(i)}}_{\substack{1,\ \text{eşit değil ise} \\ 0,\ \text{eşitse}}}]\!]$

'n' burada hatamızı hesaplamak için kullancağımız örnek sayıdır.

- Test hatası : $\mathcal{E}(h)$

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \dots, n\right\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \rightarrow \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^{+} = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^{-} = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![\underbrace{h(x^{(i)}) \neq y^{(i)}}_{}]\!]$

1, eşit değil ise
0, eşitse

'n' burada hatamızı hesaplamak için kullancağımız örnek sayıdır.

- Test hatası : $\mathcal{E}(h)$

Test hatasının teori kısmı, sınıflandırıcının eğitim setinde ne kadar iyi bir sınıflandırma yaptığını tespit etmesidir.

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \to \{-1, 1\}, \quad h(x) = 1 \; ya \; da -1$

$x^{+} = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^{-} = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![\underbrace{h(x^{(i)}) \neq y^{(i)}}]\!]$

‘n’ burada hatamızı hesaplamak için kullancağımız örnek sayıdır.

1, eşit değil ise
0, eşitse

- Test hatası : $\mathcal{E}(h)$

Test hatasının teori kısmı, sınıflandırıcının eğitim setinde ne kadar iyi bir sınıflandırma yaptığını tespit etmesidir.

Ayrıca test seti için bu tespit yapılmalıdır. (Generalization)

Perceptron : Temel Kavramlar

- Özellik vektörleri(Feature vectors) ve etiketler(labels) : $x \in \mathcal{R}^d, y \in \{-1, 1\}$

x sınıflandırma için tahmin yapmamızı sağlayan bir vektördür.

Çıkışlar, etiketler yada hedefler

- Eğitim seti(Training set) : $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$

Denetimli öğrenmenin görevi, bize bir girdi(input) ve buna karşılık gelen istediğimiz bir çıktı(output) vermesidir.

- Sınıflandırıcı : $h : \mathcal{R}^d \to \{-1, 1\}, \quad h(x) = 1 \; ya \; da - 1$

$x^+ = \{x \in \mathcal{R}^d : h(x) = 1\}$

$x^- = \{x \in \mathcal{R}^d : h(x) = -1\}$

Sınıflandırıcının görevi, uyguladığımız bir girdiyi belli bir noktaya haritalandırmasıdır.

- Eğitim hatası : $\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![\underbrace{h(x^{(i)}) \neq y^{(i)}}]\!]$

1, eşit değil ise
0, eşitse

'n' burada hatamızı hesaplamak için kullancağımız örnek sayıdır.

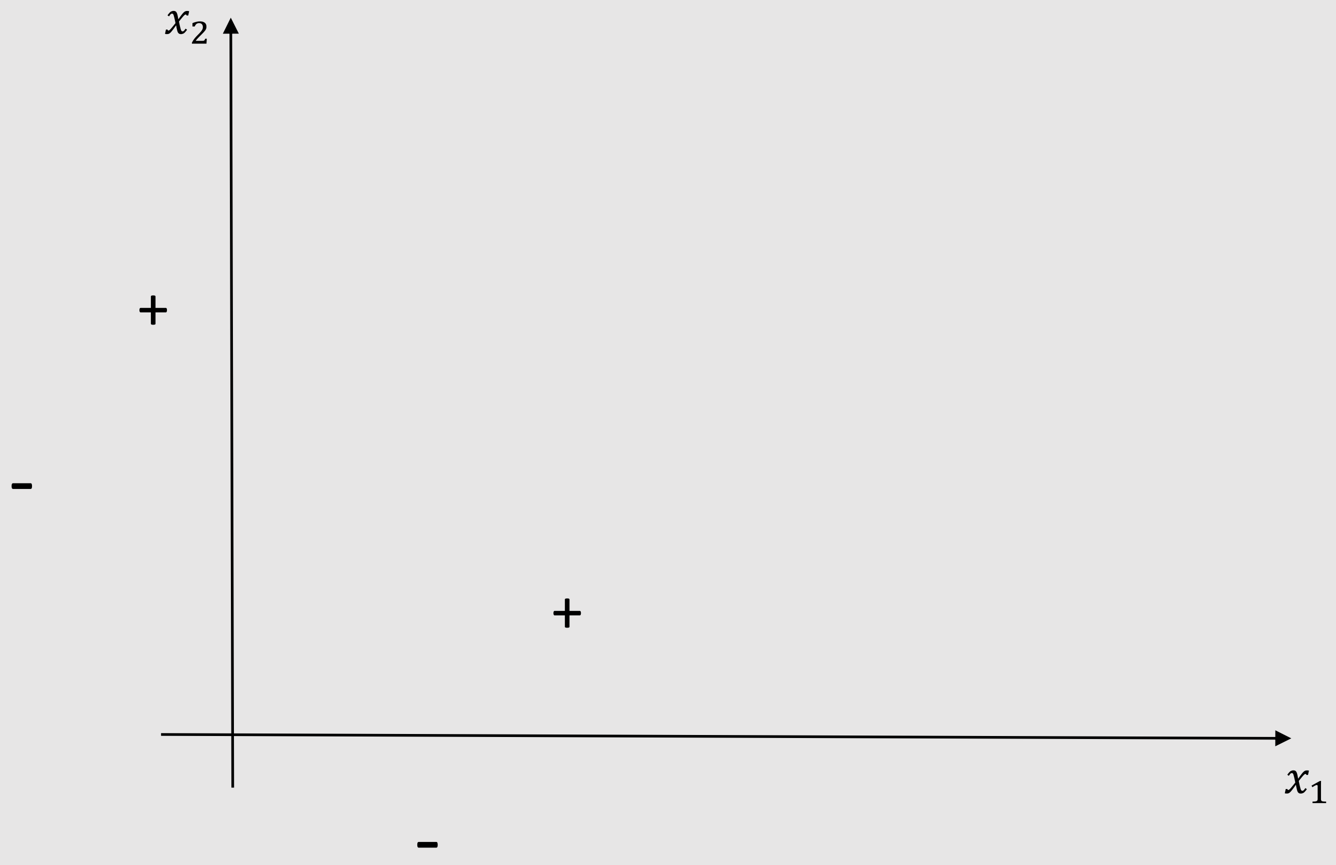
- Test hatası : $\mathcal{E}(h)$

Test hatasının teori kısmı, sınıflandırıcının eğitim setinde ne kadar iyi bir sınıflandırma yaptığını tespit etmesidir.

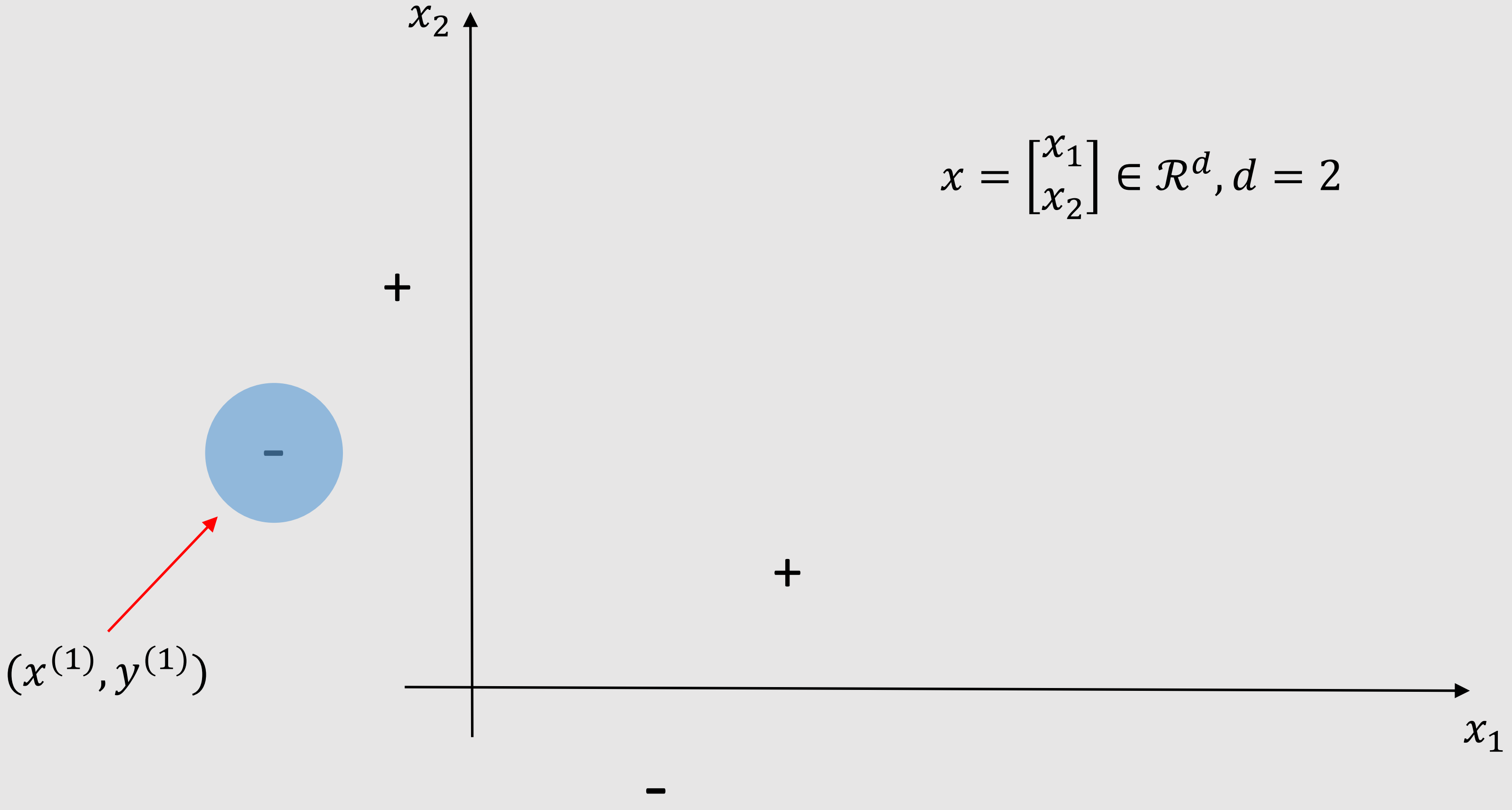
Ayrıca test seti için bu tespit yapılmalıdır. (Generalization)

- Sınıflandırıcı kümesi (set of classifiers) : $h \in \mathcal{H}$

Eğitim Seti (Train Set)

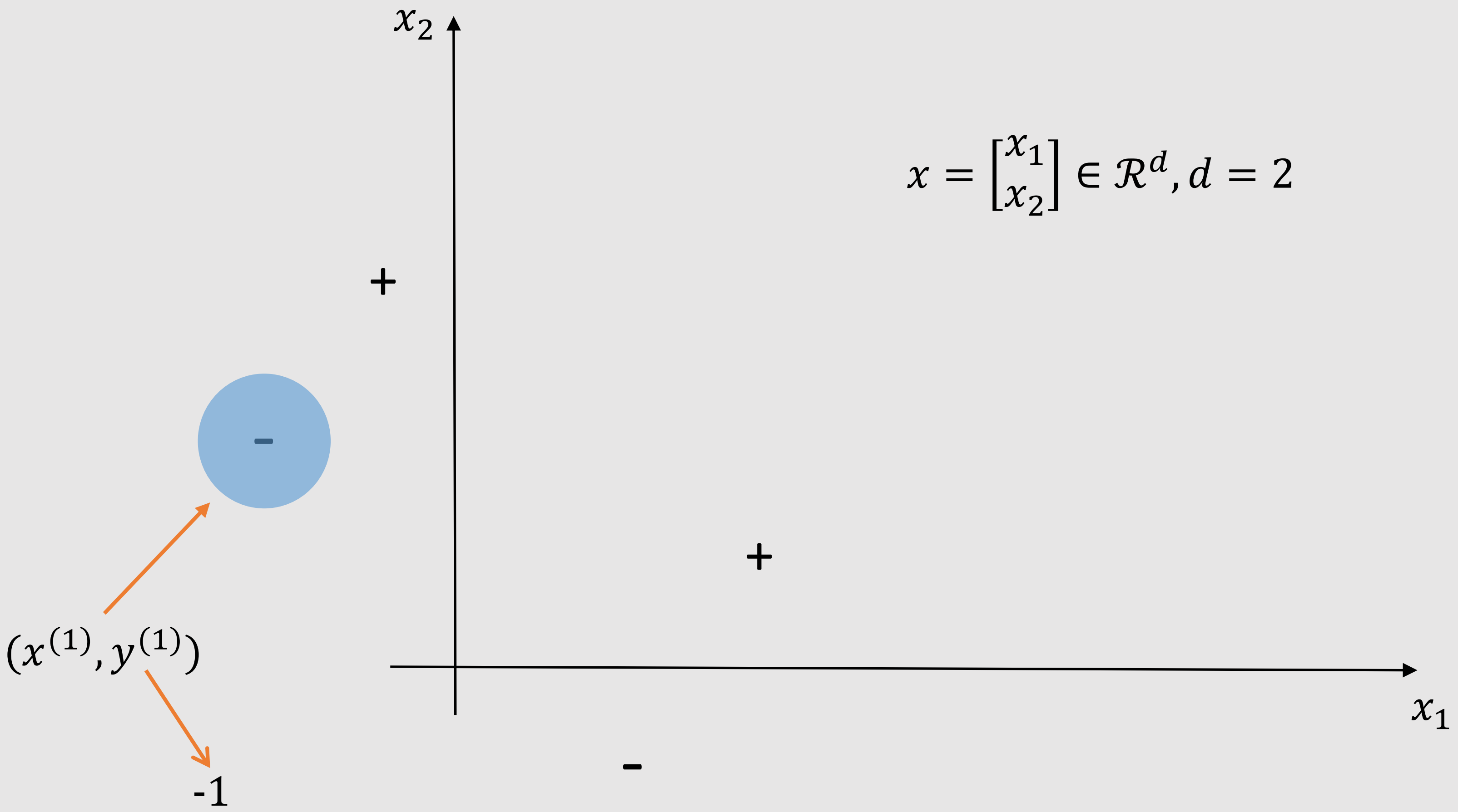


Eğitim Seti (Train Set)



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

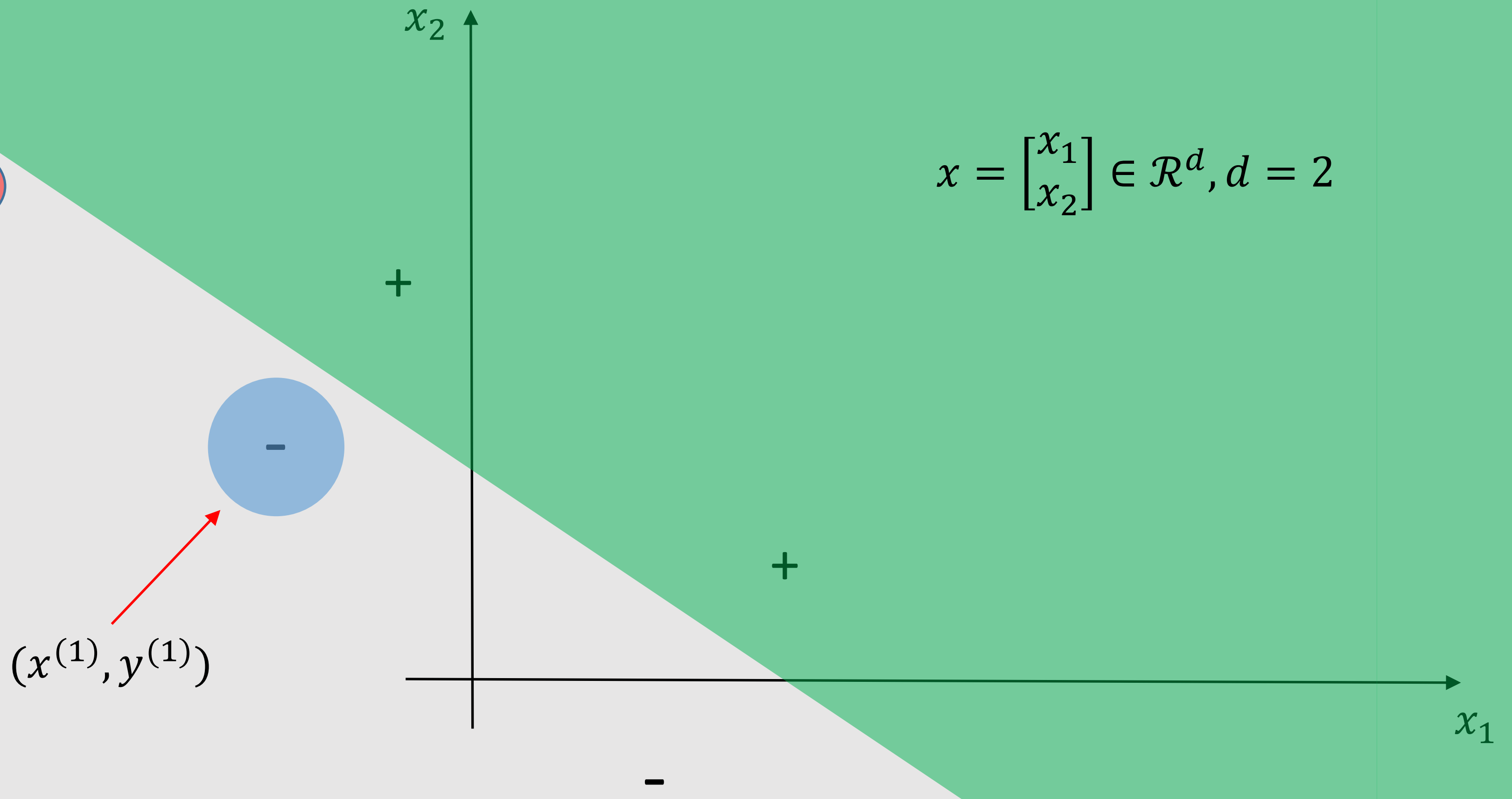
Eğitim Seti (Train Set)



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

Sınıflandırıcı (Classifier)

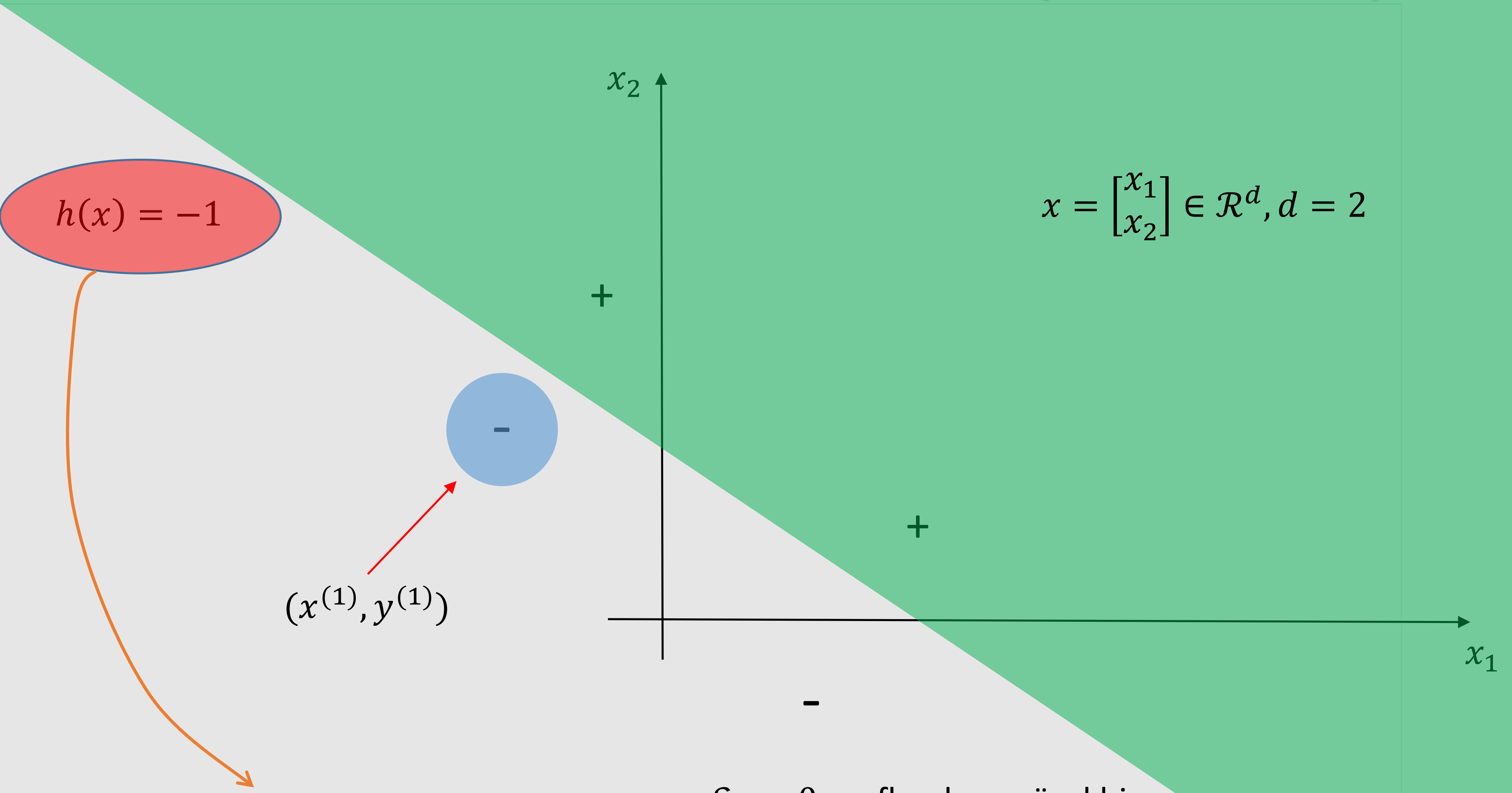
$h(x) = -1$



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

h(x) burada uzayı (düzlemi) ikiye bölen bir sınıflandırıcıdır.
Bu bölgedeki değerler -1’ eşittir.

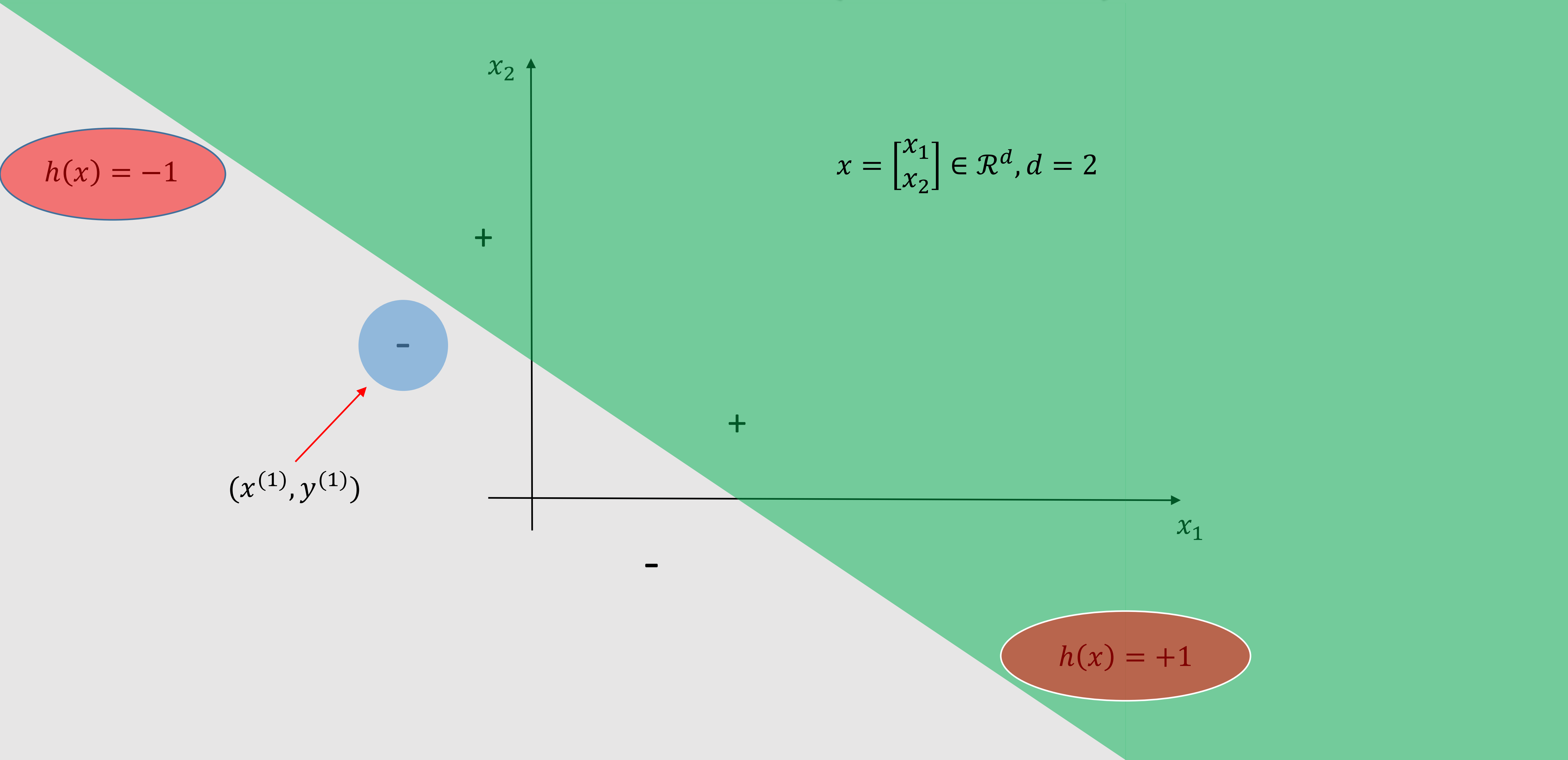
Sınıflandırıcı (Classifier)



h(x) burada uzayı (düzlemi) ikiye bölen bir sınıflandırıcıdır.
Bu bölgedeki değerler -1’ eşittir.

$\mathcal{E}_n = 0$ sınıflandırıcı güzel bir şekilde sınıflandırdı. Dolayısı ile hatamız yok.

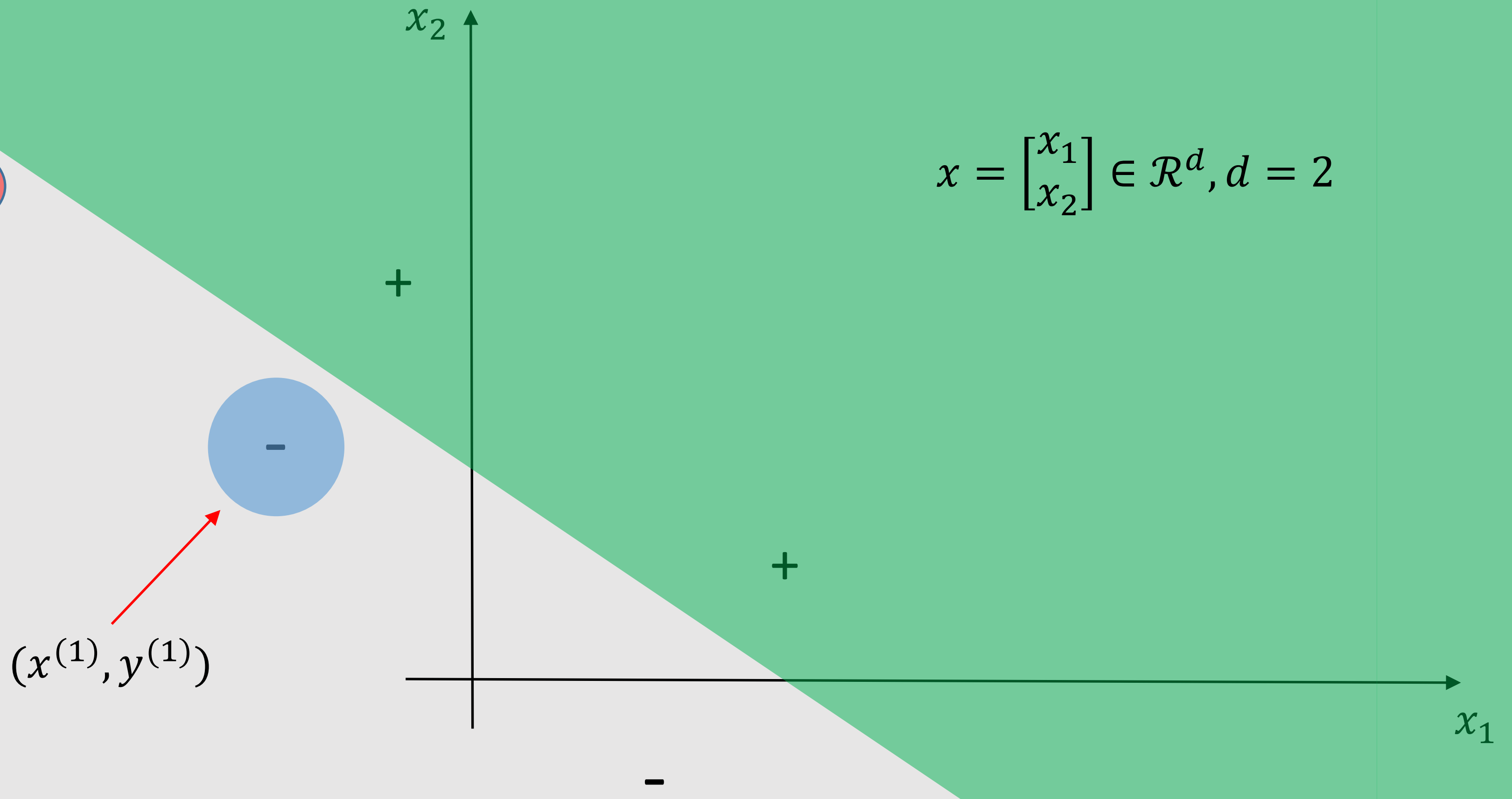
Sınıflandırıcı (Classifier)



Sınıflandırıcı (Classifier)

$h(x) = -1$

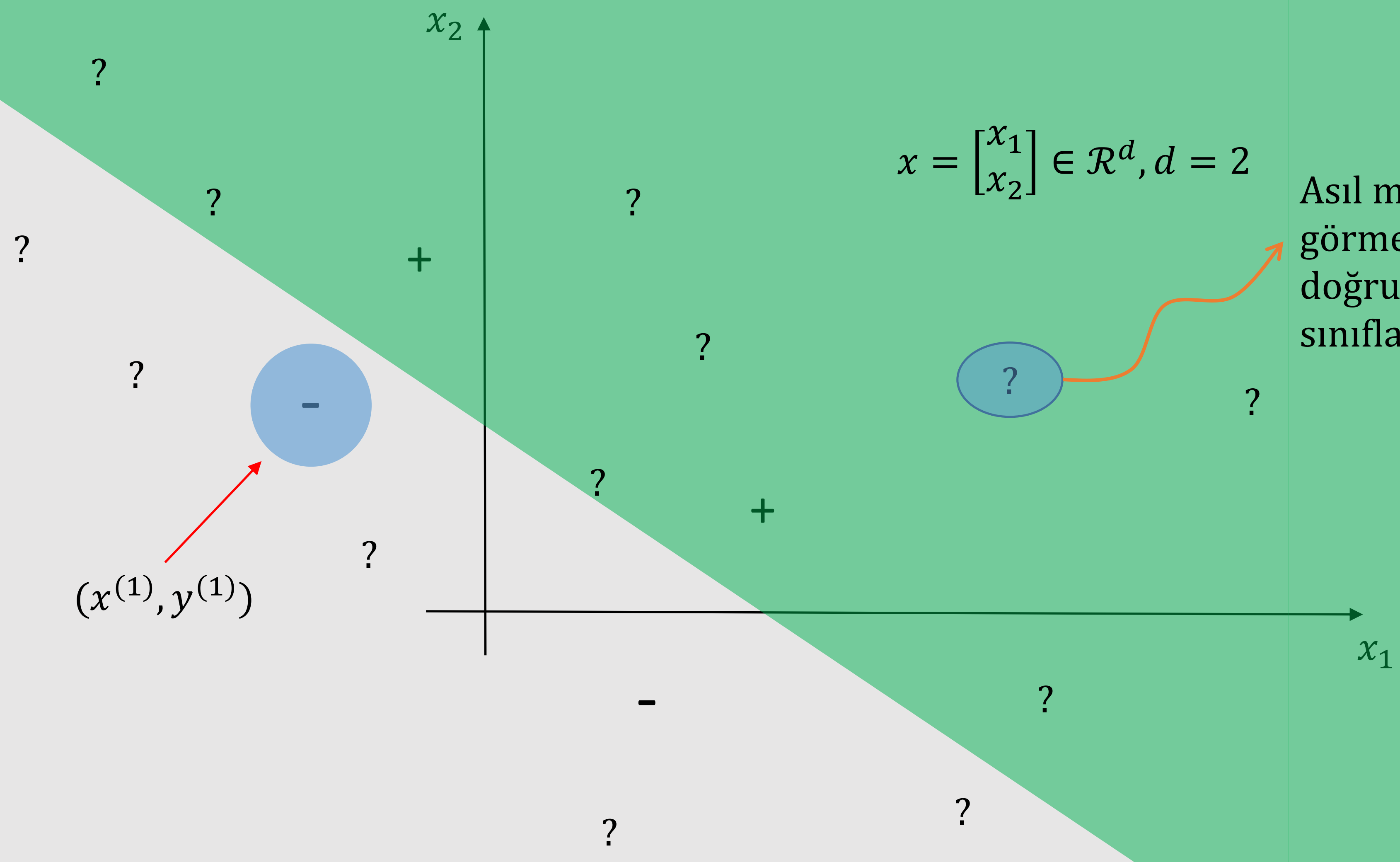
$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$



$\mathcal{E}_n = 0$: Bu sınıflandırıcı iyi bir iş yapıyor.
Koordinat sistemindeki eğitim noktalarını çok güzel ayırmakta!

