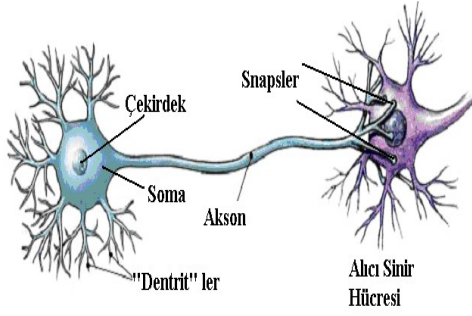
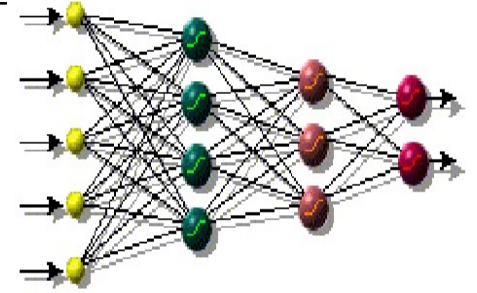


# ECE655 YAPAY SİNİR AĞLARINDA ÖĞRENME YÖNTEMLERİ

- 1) EĞİM DÜŞME (Gradient Descent)  
Geriye Yayılım (Back Propagation) Algoritması



# Çok Katmanlı İleri Sürümlü YSA'da Geri Hesaplama (EĞİTİM/Hata Geri Yayılmı)



- İleri sürümlü ve çok katmanlı bir YSA için eğitici/danışmanlı öğrenme yöntemi olan **GERİYEYAYILIM** (*Backpropagation*) algoritması tanımlanacak ve türetilecektir.
- Algoritma çıkış katmanındaki hücre çıkışları ( $y_d$ ) ile bu hücre çıkışları için istenen ( $y_d$ ) çıkışlar arasındaki hata  $(e_j^{(n)})^2$ 'ya dayalı olarak işletilir.
- **Verilenler:** Eğitilecek ağı için önceden belirlenmiş bir eğitim kümemiz var...
- **Hesaplanacaklar:** Önceden belirlenen ağı yapısı üzerinden, eğitime başlamadan rasgele atanmış ağı parametreleri ile eğitim kümesindeki  $k$  ncı giriş için ileri hesaplama ve hata belirleme...veee maliyet (cost) fonksiyonu (**eğitim başarıml ölçütü**)

$j$ . Çıkış hücresi için hata  $\rightarrow e_j^{(n)} = y_{d_j}^{(n)} - y_{a_j}^{(n)}$

$j$ . Çıkış hücresi için ani hata  $\rightarrow \frac{1}{2}(e_j^{(n)})^2$

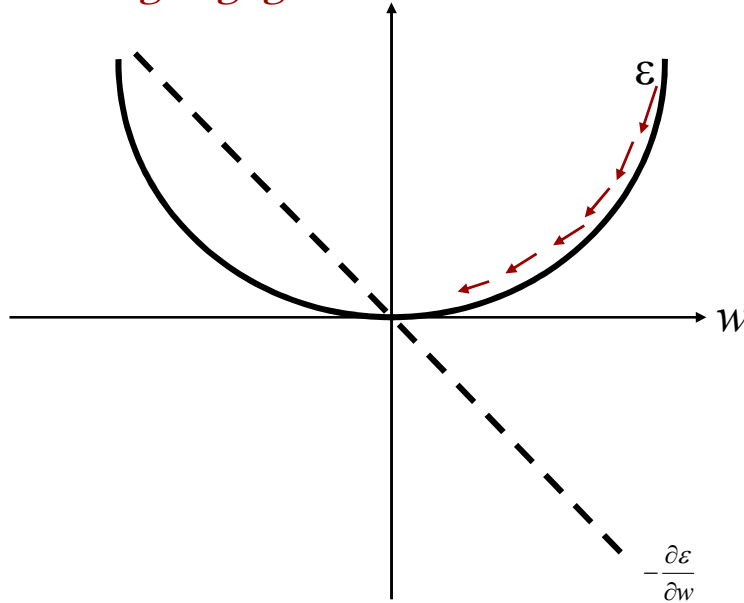
Toplam ani hata (cost  $f$ )  $\rightarrow \mathcal{E}^{(n)} = \frac{1}{2} \sum_{j \in C} (e_j^{(n)})^2$

