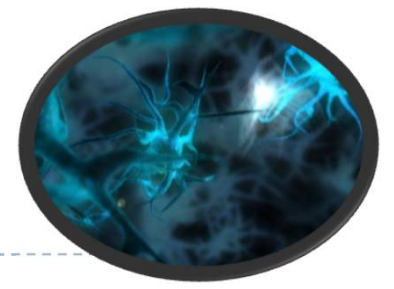
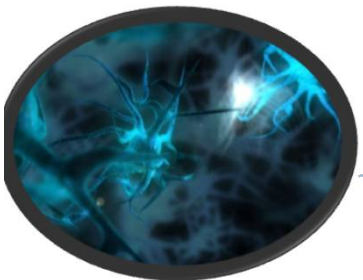


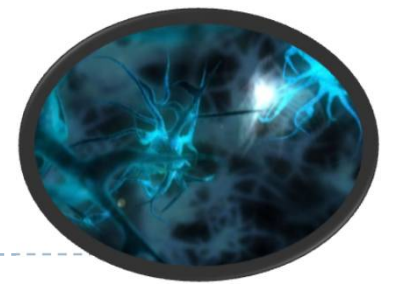
# YAPAY SİNİR AĞLARINA GİRİŞ

DR. ÖĞR. ÜYESİ BETÜL UZBAŞ

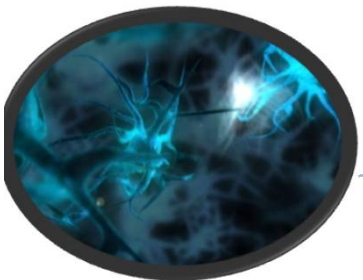
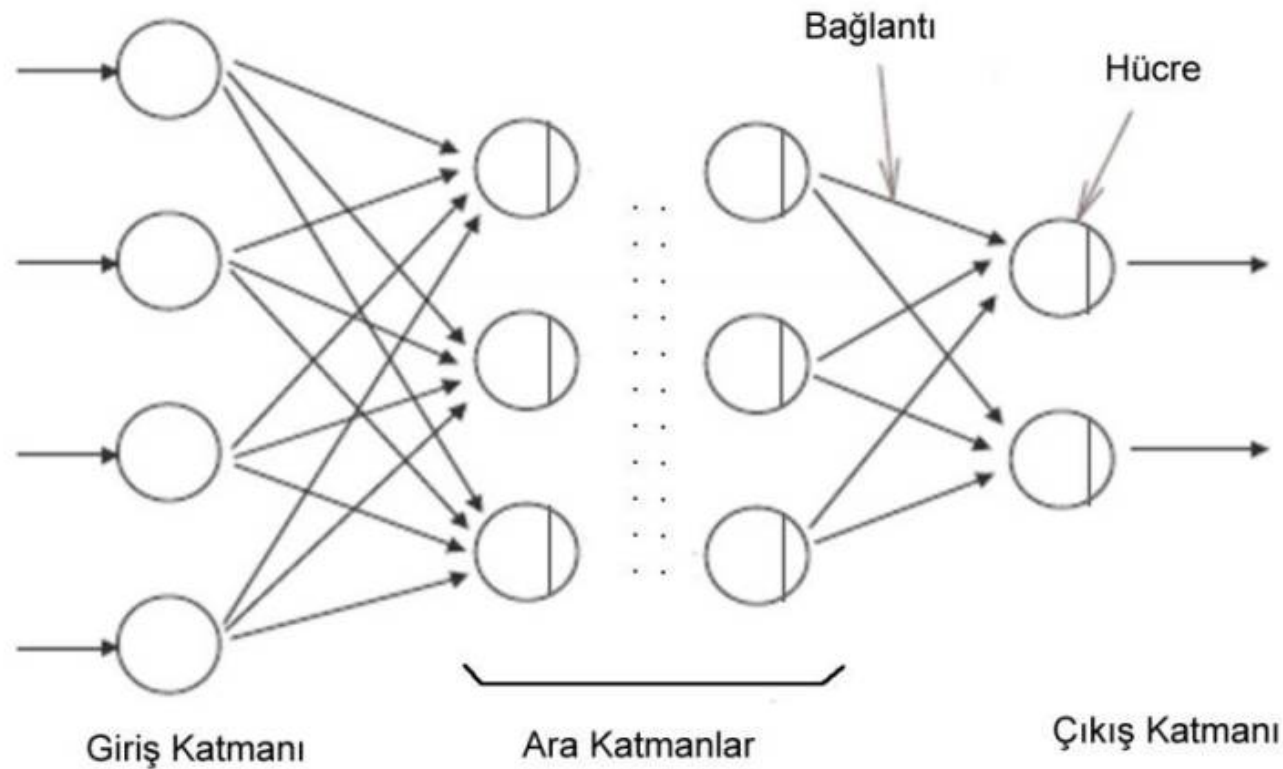


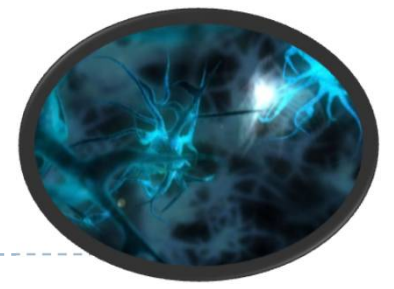
- ▶ Tek katmanlı algılayıcılar basit ve hızlı olmakla birlikte, sadece karmaşık olmayan doğrusal problemleri çözebilmektedir.
- ▶ Minsky perceptron modelinin XOR problemine çözüm üretemediğini göstermiş ve YSA'nın doğrusal olmayan problemlere çözüm üretemediği iddia edilerek bilimsel araştırmaların durmasına neden olmuştur.
- ▶ XOR problemini çözmek için Çok Katmanlı Algılayıcılar (ÇKA) modeli geliştirilmiştir. Rumelhart ve ark. tarafından geliştirilen bu modele backpropagation modeli de denmektedir. ÇKA modeli YSA'ya ilgiyi hızlı bir şekilde artırmıştır. Bu model Delta öğrenme kuralı denilen öğrenme yöntemini kullanmaktadır.





- Yapay sinir hücreleri biraraya gelerek yapay sinir ağılatını oluştururlar. Yapay sinir ağları üç ana katmanda incelenir; Giriş Katmanı, Ara (Gizli) Katmanlar ve Çıkış Katmanı.