$h(x) = +1$

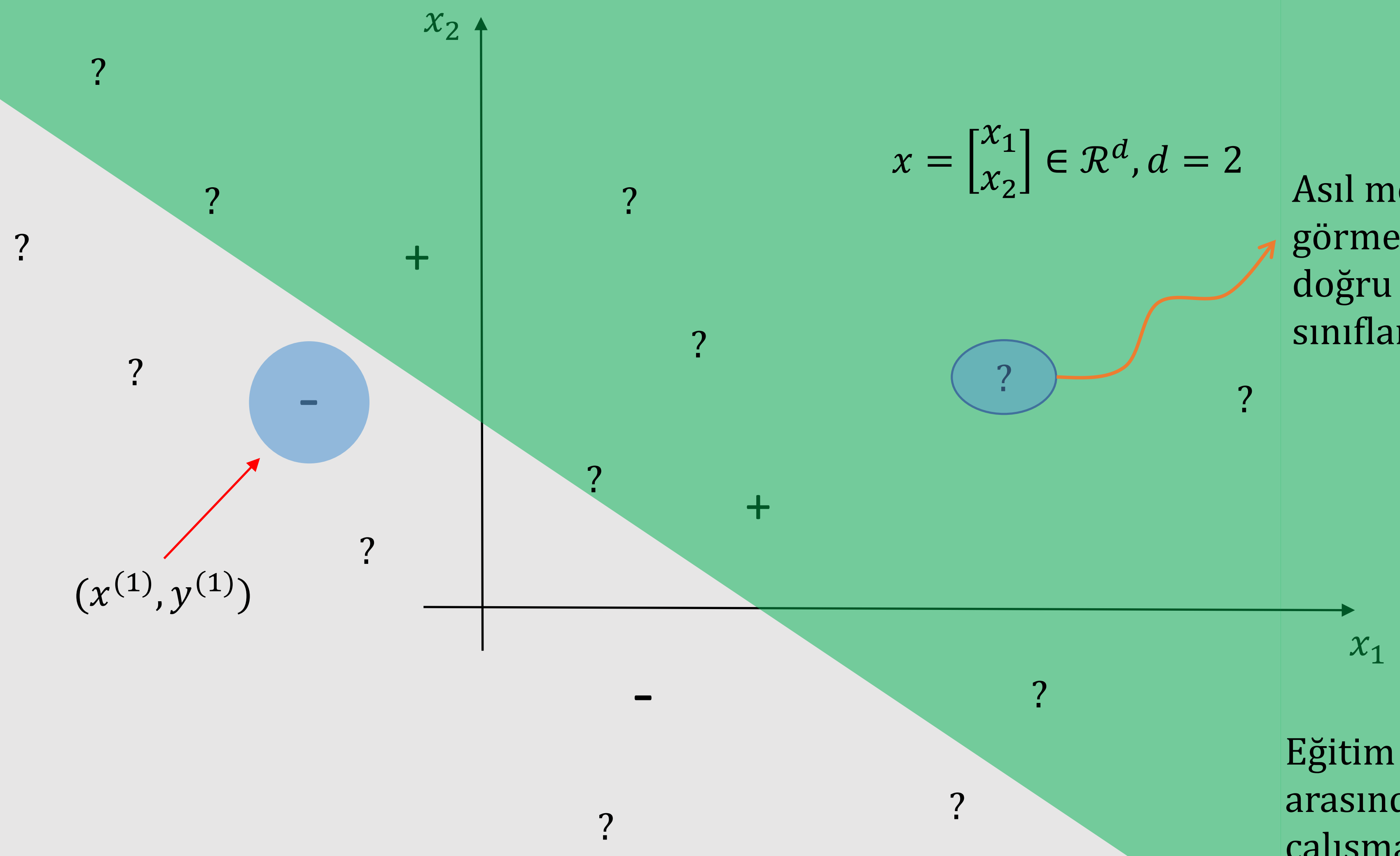
Sınıflandırıcı (Classifier)



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

Asıl mesele ise henüz görmediğimiz test örneklerini doğru bir şekilde sınıflandırabilmek.

Sınıflandırıcı (Classifier)

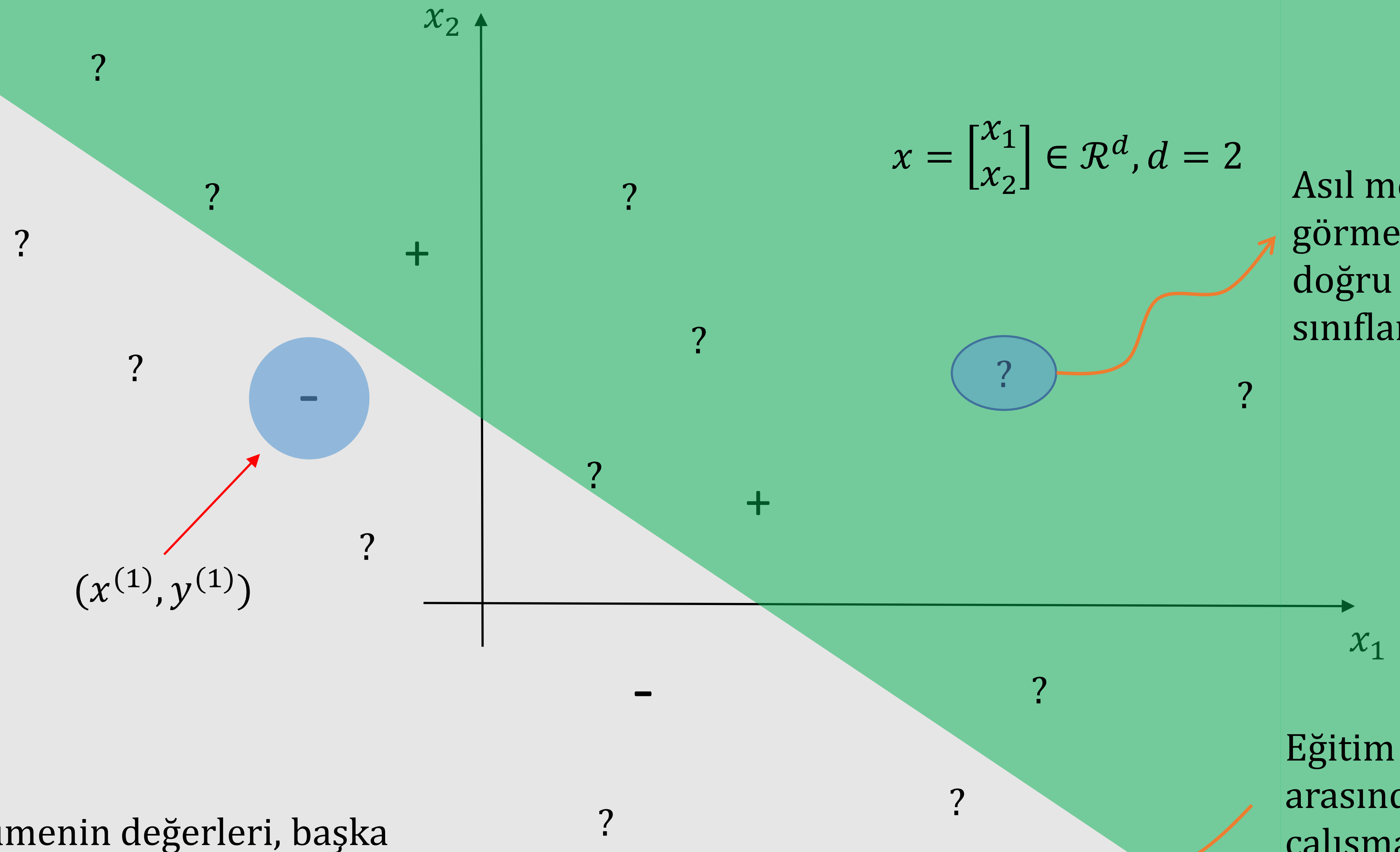


$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

Asıl mesele ise henüz görmediğimiz test örneklerini doğru bir şekilde sınıflandırabilmek.

Eğitim örneği ve test örneği arasında bir bağlantı kurmaya çalışmak.

Sınıflandırıcı (Classifier)



$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathcal{R}^d, d = 2$$

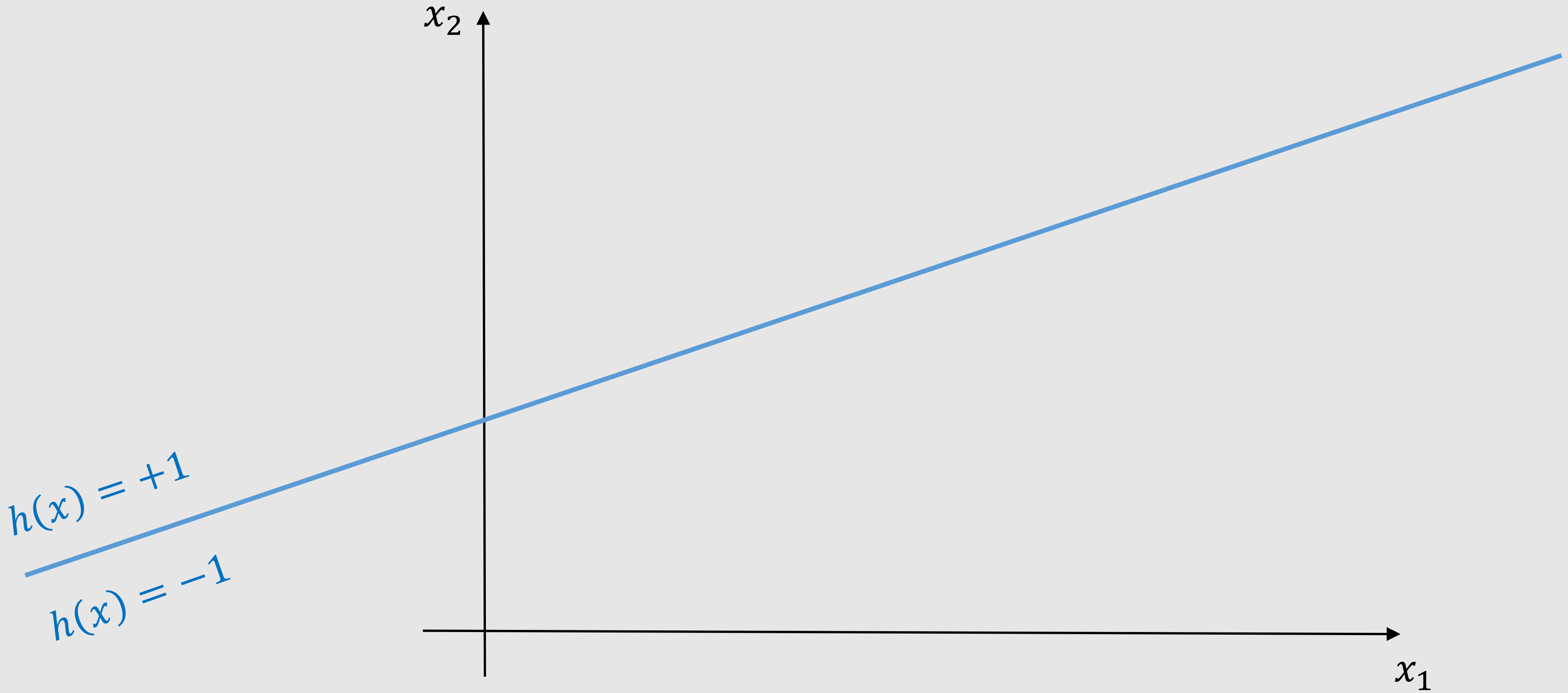
Asıl mesele ise henüz görmediğimiz test örneklerini doğru bir şekilde sınıflandırabilmek.

Eğitim örneği ve test örneği arasında bir bağlantı kurmaya çalışmak.

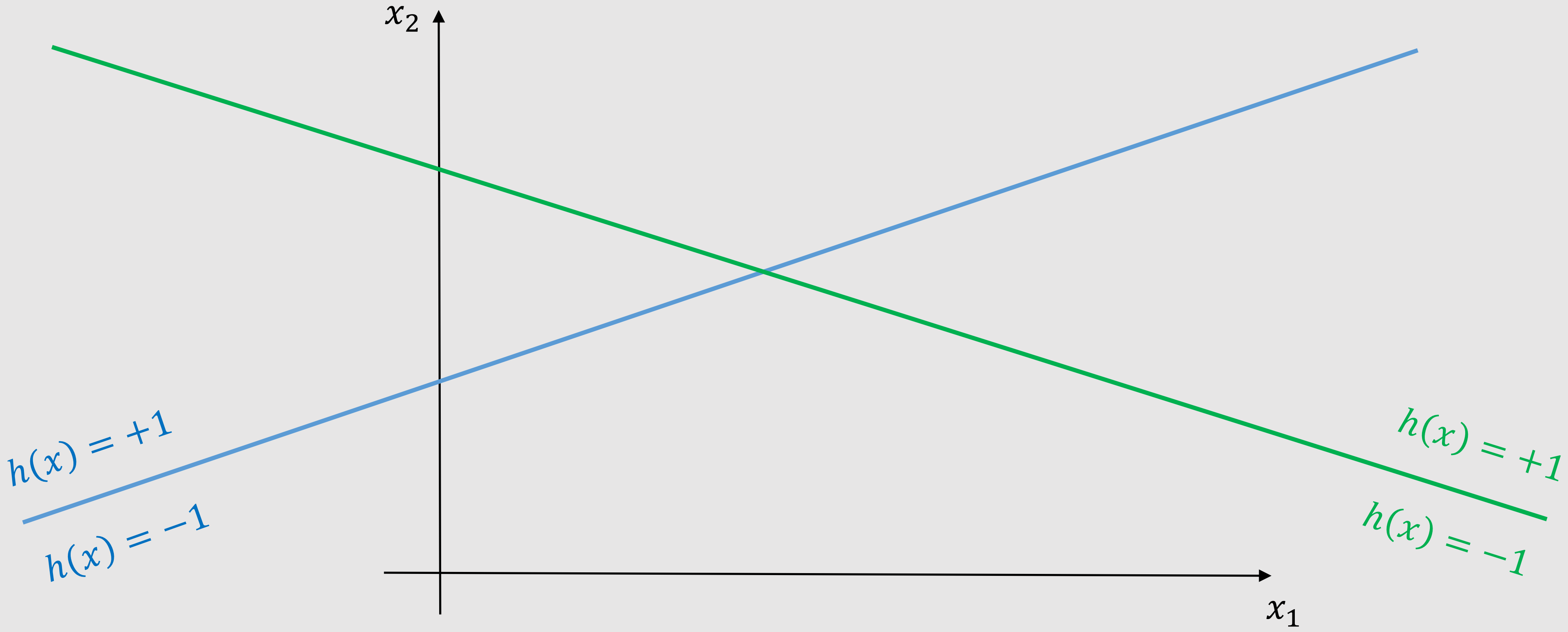
Bu iki kümenin değerleri, başka bir kümenin alt kümesinin rastgele(random) değerleri

Doğrusal Sınıflandırıcı (Linear Classifier)

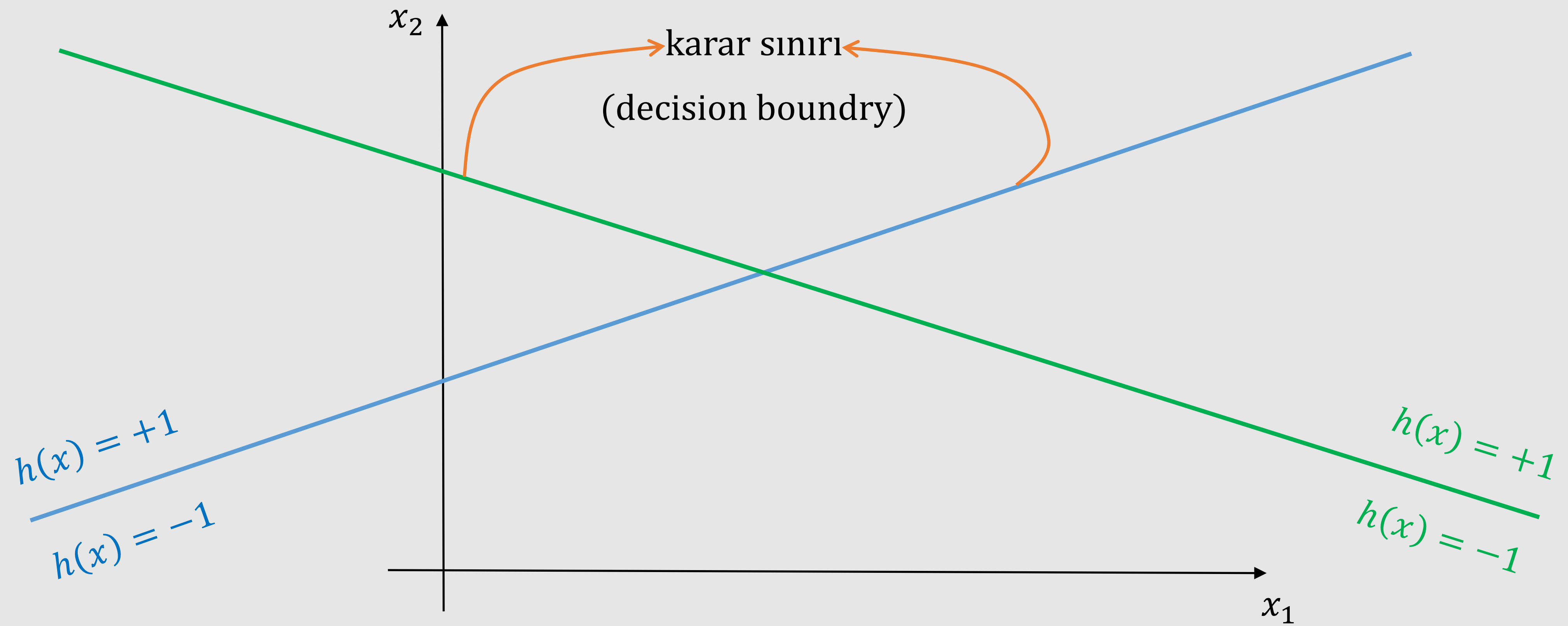
Doğrusal Sınıflandırıcı (Linear Classifier)



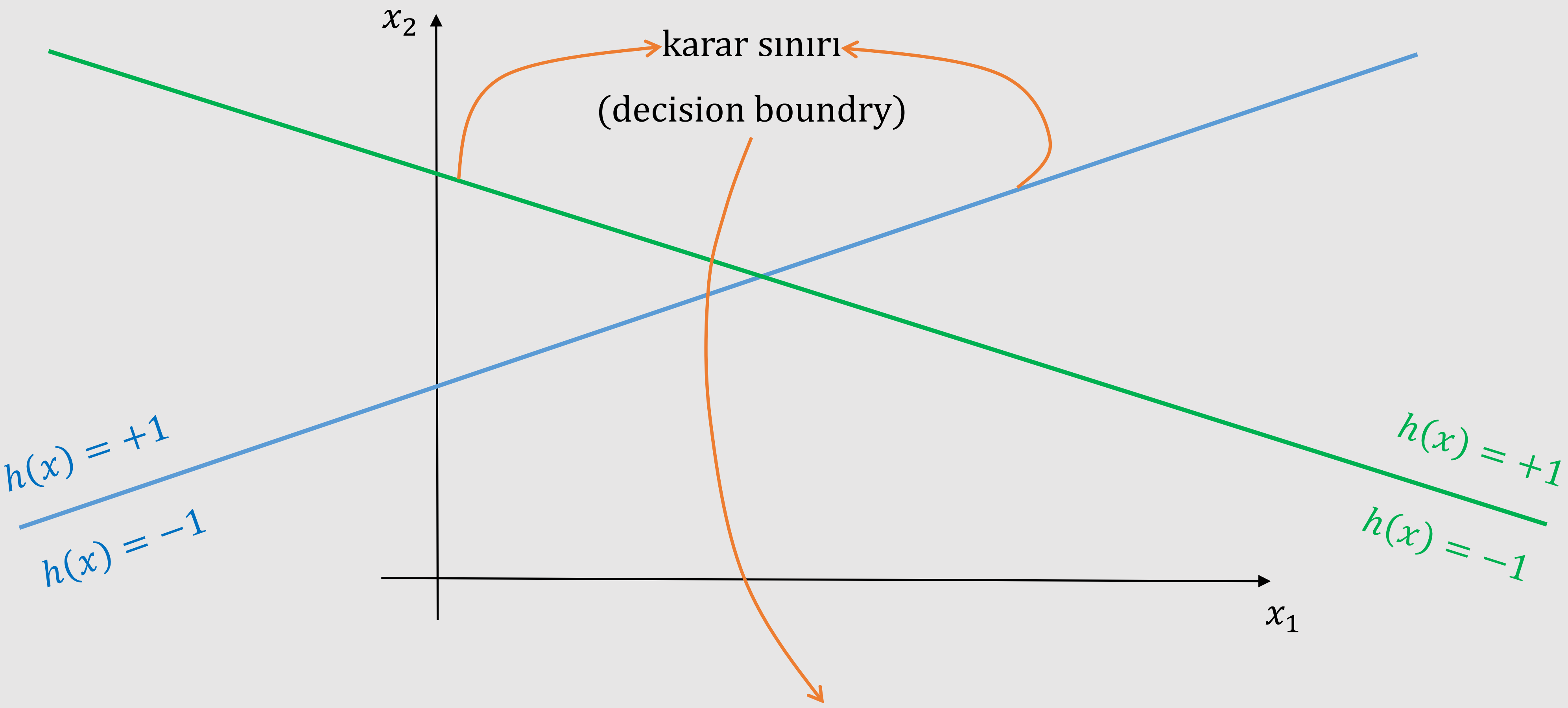
Doğrusal Sınıflandırıcı (Linear Classifier)



Doğrusal Sınıflandırıcı (Linear Classifier)

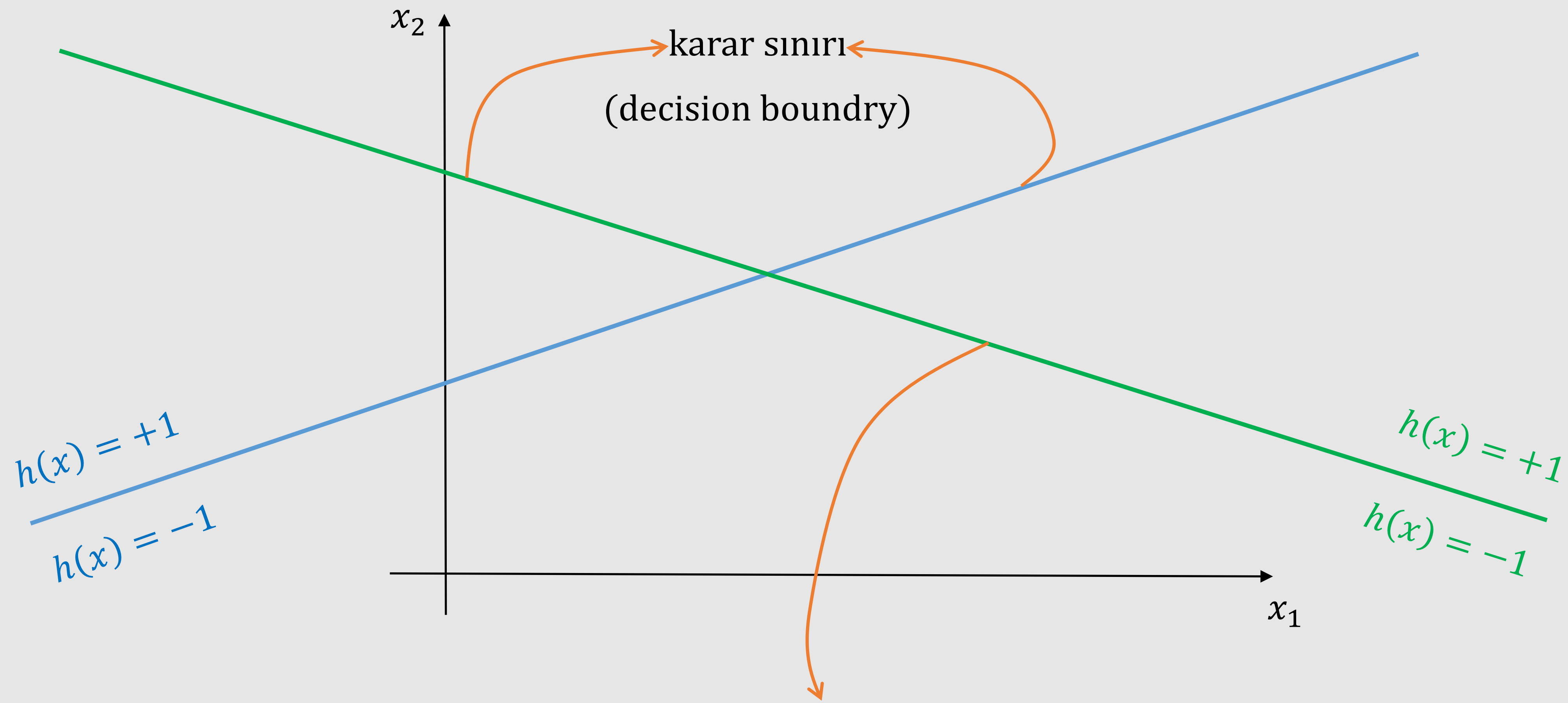


Doğrusal Sınıflandırıcı (Linear Classifier)



2 boyutta, karar sınırımız bir çizgi, x vektörü 1 boyutlu olsaydı karar sınırımız bir nokta olacaktı. 3 boyutta ise bir düzlem...

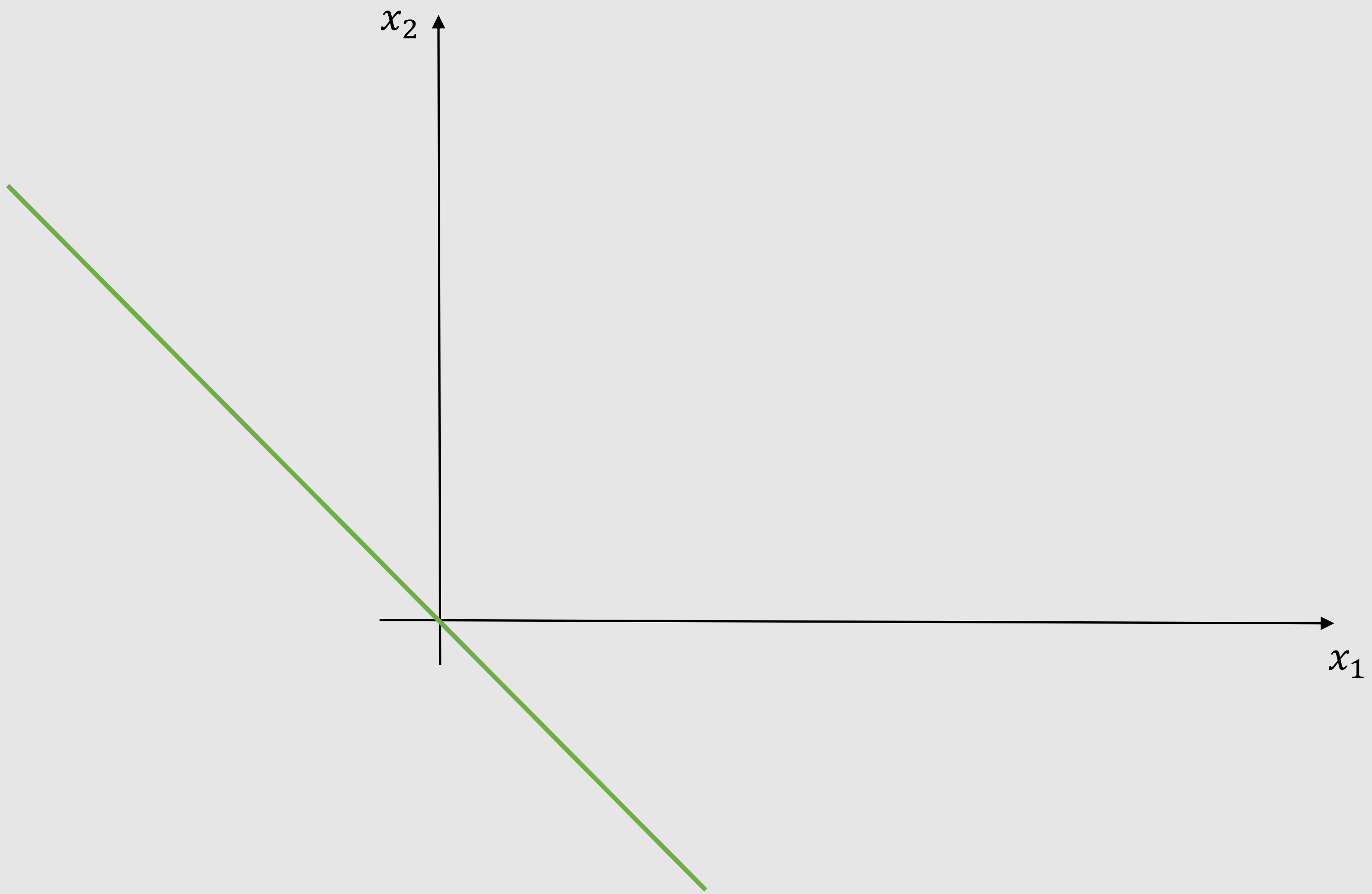
Doğrusal Sınıflandırıcı (Linear Classifier)



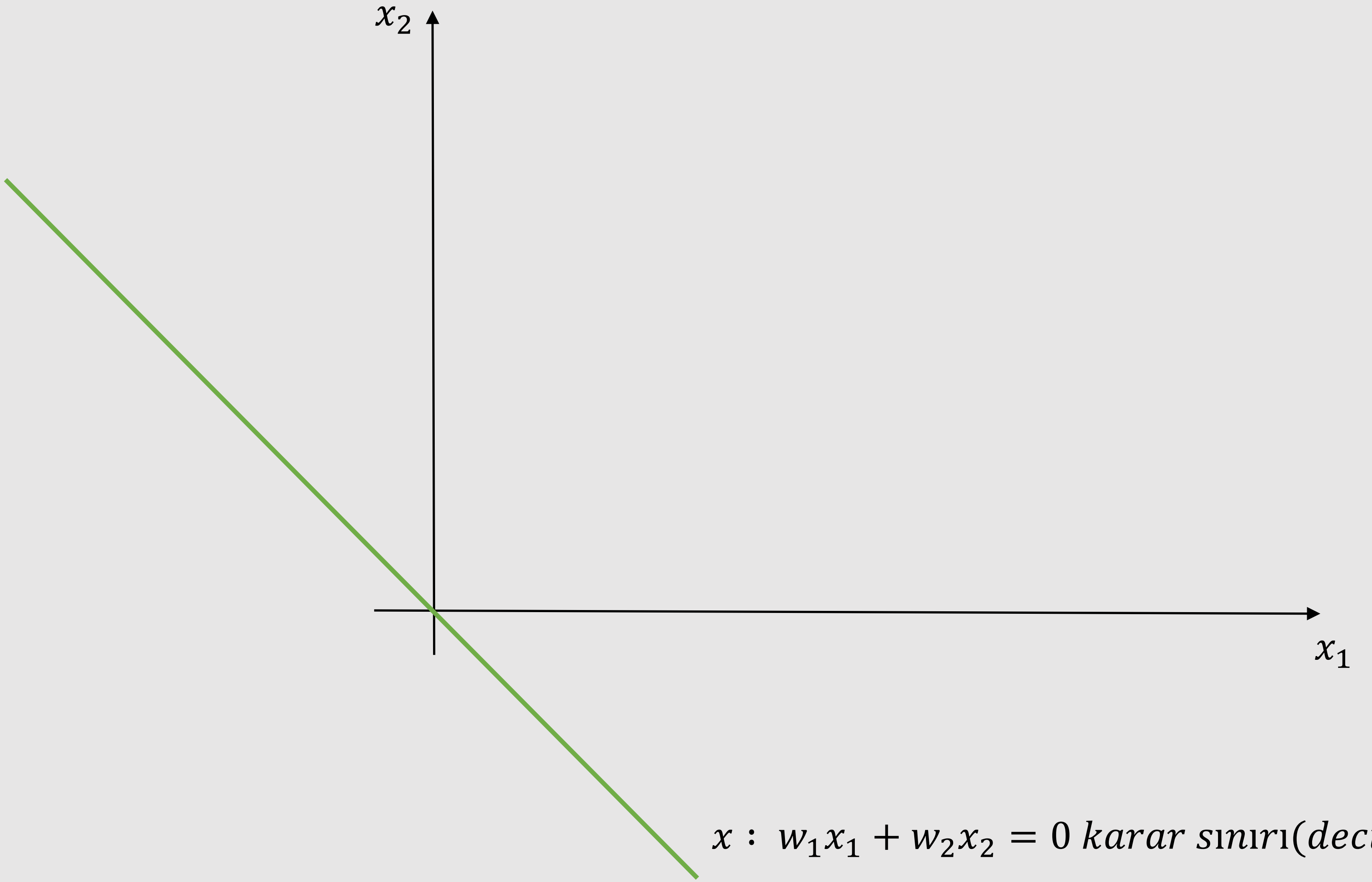
Doğrusal sınıflandırıcı için
karar sınırımızı değiştirebiliriz.