Şimdi geri hesaplama yapılacak ve parametreler bu hesaba göre güncellenecek.... **NASIL?**

- *Algoritmanın tanımına göre: Herhangi bir iterasyonda, herhangi bir katmanındaki bir j. nöron ile bir önceki katmandaki bir i. nöron arasındaki bağlantı ağırlık parametresi*

$$w_{ji}(n+1) = w_{ji}(n) - \eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \quad \{\text{Delta kuralıudik iniş (steepest - descent) yöntemi}\}$$

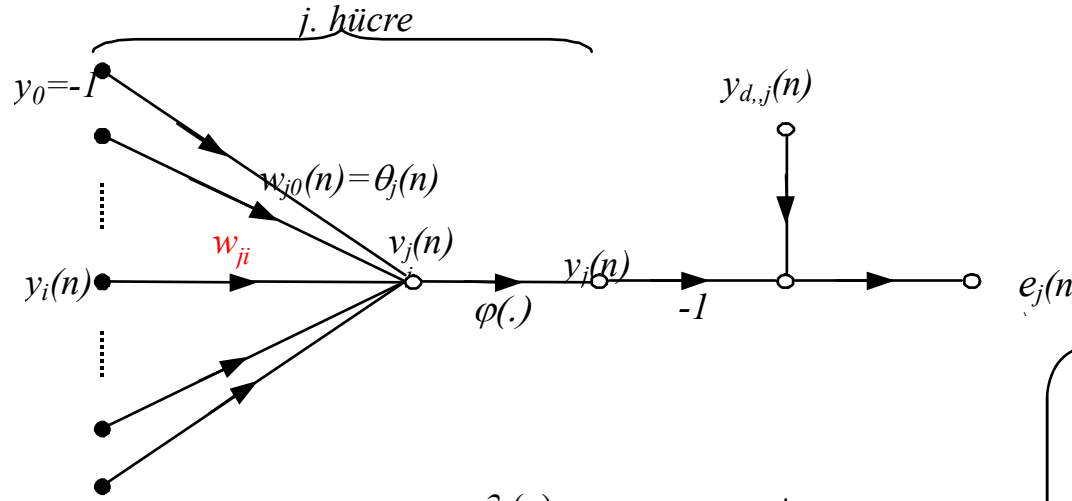
*denklemleri gereği güncellenir.*



*Yöntemin felsefesi*

*ε = 1/2 w<sup>2</sup> denklemi ile verilen tek parametrelili bir ölçüt (cost) fonksiyonunun en küçük değerini aldığı noktanın Δw = -η ∂ε<sup>(k)</sup> / ∂w<sub>ji</sub><sup>(k)</sup> bağıntısı ile verilen kural ile iteratif olarak bulunabilmesine dayalıdır.*

## ALGORİTMANIN TÜRETİMİ: “j. hücre çıkış katmanında ise”



$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \eta e_j(n) \phi_j'(v_j(n)) y_i(n)$$

$$\delta_j(n) = -\frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$$

} yerel gradyeni

$$= e_j(n) \phi_j'(v_j(n))$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

Geriye yayılım algoritması,  $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$

ani gradyeni ile orantılı olan  $w_{ji}(n)$  bağlantı ağırlığındaki  $\Delta w_{ji}(n)$  düzeltme

işlemini uygular.

Zincir kuralına göre;  $\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)}$