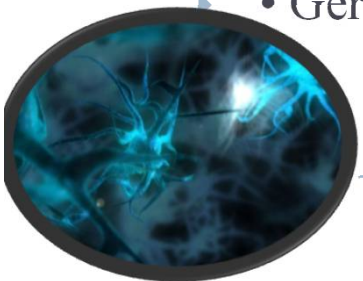


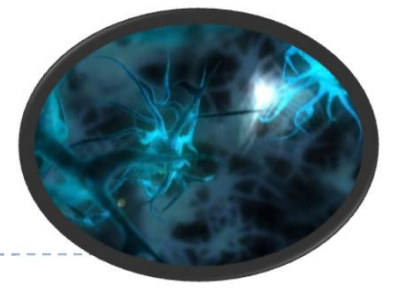


# ÇKA Ağının Öğrenme Kuralı

---

- ▶ Danışmanlı öğrenme stratejisine göre çalışır.
- ▶ ÇKA'nın öğrenme kuralı en küçük kareler yöntemine dayalı *Delta Öğrenme Kuralının* geliştirilmiş halidir. Bu nedenle *Genelleştirilmiş Delta Kuralı* da denilmektedir.
- ▶ Ağın öğrenebilmesi için eğitim seti adı verilen ve örneklerden oluşan bir sete ihtiyaç vardır. Bu set içinde her örnek için ağın hem girdiler hem de o girdiler için ağın üretmesi gereken çıktılar belirlenmiştir.
- ▶ “Genelleştirilmiş Delta Kuralı” iki aşamadan oluşur:
  - ▶ • İleri doğru hesaplama(Feed Forward)
  - ▶ • Geri doğru hesaplama (Back Propagation)

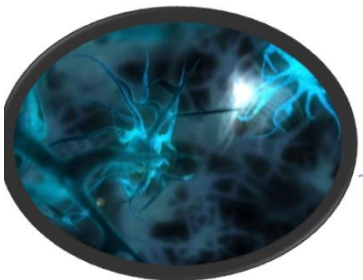


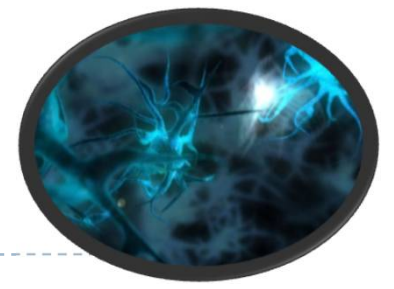


## ► I) İleri Doğru Hesaplama:

- Bu safhada bilgi işleme eğitim setindeki bir örneğin Girdi Katmanından ağı gösterilmesi ile başlar. Gelen girdiler hiç bir değişiklik olmadan ara katmana gönderilir.

$$C_k^i = G_k$$





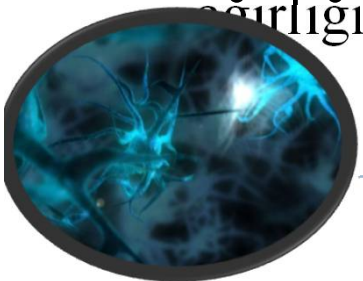
# I) İleri Doğru Hesaplama:

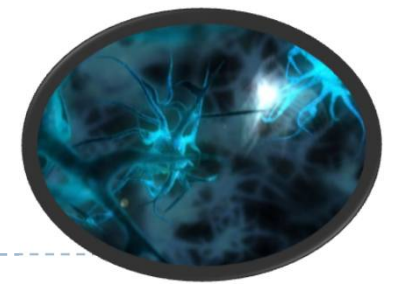
- ▶ Ara katmandaki her işlemci elemanı girdi katmandaki bütün işlemci elemanlarından gelen bilgileri ağırlıklarının etkisi ile alır.
- ▶  $A_{kj}$  k. girdi elemanının j. ara katman elemanına bağlayan bağlantının ağırlık değerini göstermektedir.
- ▶ j. ara katman elemanının çıktısı net girdinin aktivasyon fonksiyonundan (genellikle sigmoid) geçirilmesi ile hesaplanır.

$$NET_j^a = \sum_{k=1}^n A_{kj} C_i^k$$

$$C_j^a = \frac{1}{1 + e^{-(NET_j^a + \beta_j^a)}}$$

- ▶  $B_j$  = Ara katmanda bulunan j. elemana bağlanan bias değer elemanının ağırlığını göstermektedir

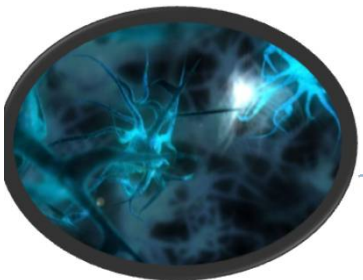




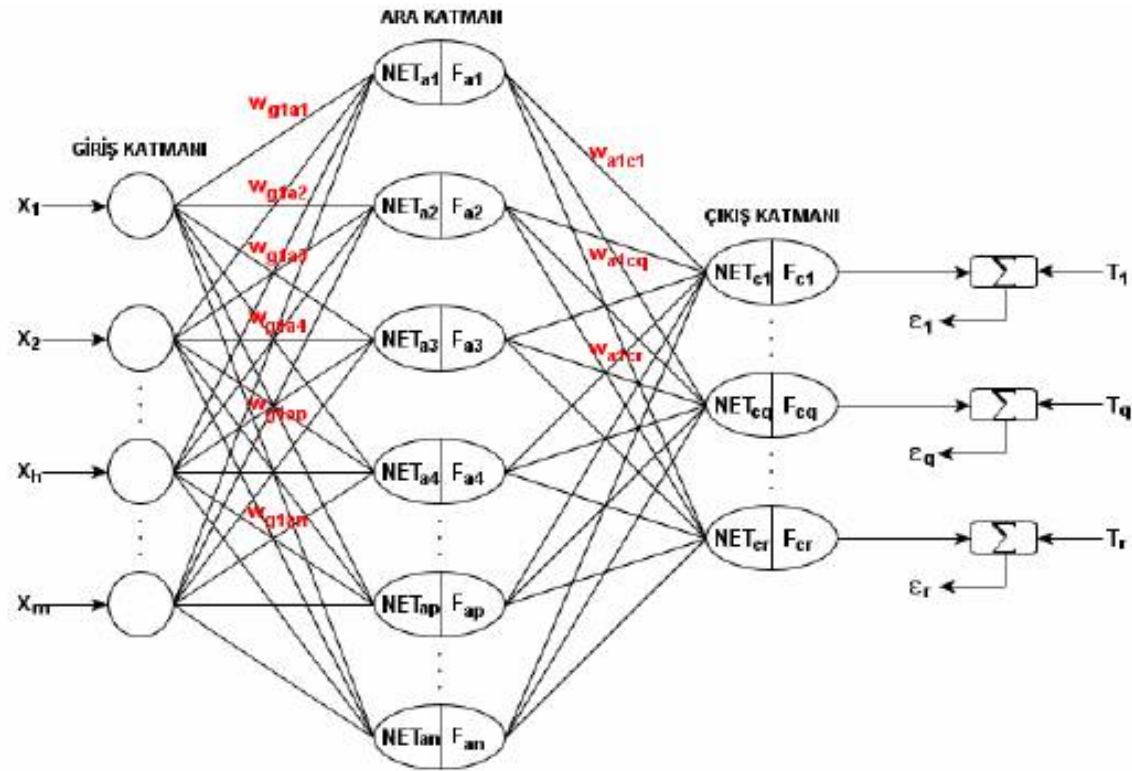
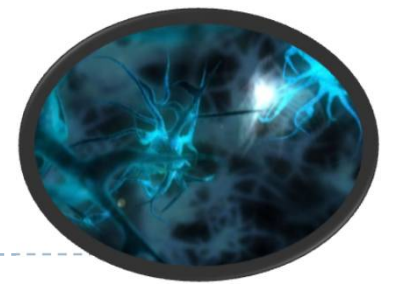
# I) İleri Doğru Hesaplama:

---

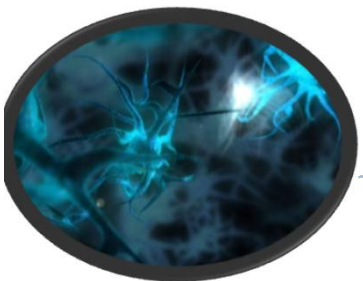
- ▶ Ara katmanın bütün elemanlar ve çıktı katmanının elemanlarının çıktıları aynı şekilde kendilerine gelen NET girdinin hesaplanması ve aktivasyon fonksiyonundan geçirilmesi sonucu belirlenirler.
- ▶ Çıktı katmanından çıkan değerler bulununca ağıın ileri doğru hesaplama işlemi tamamlanmış olur.



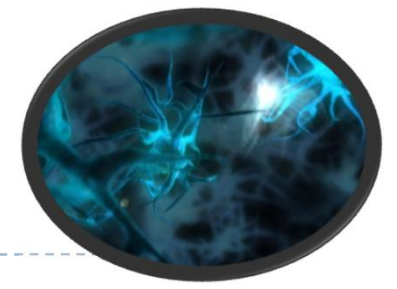




Öğücü, M. Orkun , 2006. Yapay Sinir Ağları ile Sistem Tanıma, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.





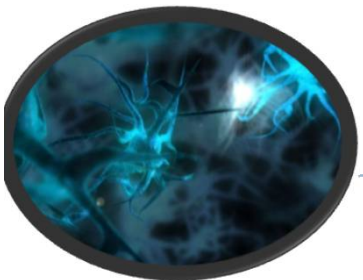


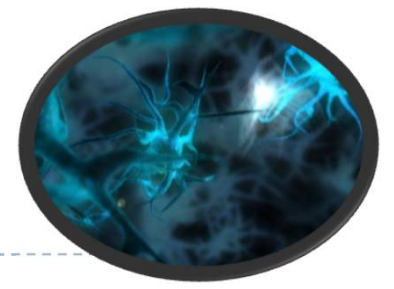
## II) Geriye Doğru Hesaplama

---

- ▶ Ağa sunulan girdi için ağın ürettiği çıktı ağın beklenen çıktıları ile karşılaştırılır. Bunların arasındaki fark hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir.
- ▶ Bu hata, ağın ağırlık değerlerine dağıtılarak bir sonraki iterasyonda hatanın azaltılması sağlanır.
- ▶ Çıktı katmanındaki m elemanı için oluşan hata  $E_m$  ;

$$E_m = B_m - C_m$$



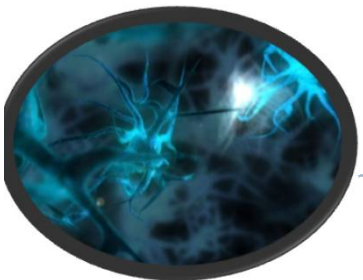


## II) Geriye Doğru Hesaplama

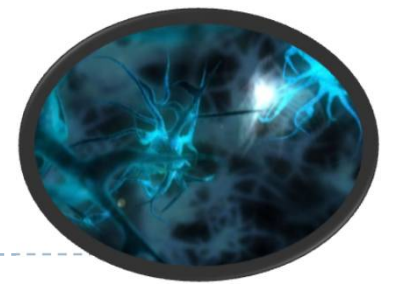
- ▶  $E_m$  hata, bir eleman için oluşan hatadır. Çıktı katmanı için oluşan toplam hatayı (TH) bulmak için bütün hataların toplanması gerekir.

$$TH = \frac{1}{2} \sum_m E_m^2$$

- ▶ Toplam hatayı en aza indirmek için bu hatanın kendisine neden olan elemanlarla dağıtılması gerekmektedir.
- ▶ Ağın ağırlıklarını değiştirmek için 2 durum söz konusudur:
  - ▶ Ara katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi
  - ▶ Ara katmanlar arası veya ara katman girdi katmanı arasındaki ağırlıkların değiştirilmesi



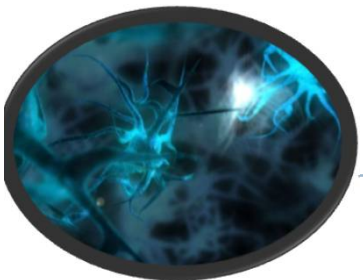
# Ara Katman ile Çıktı Katmanı Arasındaki Ağırlıkların Değiştirilmesi



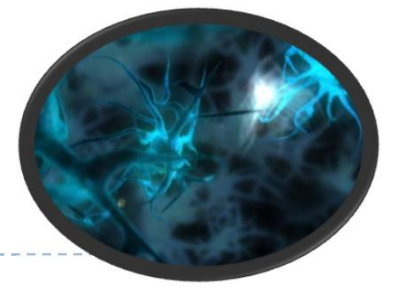
- ▶ Ara katmandaki  $j$ . Proses elemanı çıktı katmanındaki  $m$ . Proses elemanına bağlayan bağlantının ağırlığındaki değişim miktarına  $\Delta A^a$  denirse; herhangi bir  $t$  zamanında ağırlığın değişim miktarı şöyle hesaplanır:

$$\Delta A_{jm}^a(t) = \lambda \delta_m C_j^a + \alpha \Delta A_{jm}^a(t-1)$$

- ▶  $\lambda$  = öğrenme katsayısı (ağırlıkların değişim miktarını)
- ▶  $\alpha$  = momentum katsayısı (Momentum katsayısı ağırlığın öğrenmesi esnasında yerel bir optimum noktaya takılıp kalmaması için ağırlık değişim değerinin belirli bir oranda bir sonraki değişime eklenmesini sağlar.)
- ▶  $\delta$  =  $m$ . çıktı ünitesinin hatasını



