

# Makine Öğrenmesi

6. hafta

- Yapay Sinir Ağlarına Giriş
- Tek katmanlı YSA'lar
  - Algılayıcı (Perceptron)
  - Adaline (Adaptive Linear Element)

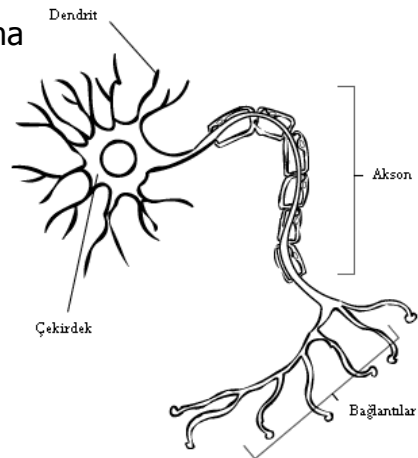
Yrd. Doç. Dr. Umut ORHAN

1

## Biyolojik Sinir Hücresi

Biyolojik sinirler dört ana bölümden oluşmaktadır. Bunlar:

- Dendrit,
- Akson,
- Çekirdek,
- Bağlantılar.

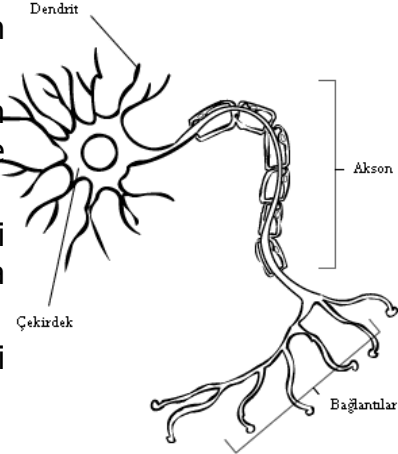


Yrd. Doç. Dr. Umut ORHAN

2

## Biyolojik Sinir Hücresi

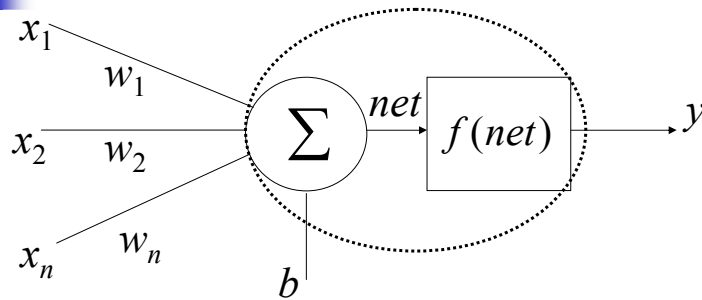
- Dendrit: Bağlı olduğu yerlerden gelen işaretleri çekirdeğe iletir.
- Çekirdek: Dendritten gelen işaretleri bir araya toplar ve aksona iletir.
- Akson: Toplanan bu işaretleri işleyerek nöronun diğer ucunda bulunan bağlantılara gönderilir.
- Bağlantılar: Üretilen bu işareti diğer nöronlara iletir.



Yrd. Doç. Dr. Umut ORHAN

3

## Yapay Sinir Hücresi



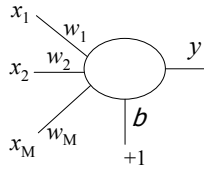
$$net = b + \sum_{i=1}^M x_i w_i \quad y = f(net)$$

Yrd. Doç. Dr. Umut ORHAN

4

## Tek Katmanlı Yapay Sinir Ağı

Tek katmanlı ileri beslemeli ağlar her biri belli ağırlıklarla tüm girişlere bağlı olan bir veya daha fazla çıkış nöronuna sahiptir. Şekilde gösterildiği gibi en basit ağ, iki giriş ve bir çıkıştan oluşur. Nöronun girişi, giriş değişkenlerinin ağırlıklı toplamına sapma ( $b$ ) değerinin eklenmesiyle hesaplanır. Ağın çıkışı, hesaplanan toplamın bir  $f$  fonksiyonu olan çıkış nöronunun aktivasyonu ile aşağıdaki gibi hesaplanır.



$$y = f\left(b + \sum_i w_i x_i\right)$$

Yrd. Doç. Dr. Umut ORHAN

5

## Aktivasyon Fonksiyonu

Tek katmanlı ağlarda  $f$  aktivasyon fonksiyonu doğrusaldır. Biyolojik hücrelerdeki gibi hesaplanan toplam değerin belirli bir eşik değerinden büyük olması durumunda çıkış aktive edilir. Diğer durumlarda çıkış pasiftir. Bu iki değer bazı çalışmalarda 0 ve 1 iken bazılarında -1 ve 1 değerlerini alır.

$$f(net) = \begin{cases} 1 & net > 0 \\ 0 & - \end{cases}$$

Yrd. Doç. Dr. Umut ORHAN

6



## Öğrenme Kuralı

İki öğrenme kuralını konuşacağız: 1958'de Rosenblatt tarafından önerilen Algılayıcı (perceptron) kuralı ve 1960'da Widrow ve Hoff'un önerdiği Adaline (Delta veya LMS kuralı). Her iki yöntem de birbirine çok benzer. Her ağırlık için eski değere bir düzeltme eklenerek yeni bir değer hesaplanır.

