

# Zeki Optimizasyon Teknikleri Intelligent Optimization Techniques

Hazırlayan: M. Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Singiresu S. Rao, Engineering Optimization: Theory and Practice, Wiley, 2009." kitabı kullanılarak hazırlanmıştır.

## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

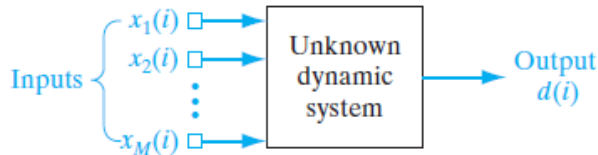
## Öğrenme nedir?

- ▶ **Öğrenme sürecinde**, model içerisindeki **parametreler** problemin çözümü için **optimize edilir**.
- ▶ **Parametre sayısı** geliştirilen modelin **öğrenme kapasitesiyle ilişkilidir**.
- ▶ Modelin parametrelerinin optimizasyonu için **farklı algoritmalar kullanılabilir**.
- ▶ Modellenen **sisteme ait veriler** eğitim sürecinde geliştirilen modele **giriş olarak verilir**.
- ▶ **Denetimli öğrenme** yapan sistemlerde **çıkış verisi de modele verilir**.

3

## Öğrenme nedir?

- ▶ Dinamik bir sistem aşağıdaki gibi M adet giriş ve bir tane çıkışa sahip olsun.



- ▶ Sistemde,  $x_1(i), x_2(i), \dots, x_M(i)$   $i$ .sıradaki girişleri gösterir.
- ▶ Eğitim veriseti aşağıdaki gibi tanımlanabilir.

$$\mathcal{T}: \{\mathbf{x}(i), d(i); i = 1, 2, \dots, n, \dots\}$$

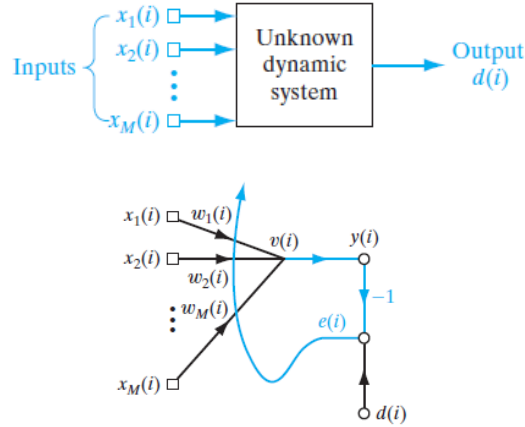
$$\mathbf{x}(i) = [x_1(i), x_2(i), \dots, x_M(i)]^T$$

- ▶ Eğitim setindeki her satır için **istenen çıkış** ise  **$d(i)$**  ile **gösterilir**.

4

## Öğrenme nedir?

- Dinamik sistem aşağıdaki gibi modellenir.



- Modelin parametreleri her girişin ağırlık değerleridir ( $w$ ).
- Öğrenme sürecinde  $w$  ağırlıkları belirlenir.

5

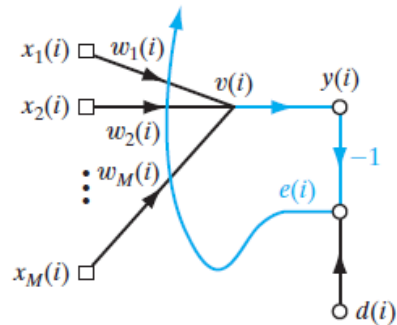
## Öğrenme nedir?