# Ara Katman ile Çıktı Katmanı Arasındaki Ağırlıkların Değiştirilmesi



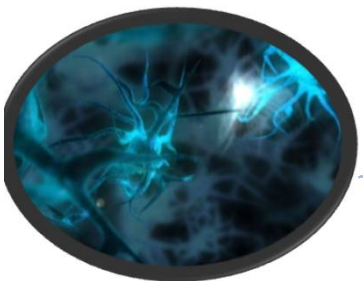
$$\delta_m = f'(NET)E_m$$

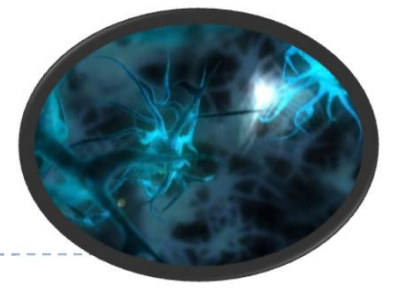
- ▶  $f'(NET)$  aktivasyon fonksiyonunun türevidir. Sigmoid fonksiyonun kullanılması durumunda

$$\delta_m = C_m (1 - C_m).E_m$$

- ▶ Değişim miktarı hesaplandıktan sonra ağırlıkların t.iterasyondaki yeni değerleri:

$$A_{jm}^a(t) = A_{jm}^a(t-1) + \Delta A_{jm}^a(t)$$



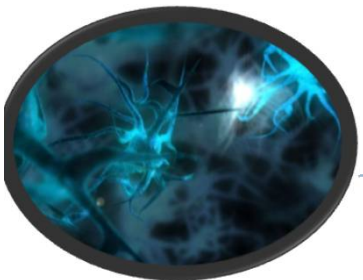


- Çıktı katmanındaki elemanların bias değer ağırlıklarının değişim miktarı

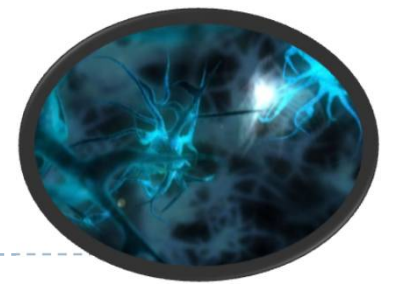
$$\Delta\beta_m^c(t) = \lambda\delta_m + \alpha\Delta\beta_m^c(t-1)$$

- t. İterasyonundaki ağırlığının yeni değeri ise ;

$$\beta_m^c(t) = \beta_m^c(t-1) + \Delta\beta_m^c(t)$$



# Ara Katmanlar Arası veya Ara Katman Girdi Katmanı Arasındaki Ağırlıkların Değiştirilmesi



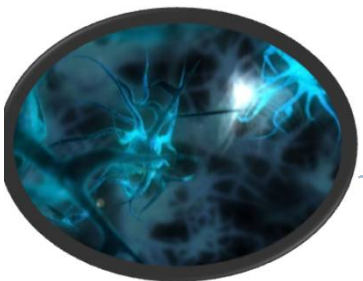
- ▶ Girdi katmanı ile ara katman arasındaki ağırlıklar

$$\Delta A_{kj}^i(t) = \lambda \delta_j^a \zeta_k^i + \alpha \Delta A_{kj}^i(t-1)$$

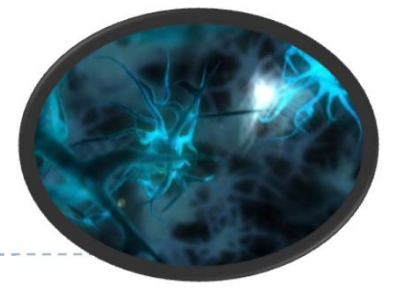
$$\delta_j^a = \zeta_j^a (1 - \zeta_j^a) \sum_m \delta_m A_{jm}^a$$

- ▶ Ağırlıkların yeni değeri:

$$A_{kj}^i(t) = A_{kj}^i(t-1) + \Delta A_{kj}^i(t)$$



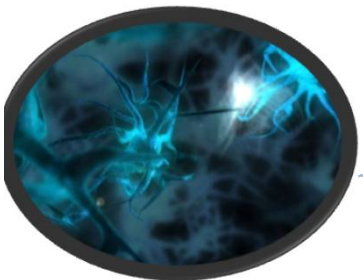
# Ara Katmanlar Arası veya Ara Katman Girdi Katmanı Arasındaki Ağırlıkların Değiştirilmesi



- Ara katman eşik değer ağırlıkları ve yeni değerleri ise sırasıyla:

$$\Delta B_j^a(t) = \lambda \delta_j^a + \alpha \Delta \beta_j^a(t-1)$$

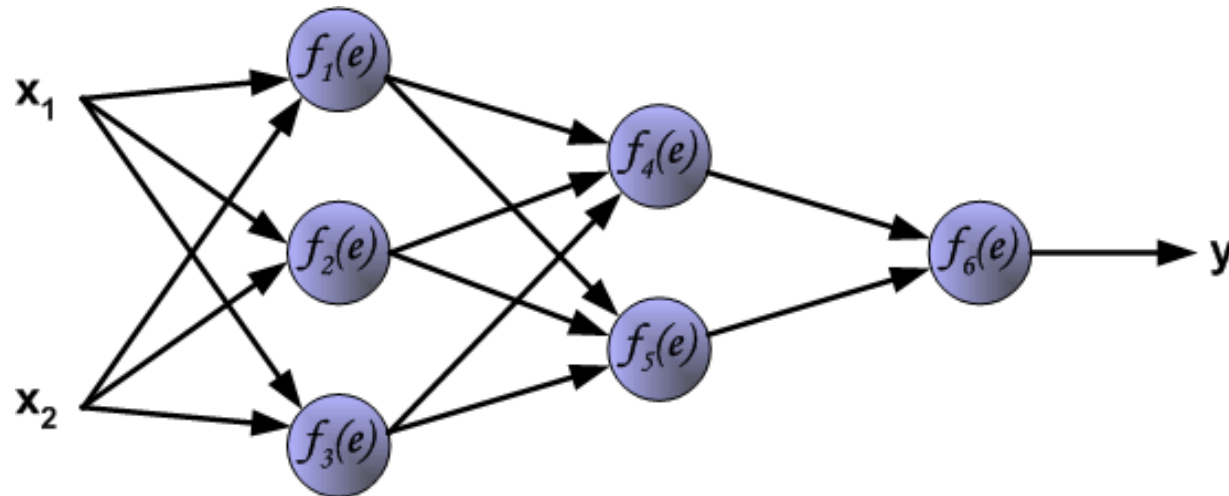
$$B_j^a(t) = B_j^a(t-1) + \Delta B_j^a(t)$$