ani gradyeni şöyle

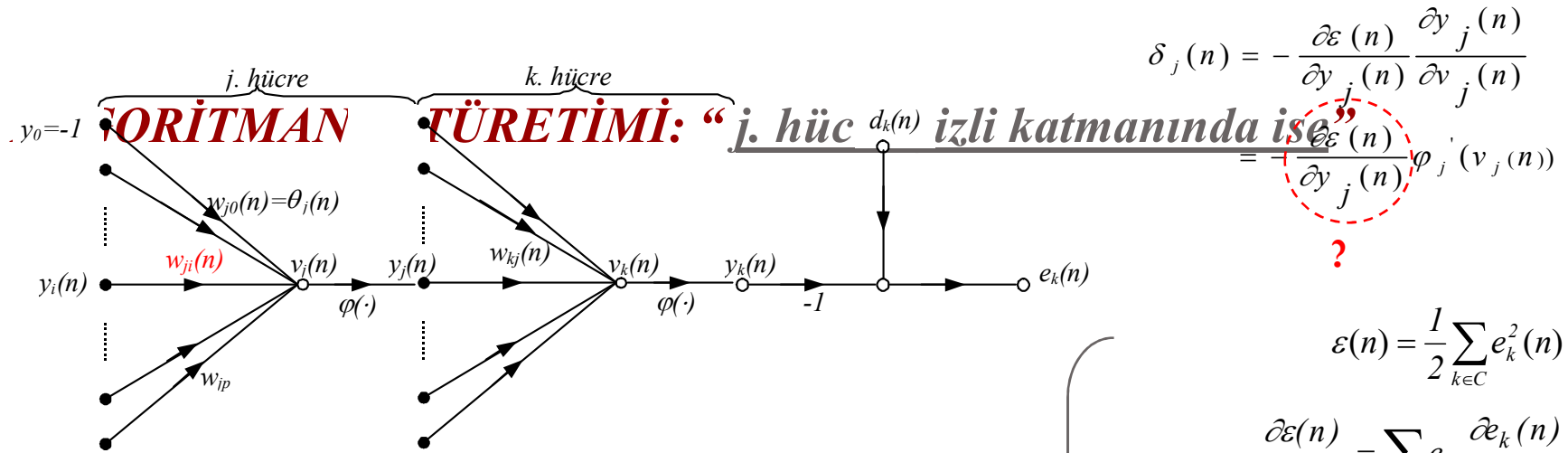
ifade edilebilir :

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n) \quad \frac{\partial e_j(n)}{\partial y_j(n)} = -1$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi_j'(v_j(n)) \quad \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n) \phi_j'(v_j(n)) y_i(n)$$



$$\delta_j(n) = - \frac{\partial \varepsilon(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$$

$$= - \frac{\partial \varepsilon(n)}{\partial y_j(n)} \varphi_j'(v_j(n))$$

$$\varepsilon(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)}$$

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)}$$

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n))$$

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n))$$

$$v_k(n) = \sum_{j=0}^q w_{kj}(n) y_j(n)$$

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n)$$

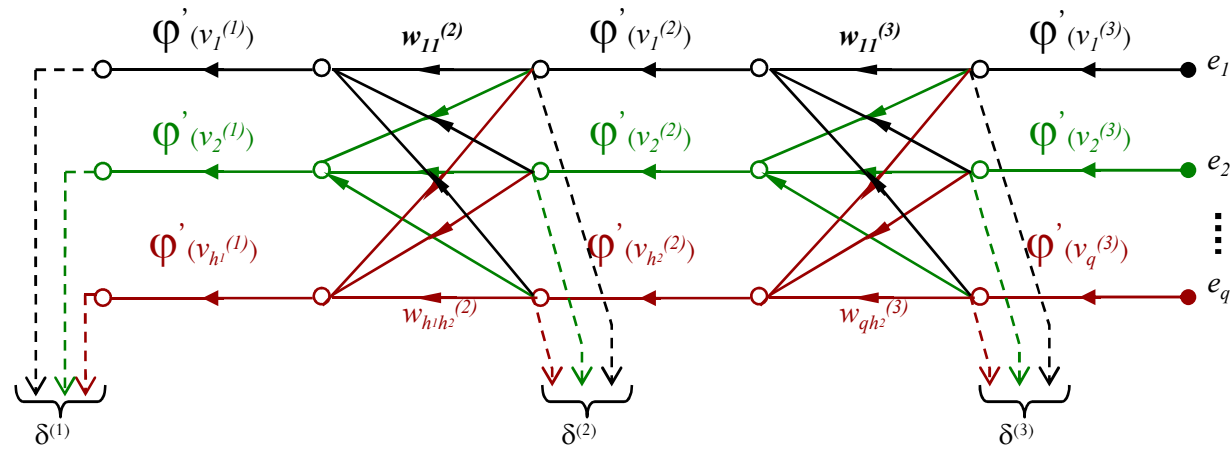
$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = - \sum_k e_k(n) \varphi_k'(v_k(n)) w_{kj}(n)$$

$$= - \sum_k \delta_k(n) w_{kj}(n)$$

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

3 katmanlı ileri sürümlü YSA için geri hesaplama (hata geri yayılımı/yerel gradyen hesabı) işaret akış diyagramı



*Transpose network*