- Modelde **ileri doğru geçişte** giriş değerlerine göre ağırlıklar kullanılarak **çıkış hesaplanır**.

$$y(i) = v(i) = \sum_{k=1}^M w_k(i) x_k(i)$$

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

$$\mathbf{w}(i) = [w_1(i), w_2(i), \dots, w_M(i)]^T$$



- İstenen çıkış ile hesaplanan çıkışın farkı alınarak **hata değeri hesaplanır**.

$$e(i) = d(i) - y(i)$$

6

## Öğrenme nedir?

- ▶ Parametrelerin optimizasyonu sürecinde ağırlıkların optimal çözümü ( $\mathbf{w}^*$ ) bulunur.

$$\mathcal{E}(\mathbf{w}^*) \leq \mathcal{E}(\mathbf{w})$$

- ▶  $\mathcal{E}(\mathbf{w}^*)$  **maliyet fonksiyonu** ağırlık vektörüne göre **minimize edilir** (unconstraint optimization).
- ▶ Optimalite için gerekli şart,

$$\nabla \mathcal{E}(\mathbf{w}^*) = \mathbf{0}$$

7

## Öğrenme nedir?

- ▶  $\nabla$  gradient operatörüdür,

$$\nabla = \left[ \frac{\partial}{\partial w_1}, \frac{\partial}{\partial w_2}, \dots, \frac{\partial}{\partial w_M} \right]^T$$

- ▶  $\nabla \mathcal{E}(\mathbf{w})$  hata vektörünün gradient operatörüdür,

$$\nabla \mathcal{E}(\mathbf{w}) = \left[ \frac{\partial \mathcal{E}}{\partial w_1}, \frac{\partial \mathcal{E}}{\partial w_2}, \dots, \frac{\partial \mathcal{E}}{\partial w_M} \right]^T$$

- ▶ Her iterasyonda hesaplanan yeni ağırlık değerleri ile maliyet fonksiyonu daha küçük değer alır.

$$\mathcal{E}(\mathbf{w}(n+1)) < \mathcal{E}(\mathbf{w}(n))$$

8

## Öğrenme nedir?

- ▶ **Steepest descent** algoritmasında ağırlıklar gradyan vektörünün ters yönünde değiştirilir.

$$\mathbf{g} = \nabla \mathcal{E}(\mathbf{w})$$

- ▶  $\nabla E(\mathbf{w})$  gradient operatörüdür,

$$\mathbf{w}(n + 1) = \mathbf{w}(n) - \eta \mathbf{g}(n)$$

- ▶ Burada,  $\eta$  ağırlıkların değişme hızını ayarlar ve **öğrenme oranı** veya adım boyutunu olarak adlandırılır.

$$\begin{aligned} \Delta \mathbf{w}(n) &= \mathbf{w}(n + 1) - \mathbf{w}(n) \\ &= -\eta \mathbf{g}(n) \end{aligned}$$

9

## İçerik

- ▶ Öğrenme nedir?
- ▶ **Perceptron öğrenme kuralı**
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

10

## Perceptron öğrenme kuralı

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \Delta \mathbf{w}_i$$

$$\Delta \mathbf{w}_i = \eta * \mathbf{x}_i * (\mathbf{d} - \mathbf{o})$$

$\eta$  = öğrenme oranı (0 - 1 arasında)

$\mathbf{w}_i$  =  $i$ . bağlantının ağırlık değeri

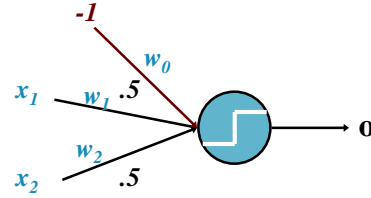
$\Delta \mathbf{w}_i$  = Ağırlık değişim değeri

$\mathbf{x}_i$  =  $i$ . girişe girilen değer

$\mathbf{d}$  = İstenen çıkış değeri

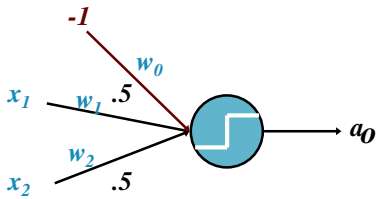
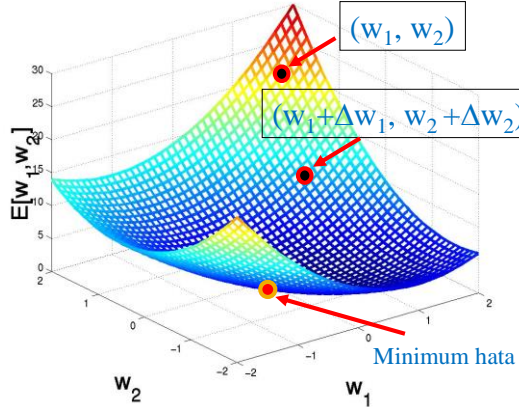
$\mathbf{o}$  = Hesaplanan çıkış değeri