# Backpropagation Adımları

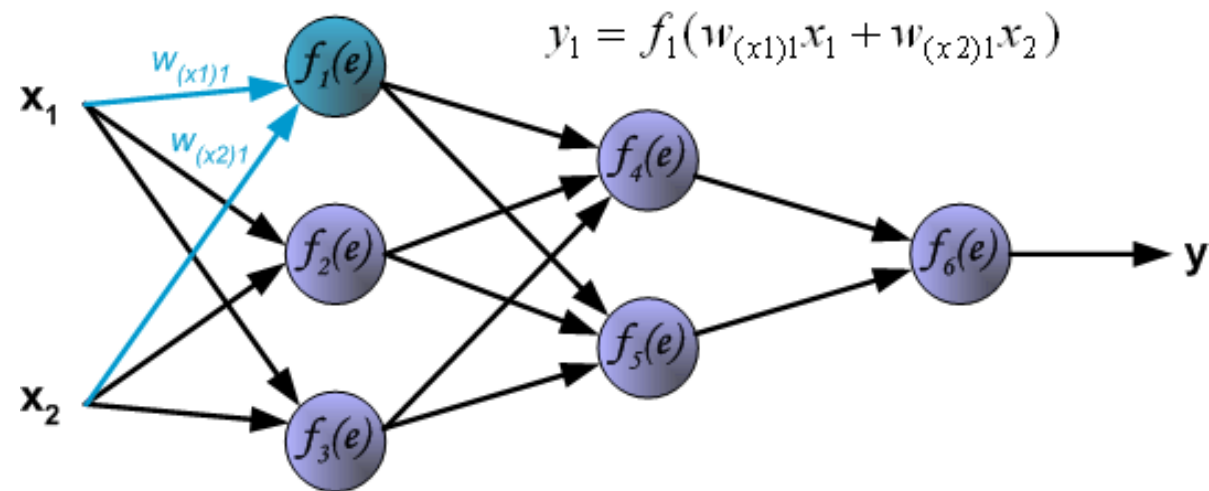
---



[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Erişim Tarihi: 19.11.2020)

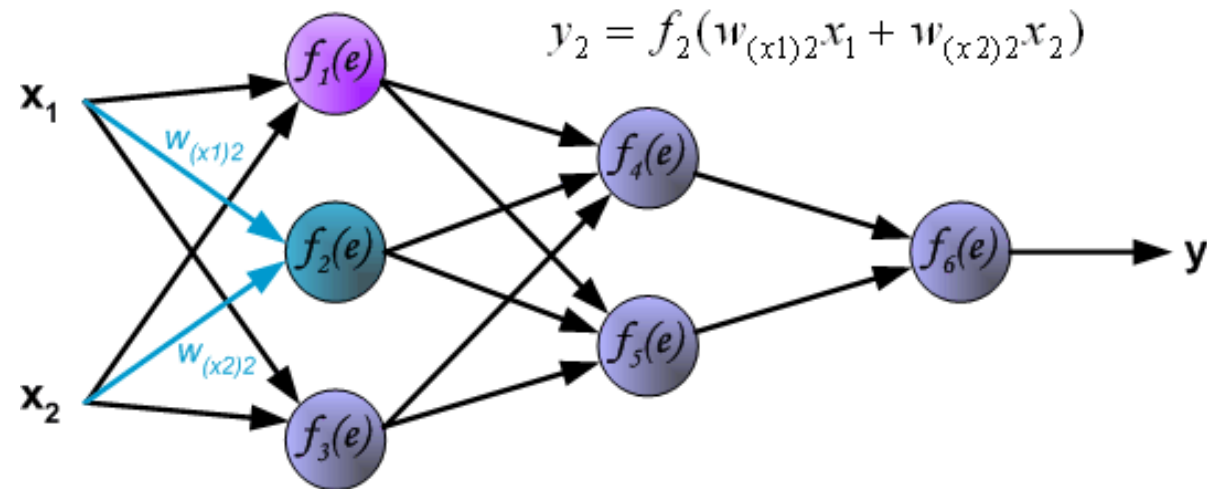


...



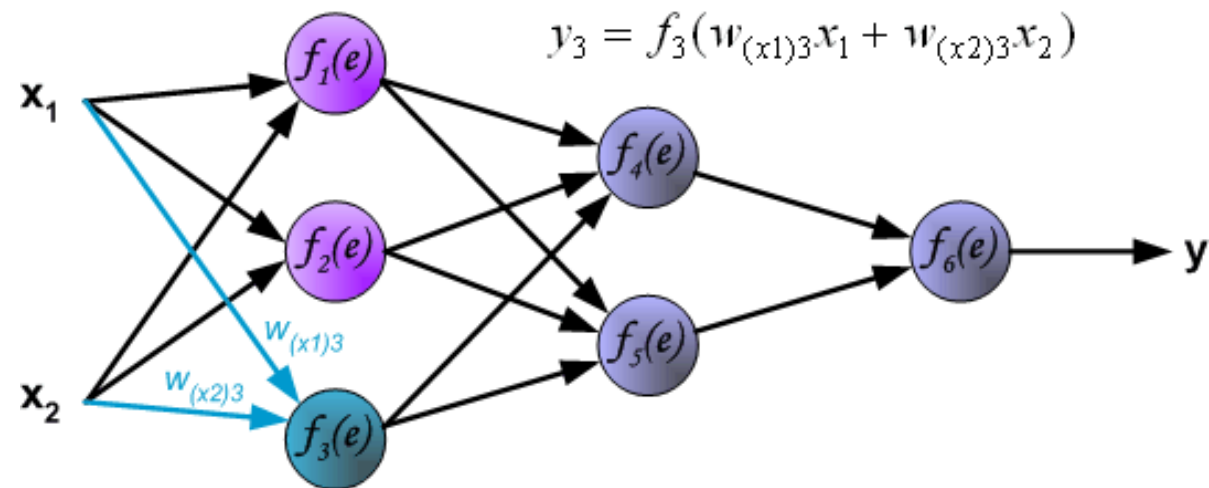
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