Orjinden Geçen Doğrusal Sınıflandırıcı

Orjinden Geçen Doğrusal Sınıflandırıcı



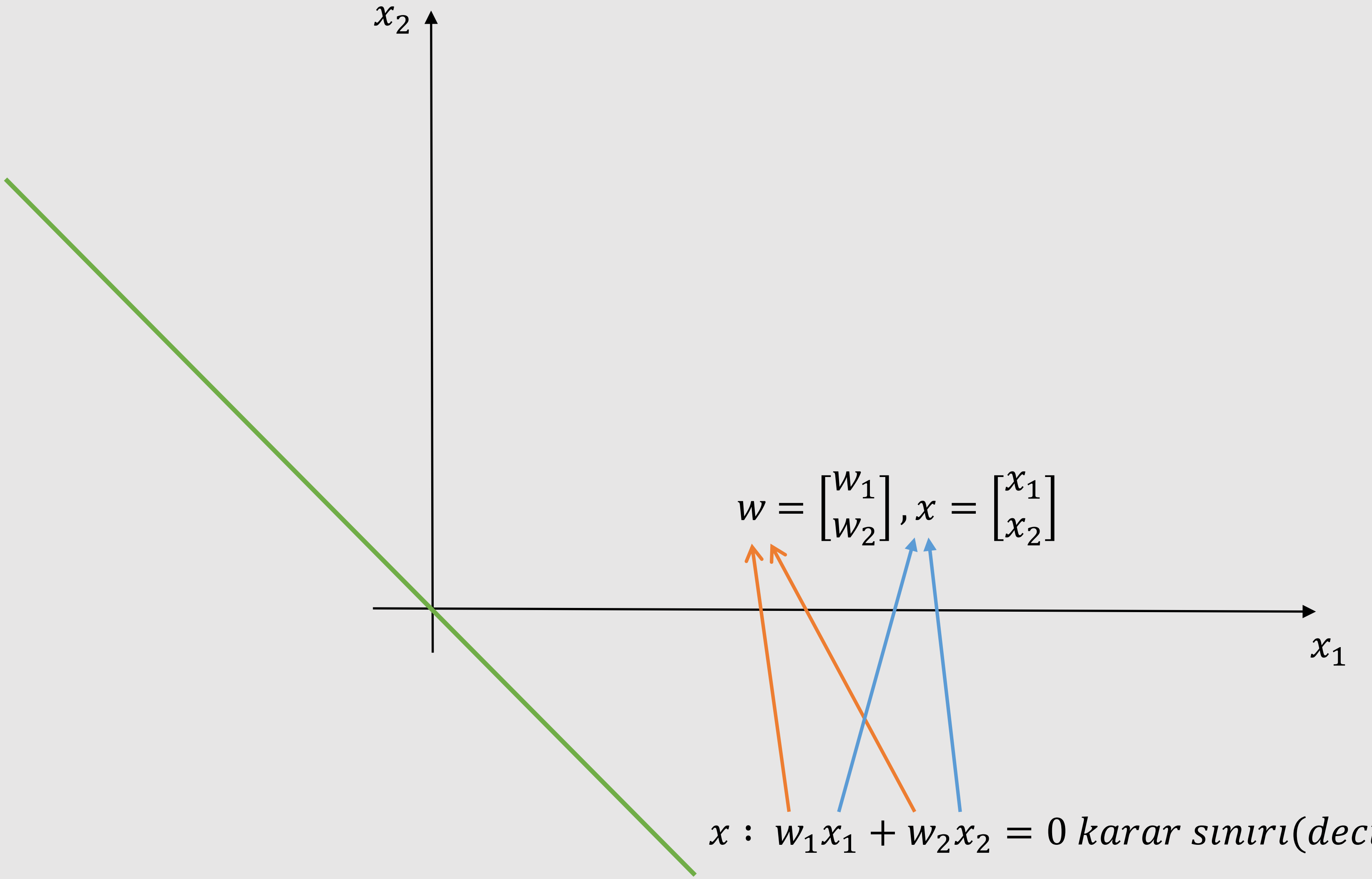
Orjinden Geçen Doğrusal Sınıflandırıcı



$x: \ w_1 x_1 + w_2 x_2 = 0 \ karar \ sınırı(decision \ boundry)$

$x: w \cdot x = 0 \ karar \ sınırı(decision \ boundry)$

Orjinden Geçen Doğrusal Sınıflandırıcı

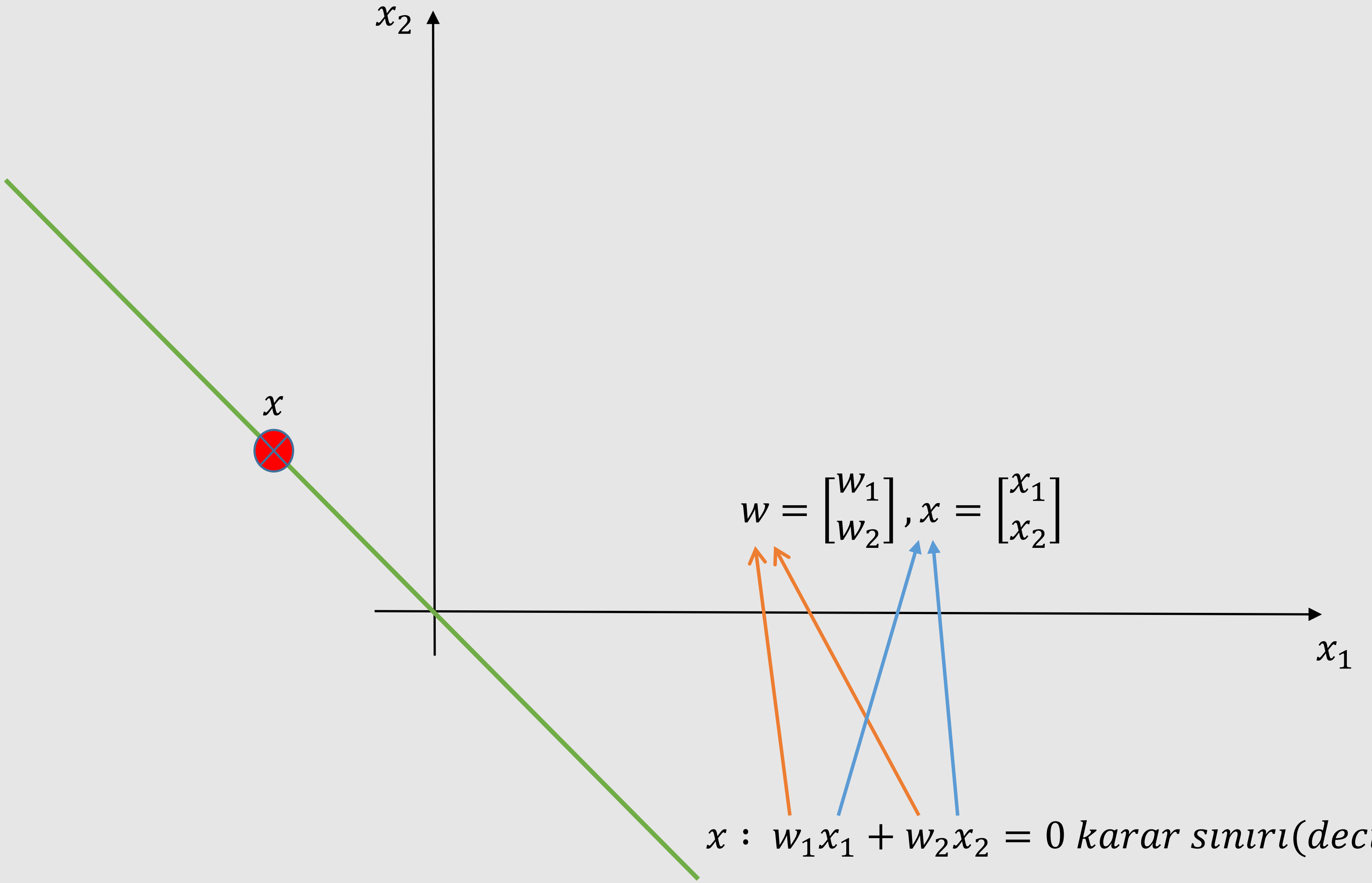


$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$x : \ w_1 x_1 + w_2 x_2 = 0 \ karar \ sınırı(decision \ boundry)$$

$$x : w \cdot x = 0 \ karar \ sınırı(decision \ boundry)$$

Orjinden Geçen Doğrusal Sınıflandırıcı

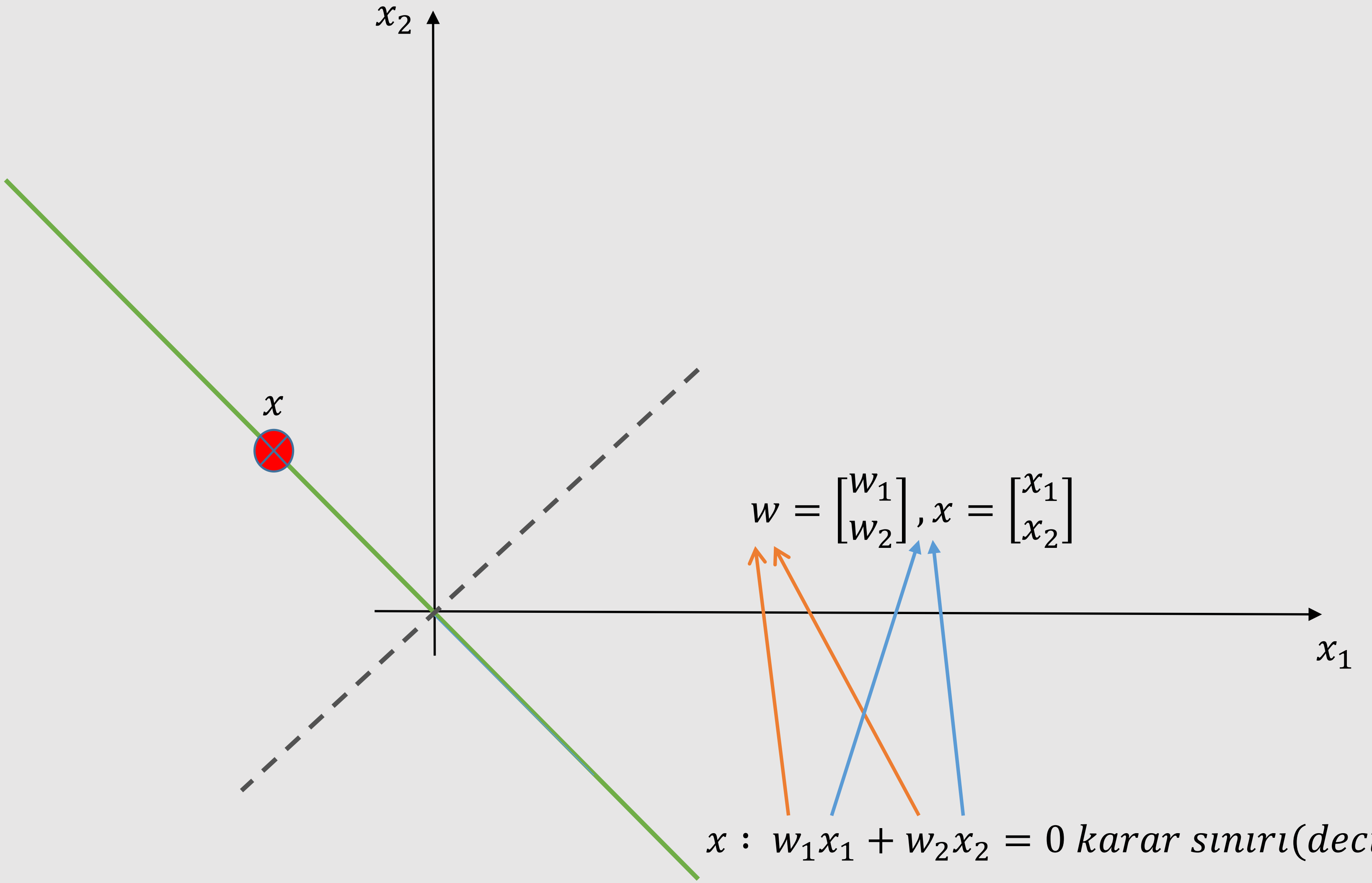


$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$x : \; w_1 x_1 + w_2 x_2 = 0$ *karar sınırı(decision boundry)*

$x : w \cdot x = 0$ *karar sınırı(decision boundry)*

Orjinden Geçen Doğrusal Sınıflandırıcı

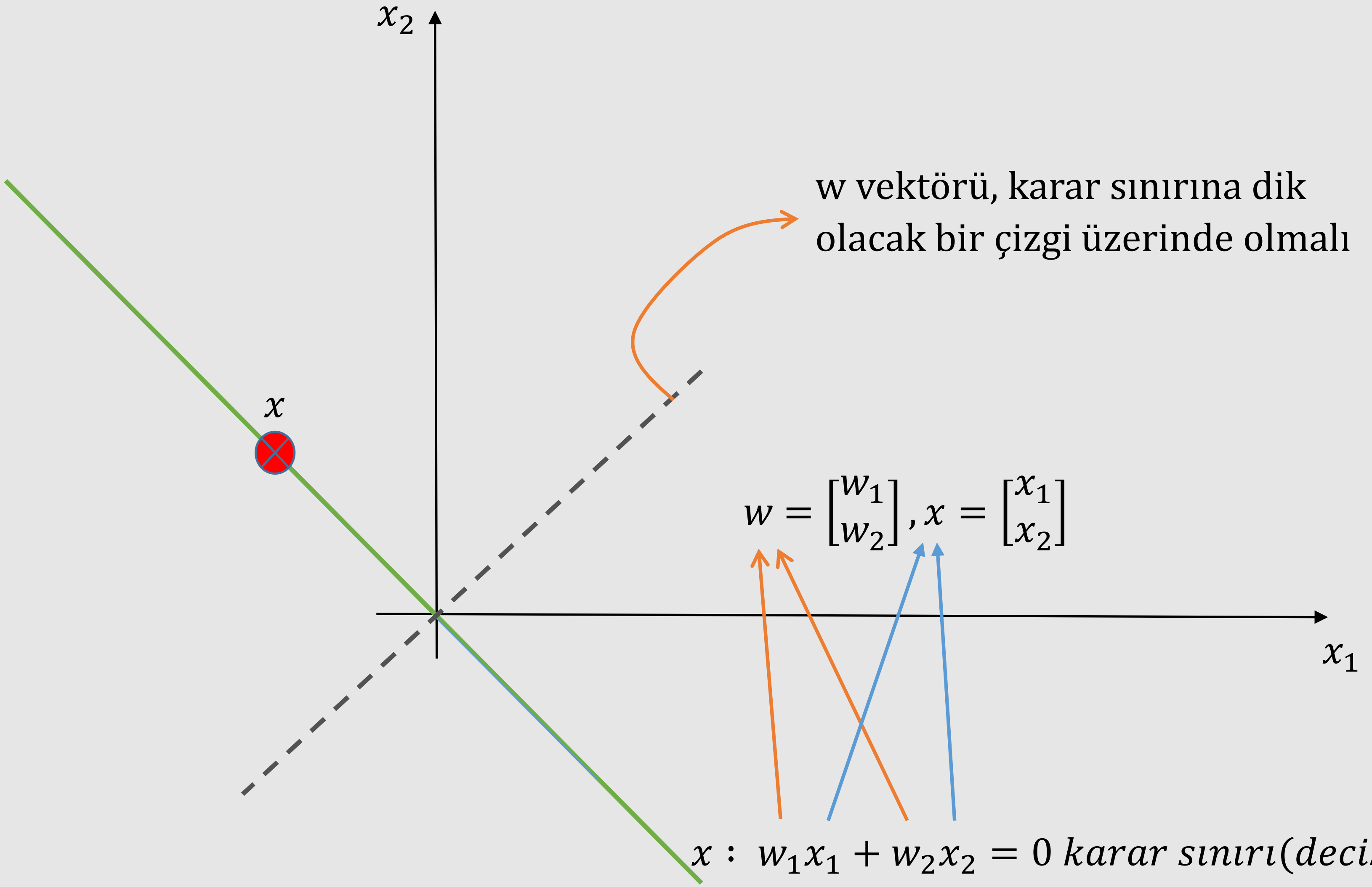


$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$x : \ w_1 x_1 + w_2 x_2 = 0 \ karar \ sınırı (decision \ boundry)$

$x : w \cdot x = 0 \ karar \ sınırı (decision \ boundry)$

Orjinden Geçen Doğrusal Sınıflandırıcı



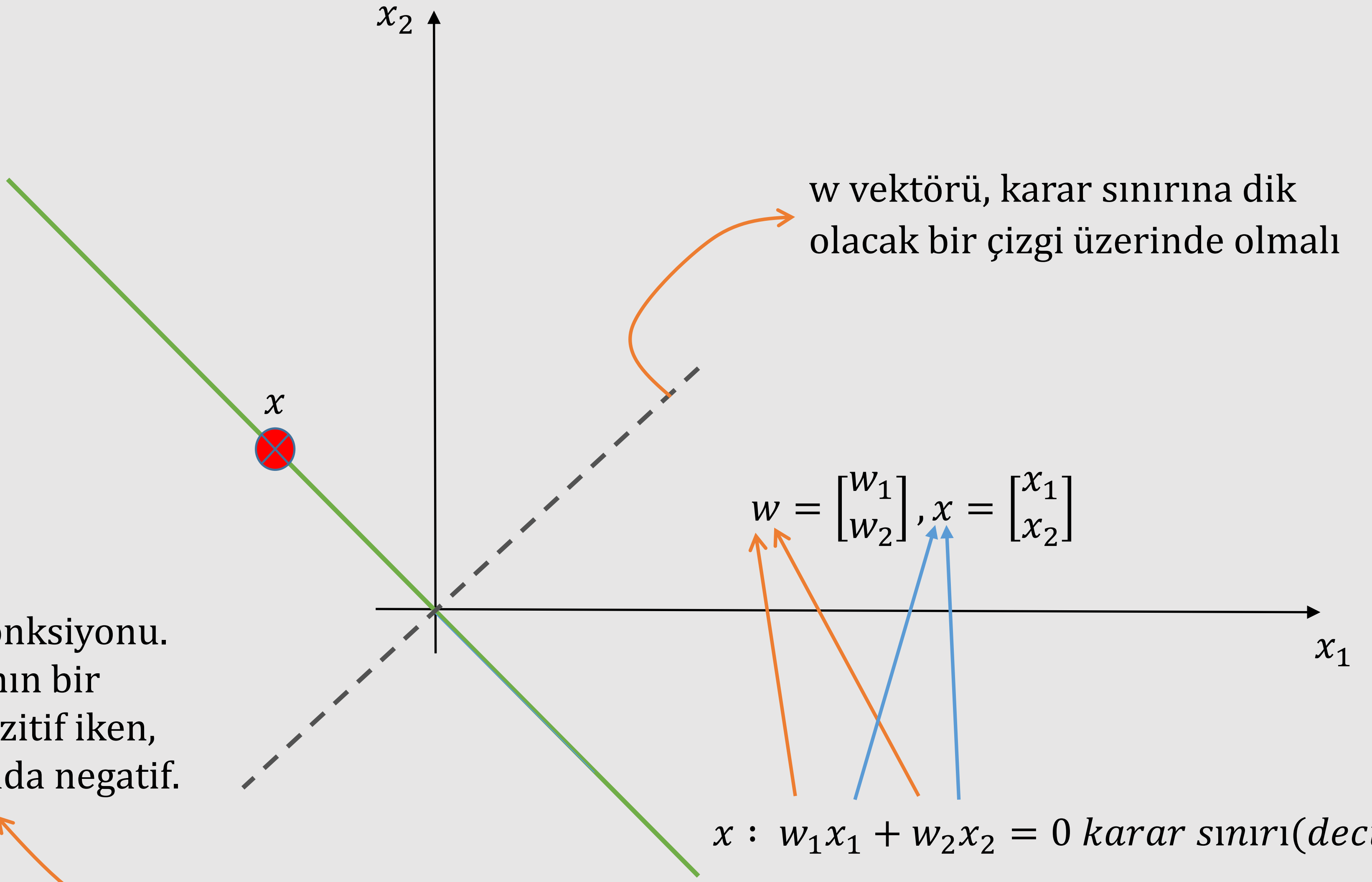
w vektörü, karar sınırına dik olacak bir çizgi üzerinde olmalı

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$x : \ w_1 x_1 + w_2 x_2 = 0 \ karar \ sınırı(decision \ boundry)$$

$$x : w \cdot x = 0 \ karar \ sınırı(decision \ boundry)$$

Orjinden Geçen Doğrusal Sınıflandırıcı



w vektörü, karar sınırına dik olacak bir çizgi üzerinde olmalı

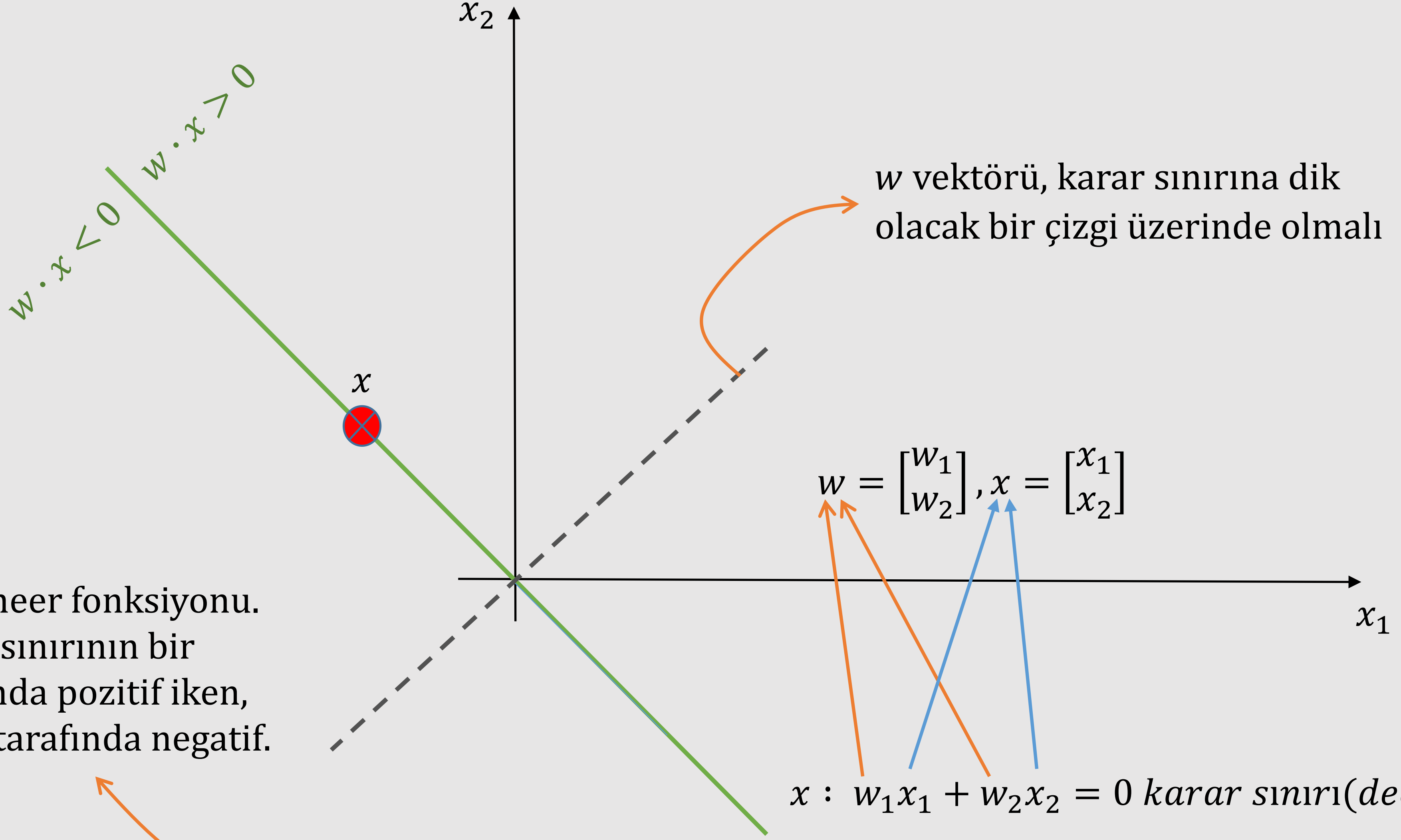
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x'in lineer fonksiyonu. Karar sınırının bir tarafında pozitif iken, diğer tarafında negatif.

$$x : \ w_1x_1 + w_2x_2 = 0 \ karar \ sınırı(decision \ boundry)$$

$$x : \mathrm{w} \cdot x = 0 \ karar \ sınırı(decision \ boundry)$$

Orjinden Geçen Doğrusal Sınıflandırıcı



w vektörü, karar sınırına dik olacak bir çizgi üzerinde olmalı

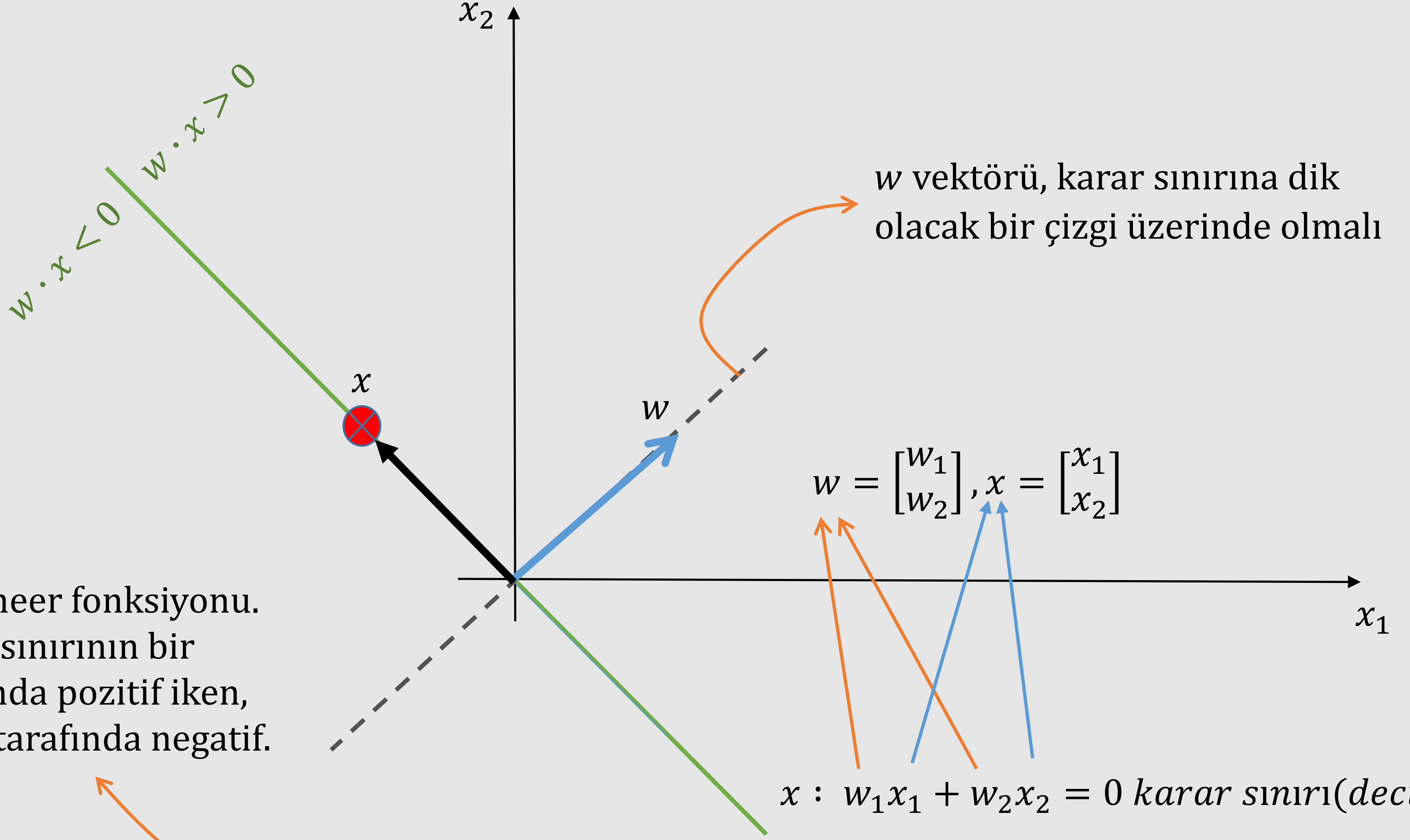
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x'in lineer fonksiyonu. Karar sınırının bir tarafında pozitif iken, diğer tarafında negatif.

$$x : \; w_1 x_1 + w_2 x_2 = 0 \; karar \; sınırı (decision \; boundry)$$

$$x : w \cdot x = 0 \; karar \; sınırı (decision \; boundry)$$

Orjinden Geçen Doğrusal Sınıflandırıcı



w vektörü, karar sınırına dik olacak bir çizgi üzerinde olmalı

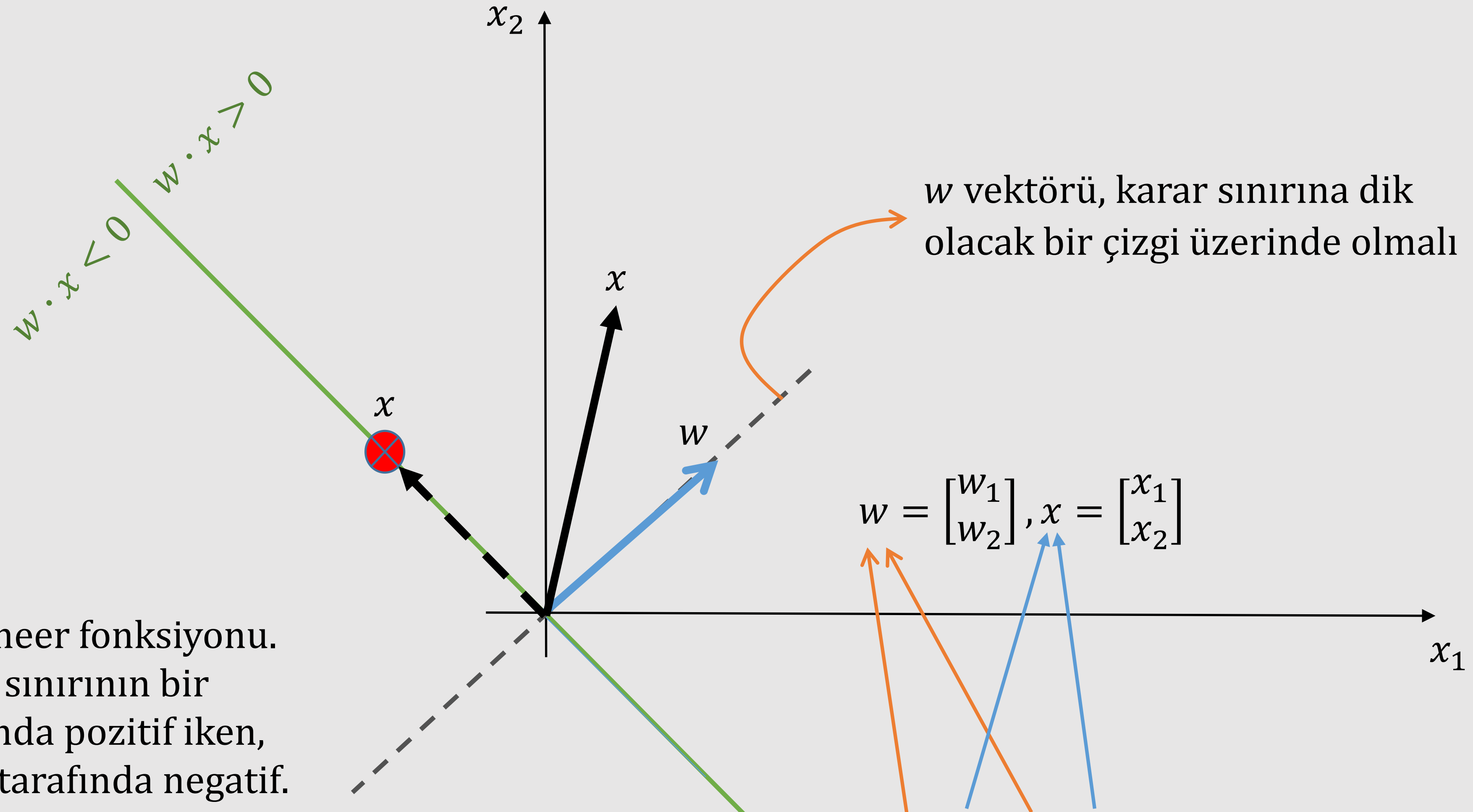
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x'in lineer fonksiyonu. Karar sınırının bir tarafında pozitif iken, diğer tarafında negatif.