$x_1$	$x_2$	$d$
1.0	1.1	3.3
0.9	1.4	2.7
...	...	...
2.2	3.1	0.8



## Perceptron öğrenme kuralı

$x_1$	$x_2$	$d$
1.0	1.1	3.3
0.9	1.4	2.7
...	...	...
2.2	3.1	0.8



## Perceptron öğrenme kuralı

Sum Square Error (SSE)

$$SSE = \sum_{p=1}^P \sum_{j=1}^K (d_{p,j} - o_{p,j})^2$$

**P** = Eğitim kümesi boyutu

**K** = Çıkış vektörü boyutu (Çıkış neuron sayısı)

Mean Squared Error (MSE)

$$MSE = \frac{1}{P} \sum_{p=1}^P \sum_{j=1}^K (d_{p,j} - o_{p,j})^2$$

13

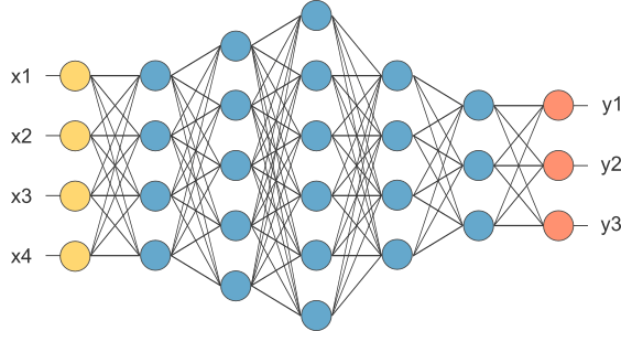
## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

14

## Çok katmanlı ileri beslemeli ağlar

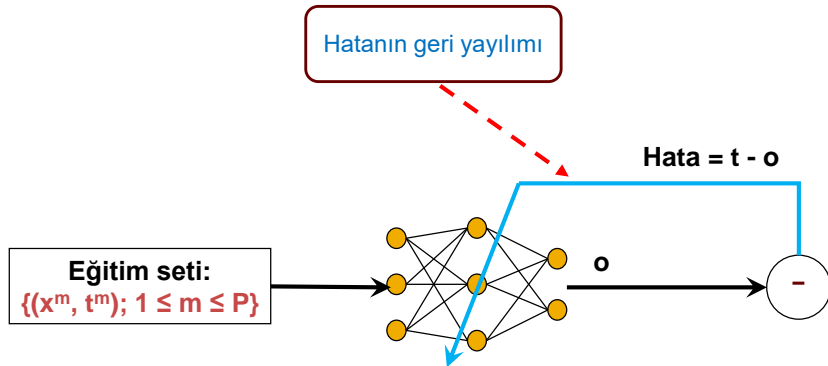
- ▶ Çok katmanlı ağlarda **geriye yayılım (backpropagation) yöntemiyle** öğrenme gerçekleştirilebilir.
- ▶ Eğitim verisetindeki **her satır için çıkış hatası hesaplanır.**
- ▶ Çıkış hatasına göre **ağırlıklar çıkıştan girişe doğru değiştirilir.**
- ▶ Ağırlıkların değişimini hesaplamak için **gradyan vektörü kullanılır.**



15

## Çok katmanlı ileri beslemeli ağlar

- ▶ Çok katmanlı ağlarda **geriye yayılım (backpropagation) yöntemiyle** ağırlıklar değiştirilir (öğrenme).