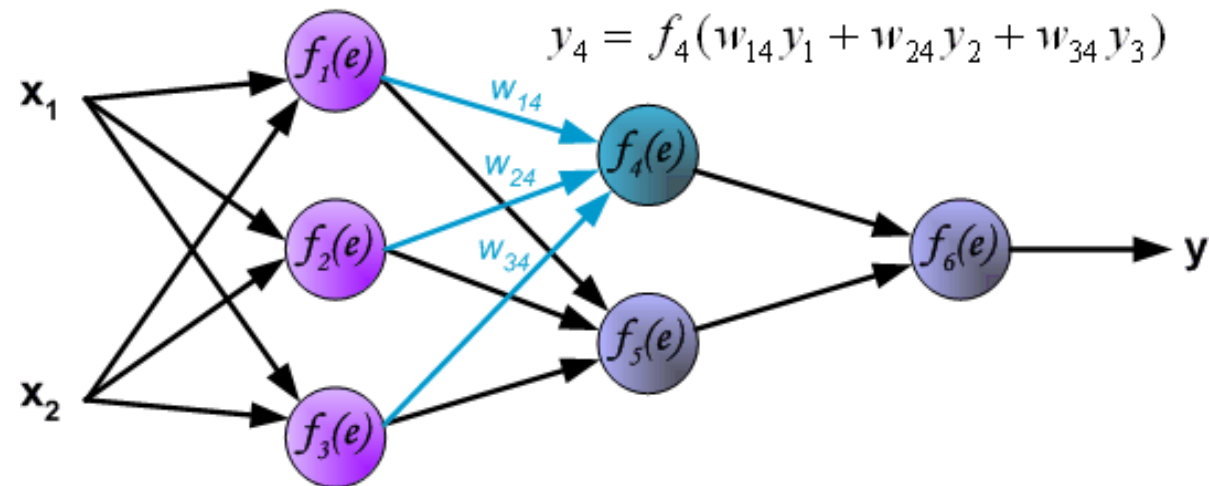
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



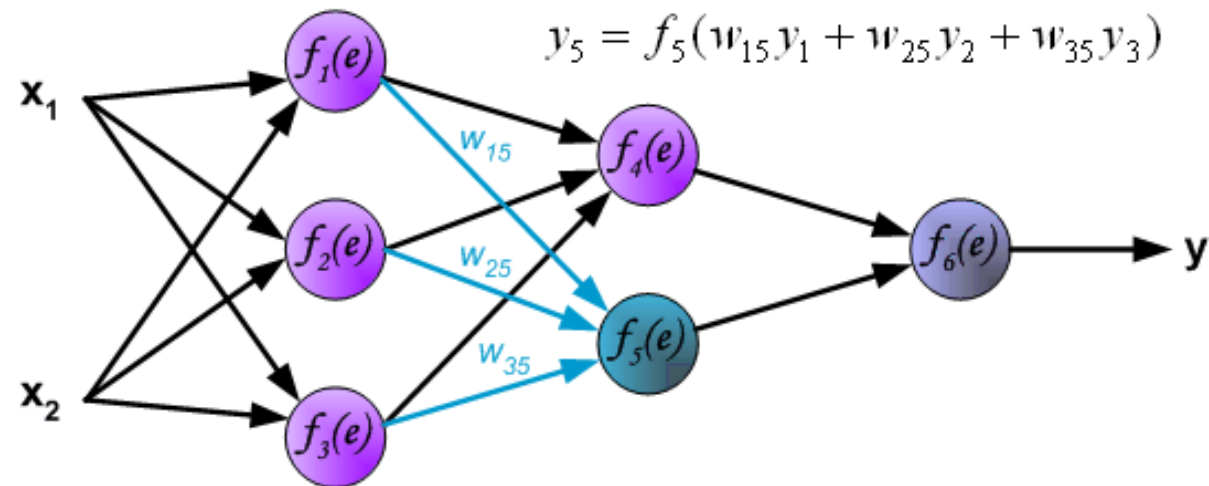
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



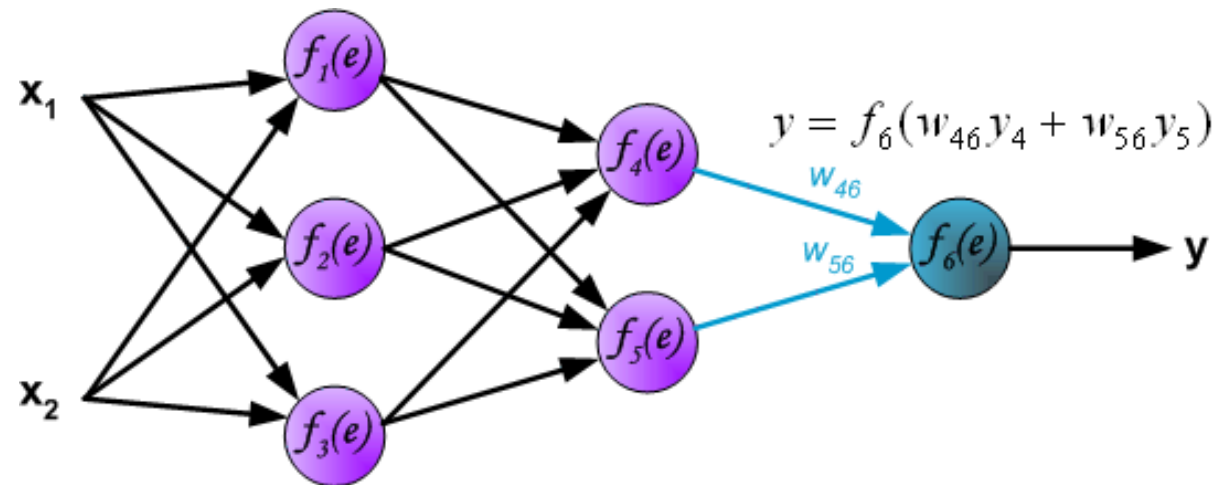
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi:19.11.2020)