$x: \ w_1x_1 + w_2x_2 = 0 \ karar \ sınırı(decision \ boundry)$

$x: w \cdot x = 0 \ karar \ sınırı(decision \ boundry)$

Orjinden Geçen Doğrusal Sınıflandırıcı



w vektörü, karar sınırına dik olacak bir çizgi üzerinde olmalı

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

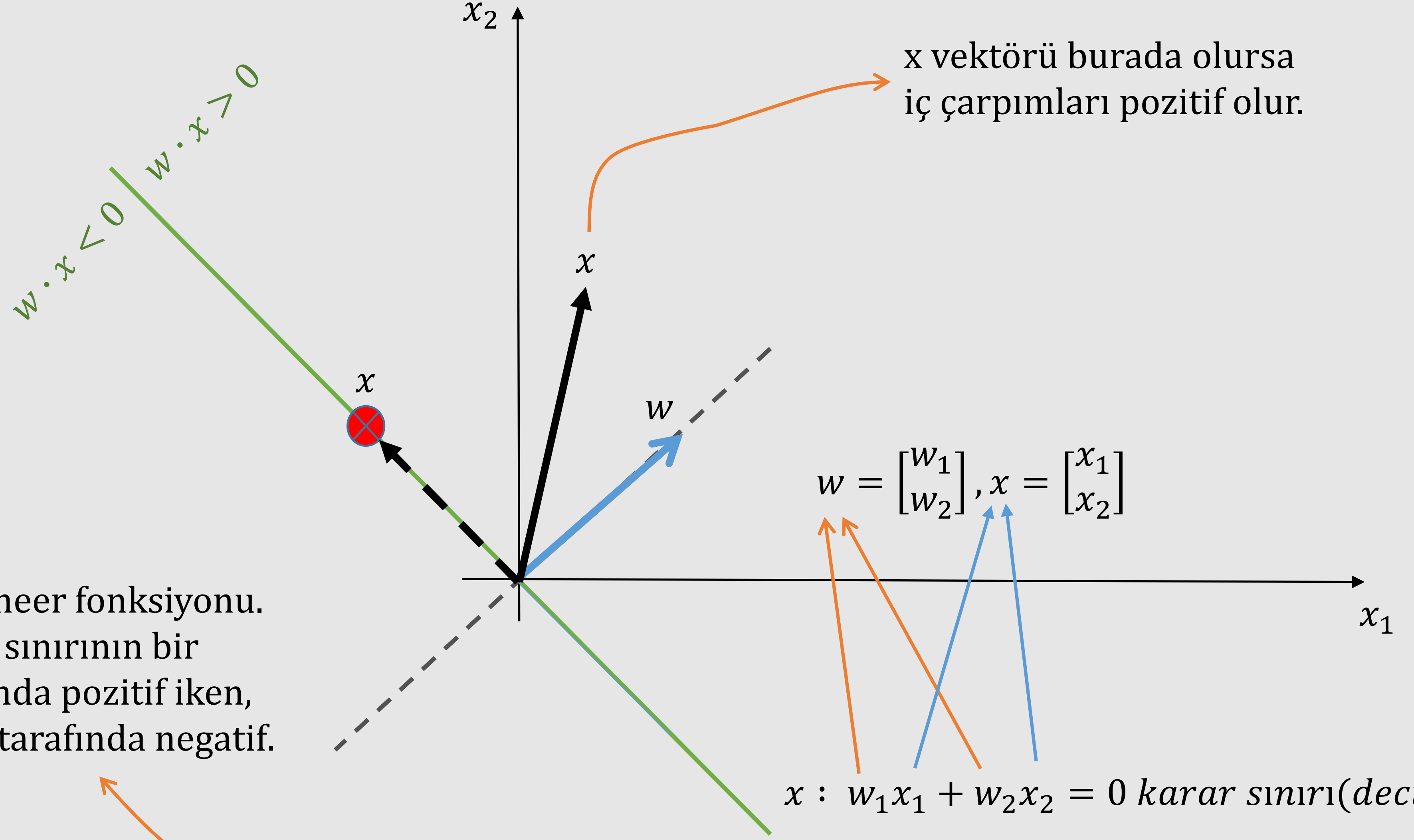
x'in lineer fonksiyonu. Karar sınırının bir tarafında pozitif iken, diğer tarafında negatif.

$x : \; w_1 x_1 + w_2 x_2 = 0 \; karar \; sınırı(decision \; boundry)$

$x : w \cdot x = 0 \; karar \; sınırı(decision \; boundry)$

Orjinden Geçen Doğrusal Sınıflandırıcı

x vektörü burada olursa
iç çarpımları pozitif olur.



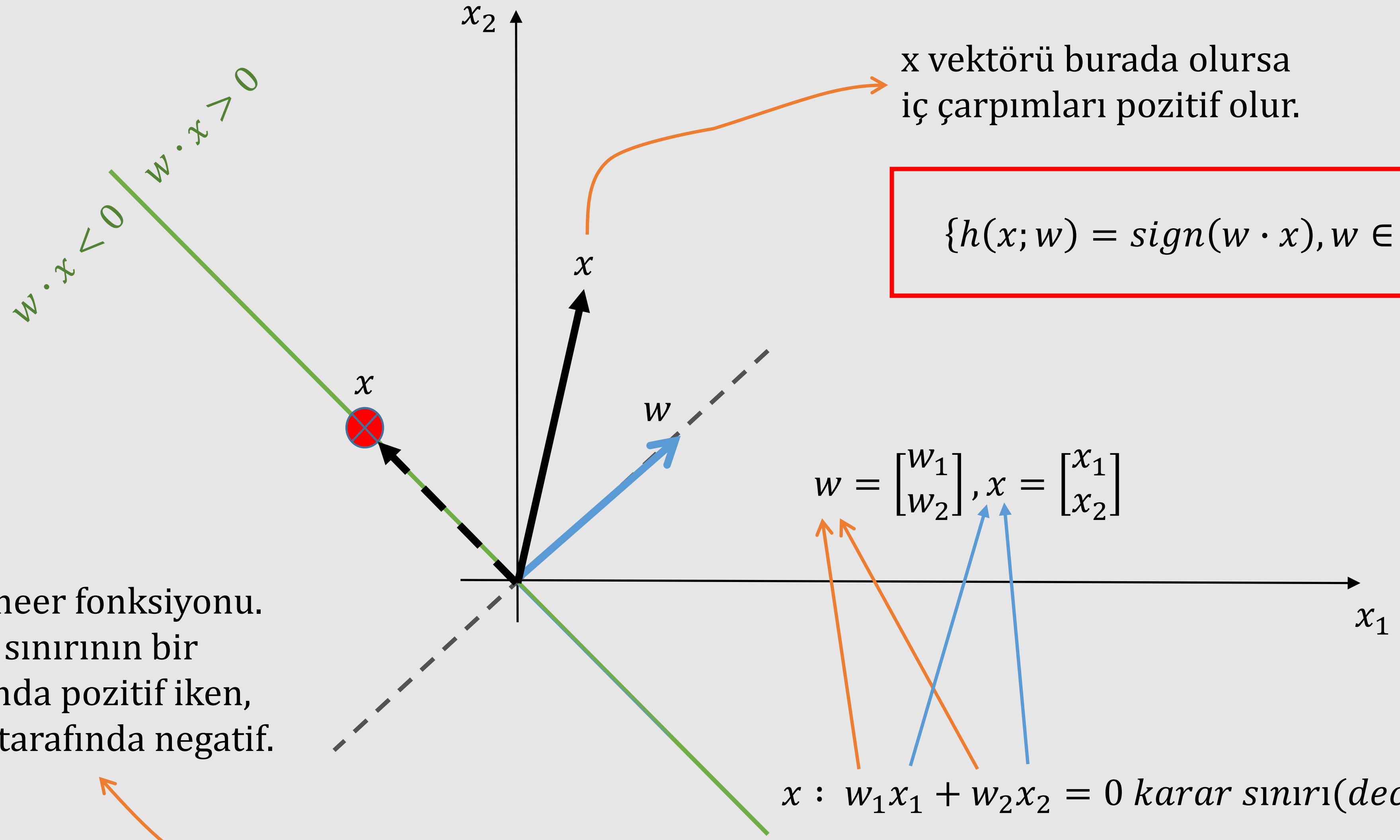
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x’in lineer fonksiyonu.
Karar sınırının bir
tarafında pozitif iken,
diğer tarafında negatif.

$x : \; w_1x_1 + w_2x_2 = 0 \; karar \; sınırı(decision \; boundry)$

$x : w \cdot x = 0 \; karar \; sınırı(decision \; boundry)$

Orjinden Geçen Doğrusal Sınıflandırıcı



x vektörü burada olursa
iç çarpımları pozitif olur.

$$\boxed{\{h(x; w) = sign(w \cdot x), w \in \mathcal{R}^d\}}$$

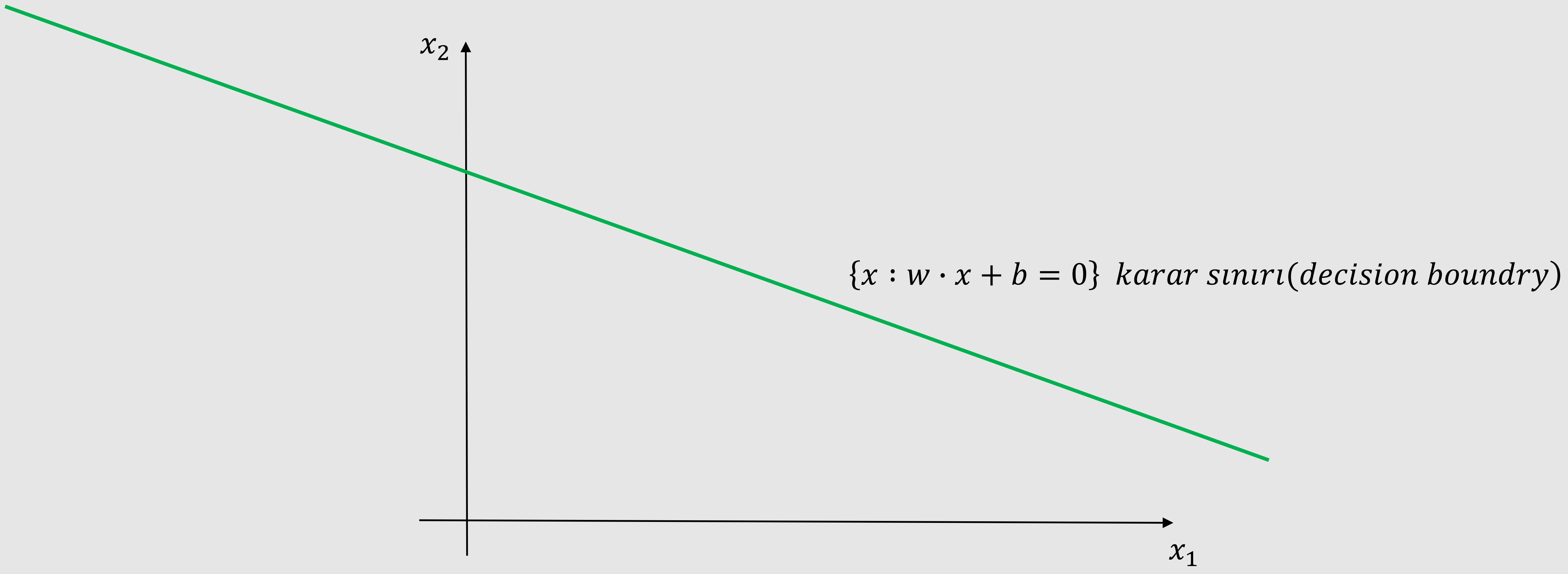
$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

x'in lineer fonksiyonu.
Karar sınırının bir
tarafında pozitif iken,
diğer tarafında negatif.

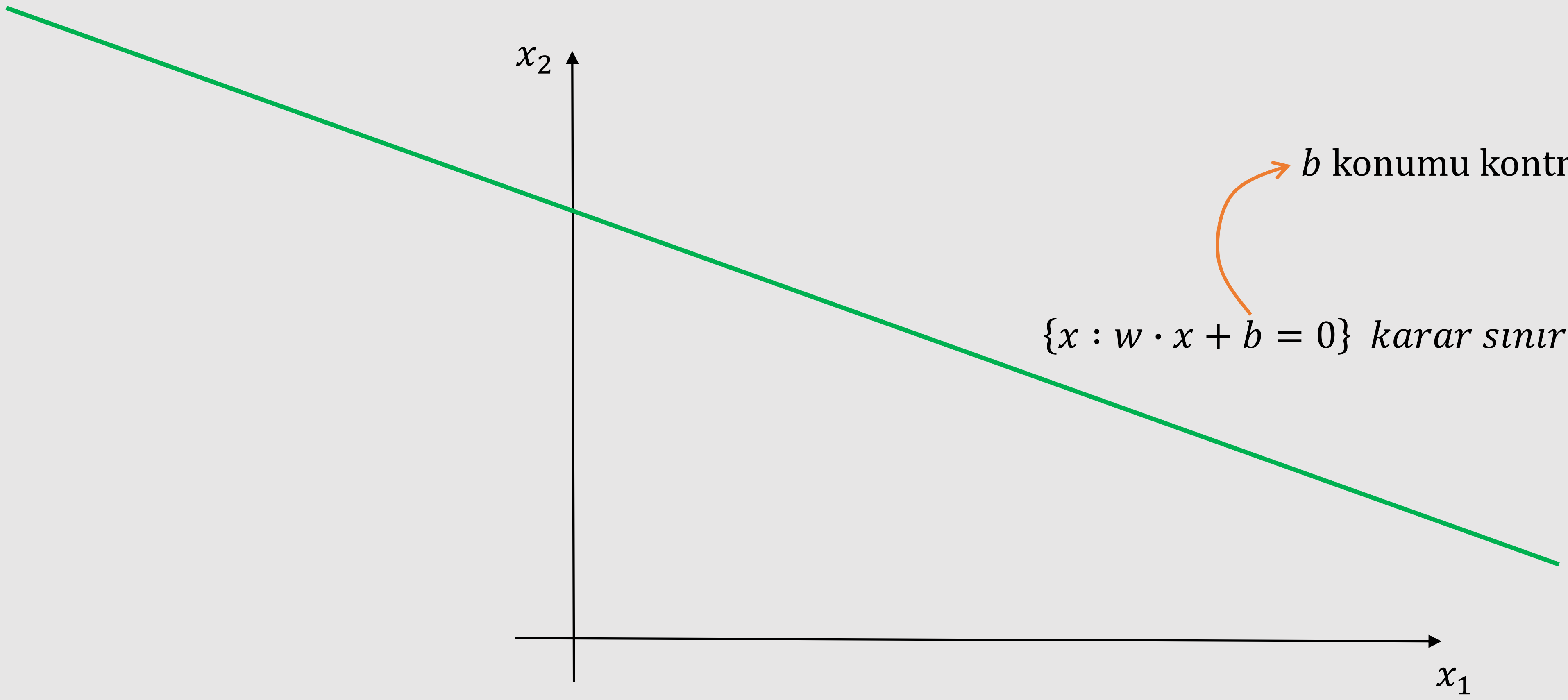
$$x: \; w_1x_1 + w_2x_2 = 0 \; karar \; sınırı(decision \; boundry)$$

$$x: w \cdot x = 0 \; karar \; sınırı(decision \; boundry)$$

Doğrusal Sınıflandırıcılar



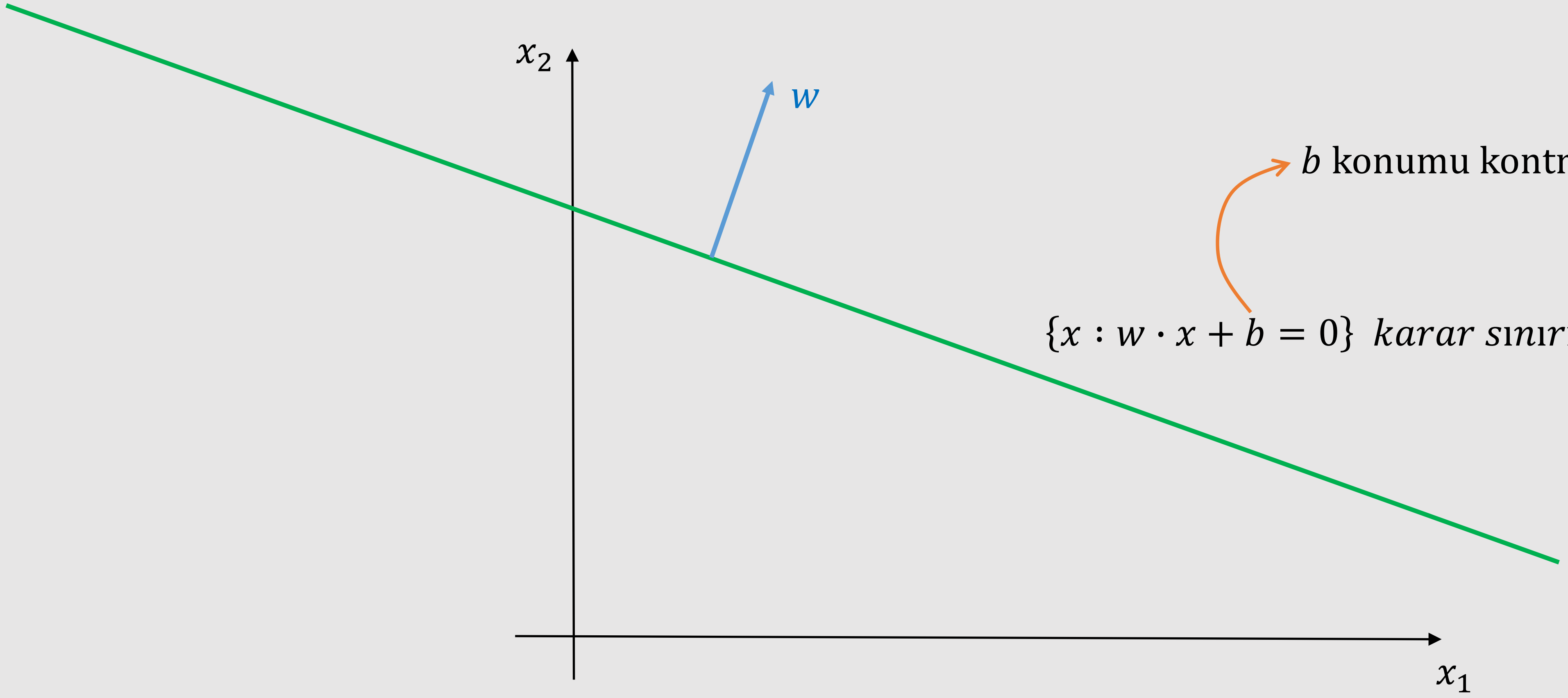
Doğrusal Sınıflandırıcılar



b konumu kontrol ediyor.

$\{x : w \cdot x + b = 0\}$ *karar sınırı(decision boundry)*

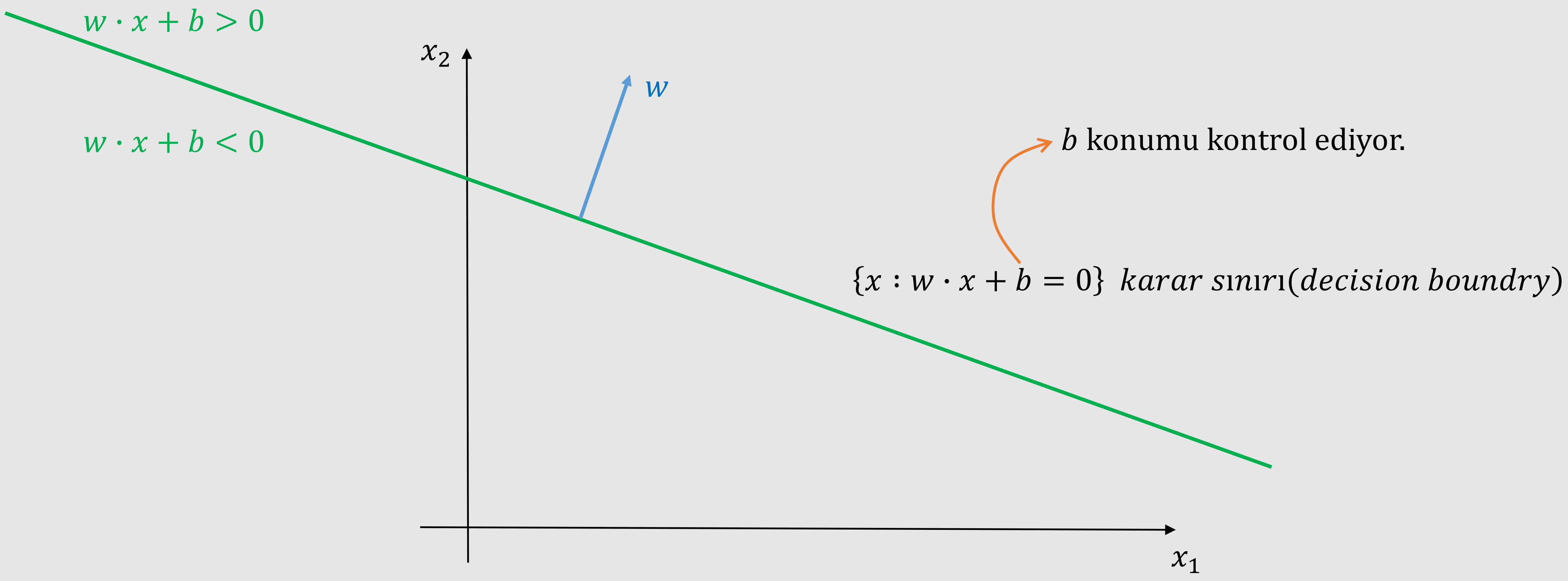
Doğrusal Sınıflandırıcılar



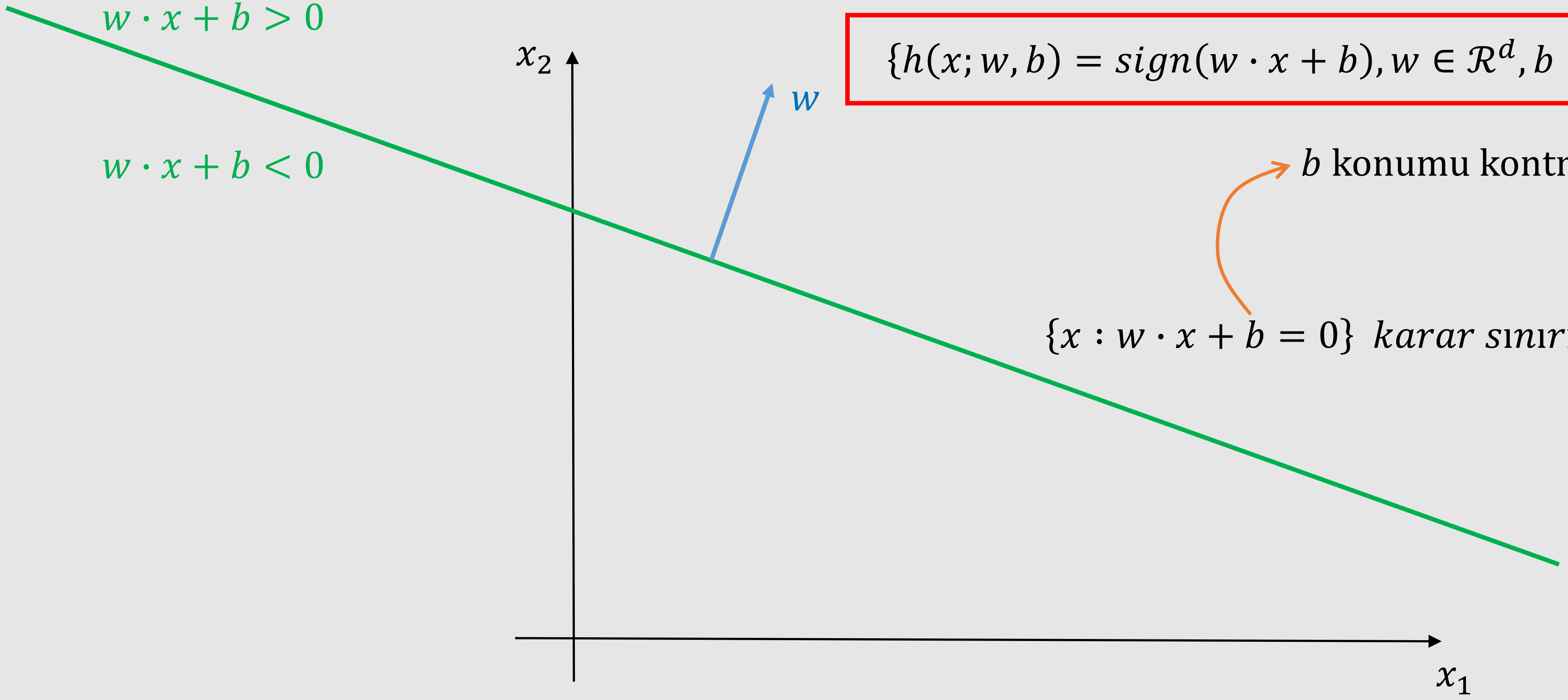
b konumu kontrol ediyor.

$\{x : w \cdot x + b = 0\}$ *karar sınırı(decision boundry)*

Doğrusal Sınıflandırıcılar



Doğrusal Sınıflandırıcılar



$$\boxed{\{h(x; w, b) = sign(w \cdot x + b), w \in \mathcal{R}^d, b \in \mathcal{R}\}}$$

b konumu kontrol ediyor.

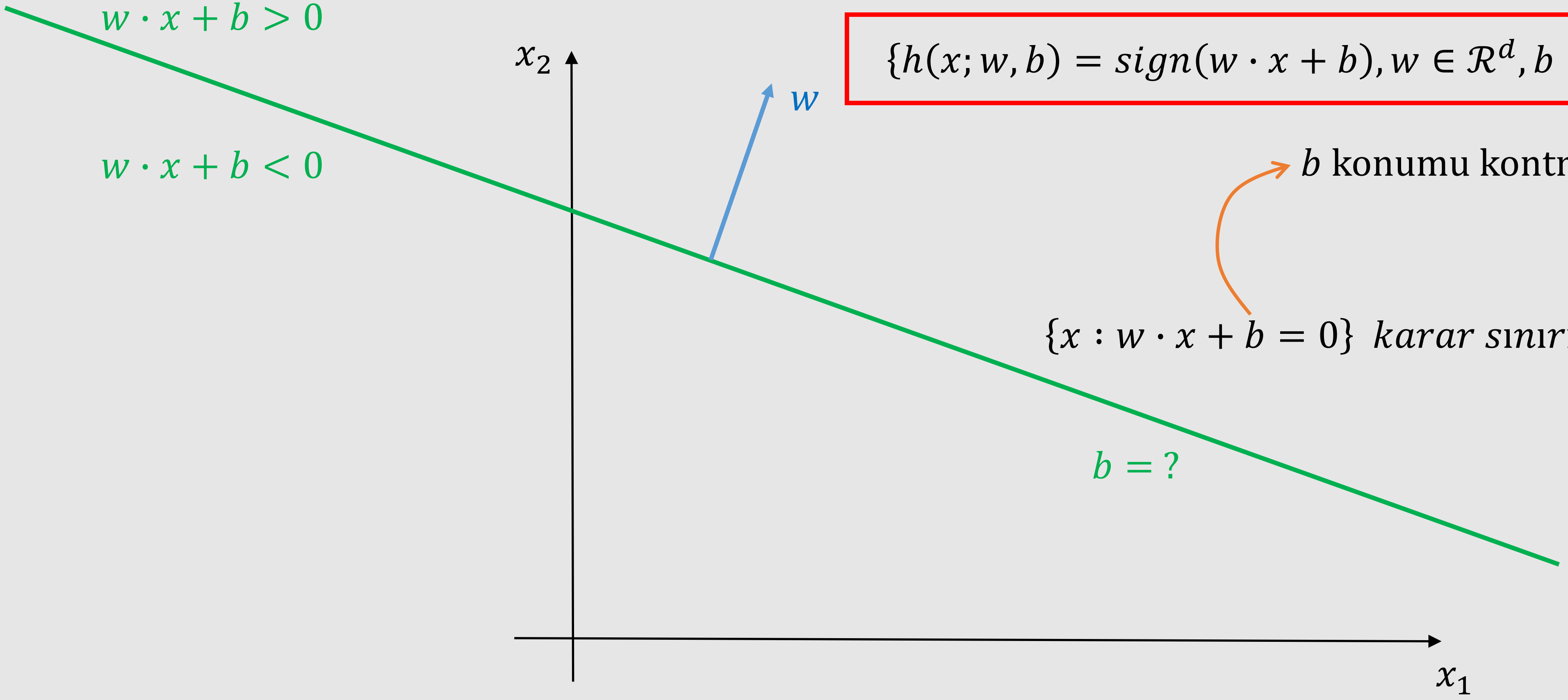
$\{x : w \cdot x + b = 0\}$ *karar sınırı(decision boundry)*

Doğrusal Sınıflandırıcılar

$$\{h(x; w, b) = sign(w \cdot x + b), w \in \mathcal{R}^d, b \in \mathcal{R}\}$$

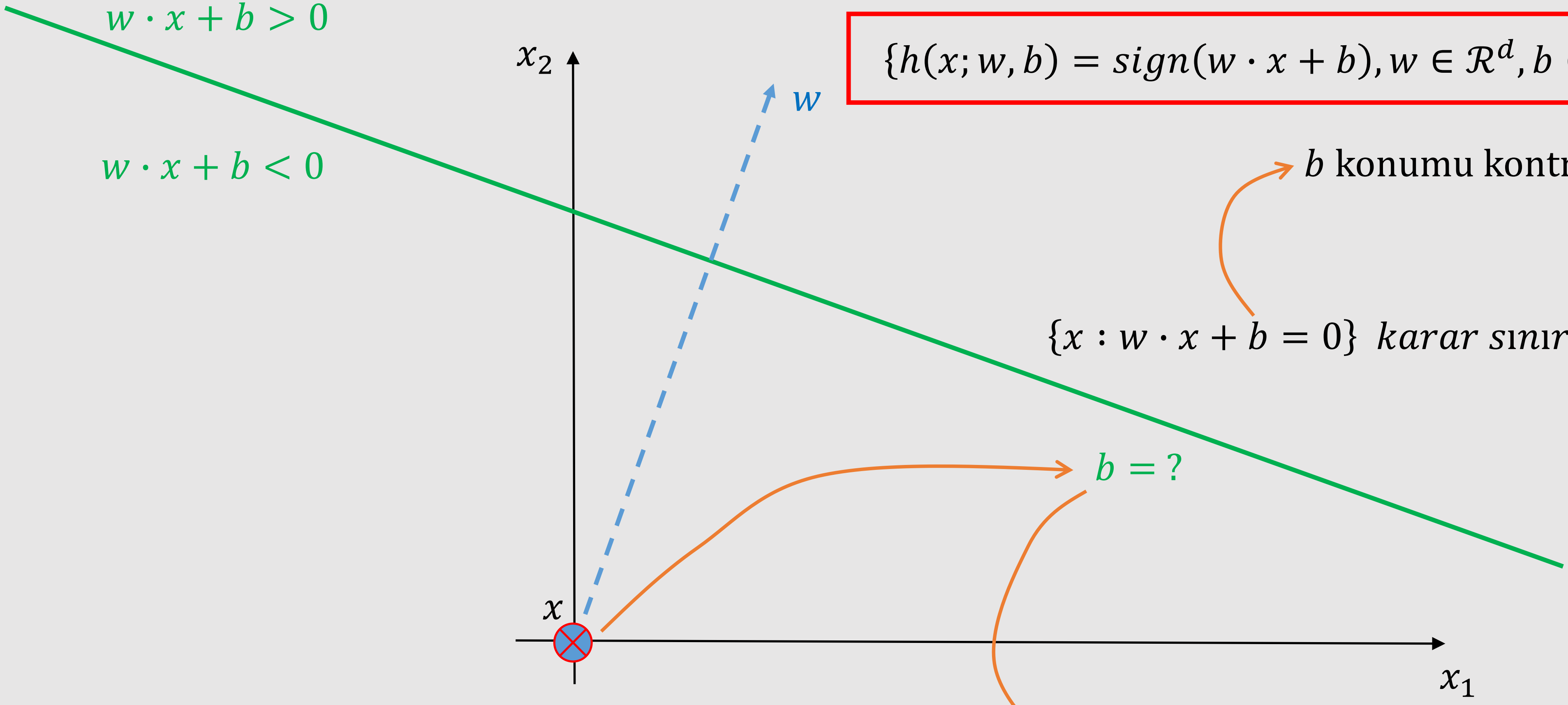
b konumu kontrol ediyor.