$$w_i(t+1) = w_i(t) + \Delta w_i(t),$$

Yrd. Doç. Dr. Umut ORHAN

7



## Genel Algoritma

Hem algılayıcı hem de adaline için algoritma adımları şöyledir:

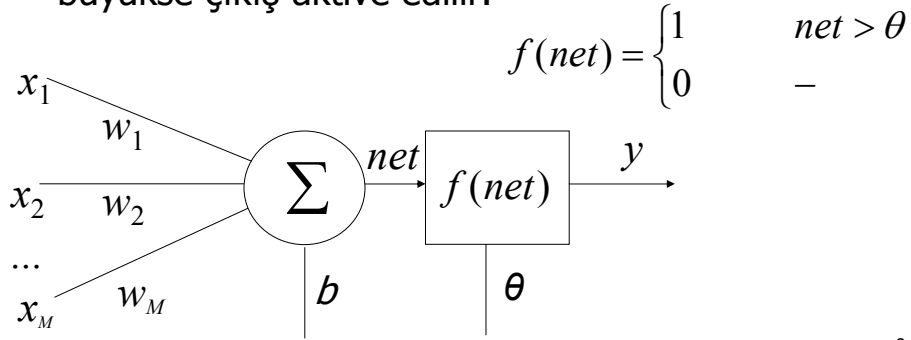
1. Bağlantılar için rasgele ağırlıklarla başla.
2. Eğitim kümesinden bir örnek seç ( $x$  örneğin girişi ve  $d$  örneğin sınıfı).
3. Çıkışı ( $y$ ) hesapla. Eğer  $y \neq d$  ise, tüm  $w_j$  bağlantı ağırlıklarını öğrenme kuralının bulduğu  $\Delta w_j$  değerine göre değiştir.
4. 2. adıma git.

Yrd. Doç. Dr. Umut ORHAN

8

## Algılayıcı (Perceptron)

Aktivasyon fonksiyonu için bir eşik ( $\theta$ ) değeri kullanılır. Hesaplanan 'net' değeri  $\theta$  değerinden büyükse çıkış aktive edilir.



Yrd. Doç. Dr. Umut ORHAN

9

## Algılayıcı (Perceptron)

Ağın bulduğu çıkış değeri gerçek değere uymuyorsa (mutlak hata 1 ise)  $\Delta w_i(t)$  değerleri aşağıdaki gibi hesaplanır.

$$e(t) = d(t) - y(t) = \{-1, 0, 1\}$$

$$\Delta w_i(t) = \eta e(t) x_i(t)$$

$$\Delta b(t) = \eta e(t)$$

burada  $\eta$  öğrenme katsayısı değeri özenle seçilmiş bir sabittir.

Yrd. Doç. Dr. Umut ORHAN

10

## Örnek

$X$  ve  $D$  verisine göre algılayıcı eğitimi sonucu ağırlıkları bulalım.

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Varsayımsal başlangıç değerleri şöyle olsun.

$$w = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad b = 0 \quad \theta = -1 \quad \eta = 0.5$$

Yrd. Doç. Dr. Umut ORHAN

11

## Örnek

İki boyutlu veri kümemizde 2 örnek mevcuttur. Buna göre sırayla ileri beslemeleri hesaplayalım.

$$\begin{aligned} net_1 &= b + x_{1,1}w_1 + x_{1,2}w_2 \\ &= 0 + 1*1 + 0*2 = 1 \\ y &= f(1) = 1 \quad e = 0 \end{aligned}$$

İlk veri örneği doğru sonuç üretti. Ağırlıklar güncellenmez.

$$\begin{aligned} x_1 &= \begin{bmatrix} 1 & 0 \end{bmatrix} \\ x_2 &= \begin{bmatrix} 0 & 1 \end{bmatrix} \\ d_1 &= 1 \\ d_2 &= 0 \\ w_1 &= 1 \\ w_2 &= 2 \\ b &= 0 \\ \theta &= -1 \\ \eta &= 0.5 \end{aligned}$$

Yrd. Doç. Dr. Umut ORHAN

12



## Örnek

İkinci örneğimizin sonucunu hesaplayalım.

$$net_2 = b + x_{2,1}w_1 + x_{2,2}w_2$$

$$= 0 + 0*1 + 1*2 = 2$$

$$y_2 = f(2) = 1 \quad e = -1$$

Yanlış sonuç ürettiği için ağırlıklar güncellenir.