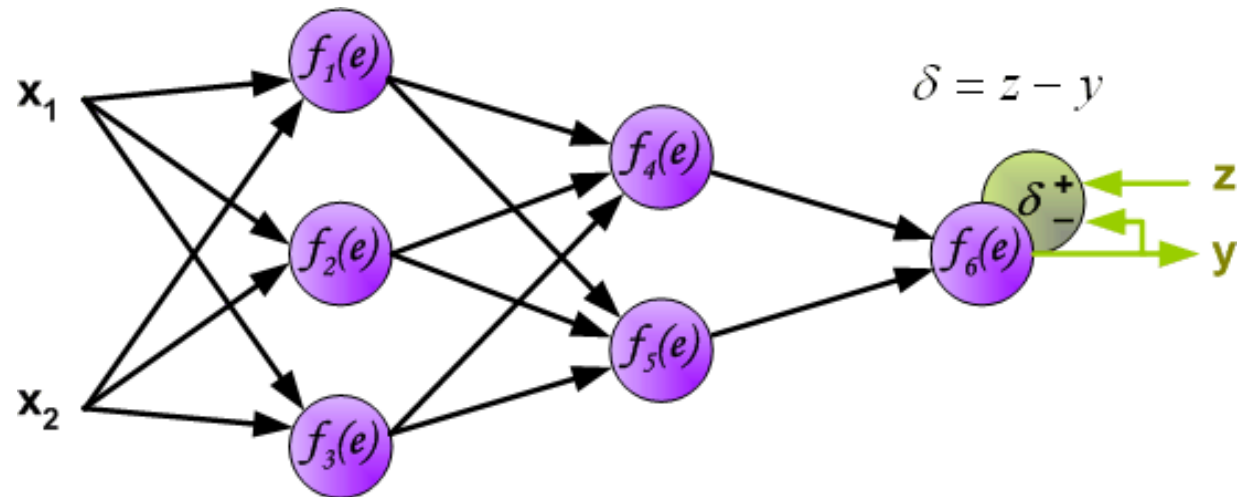
...



[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

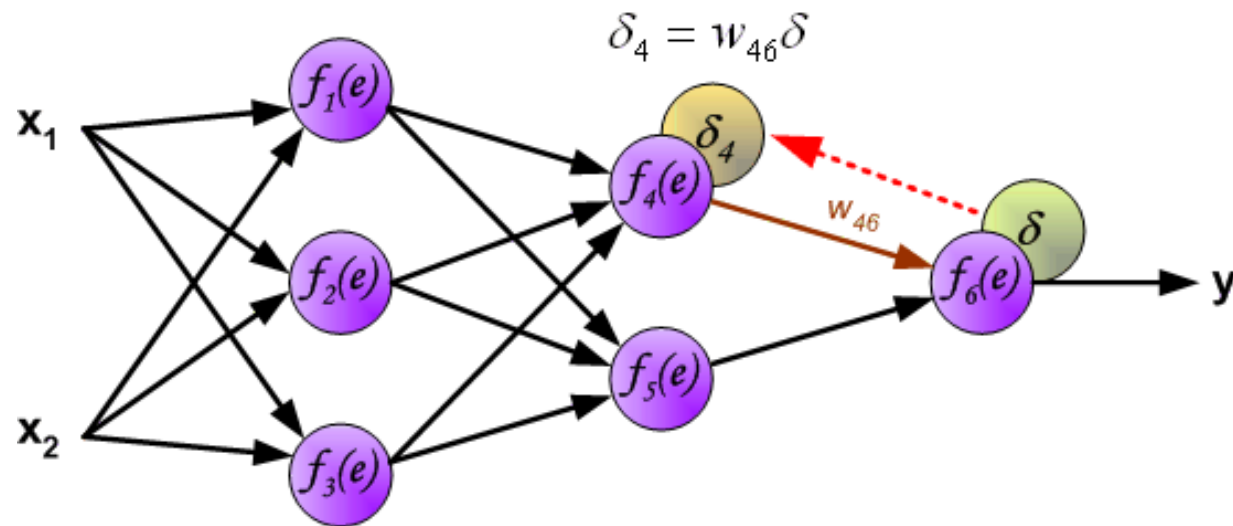


...



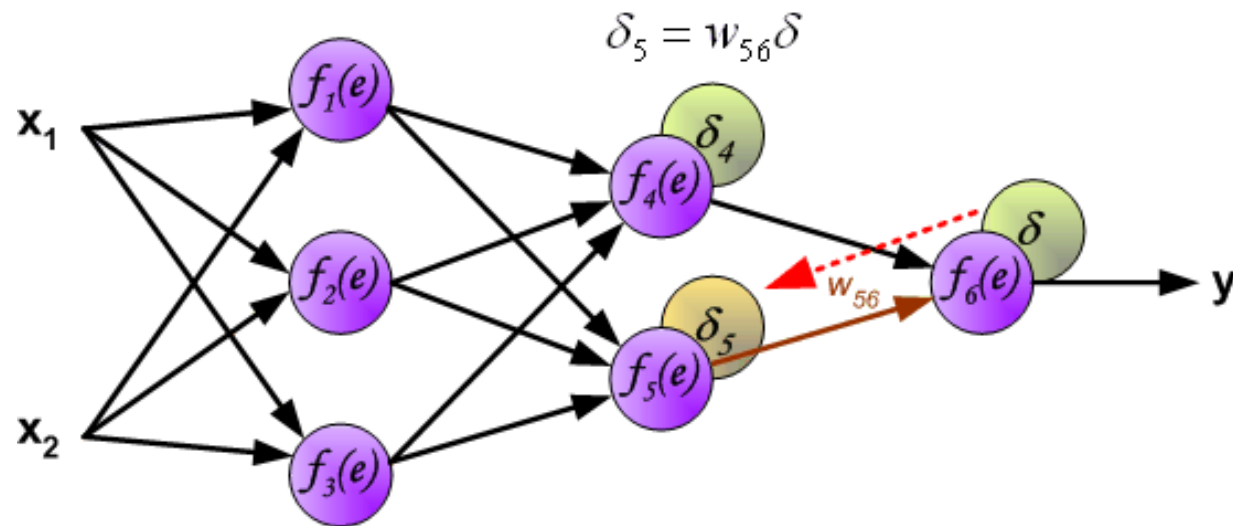
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi:19.11.2020)

...