$\{x : w \cdot x + b = 0\}$ *karar sınırı(decision boundry)*



Doğrusal Sınıflandırıcılar

$$\boxed{\{h(x; w, b) = sign(w \cdot x + b), w \in \mathcal{R}^d, b \in \mathcal{R}\}}$$



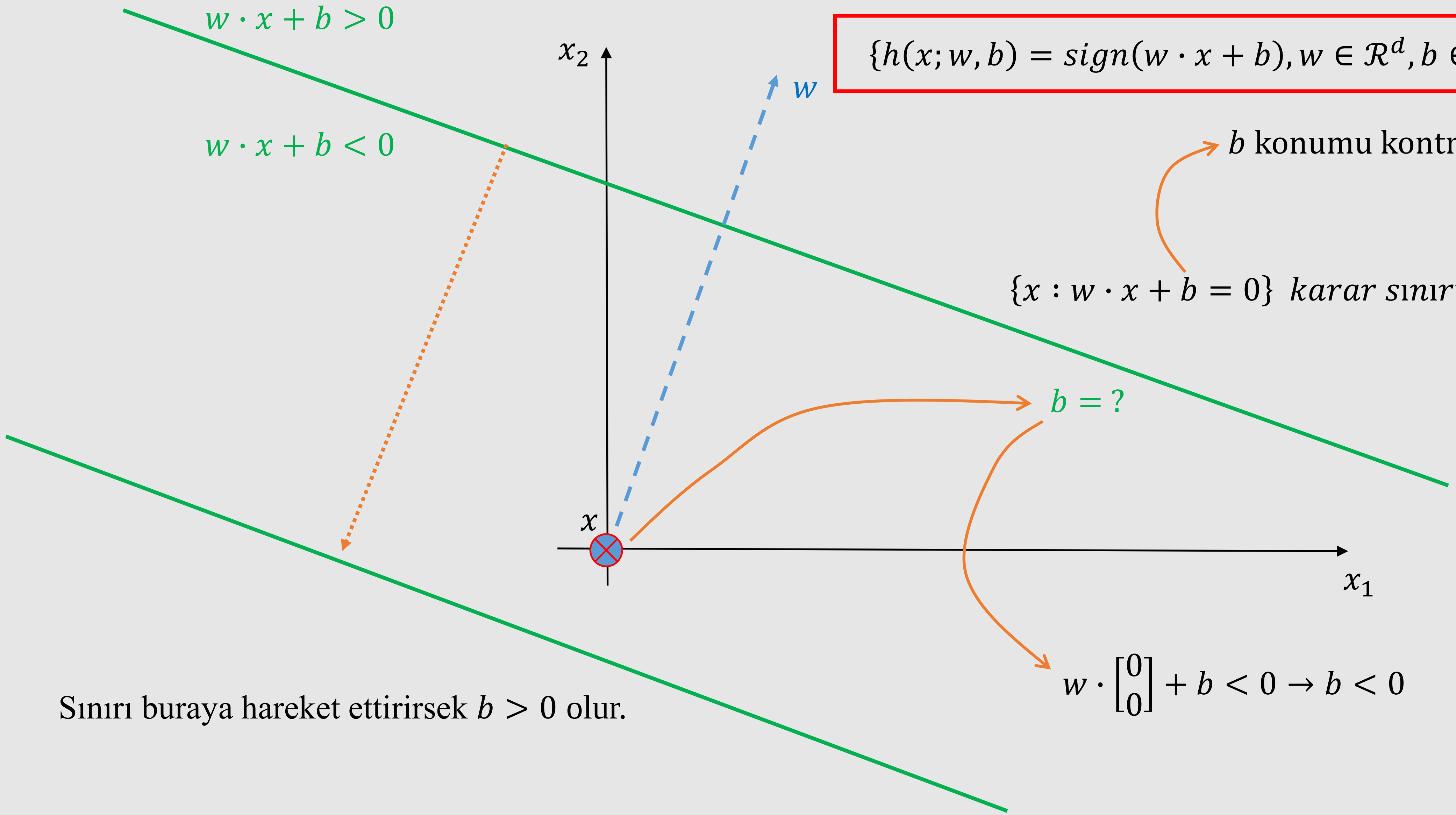
b konumu kontrol ediyor.

$\{x : w \cdot x + b = 0\}$ *karar sınırı(decision boundry)*

$$w \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + b < 0 \rightarrow b < 0$$

Doğrusal Sınıflandırıcılar

$$\{h(x; w, b) = sign(w \cdot x + b), w \in \mathcal{R}^d, b \in \mathcal{R}\}$$



b konumu kontrol ediyor.

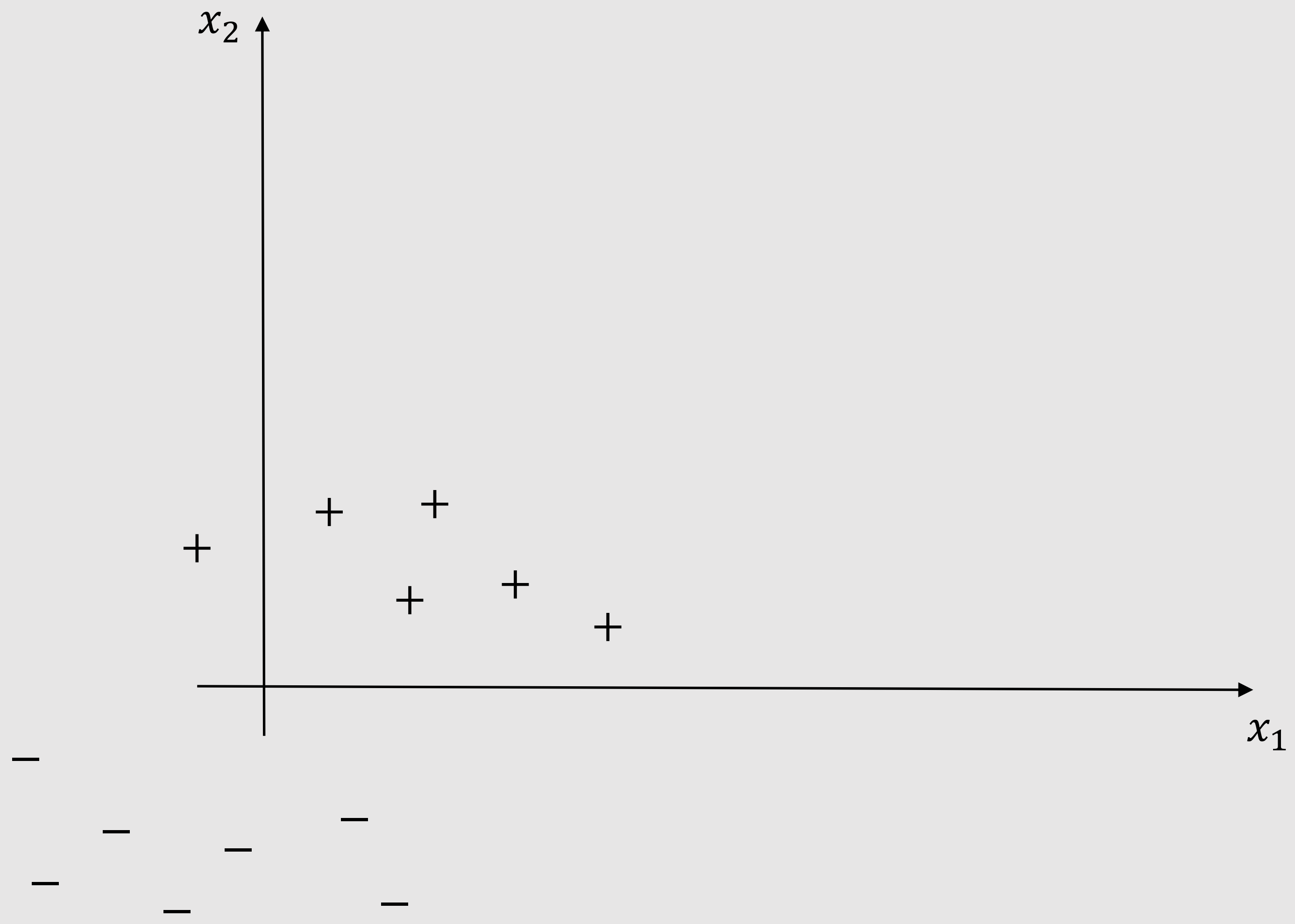
$\{x : w \cdot x + b = 0\}$ $karar\ sınırı(decision\ boundry)$

$$w \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + b < 0 \rightarrow b < 0$$

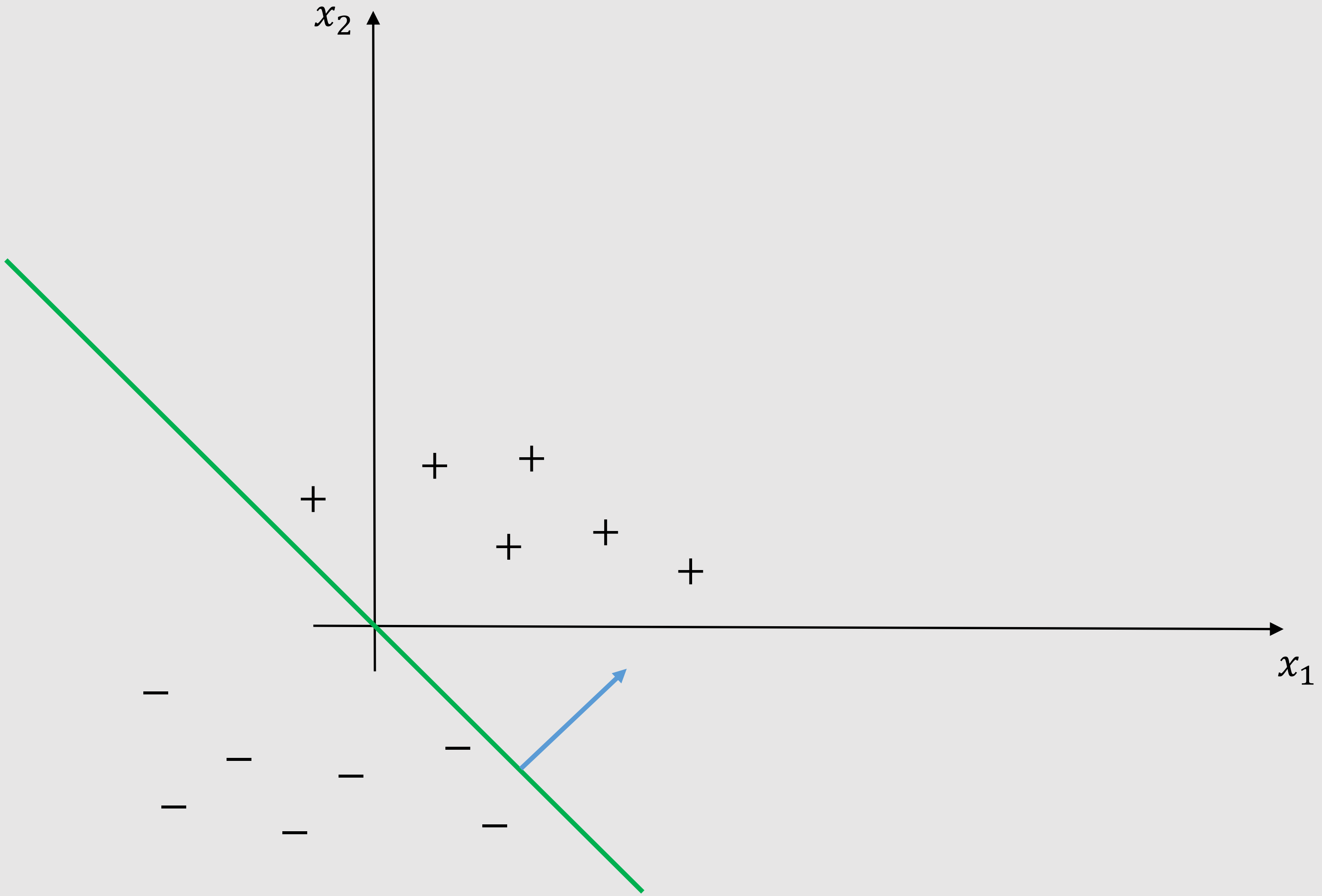
Sınırı buraya hareket ettirirsek $b > 0$ olur.

Doğrusal Ayırma : Örnek

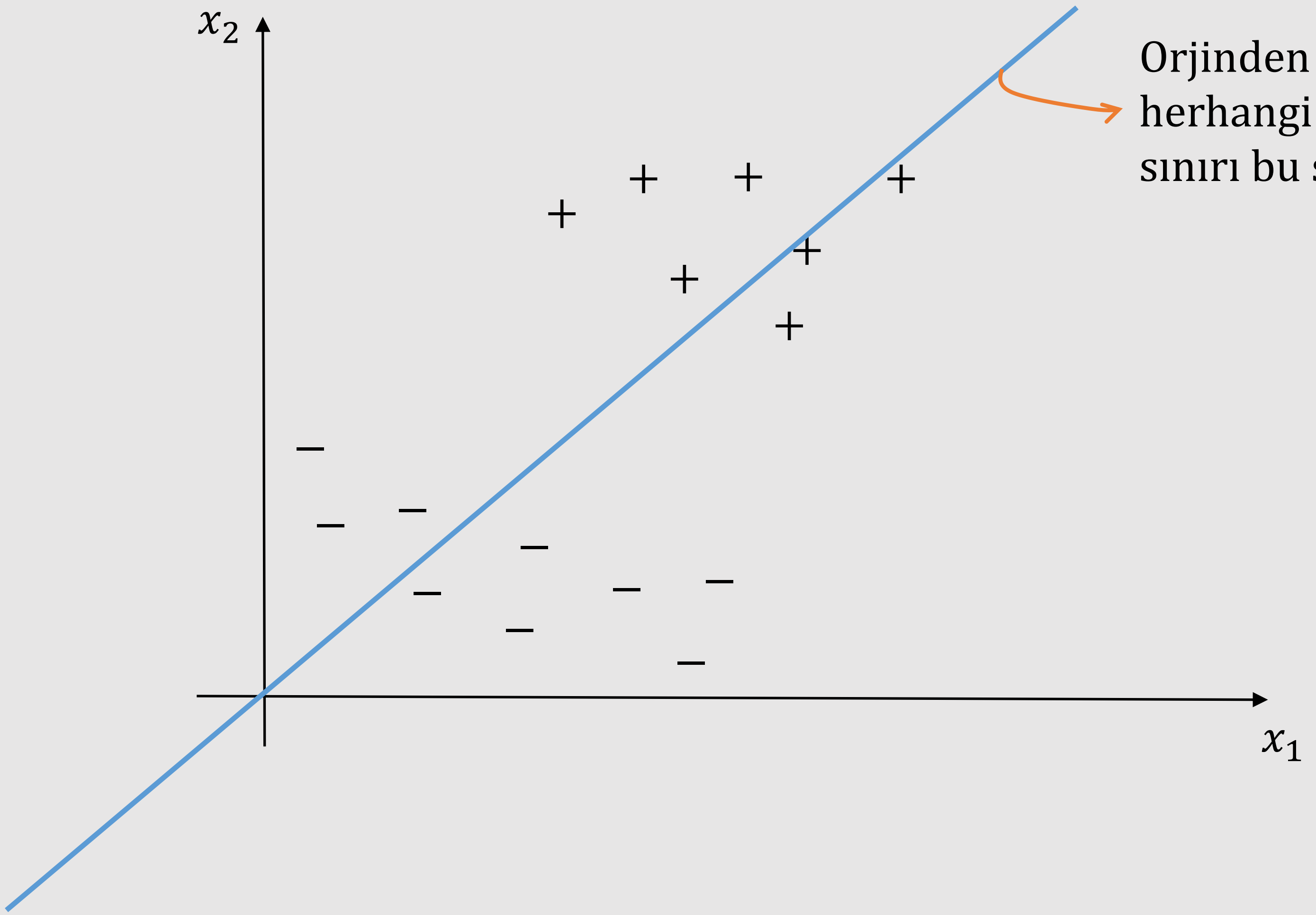
Doğrusal Ayırma : Örnek



Doğrusal Ayırma : Örnek

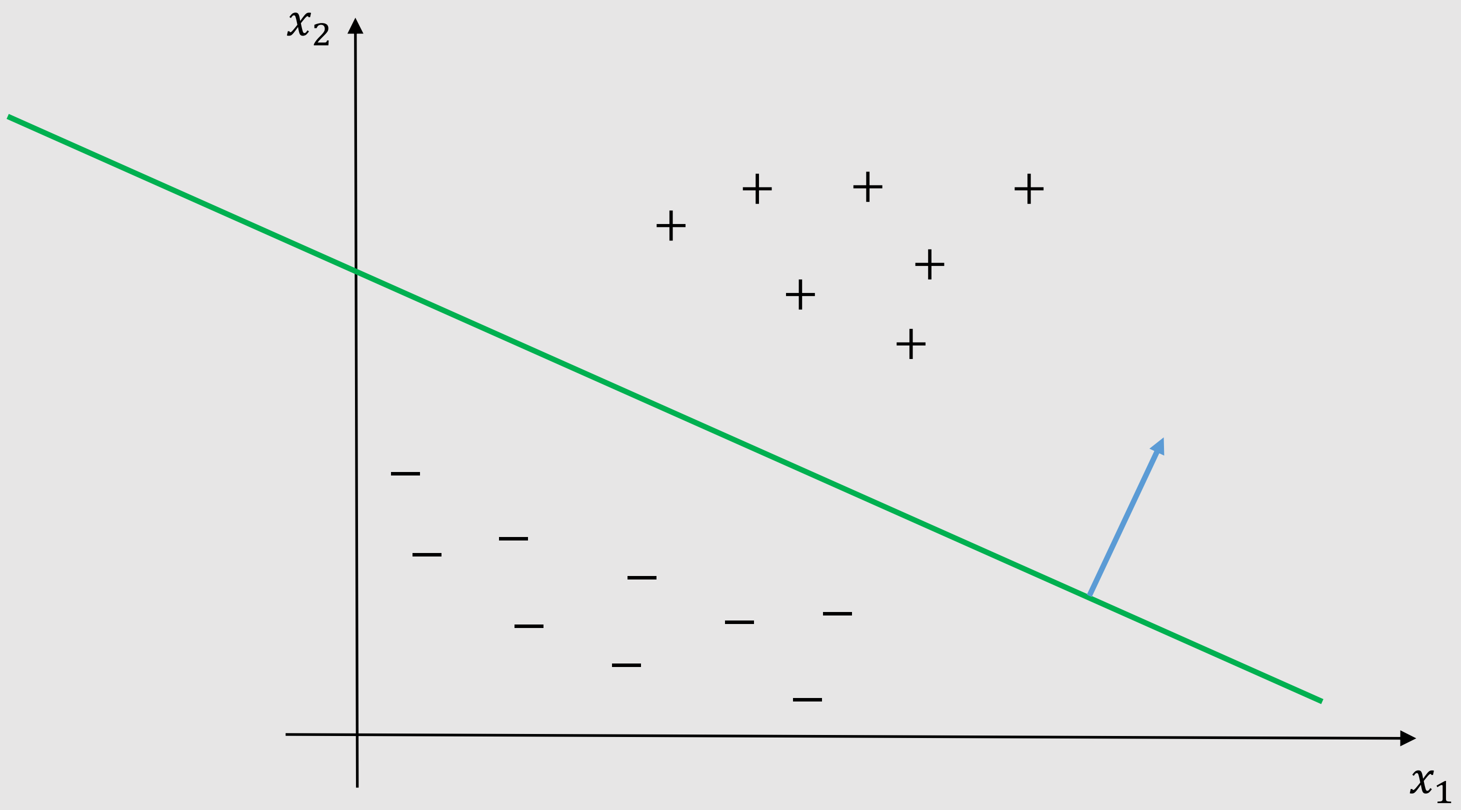


Doğrusal Ayırma : Örnek

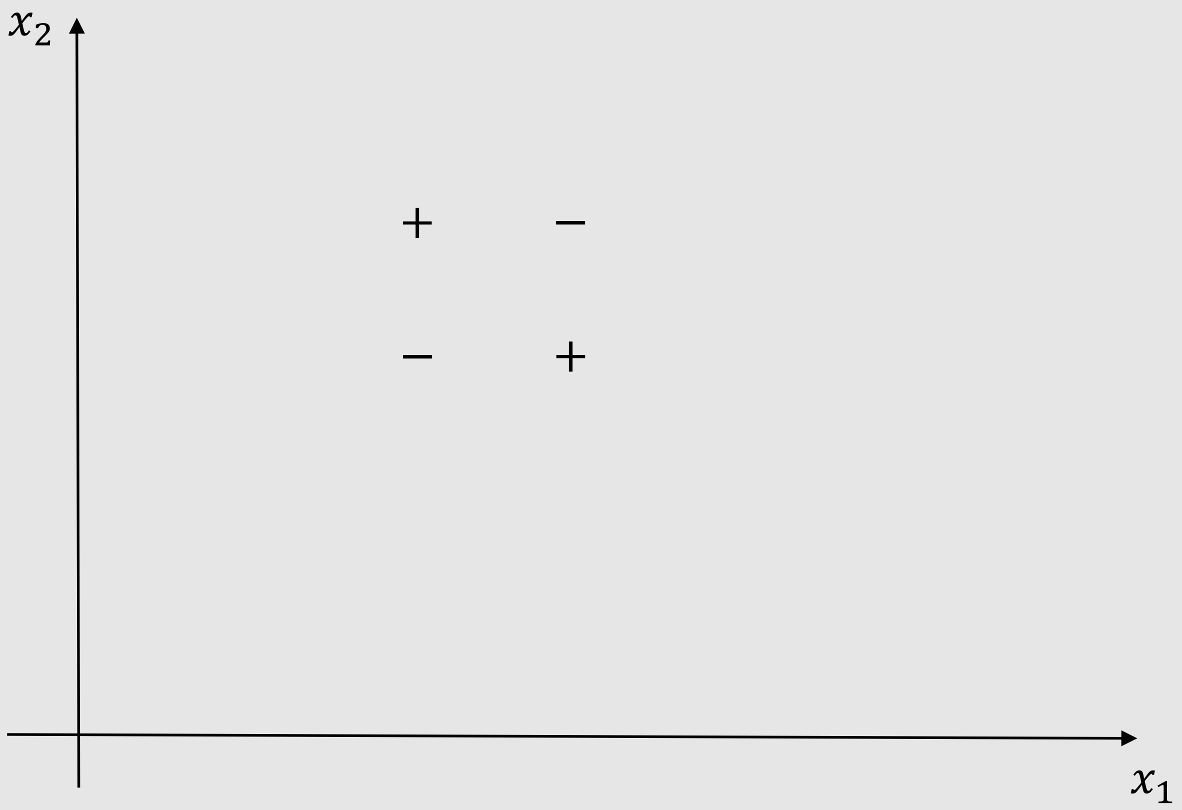


Orjinden geçen
herhangi bir karar
sınırı bu seti ayıramaz !

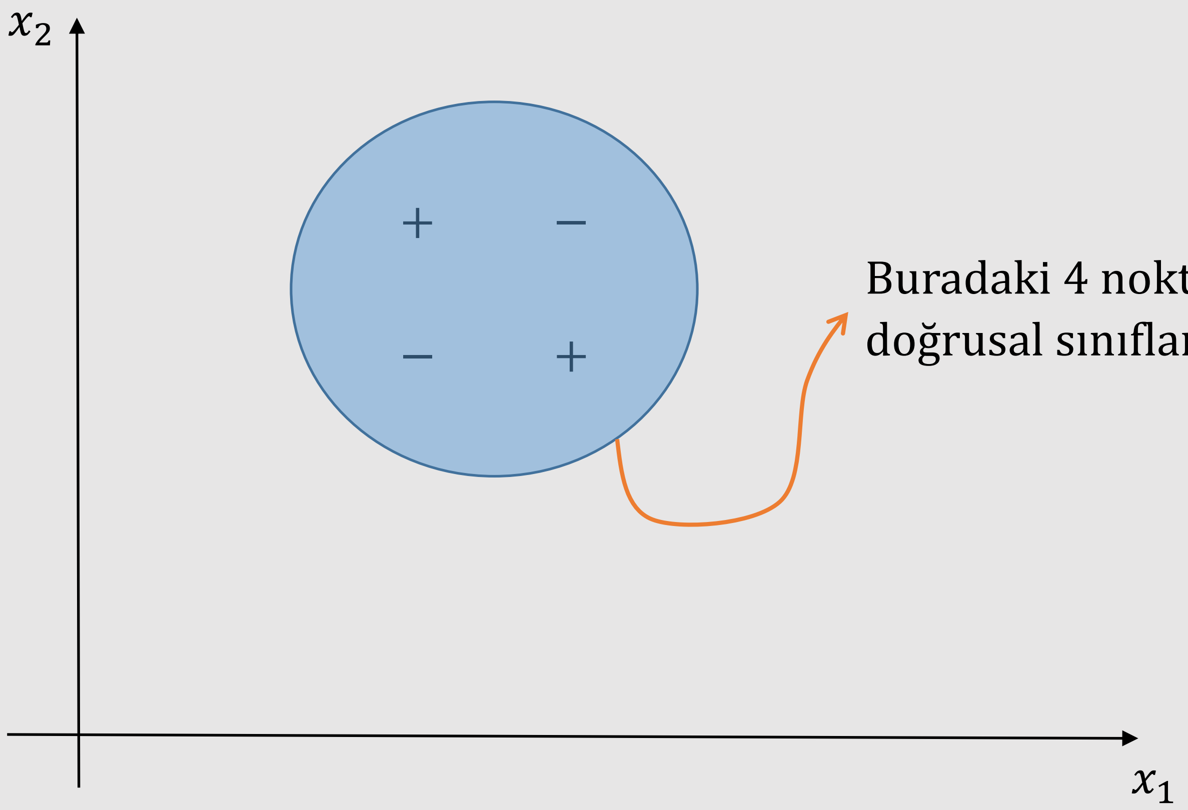
Doğrusal Ayırma : Örnek



Doğrusal Ayırma : Örnek



Doğrusal Ayırma : Örnek



Buradaki 4 noktayı da ayıracak bir doğrusal sınıflandırıcı bulunmamakta.

Doğrusal Ayırma

Tanım:

Eğitim örnekleri $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \dots n\}$ *doğrusal ayrılabilir* eğer ki uygun bir parametre vektörü $\hat{w}$ ve denkleştirme parametresi(offset parameter) $\hat{b}$ aşağıda bulunan denklemdeki koşula uyuyor ise;

$$y^{(i)}(\hat{w} \cdot x^{(i)} + \hat{b}\,) > 0, \qquad i = 1, \dots, n$$

Doğrusal Ayırma

Tanım:

Eğitim örnekleri $S_n = \{(x^{(i)}, y^{(i)}), i = 1, \dots n\}$ *doğrusal ayrılabilir* eğer ki uygun bir parametre vektörü $\hat{w}$ ve denkleştirme parametresi(offset parameter) $\hat{b}$ aşağıda bulunan denklemdeki koşula uyuyor ise;

$$y^{(i)}\left(\hat{w} \cdot x^{(i)} + \hat{b}\right) > 0, \qquad i = 1, \dots, n$$

Sonuçların işaretleri birbirlerine eşit olmalı, aksi halde doğrusal ayırmamız mümkün değil.

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n}[\![h(x^{(i)}) \neq y^{(i)}]\!]$$

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n}[\![h(x^{(i)}) \neq y^{(i)}]\!]$$

Eğitim hatası

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![h(x^{(i)}) \neq y^{(i)}]\!]$$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n} [\![h(x^{(i)}) \neq y^{(i)}]\!]$$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

Orjinden geçen doğrusal sınıflandırıcının hatası

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n}[\![h(x^{(i)}) \neq y^{(i)}]\!]$$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n}[\![\underbrace{y^{(i)} w \cdot x^{(i)} \leq 0}]\!]$$

Orjinden geçen doğrusal sınıflandırıcının hatası

Verilen $y^{(i)}$ etiketinde, bu etiketin işareti $(w \cdot x^{(i)})$ iç çarpımının sonucunda elde edilen işaret ile aynı olmalı.

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası(orjinden)

$$\mathcal{E}_n(h) = \frac{1}{n}\sum_{i=1}^{n}[\![h(x^{(i)}) \neq y^{(i)}]\!]$$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n}[\![\underbrace{y^{(i)} w \cdot x^{(i)} \leq 0}]\!]$$

Orjinden geçen doğrusal sınıflandırıcının hatası

Verilen $y^{(i)}$ etiketinde, bu etiketin işareti $(w \cdot x^{(i)})$ iç çarpımının sonucunda elde edilen işaret ile aynı olmalı.

Eğer $y^{(i)} w \cdot x^{(i)} = 0$ ise bu örnek tam karar sınırının üzerinde olduğu anlamına gelir. Ve bunu bir hata olarak kabul ederiz. Çünkü hangi yöne doğru sınıflandırma yapmamız gerektiğini bilmiyoruz.

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası

$$\mathcal{E}_n(w, b) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)}(w \cdot x^{(i)} + b) \leq 0]\!]$$

Doğrusal Sınıflandırıcıyı Öğrenme

- Doğrusal sınıflandırma için eğitim hatası

$$\mathcal{E}_n(w,b) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{[\![y^{(i)}(w \cdot x^{(i)} + b) \leq 0]\!]}$$

0 yada negatif olursa hata
olarak kabul edilir.
Pozitif olursa hata yoktur.

Perceptron Öğrenme Algoritması

$$w = 0 \ (vektör)$$

Perceptron Öğrenme Algoritması

$w = 0\ (vektör)$

Parametremizi sıfıra eşitleriz.

Perceptron Öğrenme Algoritması

$$w = 0 \ (vektör)$$

$$\boldsymbol{if} \ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0 \ \boldsymbol{then}$$

$$w = w + y^{(i)} x^{(i)}$$

Perceptron Öğrenme Algoritması

$$w = 0\ (vektör)$$

$$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$$

$$w = w + y^{(i)} x^{(i)}$$

$i.$ örnek $\left(i^{th} example\right)$

Perceptron Öğrenme Algoritması

$w = 0 \; (vektör)$

$\boldsymbol{if} \; y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0 \; \boldsymbol{then}$

$w = w + y^{(i)}x^{(i)}$

Doğrusal sınıflandırma bir hata yapar.

$i.\,örnek \; \left(i^{th} example\right)$

Perceptron Öğrenme Algoritması

$$w = 0\ (vektör)$$

$$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$$

$$w = w + y^{(i)}x^{(i)}$$

Doğrusal sınıflandırma bir hata yapar.

$i.$ örnek $\left(i^{th} example\right)$

Biz de bu hatayı düzeltmek için
parametreleri güncelleriz.

Perceptron Öğrenme Algoritması

$$w = 0\ (vektör)$$

$$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$$

$$w = w + y^{(i)}x^{(i)}$$

Doğrusal sınıflandırma bir hata yapar.

$i.$ örnek $(i^{th} example)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Perceptron Öğrenme Algoritması

$$w = 0\ (vektör)$$

$$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \le 0\ \boldsymbol{then}$$

$$w = w + y^{(i)}x^{(i)}$$

Doğrusal sınıflandırma bir hata yapar.

$i.\ örnek\ (i^{th} example)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Sonraki adım → $y_{yeni}^{(i)}\left(w + y^{(i)}x^{(i)}\right) \cdot x_{yeni}^{(i)}$

Perceptron Öğrenme Algoritması

$w = 0\ (vektör)$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)}x^{(i)}$

Doğrusal sınıflandırma bir hata yapar.

$i.\ örnek\ (i^{th} example)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Sonraki adım → $y_{yeni}^{(i)}\left(w + y^{(i)}x^{(i)}\right) \cdot x_{yeni}^{(i)}$

0

Perceptron Öğrenme Algoritması

$w = 0 \ (vektör)$

$\boldsymbol{if} \ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0 \ \boldsymbol{then}$

$w = w + y^{(i)}x^{(i)}$

Doğrusal sınıflandırma bir hata yapar.

$i.\,örnek \ (i^{th} example)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Sonraki adım → $y_{yeni}^{(i)}\underbrace{\left(w + y^{(i)}x^{(i)}\right)}_{w_{yeni}} \cdot x_{yeni}^{(i)}$

Perceptron Öğrenme Algoritması

$$w = 0 \ (vektör)$$

$$\boldsymbol{if} \ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0 \ \boldsymbol{then}$$

$$w = w + y^{(i)}x^{(i)}$$

Doğrusal sınıflandırma bir hata yapar.

$i.$ örnek $(i^{th} example)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Sonraki adım → $y_{yeni}^{(i)}\left(w + y^{(i)}x^{(i)}\right) \cdot x_{yeni}^{(i)} = \left\|x^{(i)}\right\|^2 > 0$

Perceptron Öğrenme Algoritması

$w = 0 \; (vektör)$

$\quad \boldsymbol{for} \; i = 1, \ldots, n \; \boldsymbol{do}$

$\quad\quad \boldsymbol{if} \; y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0 \; \boldsymbol{then}$

$\quad\quad w = w + y^{(i)} x^{(i)}$

Perceptron Öğrenme Algoritması

Bütün eğitim örnek sayısı

$w = 0\ (vektör)$

$\boldsymbol{for}\ i = 1, \dots, n\ \boldsymbol{do}$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)} x^{(i)}$

Perceptron Öğrenme Algoritması

Bütün eğitim örnek sayısı

$w = 0\ (vektör)$

$\boldsymbol{for}\ i = 1, \dots, n\ \boldsymbol{do}$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)} x^{(i)}$

Eğer i. Örneğimiz bir hata ise(ilk başta her zaman hata!), güncelleme yaparız.

Perceptron Öğrenme Algoritması

$w = 0\ (vektör)$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)}x^{(i)}$

Doğrusal sınıflandırma bir hata yapar.

$i.\, örnek\ \left(i^{th} example\right)$

Biz de bu hatayı düzeltmek için parametreleri güncelleriz.

İlk örnek <u>her zaman</u> hata olarak varsayılır(yaratılır).

Sonraki adım → $y_{yeni}^{(i)}\left(w + y^{(i)}x^{(i)}\right) \cdot x_{yeni}^{(i)} = {y^{(i)}}^2 \left\|x^{(i)}\right\|^2$

1 yada -1. Her şekilde karesi bize 1'i verir

Perceptron Öğrenme Algoritması

$\boldsymbol{procedure}\; PERCEPTRON\left(\left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \ldots, n\right\}, T\right)$

$w = 0\ (vekt\ddot{o}r)$

$\boldsymbol{for}\; t = 1, \ldots, T\; \boldsymbol{do}$

$\boldsymbol{for}\; i = 1, \ldots, n\; \boldsymbol{do}$

$\boldsymbol{if}\; y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\; \boldsymbol{then}$

$w = w + y^{(i)} x^{(i)}$

$\boldsymbol{return}\; w$

Perceptron Öğrenme Algoritması

$$\textbf{procedure } PERCEPTRON(\{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}, T)$$

$$w = 0 \ (vektör)$$

$$\textbf{for } t = 1, \dots, T \textbf{ do}$$

$$\textbf{for } i = 1, \dots, n \textbf{ do}$$

$$\textbf{if } y^{(i)}(w \cdot x^{(i)}) \leq 0 \textbf{ then}$$

$$w = w + y^{(i)} x^{(i)}$$

$$\textbf{return } w$$

Birbirinden farklı eğitim örnekleri parametreleri farklı yönlere güncelleyebileceğinden dolayı, sonraki güncellemeler bir önceki güncellemelerin bulduğu sınıflandırmaları ortadan kaldırabilir.