$$b = 0 - 0.5 = -0.5$$

$$w_1 = 1 - 0.5*0 = 1$$

$$w_2 = 2 - 0.5*1 = 1.5$$

$$x_1 = [1 \quad 0]$$

$$x_2 = [0 \quad 1]$$

$$d_1 = 1$$

$$d_2 = 0$$

$$w_1 = 1$$

$$w_2 = 1.5$$

$$b = -0.5$$

$$\theta = -1$$

$$\eta = 0.5$$

Yrd. Doç. Dr. Umut ORHAN

13



## MATLAB Uygulaması

```
>edit Perceptron_ornek.m
```

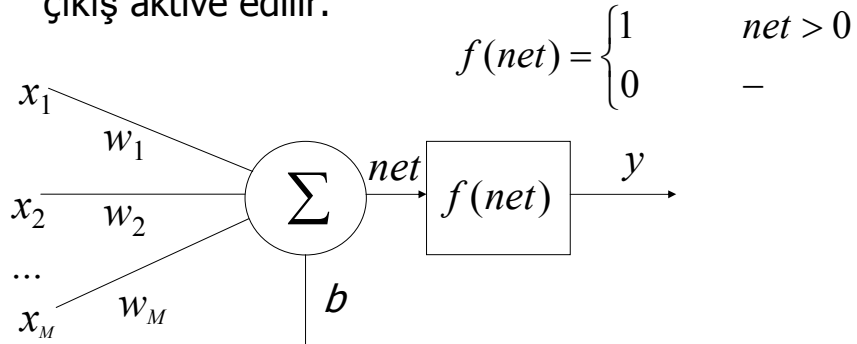
Hazırlanmış olan farklı datasetler yüklenerek Perceptron algoritması deneyi yapılmaktadır. Matlab komutları ile Perceptron benzetimi incelenmeli ve bazı değişiklikler yapılarak kodlar irdelenmelidir.

Yrd. Doç. Dr. Umut ORHAN

14

## Adaline (Adaptive Linear Element)

Eşik değeri, aktivasyon fonksiyonuna giren toplama eklenir. Hesaplanan net değeri sıfırdan büyükse çıkış aktive edilir.



Yrd. Doç. Dr. Umut ORHAN

15

## Adaline (Adaptive Linear Element)

Widrow ve Hoff tarafından geliştirilen “en küçük ortalama kare” (LMS) yöntemiyle eğim azaltmayı hedefleyen öğrenme algoritmasıdır. Karesel hatalar toplamının minimum olmasına dayanır. Ağırlıkları güncellemek için aşağıdaki temel denklem kullanılır:

$$w_i(t+1) = w_i(t) - \frac{1}{2} \eta [\nabla(E\{e^2(t)\})]$$

burada  $\nabla(E\{e^2(t)\})$  ifadesi eğim vektörüdür.

Yrd. Doç. Dr. Umut ORHAN

16





## Adaline (Adaptive Linear Element)

Eğim vektörünün hesaplanması için aşağıdaki denklem kullanılabilir:

$$\nabla(E\{e^2(t)\}) = -2x_i(t)e(t)$$

Bu açılım önceki denklemde yerine konulursa ağırlık güncelleme denklemi şu şekli alır:

$$w_i(t+1) = w_i(t) + \eta x_i(t)e(t)$$

Yrd. Doç. Dr. Umut ORHAN

17



## Adaline (Adaptive Linear Element)

Bulunan ağırlık güncelleme denklemi aslında algılayıcı ile aynıdır. Tek fark algılayıcıda hata hesaplanırken aktivasyon fonksiyonun çıkışı ile istenen hedef farkı alınırken adaline için aktivasyon fonksiyonunun girişi olan "net" değeri ile istenen hedef farkı hesaplanır.

$$e_{\text{perceptron}}(t) = d(t) - f(\sum x_i w_i + b)$$

$$e_{\text{adaline}}(t) = d(t) - \sum x_i w_i + b$$

Yrd. Doç. Dr. Umut ORHAN

18

## Örnek

Önceki örneğimizin 2. iterasyonunu Adaline için tekrarlayalım.

$$net_2 = 0 + 0 * 1 + 1 * 2 = 2$$

$$y_2 = 1 \quad e = 0 - 2 = -2$$

İkinci veri örneği yanlış sonuç üretti. Ağırlıklar güncellenmelidir.

$$b = 0 - 2 * 0.5 = -1$$

$$w_1 = 1 - 2 * 0.5 * 0 = 1$$

$$w_2 = 2 - 2 * 0.5 * 1 = 1$$

$$x_1 = [1 \quad 0]$$

$$x_2 = [0 \quad 1]$$

$$d_1 = 1$$

$$d_2 = 0$$

$$w_1 = 1$$

$$w_2 = 1$$

$$b = -1$$

$$\theta = -1$$

$$\eta = 0.5$$

Yrd. Doç. Dr. Umut ORHAN

19

## MATLAB Uygulaması

```
>edit Adaline_ornek.m
```

Hazırlanmış olan farklı datasetler yüklenerek Adaline algoritması deneyi yapılmaktadır. Matlab komutları ile Adaline benzetimi incelenmeli ve bazı değişiklikler yapılarak kodlar irdelenmelidir.

Yrd. Doç. Dr. Umut ORHAN

20



## ÖDEV

---

Algılayıcı ve Adaline için hazırlanmış MATLAB kodları ile üç farklı çapraz geçerlik yöntemi kullanılarak önceki derslerde verilen sentetik veri kümeleri analiz edilecek.