16



## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

17

## Çok katmanlı ileri beslemeli ağlarda öğrenme

- ▶ Çok katmanlı ağlarda **hataların farklarının karelerinin toplamı** ağırlıkları değiştirmek için **geri yayılım yapılır**.
- ▶ Eğitim verisetindeki **her satır için ağırlıklar** çıkıştan girişe doğru **değiştirilir**.

$$E = \sum (d_k - o_k)^2 \quad \nabla E(w) = \left[ \frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

$$w \leftarrow w + \Delta w$$

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w = -\eta \nabla E(w)$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

18

## Çok katmanlı ileri beslemeli ağlarda öğrenme

$$w_{1,2} = w_{1,2} + \Delta w_{1,2}$$

$$\Delta w_{1,2} = \eta (\partial E / \partial O_2) (\partial O_2 / \partial net_2) (\partial net_2 / \partial w_{1,2})$$

$$\partial E / \partial O_2 = (\partial / \partial O_2) (T_2 - O_2)^2$$

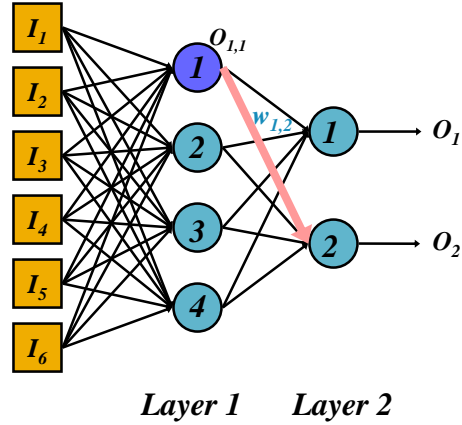
$$\partial E / \partial O_2 = -2(T_2 - O_2)$$

$$\partial O_2 / \partial net_2 = (\partial / \partial net_2) f(net_2)$$

$$\partial O_2 / \partial net_2 = f'(net_2)$$

$$\partial net_2 / \partial w_{1,2} = O_{1,1}$$

$$\Delta w_{1,2} = \eta (T_2 - O_2) f'(net_2) O_{1,1}$$



Sigmoid  $\rightarrow f(x) = 1/(1+\exp(-x))$   
 $f'(x) = \exp(-x)/(1+\exp(-x))^2$   
 $= 1/(1+\exp(-x)) - 1/(1+\exp(-x))^2 = f(x)(1-f(x))$

19

## Çok katmanlı ileri beslemeli ağlarda öğrenme

$$w_{i1,2} = w_{i1,2} + \Delta w_{i1,2}$$

$$\Delta w_{i1,2} = \eta (\partial E / \partial O_2) (\partial O_2 / \partial net_2) (\partial net_2 / \partial O_{1,2}) (\partial O_{1,2} / \partial net_{1,2}) (\partial net_{1,2} / \partial w_{i1,2})$$


---


$$\partial net_2 / \partial O_{1,2} = w_{2,2}$$

$$\partial O_{1,2} / \partial net_{1,2} = f'(net_{1,2})$$

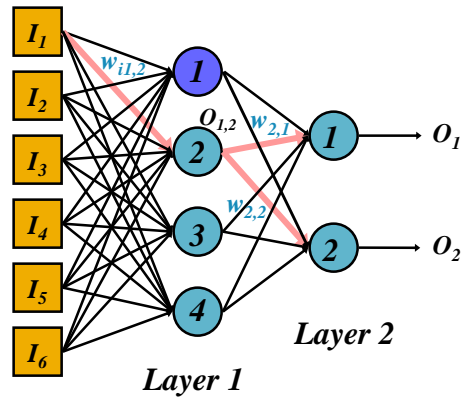
$$\partial net_{1,2} / \partial w_{i1,2} = I_i$$