Perceptron Öğrenme Algoritması

$\boldsymbol{procedure}\ PERCEPTRON\left(\left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \dots, n\right\}, T\right)$

$w = 0\ (vektör)$

$\boldsymbol{for}\ t = 1, \dots, T\ \boldsymbol{do}$

$\boldsymbol{for}\ i = 1, \dots, n\ \boldsymbol{do}$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)} x^{(i)}$

$\boldsymbol{return}\ w$

Birbirinden farklı eğitim örnekleri parametreleri farklı yönlere güncelleyebileceğinden dolayı, sonraki güncellemeler bir önceki güncellemelerin bulduğu sınıflandırmaları ortadan kaldırabilir.

T parametresi bize eğitim setinde kaç defa tekrar ettiğimizi belirtir.

Perceptron Öğrenme Algoritması

$\boldsymbol{procedure}\ PERCEPTRON(\{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}, T)$

$\quad w = 0\ (vektör)$

$\quad \boldsymbol{for}\ t = 1, \dots, T\ \boldsymbol{do}$

$\qquad \boldsymbol{for}\ i = 1, \dots, n\ \boldsymbol{do}$

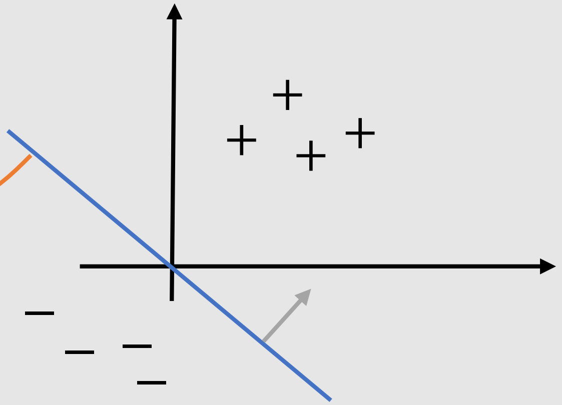
$\qquad\quad \boldsymbol{if}\ y^{(i)}(w \cdot x^{(i)}) \leq 0\ \boldsymbol{then}$

$\qquad\qquad w = w + y^{(i)}x^{(i)}$

$\quad \boldsymbol{return}\ w$

Birbirinden farklı eğitim örnekleri parametreleri farklı yönlere güncelleyebileceğinden dolayı, sonraki güncellemeler bir önceki güncellemelerin bulduğu sınıflandırmaları ortadan kaldırabilir.

T parametresi bize eğitim setinde kaç defa tekrar ettiğimizi belirtir.



Perceptron, bunu bir çözüm olarak bulabilir.

Perceptron Öğrenme Algoritması

epoch

$$\textbf{procedure}\ PERCEPTRON\left(\left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \ldots, n\right\}, T\right)$$

$$w = 0\ (vektör)$$

$$\textbf{for}\ t = 1, \ldots, T\ \textbf{do}$$

$$\textbf{for}\ i = 1, \ldots, n\ \textbf{do}$$

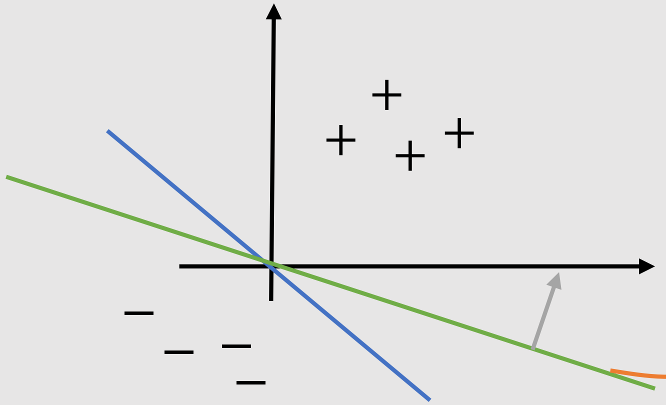
$$\textbf{if}\ y^{(i)}\left(w \cdot x^{(i)}\right) \leq 0\ \textbf{then}$$

$$w = w + y^{(i)}x^{(i)}$$

$$\textbf{return}\ w$$

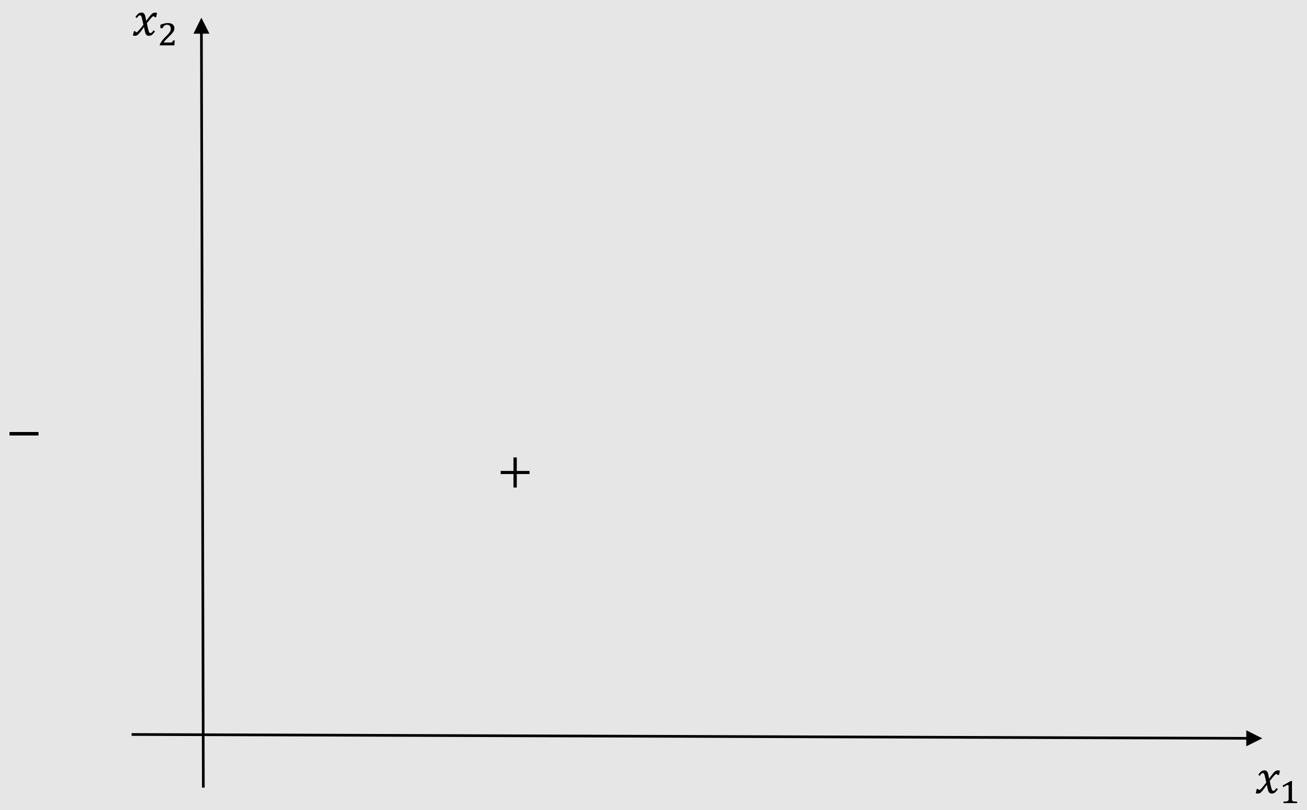
Birbirinden farklı eğitim örnekleri parametreleri farklı yönlere güncelleyebileceğinden dolayı, sonraki güncellemeler bir önceki güncellemelerin bulduğu sınıflandırmaları ortadan kaldırabilir.

T parametresi bize eğitim setinde kaç defa tekrar ettiğimizi belirtir.

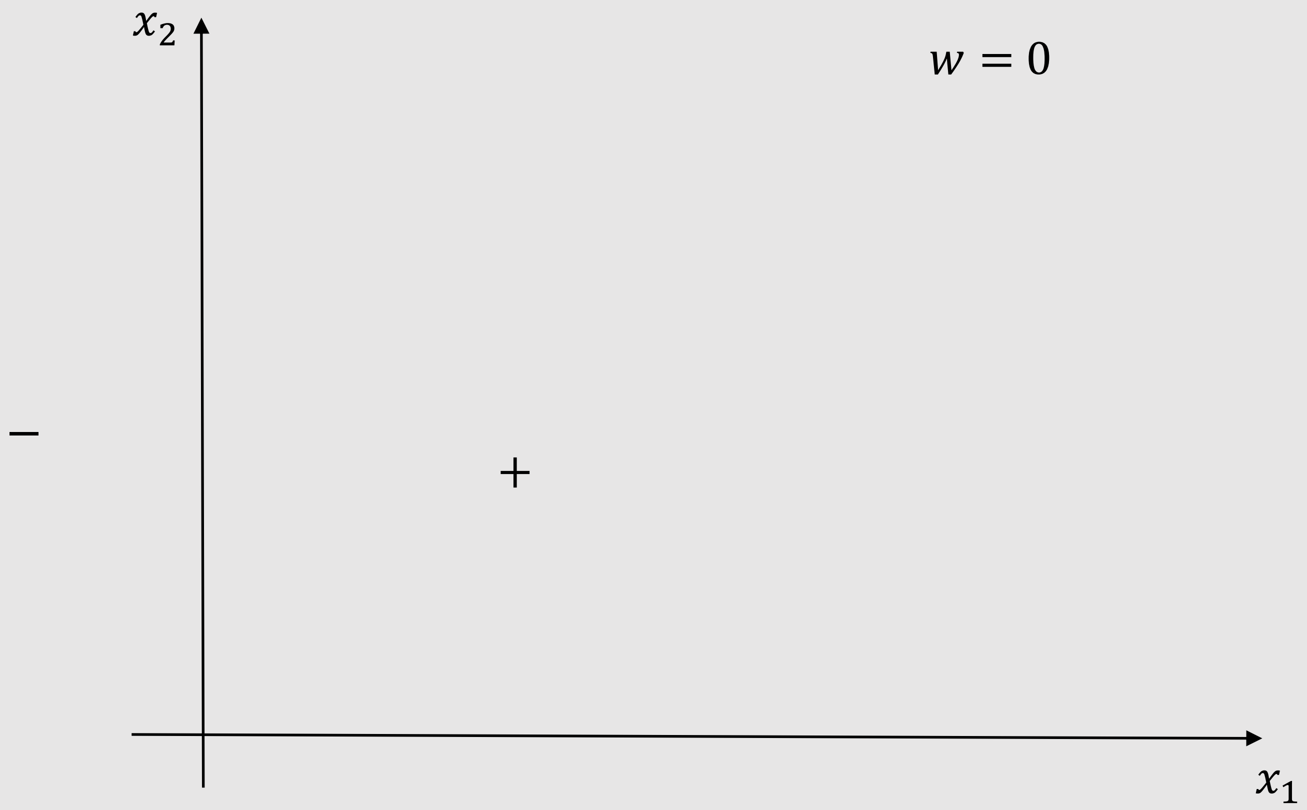


Bunu da bir çözüm olarak bulabilir.

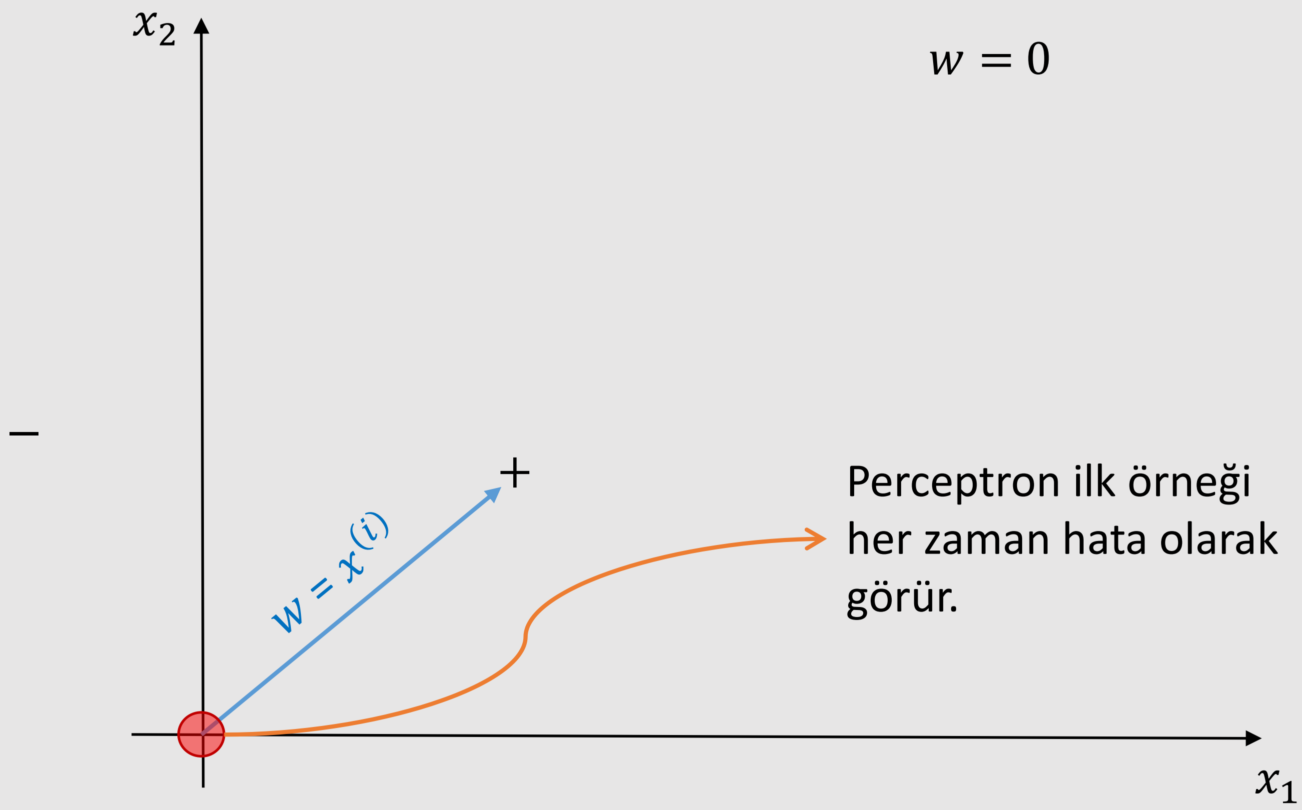
Perceptron Algoritması : Geometrik Örnek



Perceptron Algoritması : Geometrik Örnek



Perceptron Algoritması : Geometrik Örnek

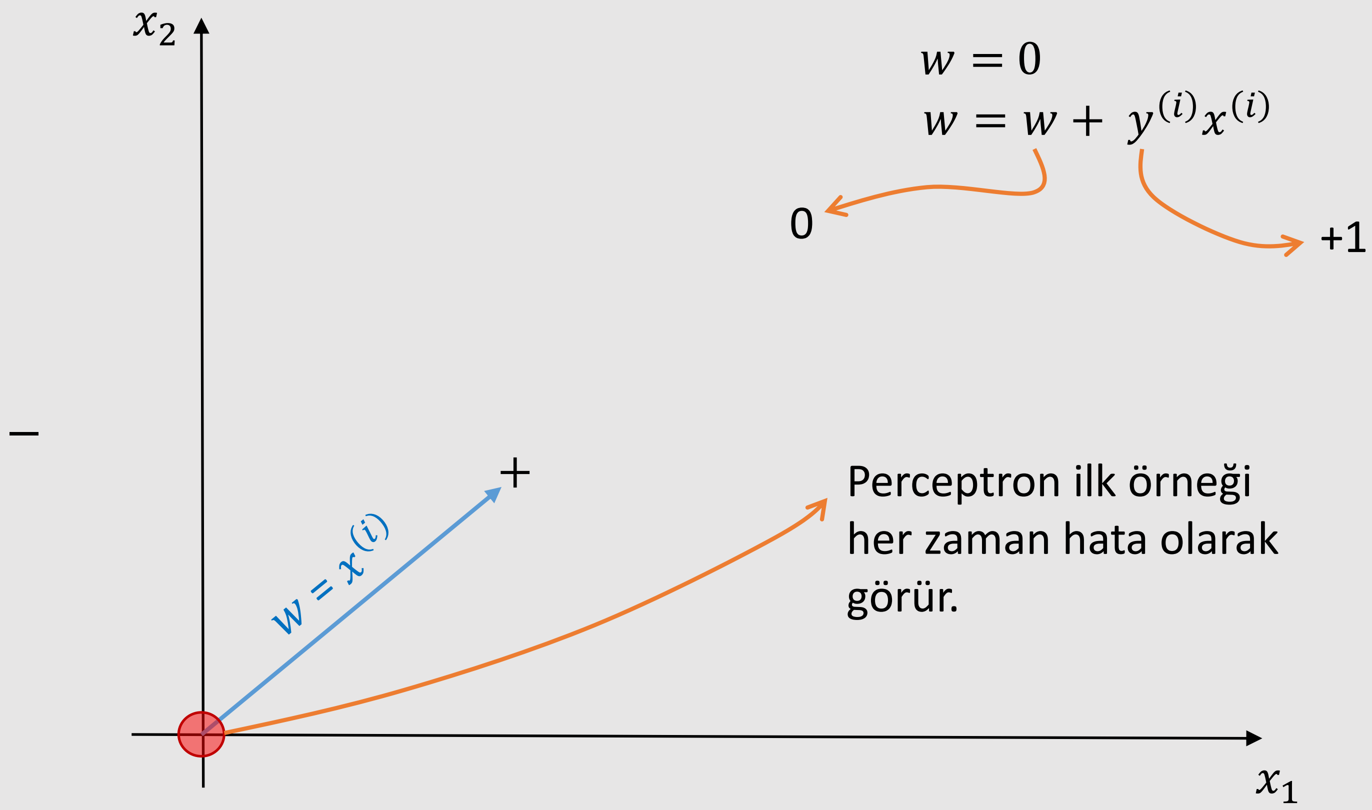


$w = 0$

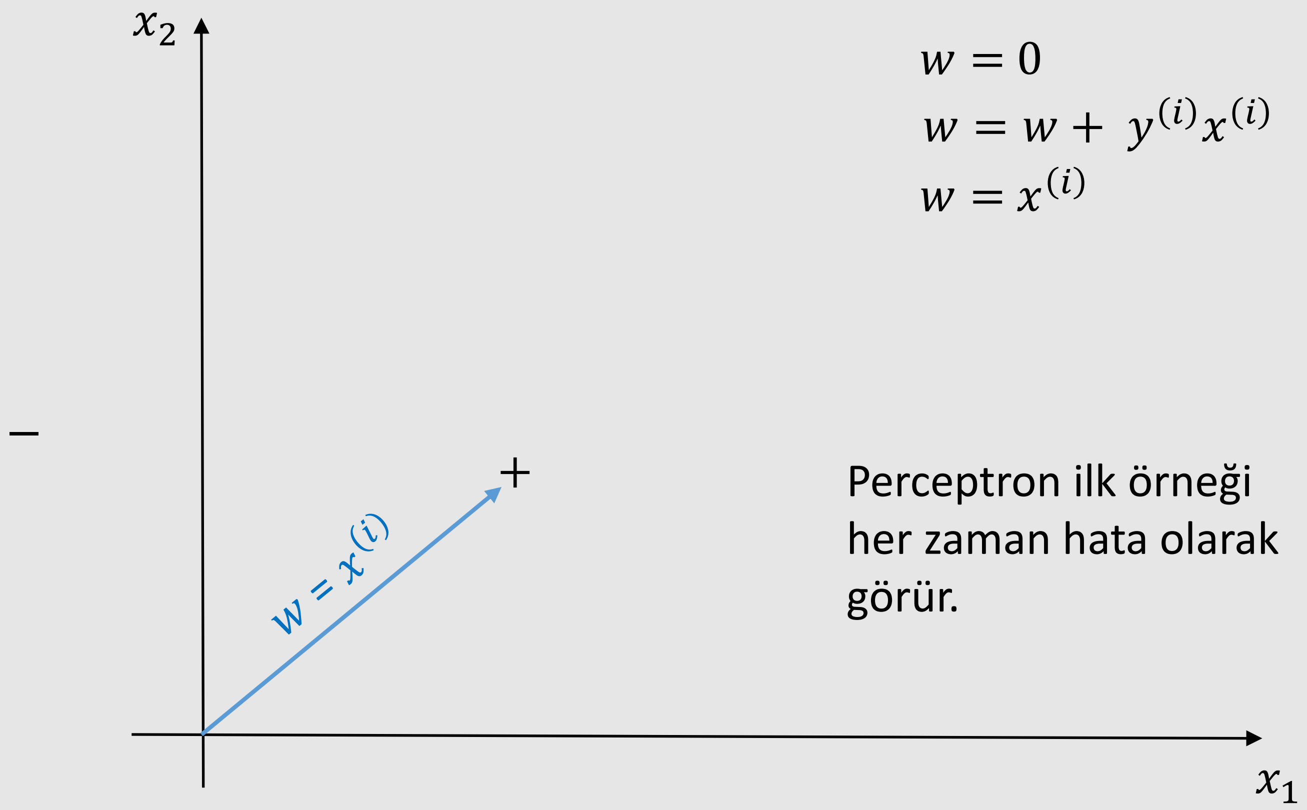
Perceptron ilk örneği her zaman hata olarak görür.

Perceptron Algoritması : Geometrik Örnek

$$w = 0$$
$$w = w + y^{(i)} x^{(i)}$$



Perceptron Algoritması : Geometrik Örnek



$$w = 0$$
$$w = w + y^{(i)} x^{(i)}$$
$$w = x^{(i)}$$

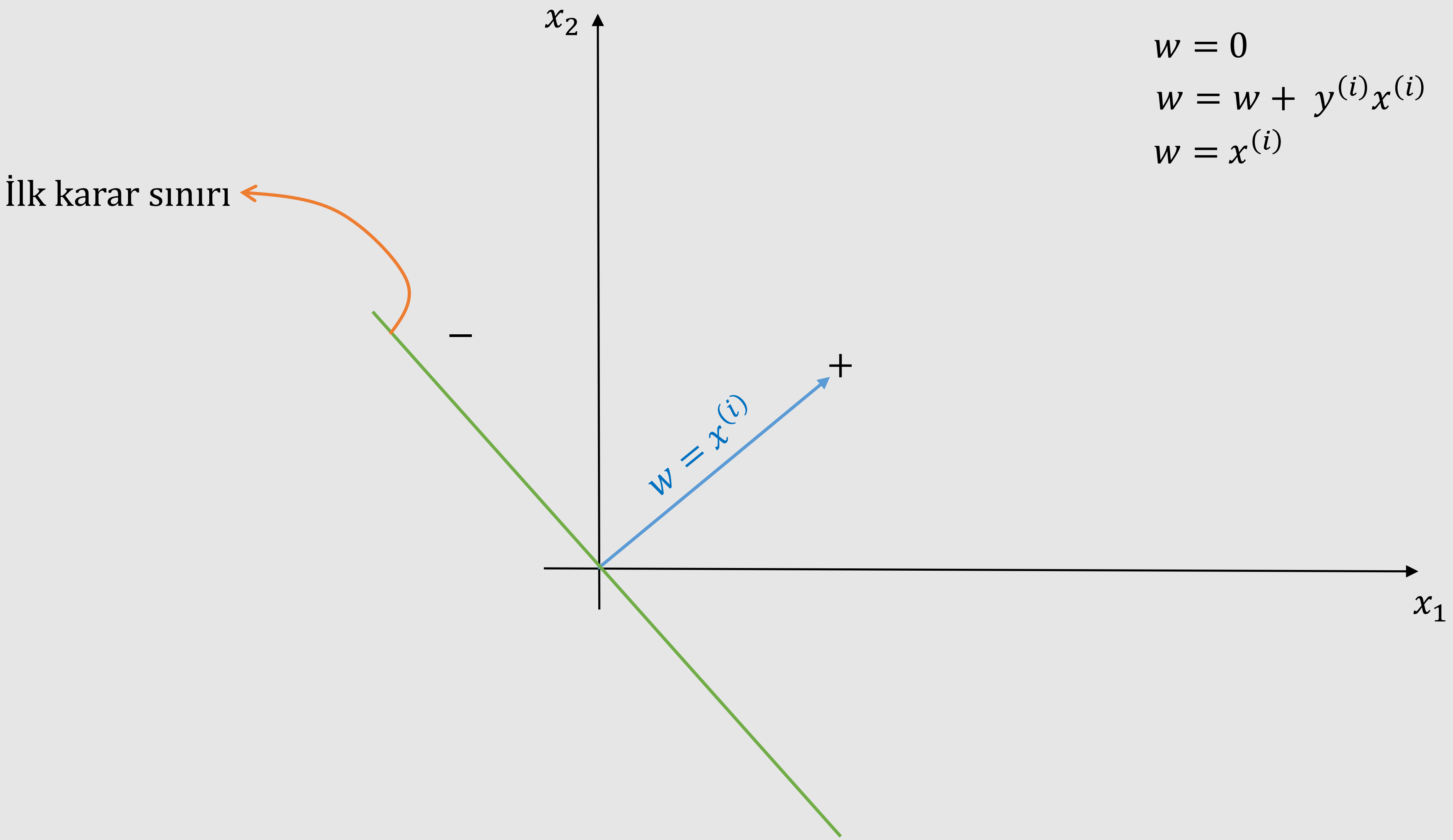
Perceptron ilk örneği her zaman hata olarak görür.

Perceptron Algoritması : Geometrik Örnek

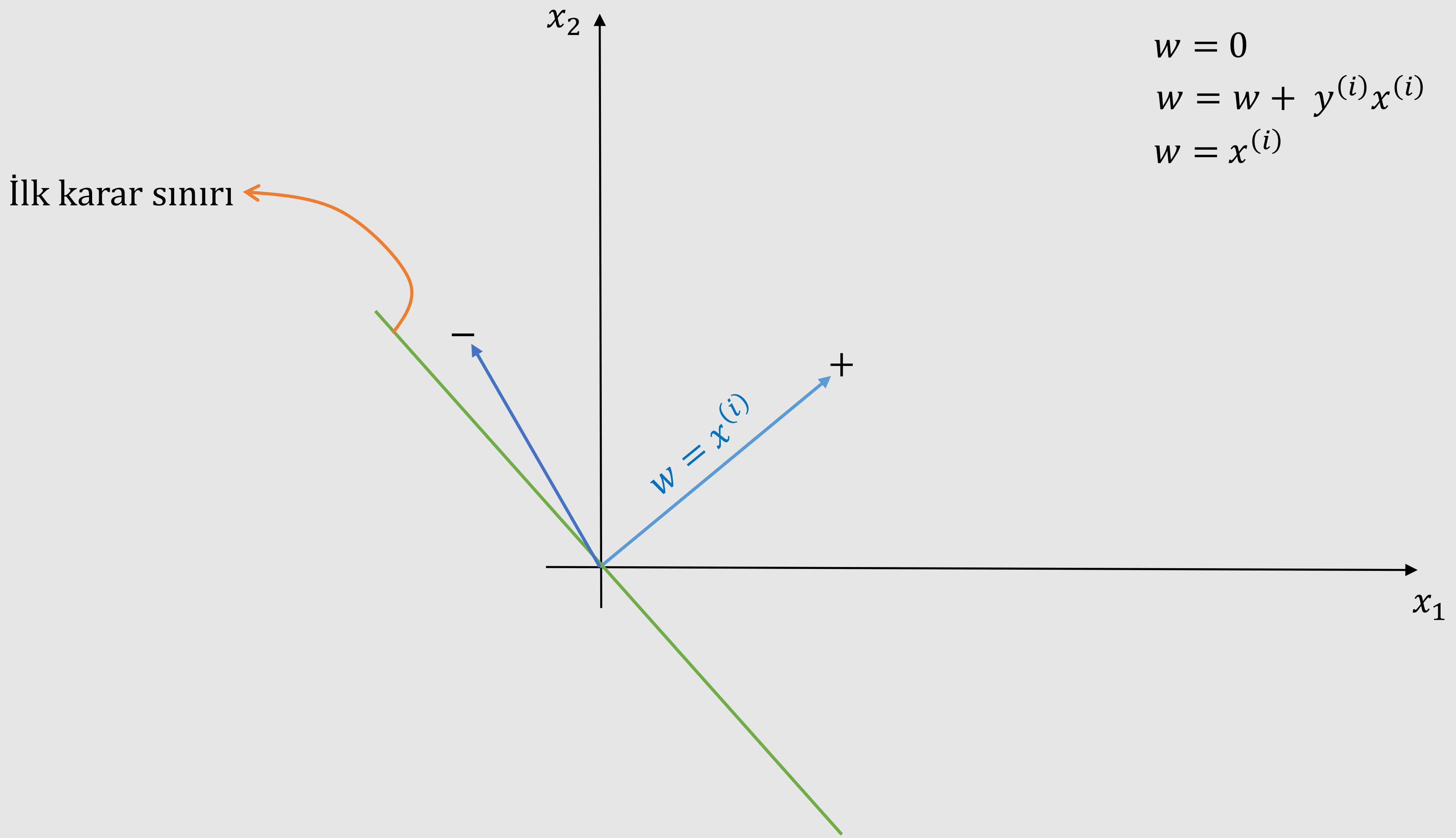
$w = 0$

$w = w + \ y^{(i)} x^{(i)}$

$w = x^{(i)}$



Perceptron Algoritması : Geometrik Örnek

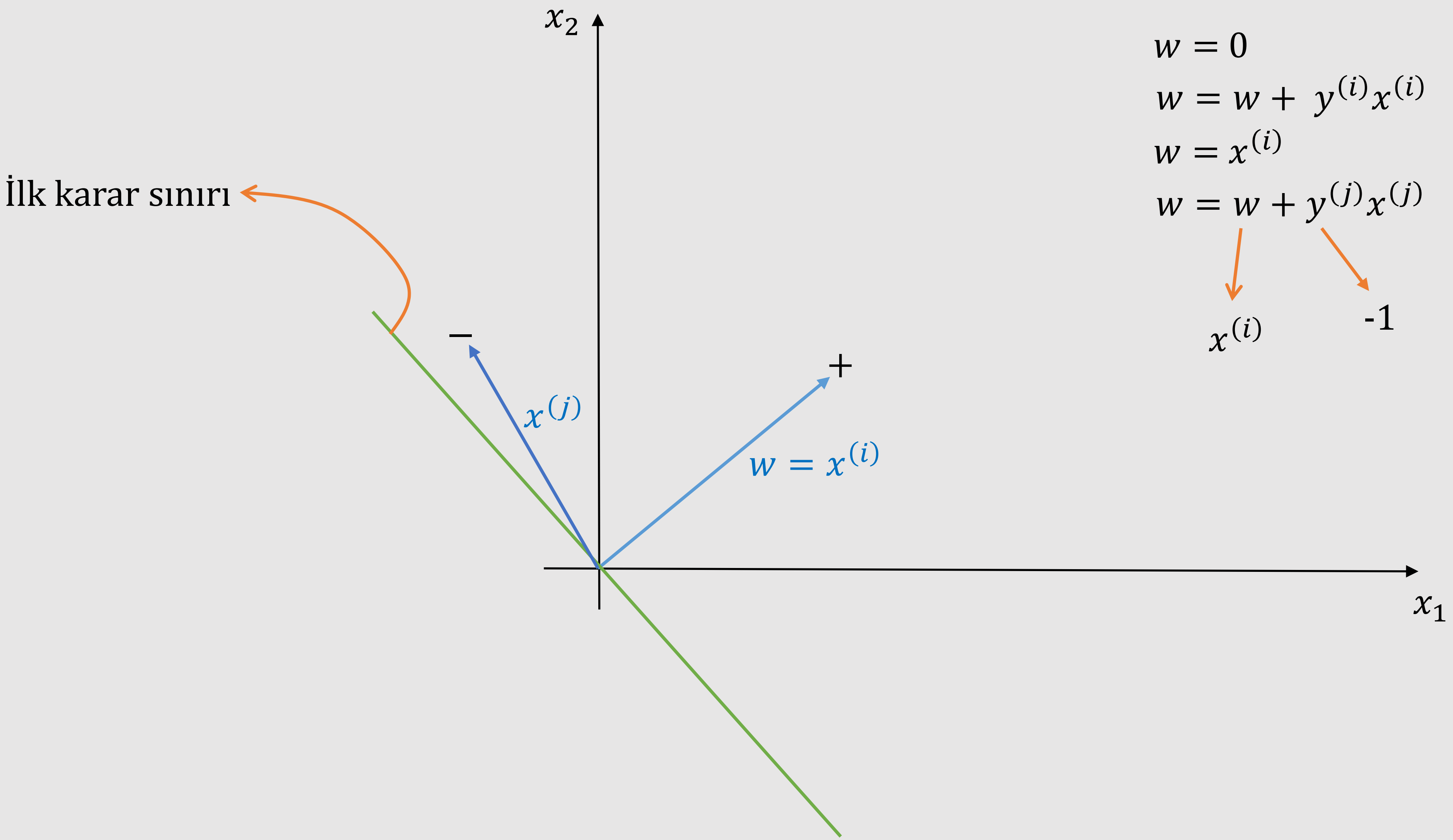


$w = 0$

$w = w + \ y^{(i)} x^{(i)}$

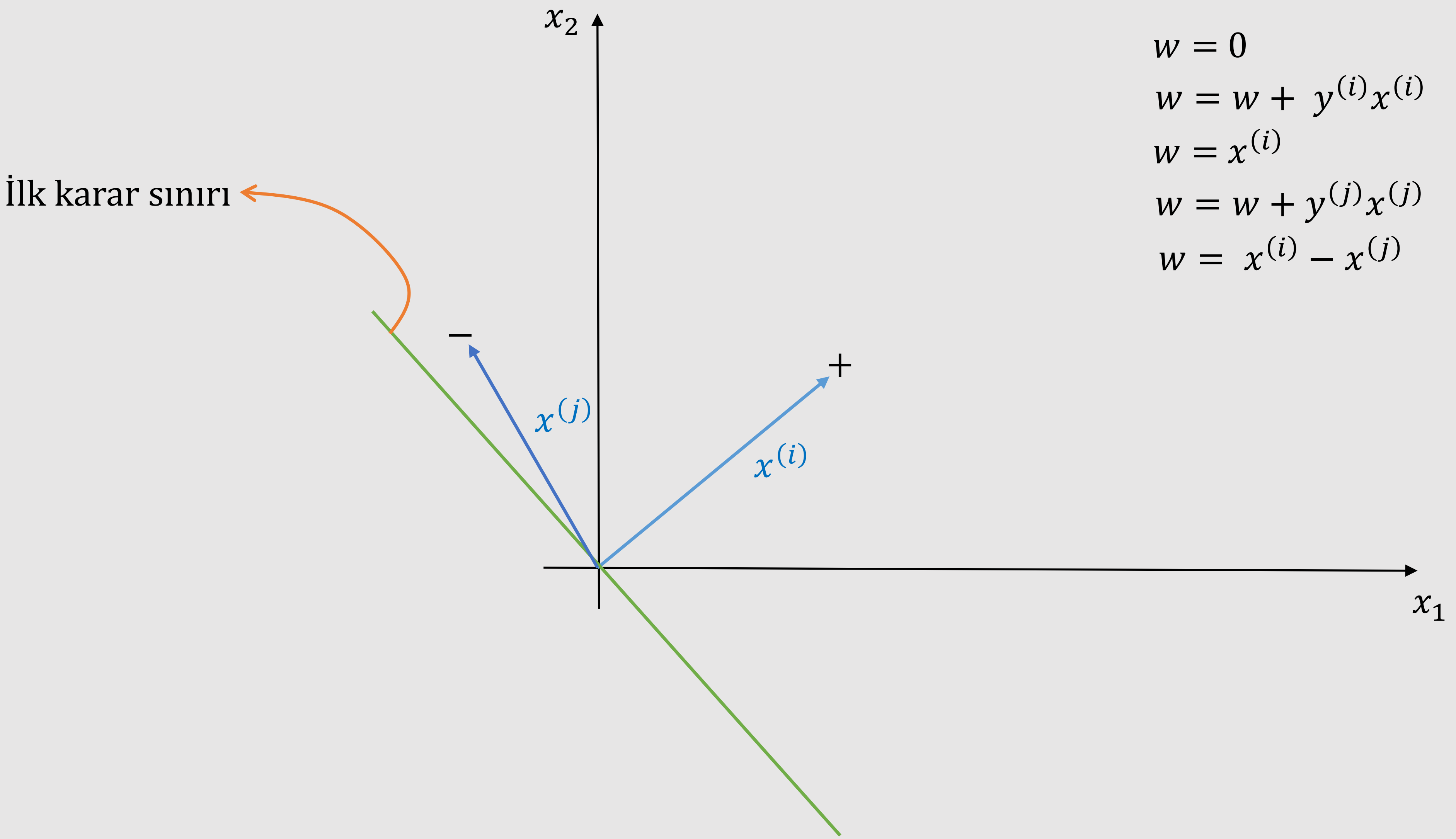
$w = x^{(i)}$

Perceptron Algoritması : Geometrik Örnek



$$w = 0$$
$$w = w + \ y^{(i)} x^{(i)}$$
$$w = x^{(i)}$$
$$w = \underbrace{w}_{x^{(i)}} + \underbrace{y^{(j)}}_{-1} x^{(j)}$$