---


$$\Delta w_{i1,2} = \eta (T_2 - O_2) f'(net_2) w_{2,2} f'(net_{1,2}) I_i$$


---


$$\Delta w_{i1,2} = \eta (\partial E / \partial O_1) (\partial O_1 / \partial net_1) (\partial net_1 / \partial O_{1,2}) (\partial / \partial net_{1,2}) (\partial net_{1,2} / \partial w_{i1,2})$$



$$\Delta w_{i1,2} = \eta (T_1 - O_1) f'(net_1) w_{2,1} f'(net_{1,2}) I_i$$

20

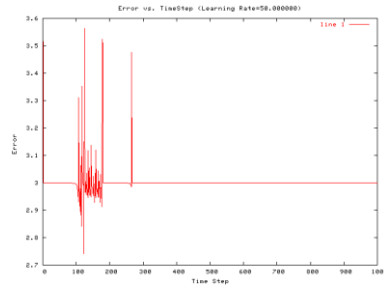
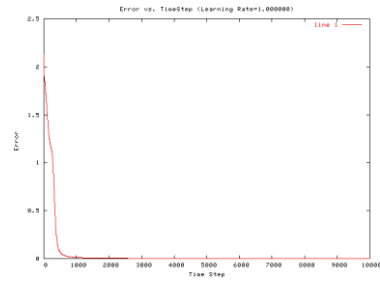
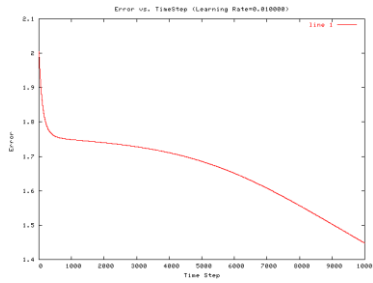
## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

21

## Öğrenme hızı

- ▶ Öğrenme hızı çok yüksek olursa kararsızlık olur, çok düşük olursa öğrenme çok uzun süre alır.



22

## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

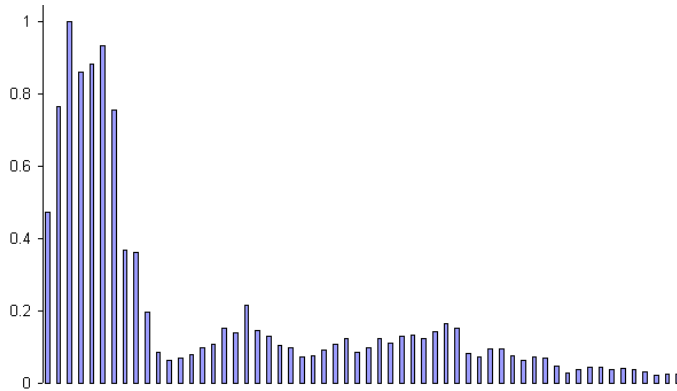
23

## Ses tanıma uygulaması

**Problem:** İki farklı kişinin “merhaba” kelimesini söylemesinin öğretilmesi

1. Kişi = Ahmet
2. Kişi = Mehmet

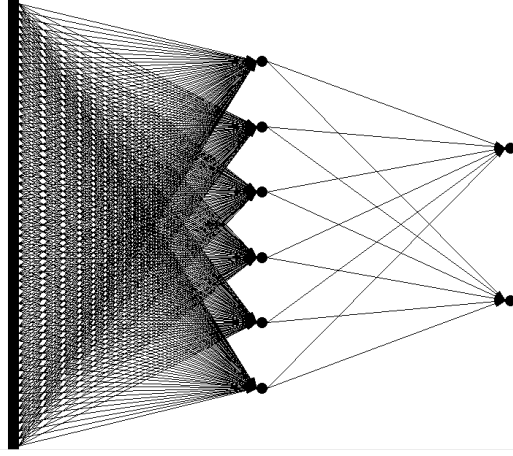
Frekans dağılımını 60 örnek ile alınsın.



24

## Ses tanıma uygulaması

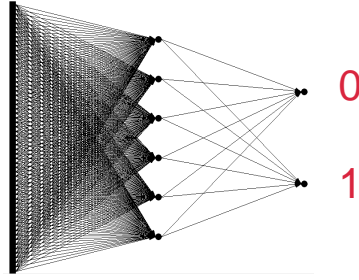
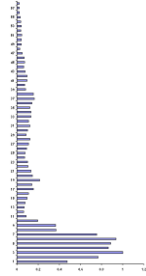
Ağ yapısı = ileri beslemeli çok katmanlı  
60 giriş (her frekans örneği için)  
6 gizli node  
2 çıkış (0-1 ise "Ahmet", 1-0 ise "Mehmet")