Perceptron Algoritması : Geometrik Örnek



$w = 0$

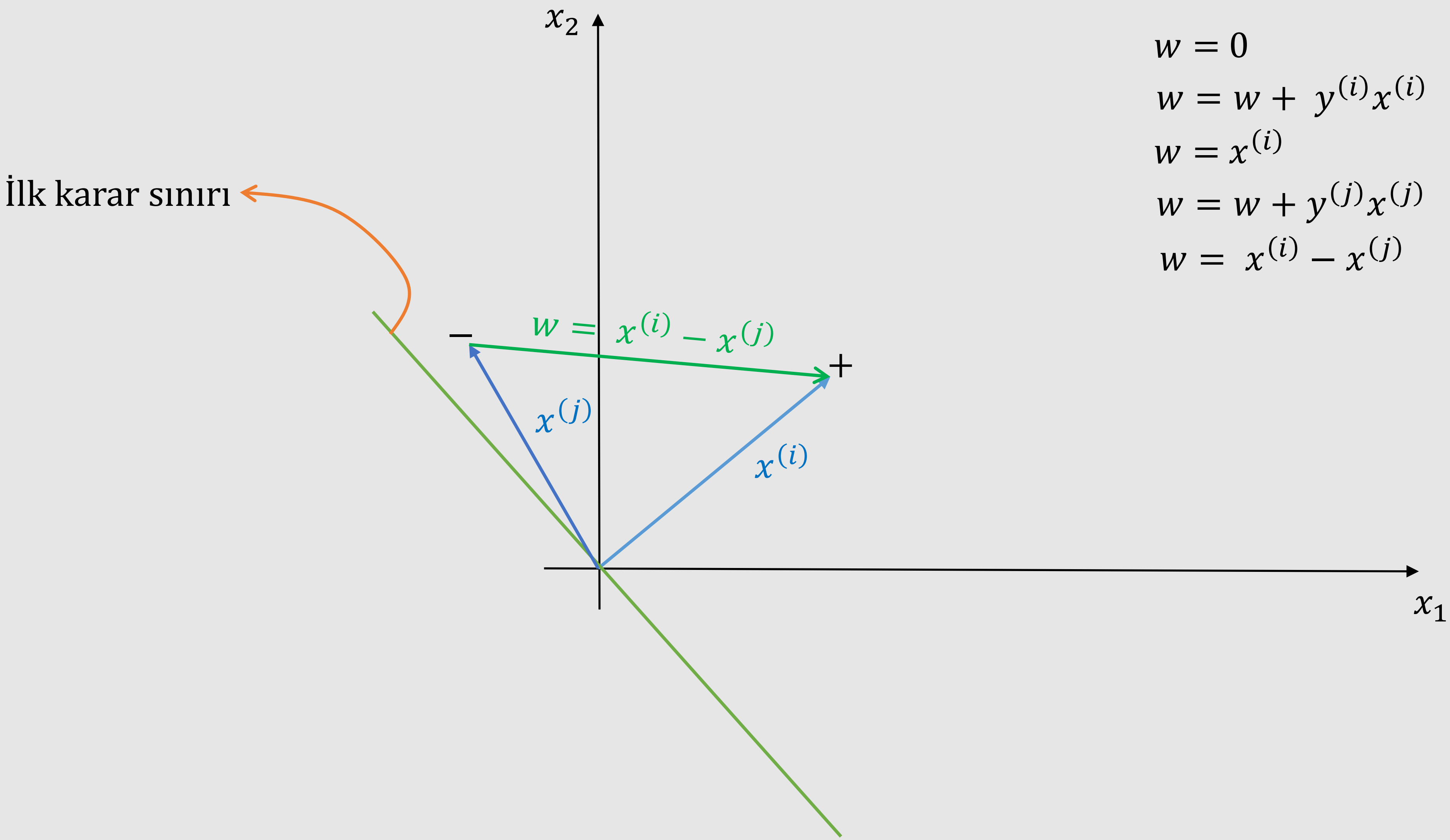
$w = w + \ y^{(i)} x^{(i)}$

$w = x^{(i)}$

$w = w + y^{(j)} x^{(j)}$

$w = \ x^{(i)} - x^{(j)}$

Perceptron Algoritması : Geometrik Örnek



$w = 0$

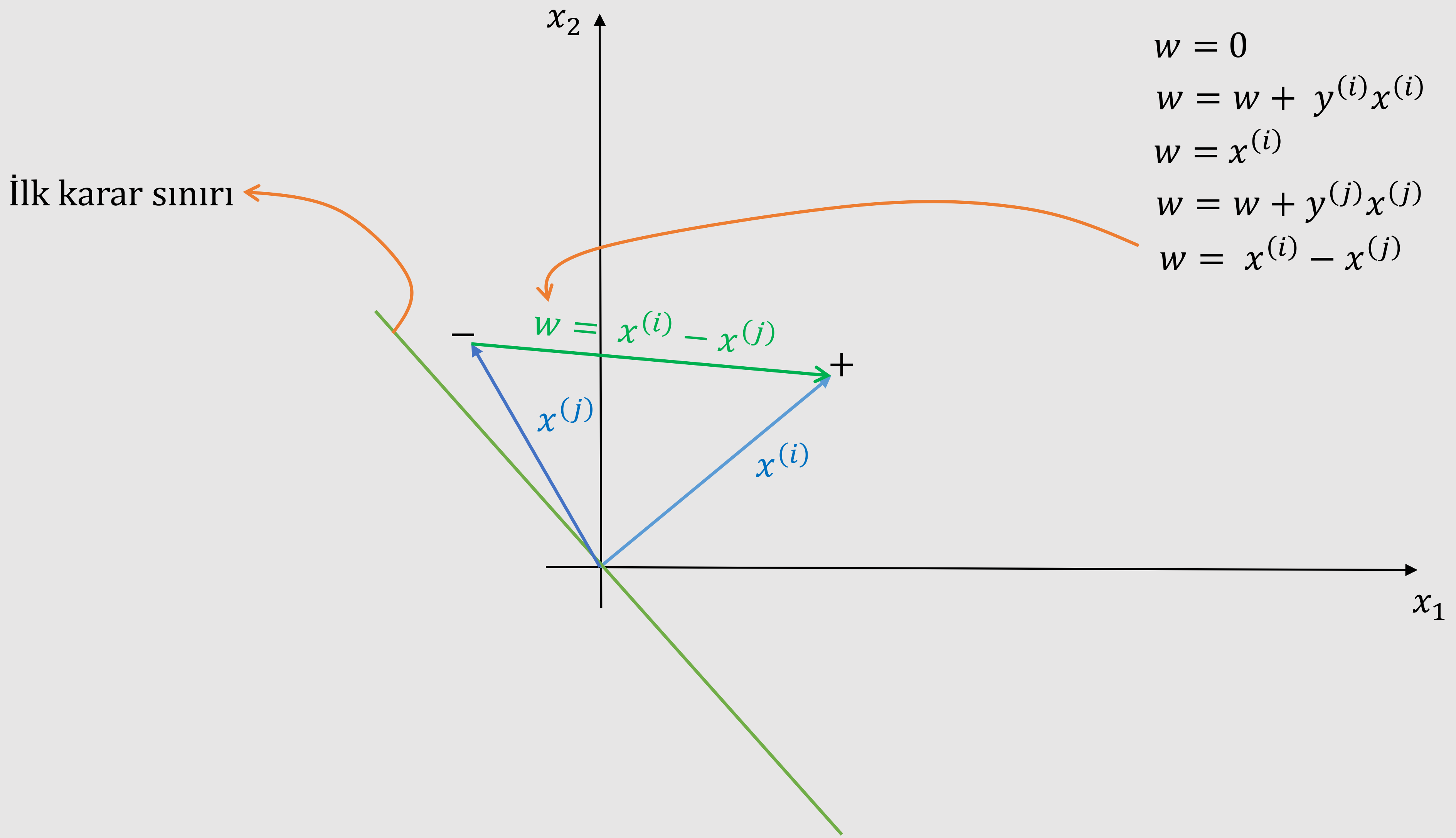
$w = w + y^{(i)} x^{(i)}$

$w = x^{(i)}$

$w = w + y^{(j)} x^{(j)}$

$w = x^{(i)} - x^{(j)}$

Perceptron Algoritması : Geometrik Örnek



$w = 0$

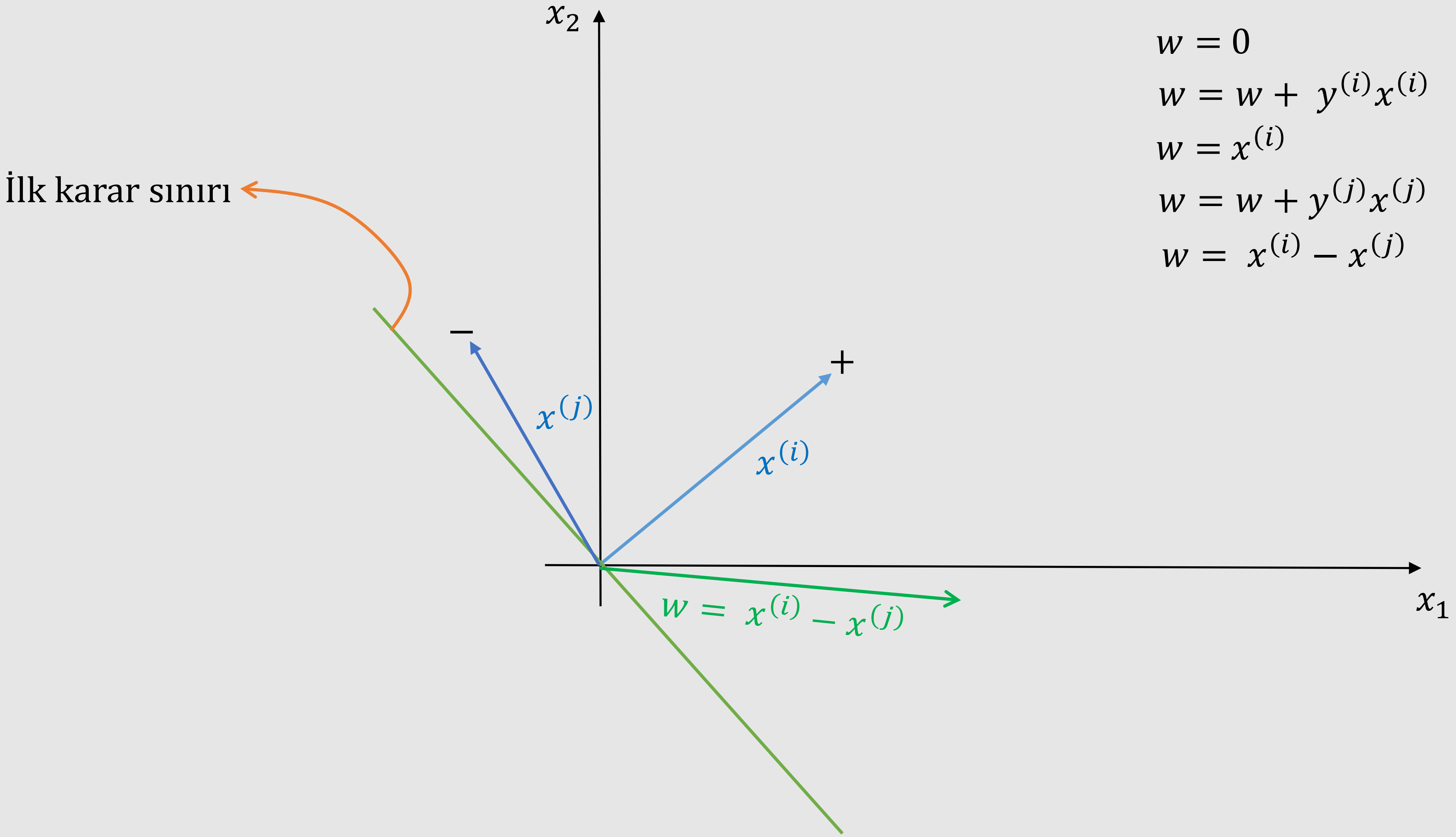
$w = w + y^{(i)} x^{(i)}$

$w = x^{(i)}$

$w = w + y^{(j)} x^{(j)}$

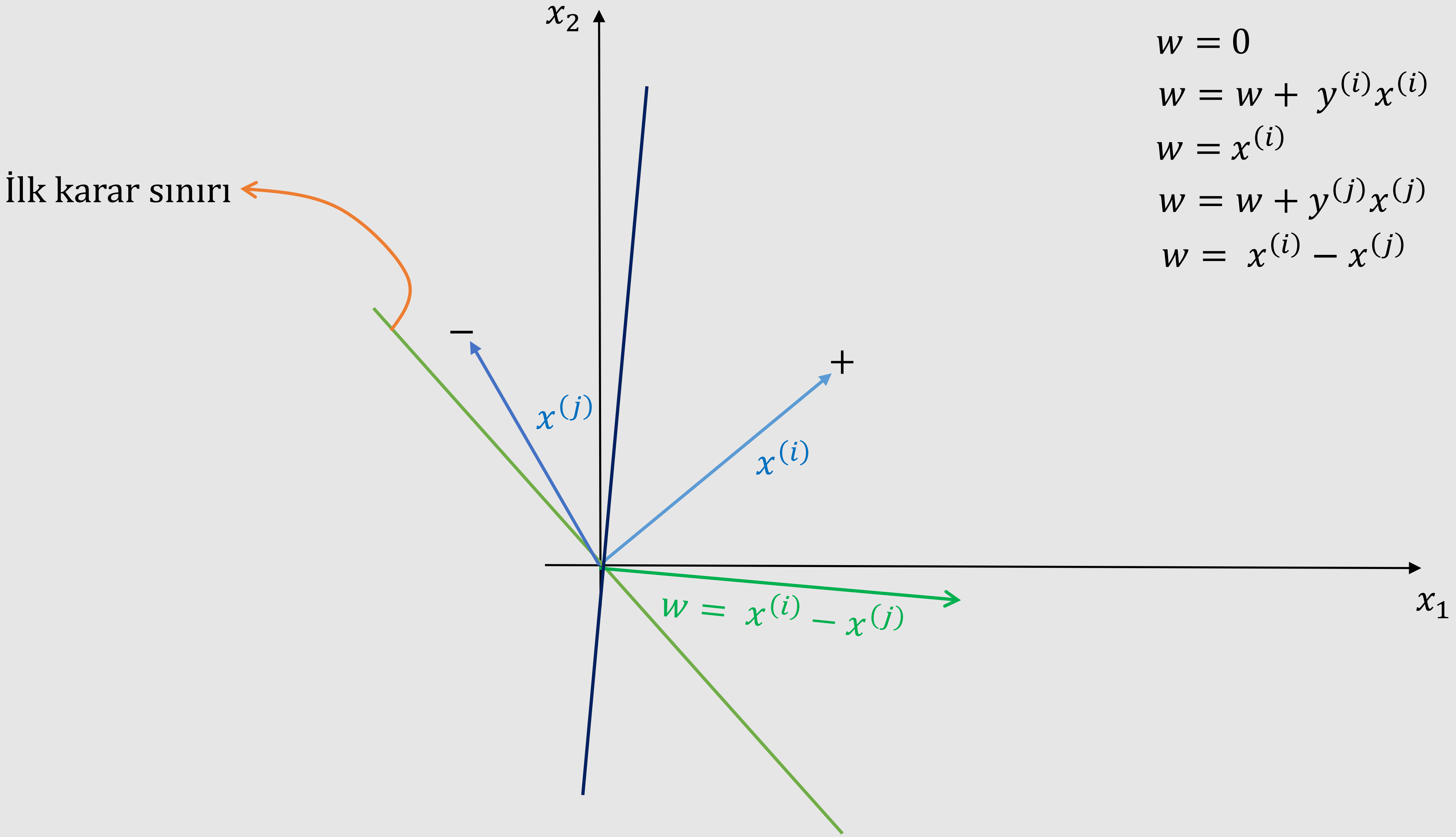
$w = x^{(i)} - x^{(j)}$

Perceptron Algoritması : Geometrik Örnek



$$w = 0$$
$$w = w + y^{(i)}x^{(i)}$$
$$w = x^{(i)}$$
$$w = w + y^{(j)}x^{(j)}$$
$$w = x^{(i)} - x^{(j)}$$

Perceptron Algoritması : Geometrik Örnek



$w = 0$

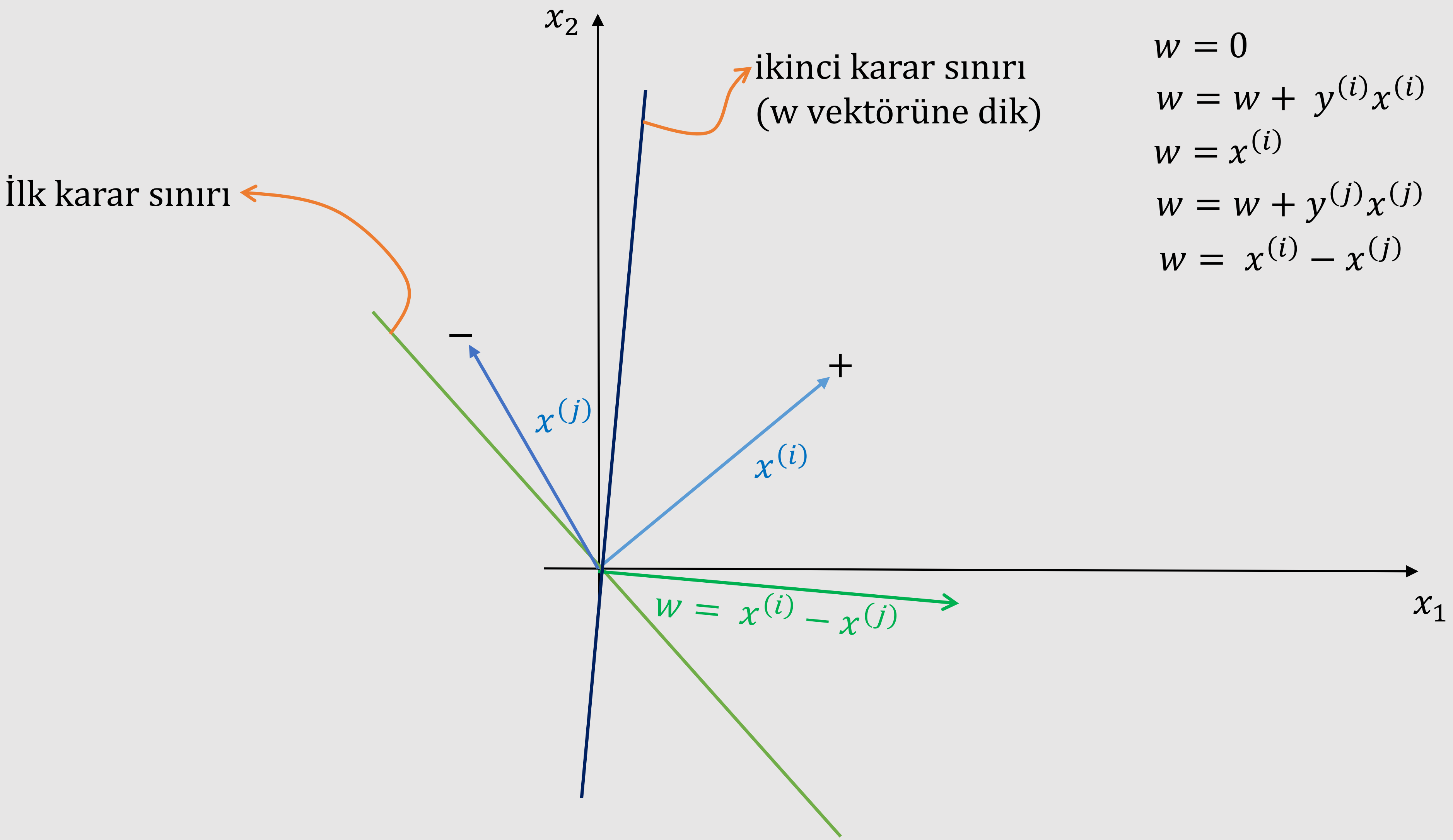
$w = w + y^{(i)} x^{(i)}$

$w = x^{(i)}$

$w = w + y^{(j)} x^{(j)}$

$w = x^{(i)} - x^{(j)}$

Perceptron Algoritması : Geometrik Örnek



$w = 0$

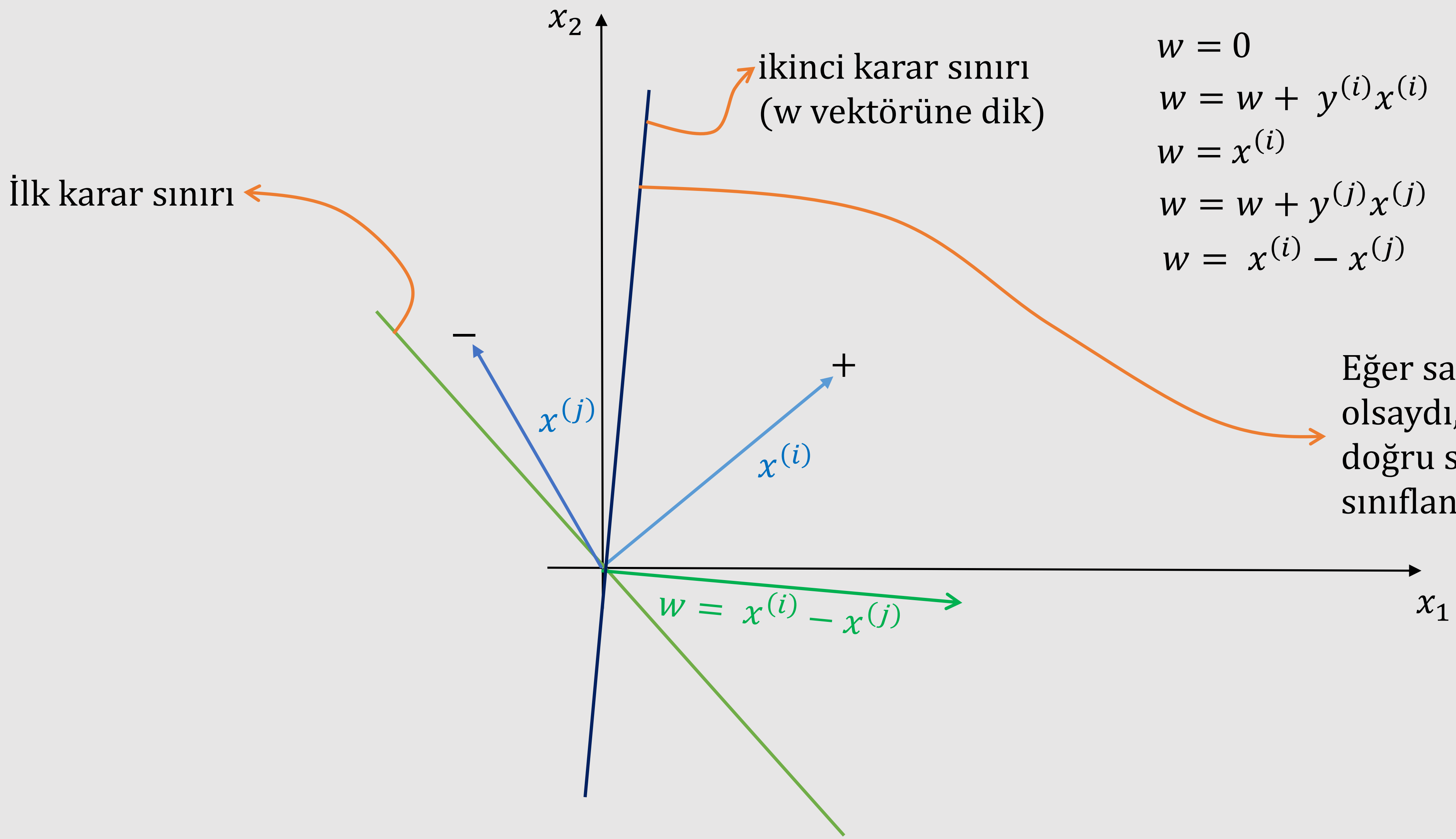
$w = w + y^{(i)} x^{(i)}$

$w = x^{(i)}$

$w = w + y^{(j)} x^{(j)}$

$w = x^{(i)} - x^{(j)}$

Perceptron Algoritması : Geometrik Örnek



$$w = 0$$
$$w = w + y^{(i)}x^{(i)}$$
$$w = x^{(i)}$$
$$w = w + y^{(j)}x^{(j)}$$
$$w = x^{(i)} - x^{(j)}$$

Eğer sadece bu iki nokta olsaydı, bu algoritma bize doğru sonucu yani doğru sınıflandırmayı verecekti.

Perceptron Fonksiyonu

$$f(x) = \begin{cases} 1, & w \cdot x + b > 0 \\ 0, & diğer \end{cases}$$

Perceptron Öğrenme Algoritması (Offset ile)

$\boldsymbol{procedure}\ PERCEPTRON\left(\left\{\left(x^{(i)}, y^{(i)}\right), i = 1, \dots, n\right\}, T\right)$

$w = 0\ (vekt\ddot{o}r)$

$\boldsymbol{for}\ t = 1, \dots T\ \ \boldsymbol{do}$

$\boldsymbol{for}\ i = 1, \dots, n\ \boldsymbol{do}$

$\boldsymbol{if}\ y^{(i)}\left(w \cdot x^{(i)} + b\right) \leq 0\ \boldsymbol{then}$

$w = w + y^{(i)} x^{(i)}$

$b = b + y^{(i)}$

$\boldsymbol{return}\ w, b$

Perceptron Öğrenme Algoritması (Offset ile)

$$\textbf{\textit{procedure}}\ PERCEPTRON(\{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}, T)$$

$$w = 0\ (vektör)$$

$$\textbf{\textit{for}}\ t = 1, \dots T\ \ \textbf{\textit{do}}$$

$$\textbf{\textit{for}}\ i = 1, \dots, n\ \textbf{\textit{do}}$$

$$\textbf{\textit{if}}\ y^{(i)}(w \cdot x^{(i)} + b) \le 0\ \textbf{\textit{then}}$$

$$w = w + y^{(i)} x^{(i)}$$

$$b = b + y^{(i)}$$

$$\textbf{\textit{return}}\ w, b$$

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} + y^{(i)} \begin{bmatrix} x^{(i)} \\ 1 \end{bmatrix}$$

Perceptron Öğrenme Algoritması (Offset ile)

$\textbf{procedure}\ PERCEPTRON(\{(x^{(i)}, y^{(i)}), i = 1, \dots, n\}, T)$

$w = 0\ (vektör)$

$\textbf{for}\ t = 1, \dots T\ \ \textbf{do}$

$\textbf{for}\ i = 1, \dots, n\ \textbf{do}$

$\textbf{if}\ y^{(i)}(w \cdot x^{(i)} + b) \leq 0\ \textbf{then}$

$w = w + y^{(i)} x^{(i)}$

$b = b + y^{(i)}$

$\textbf{return}\ w, b$

$$\begin{bmatrix} w \\ b \end{bmatrix} = \begin{bmatrix} w \\ b \end{bmatrix} + y^{(i)} \begin{bmatrix} x^{(i)} \\ 1 \end{bmatrix}$$

$$\boxed{w \cdot x + b = \begin{bmatrix} w \\ b \end{bmatrix} \cdot \begin{bmatrix} x \\ 1 \end{bmatrix}}$$

Perceptron Öğrenme Algoritması (Gradient Descent)

$\textbf{\textit{procedure}}\ PERCEPTRON(\{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}, \alpha = 0.01, epochs = 10)$

$w = 0\ (vektör)$

$\textbf{\textit{for}}\ t = 1, \ldots epochs\ \ \textbf{\textit{do}}$

$\textbf{\textit{for}}\ i = 1, \ldots, n\ \textbf{\textit{do}}$

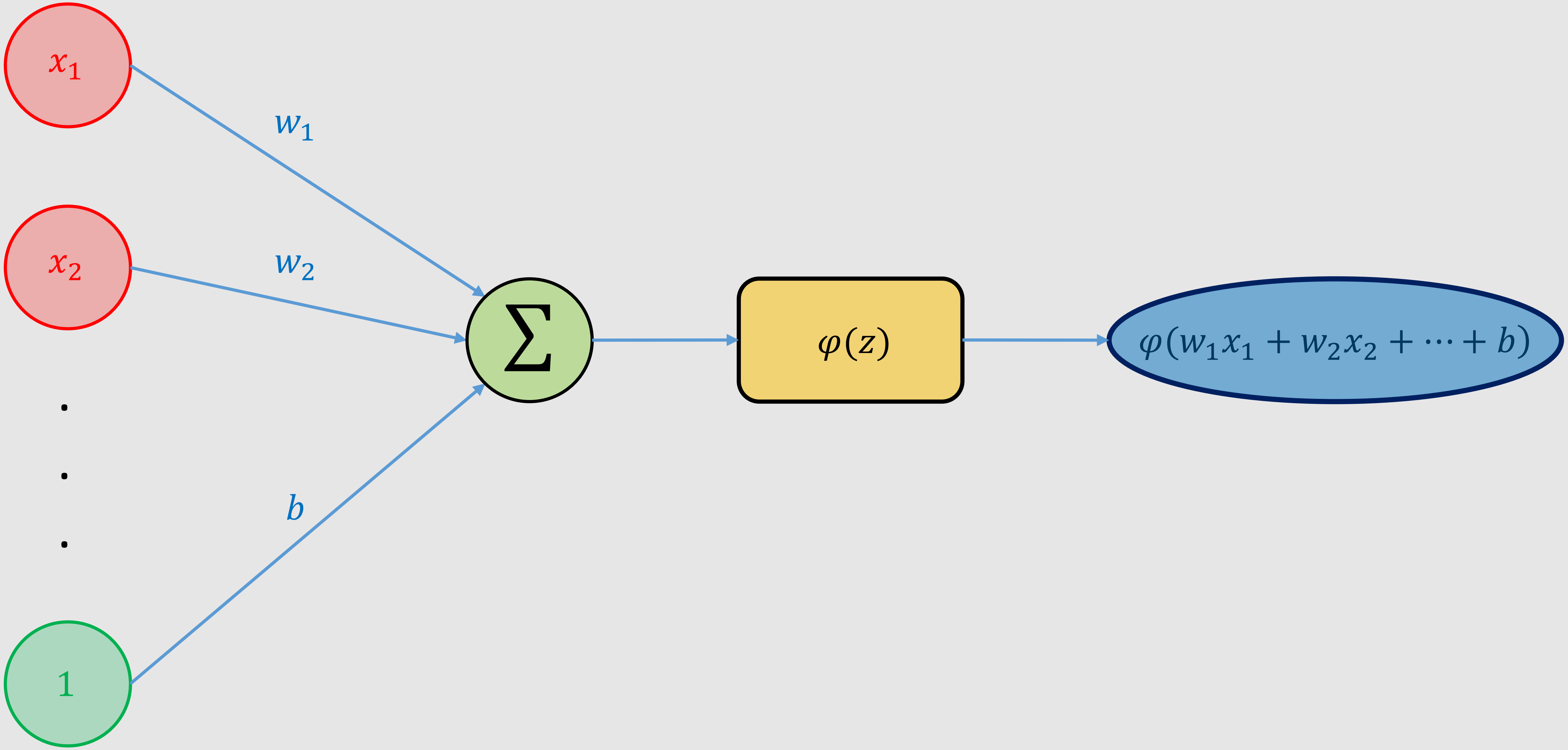
$\textbf{\textit{if}}\ y^{(i)}(w \cdot x^{(i)} + b) \leq 0\ \textbf{\textit{then}}$

$$w = w - \frac{1}{n} * \alpha \sum_{i}^{n} y^{(i)}(w \cdot x^{(i)} + b) \cdot x^{(i)}$$

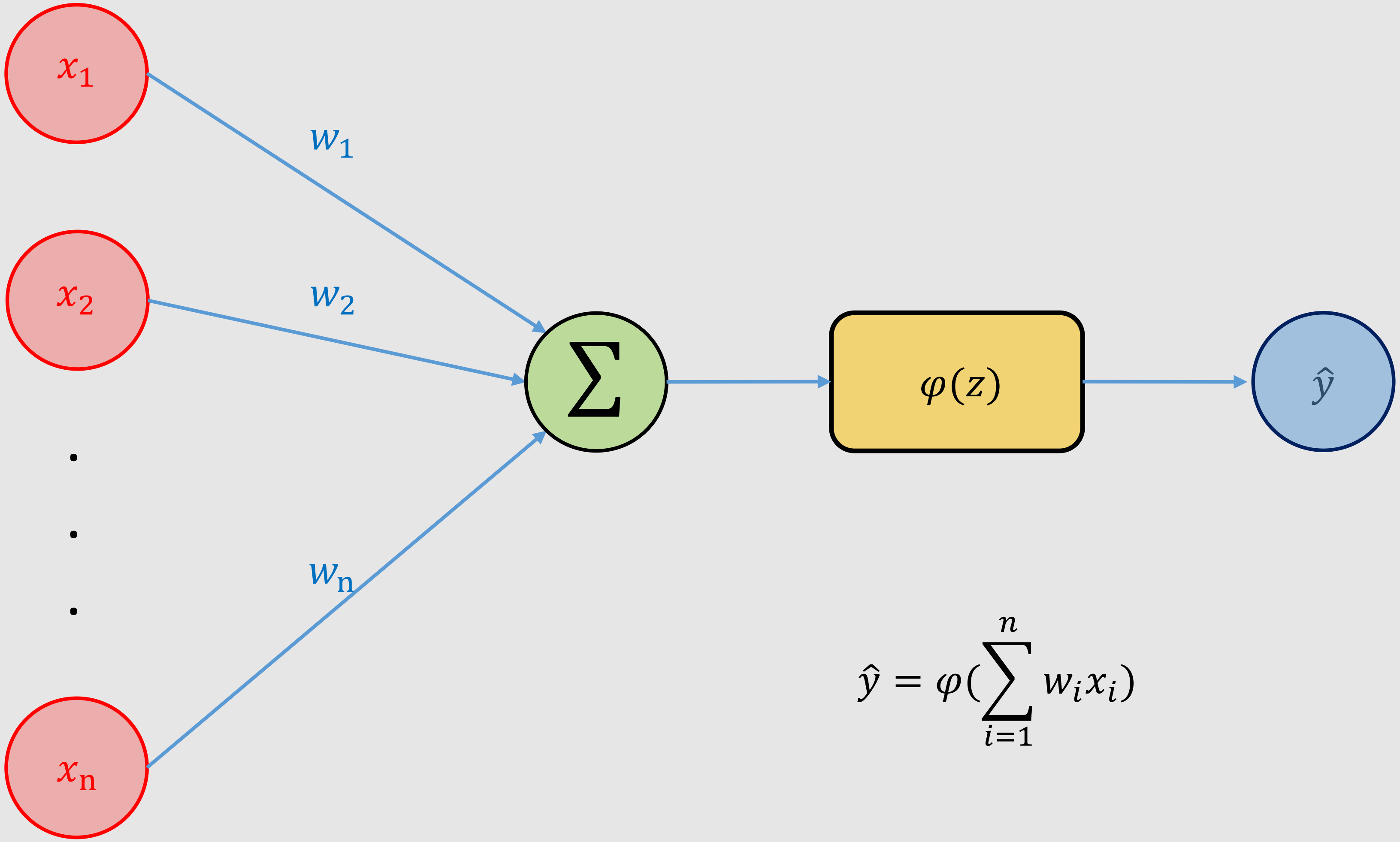
$$b = b - \frac{1}{n} * \alpha \sum_{i}^{n} y^{(i)}(w \cdot x^{(i)} + b)$$

$\textbf{\textit{return}}\ w, b$

Perceptron : YSA ile gösterilişi

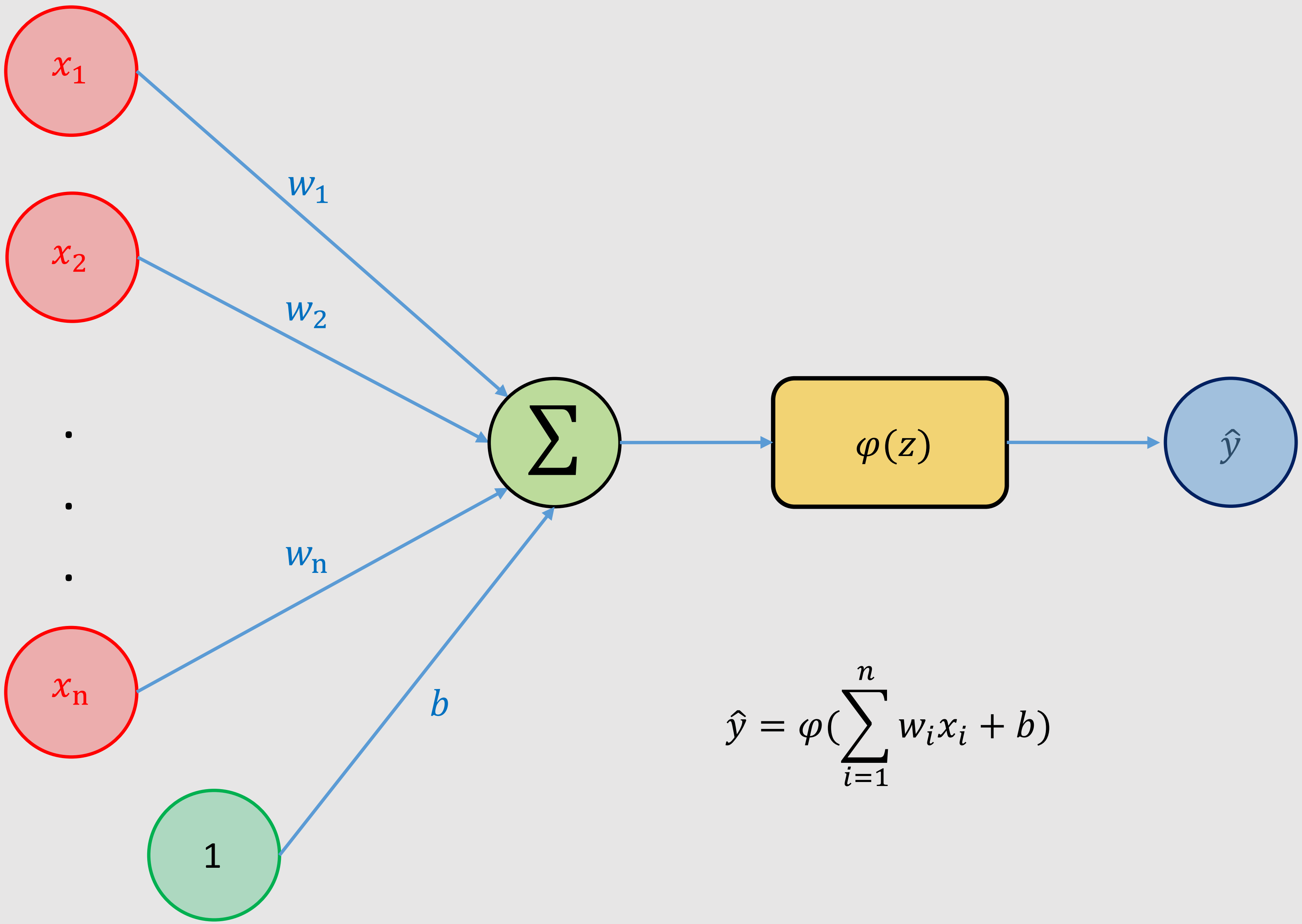


Perceptron : Forward Propagation



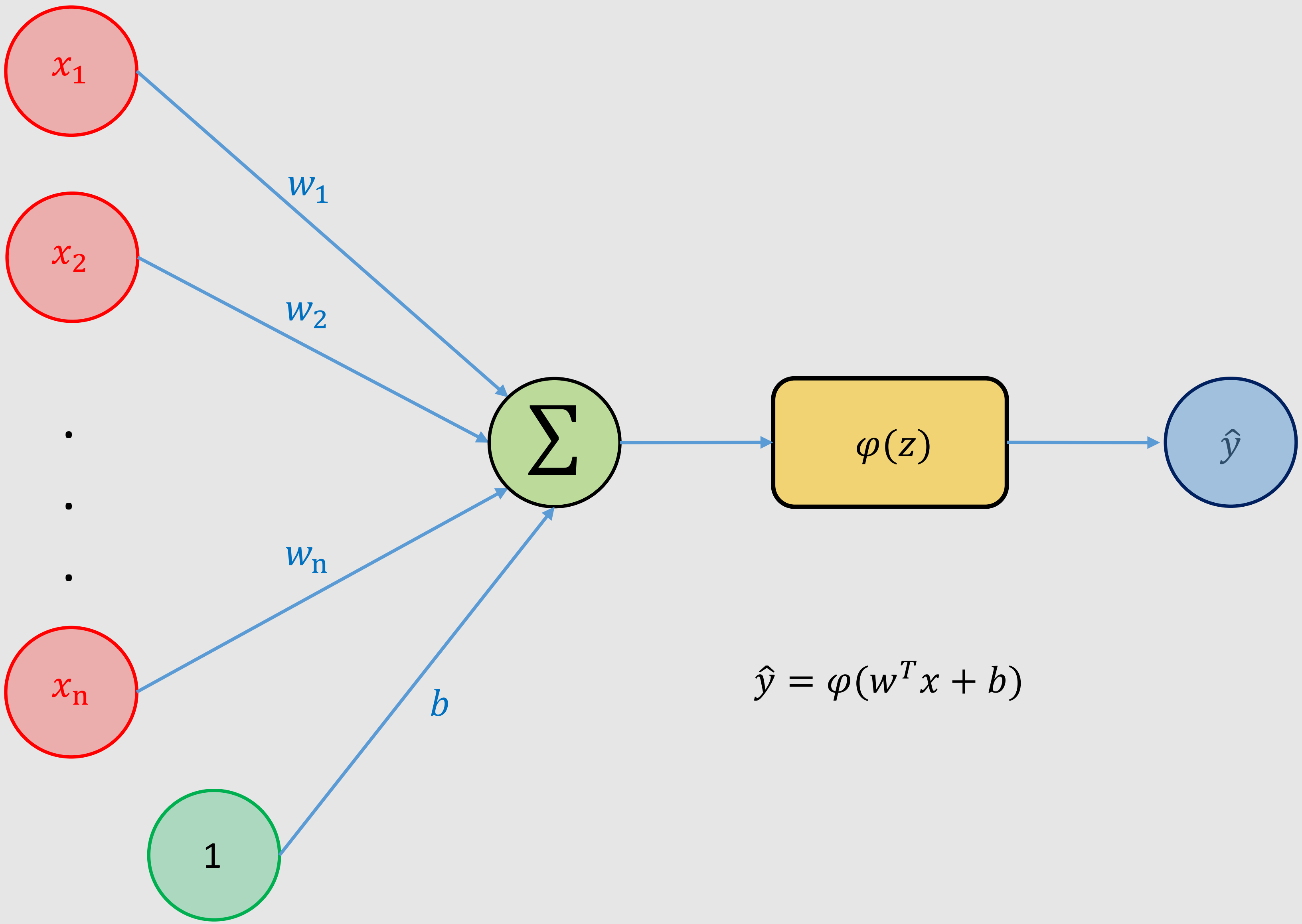
$$\hat{y} = \varphi(\sum_{i=1}^{n} w_i x_i)$$

Perceptron : Forward Propagation



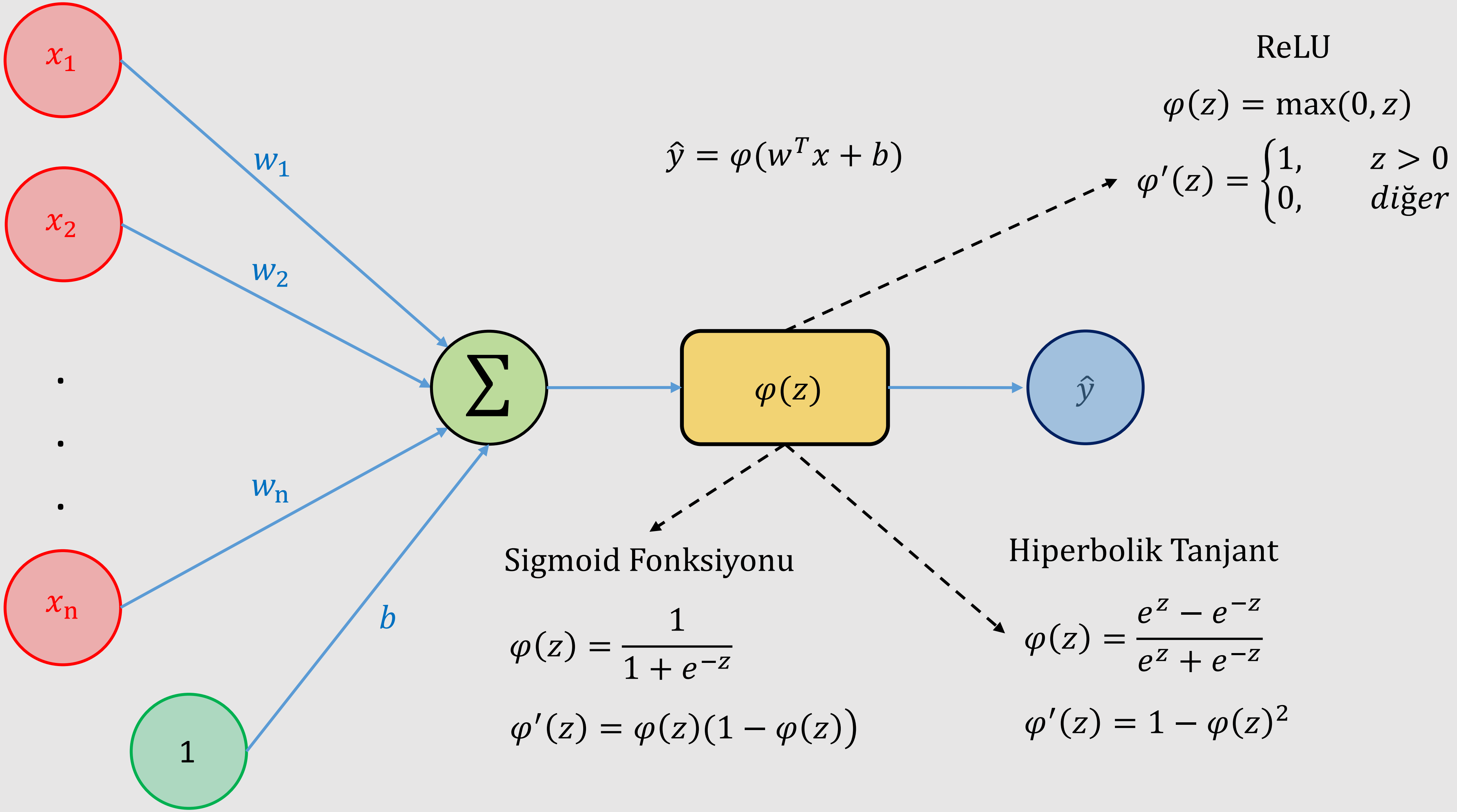
$$\hat{y} = \varphi(\sum_{i=1}^{n} w_i x_i + b)$$

Perceptron : Forward Propagation

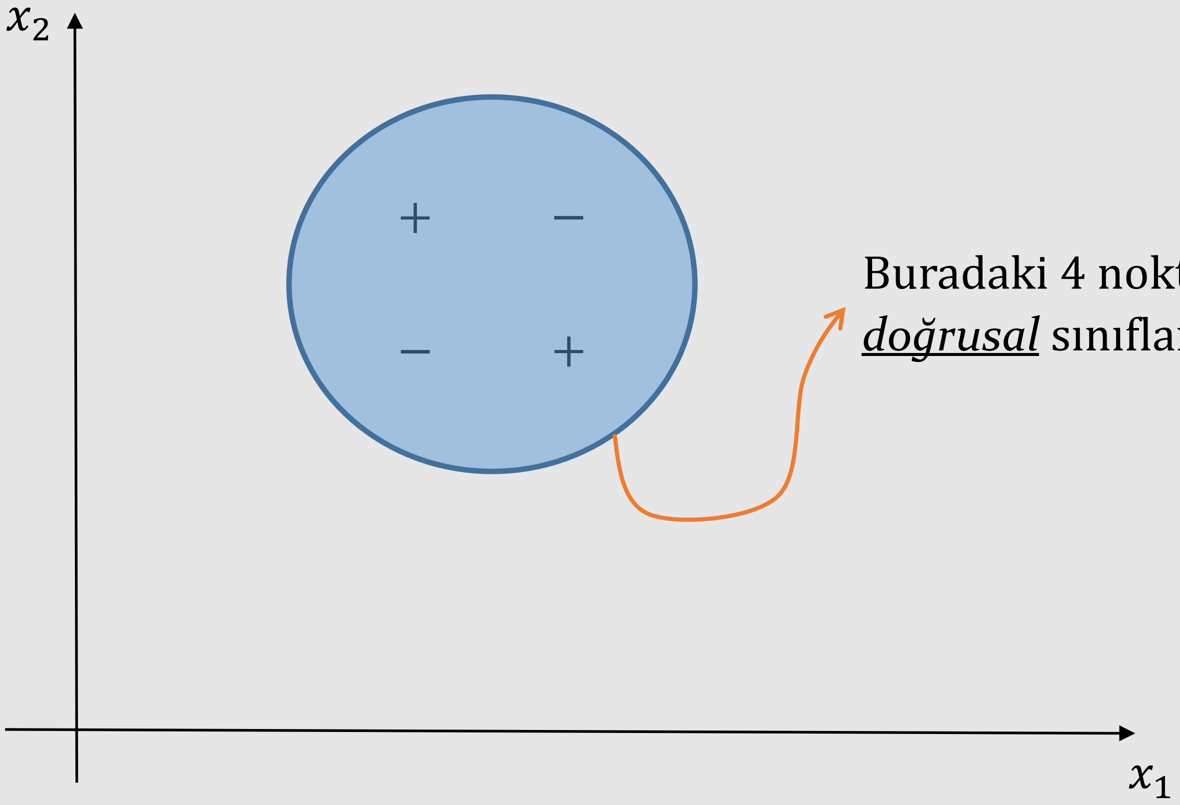


$$\hat{y} = \varphi(w^T x + b)$$

Perceptron : Forward Propagation

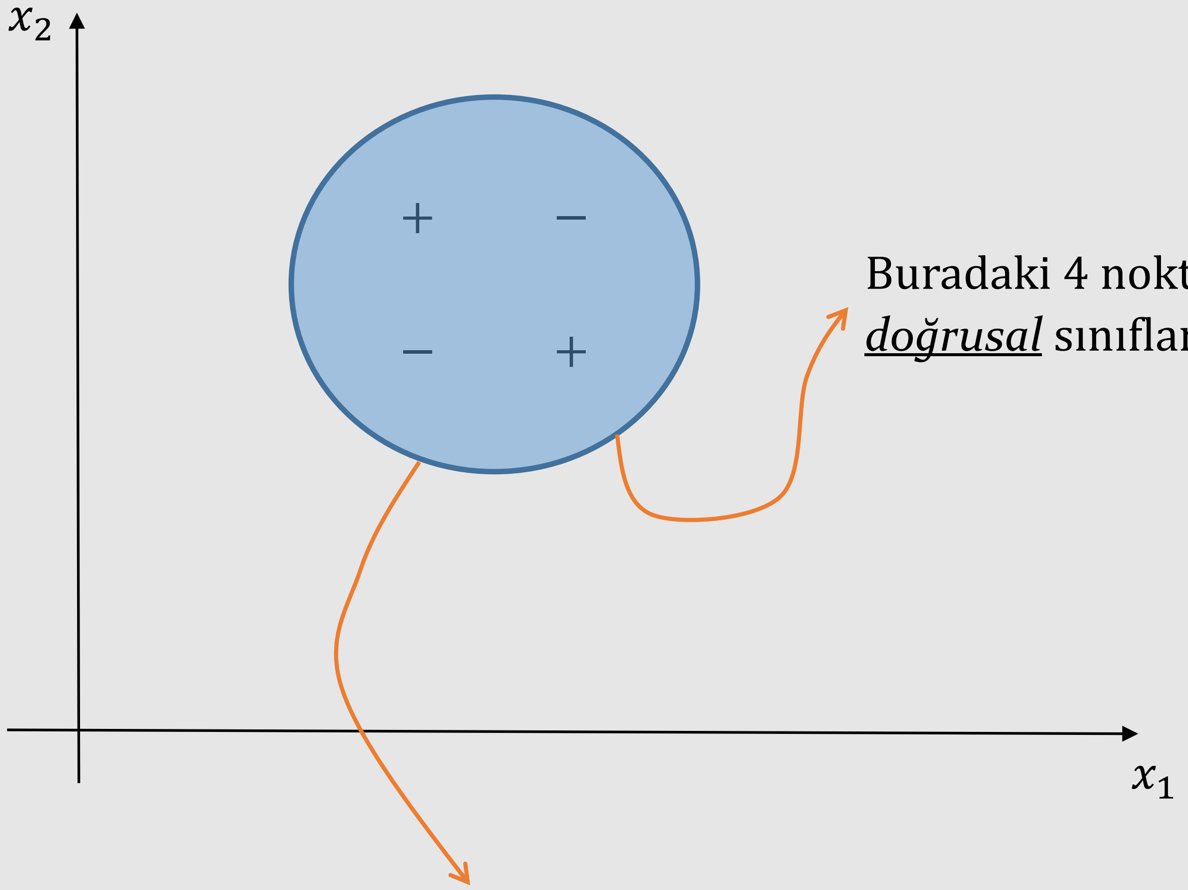


Aktivasyon Fonksiyonlarının Önemi



Buradaki 4 noktayı da ayıracak bir *doğrusal* sınıflandırıcı bulunmamaktadır.

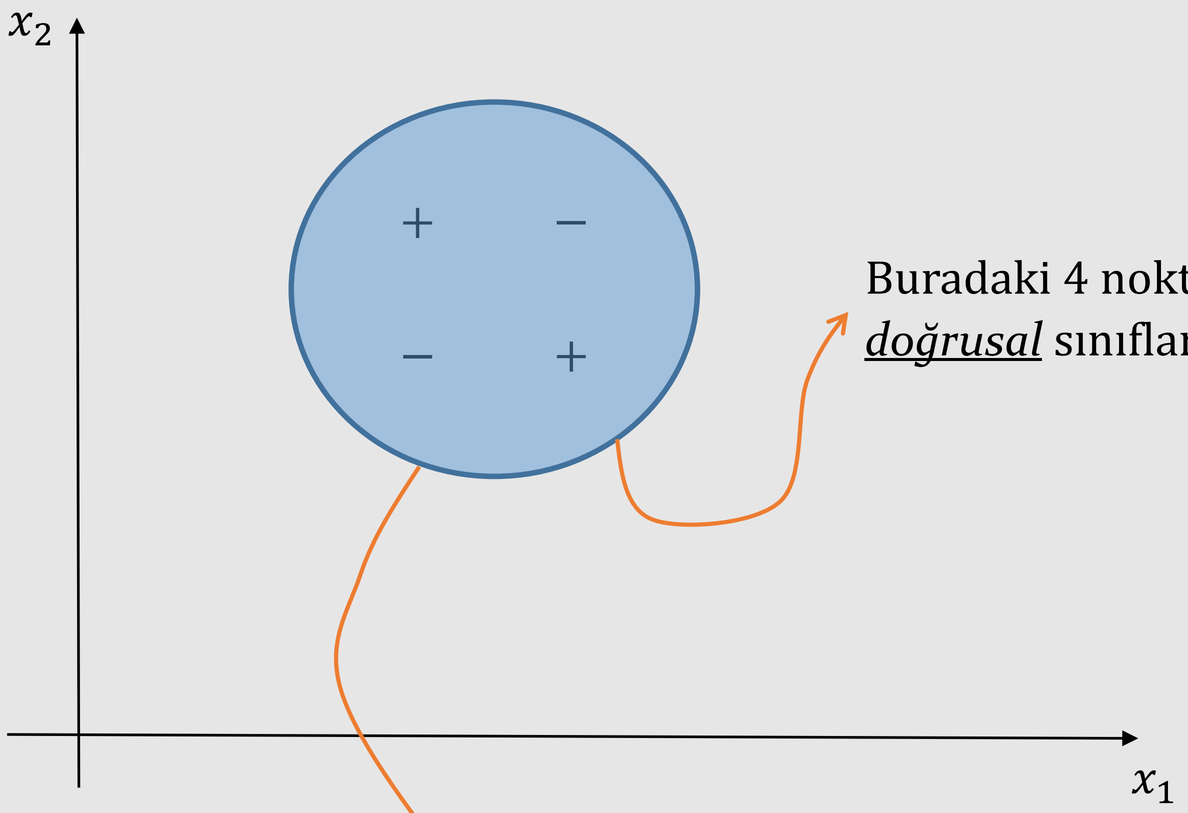
Aktivasyon Fonksiyonlarının Önemi



Buradaki 4 noktayı da ayıracak bir *doğrusal* sınıflandırıcı bulunmamakta.

Bu sınıflandırmayı doğru bir şekilde yapabilmek için aktivasyon fonksiyonlarına ihtiyaç duyarız.

Aktivasyon Fonksiyonlarının Önemi

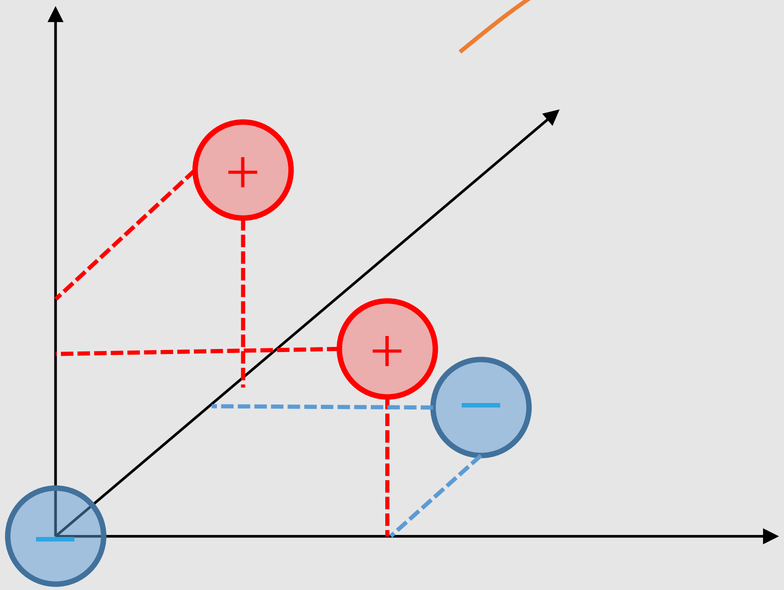


Buradaki 4 noktayı da ayıracak bir *doğrusal* sınıflandırıcı bulunmamakta.

Bu sınıflandırmayı doğru bir şekilde yapabilmek için aktivasyon fonksiyonlarına ihtiyaç duyarız.

Farklı bir boyuta dönüştürüp analiz yapabilmek için

Aktivasyon Fonksiyonlarının Önemi



Doğrusal olmayan bir aktivasyon fonksiyonu uyguladığımızda, ayırma işlemini bu şekilde ayırabiliriz. (Aktivasyon Fonksiyonu)

Örnek

Aşağıda verilen küçük veri seti bilgilerine göre orjinden geçen bir perceptron algoritması oluşturmaya çalışın.

$$x^{(1)} = [-1, -1], y^{(1)} = 1$$

$$x^{(1)} = [1, 0], \quad y^{(2)} = -1$$

$$x^{(1)} = [-1, 1.5], y^{(3)} = 1$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \qquad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Weight vektörü ilk değeri 0.

Algoritma otomatikmen hata olarak algılayacak.

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Weight vektörü ilk değeri 0.

Algoritma otomatikmen hata olarak algılayacak.

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Weight vektörü ilk değeri 0.

Algoritma otomatikmen hata olarak algılayacak.

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n}[\![y^{(i)}w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(0)} = \begin{bmatrix}0\\0\end{bmatrix}, x^{(1)} = \begin{bmatrix}-1\\-1\end{bmatrix} \longrightarrow w^{(0)} \cdot x^{(1)} = 0$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Weight vektörü ilk değeri 0.

Algoritma otomatikmen hata olarak algılayacak.

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

$w^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, x^{(1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \longrightarrow w^{(0)} \cdot x^{(1)} = 0 \longrightarrow$ 0'a eşit. Hata var!

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

Weight vektörü ilk değeri 0.

Algoritma otomatikmen hata olarak algılayacak.

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, x^{(1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \longrightarrow w^{(0)} \cdot x^{(1)} = 0 \longrightarrow$$ 0'a eşit. Hata var!

$$w^{(1)} = w^{(0)} + y^{(1)} x^{(1)} = 0 + 1 \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n}[\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$w^{(1)} \cdot x^{(2)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -1 \longrightarrow -1 > 0, hata\ yok$$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n}[\![y^{(i)}w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(1)} = \begin{bmatrix}-1\\-1\end{bmatrix}$$

$w^{(1)} \cdot x^{(2)} = \begin{bmatrix}-1\\-1\end{bmatrix} \cdot \begin{bmatrix}1\\0\end{bmatrix} = -1 \longrightarrow -1 > 0, hata\ yok$

$w^{(1)} \cdot x^{(3)} = \begin{bmatrix}-1\\-1\end{bmatrix} \cdot \begin{bmatrix}-1\\1.5\end{bmatrix} = -0.5 \longrightarrow -0.5 < 0, hata\ var, güncelle!$

Çözüm

$x^{(1)} = [-1, -1], y^{(1)} = 1$

$x^{(1)} = [1, 0], \quad y^{(2)} = -1$

$x^{(1)} = [-1, 1.5], y^{(3)} = 1$

$w^{(0)} = 0$

$$\mathcal{E}_n(w) = \frac{1}{n}\sum_{i=1}^{n} [\![y^{(i)} w \cdot x^{(i)} \leq 0]\!]$$

$$w^{(1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$w^{(1)} \cdot x^{(2)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = -1 \longrightarrow -1 > 0, hata\ yok$

$w^{(1)} \cdot x^{(3)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 1.5 \end{bmatrix} = -0.5 \longrightarrow -0.5 < 0, hata\ var, güncelle!$

$$\boxed{w^{(2)} = w^{(1)} + y^{(3)} x^{(3)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 1 \begin{bmatrix} -1 \\ 1.5 \end{bmatrix} = \begin{bmatrix} -2 \\ 0.5 \end{bmatrix}}$$