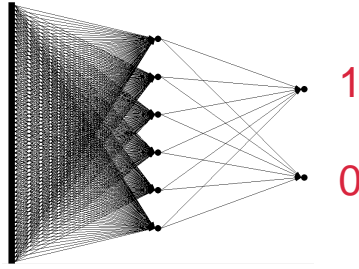
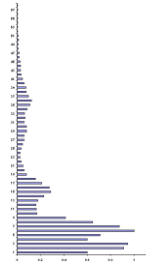
25

## Ses tanıma uygulaması

Ahmet



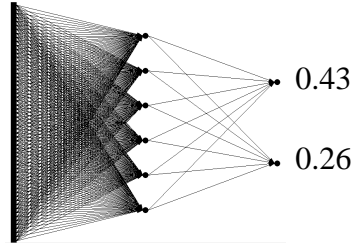
Mehmet



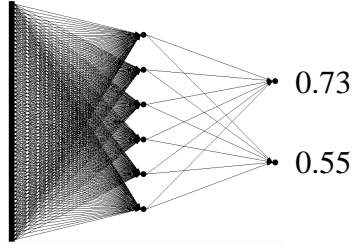
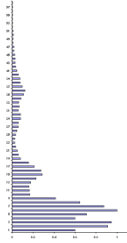
26

## Ses tanıma uygulaması

Ahmet



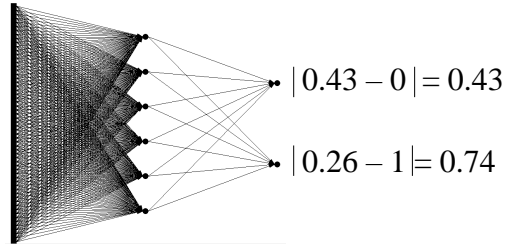
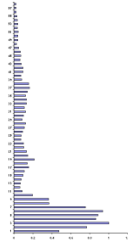
Mehmet



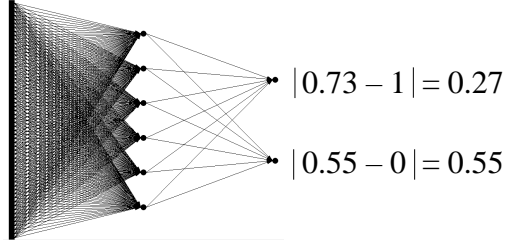
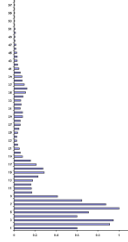
27

## Ses tanıma uygulaması

Ahmet



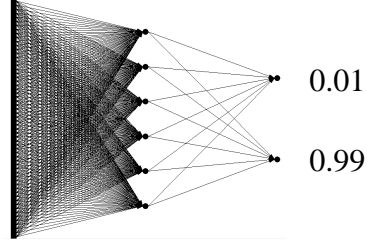
Mehmet



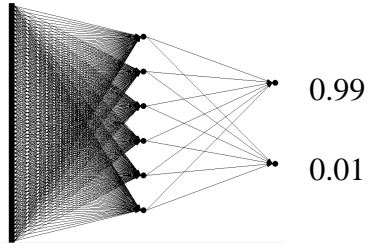
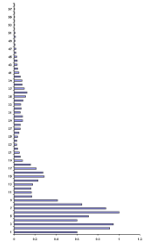
28

## Ses tanıma uygulaması

Ahmet



Mehmet



29

## İçerik

- ▶ Öğrenme nedir?
- ▶ Perceptron öğrenme kuralı
- ▶ Çok katmanlı ileri beslemeli ağlar
- ▶ Çok katmanlı ileri beslemeli ağlarda öğrenme
- ▶ Öğrenme hızı
- ▶ Ses tanıma uygulaması
- ▶ Karakter tanıma uygulaması

30

## Karakter tanıma uygulaması

- İleri beslemeli çok katmanlı ağ
- Backpropagation öğrenme metodu