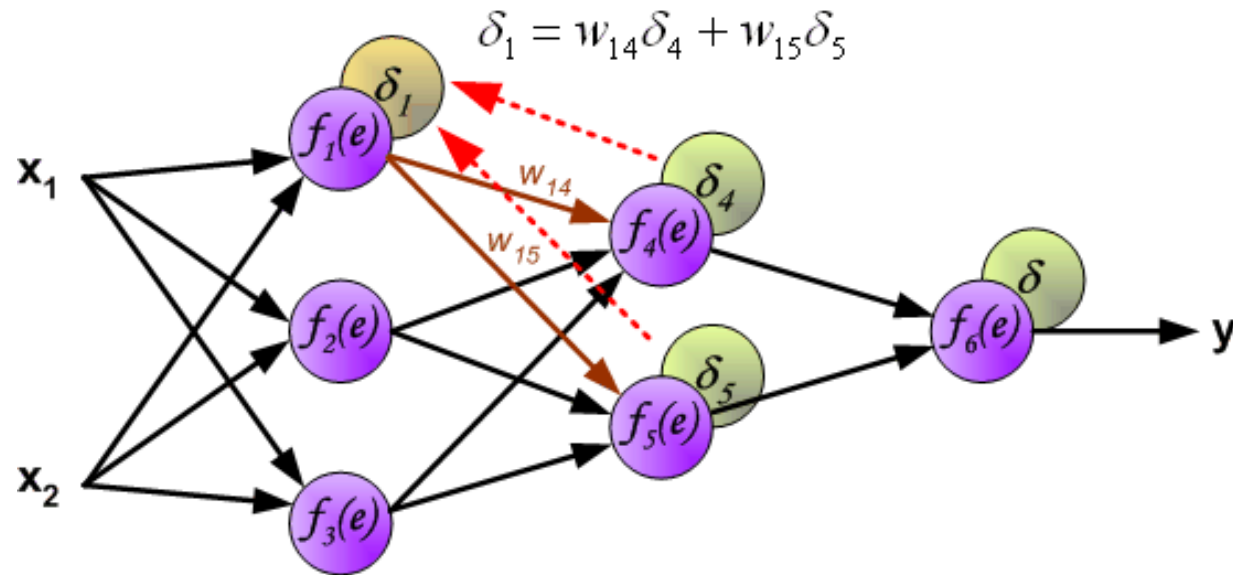
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



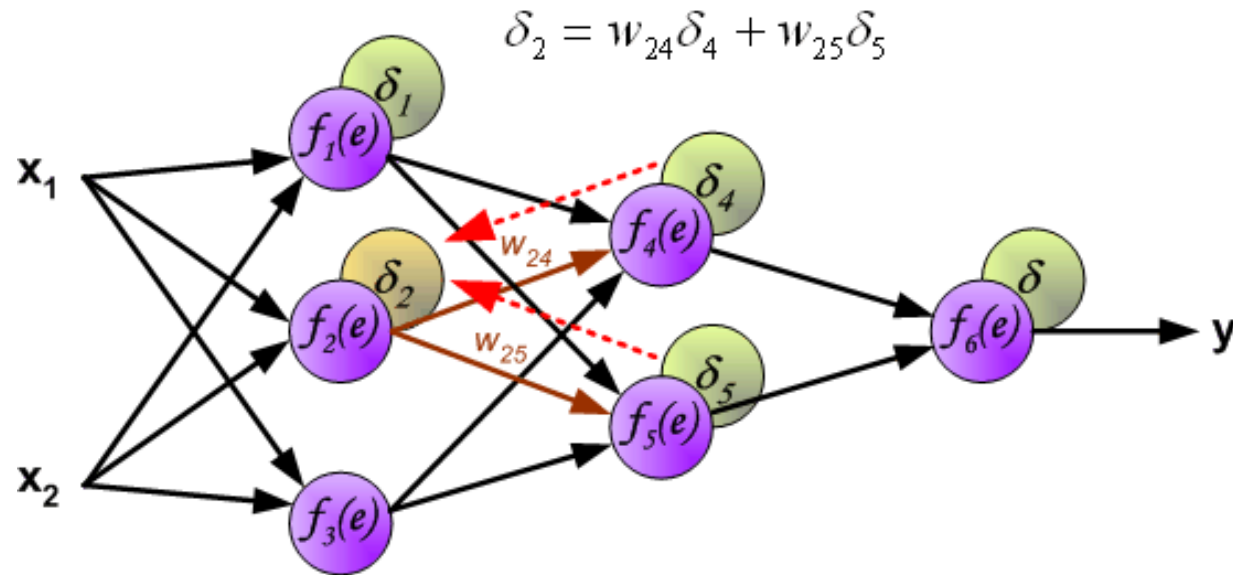
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) (Eriřim Tarihi: 19.11.2020)

...



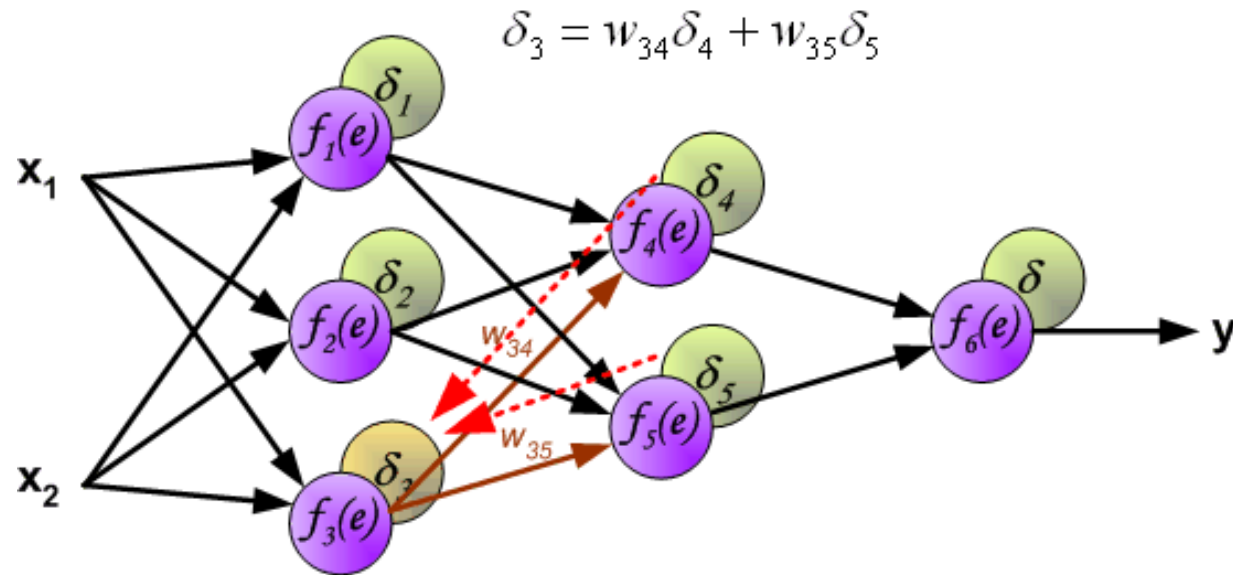
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)

...



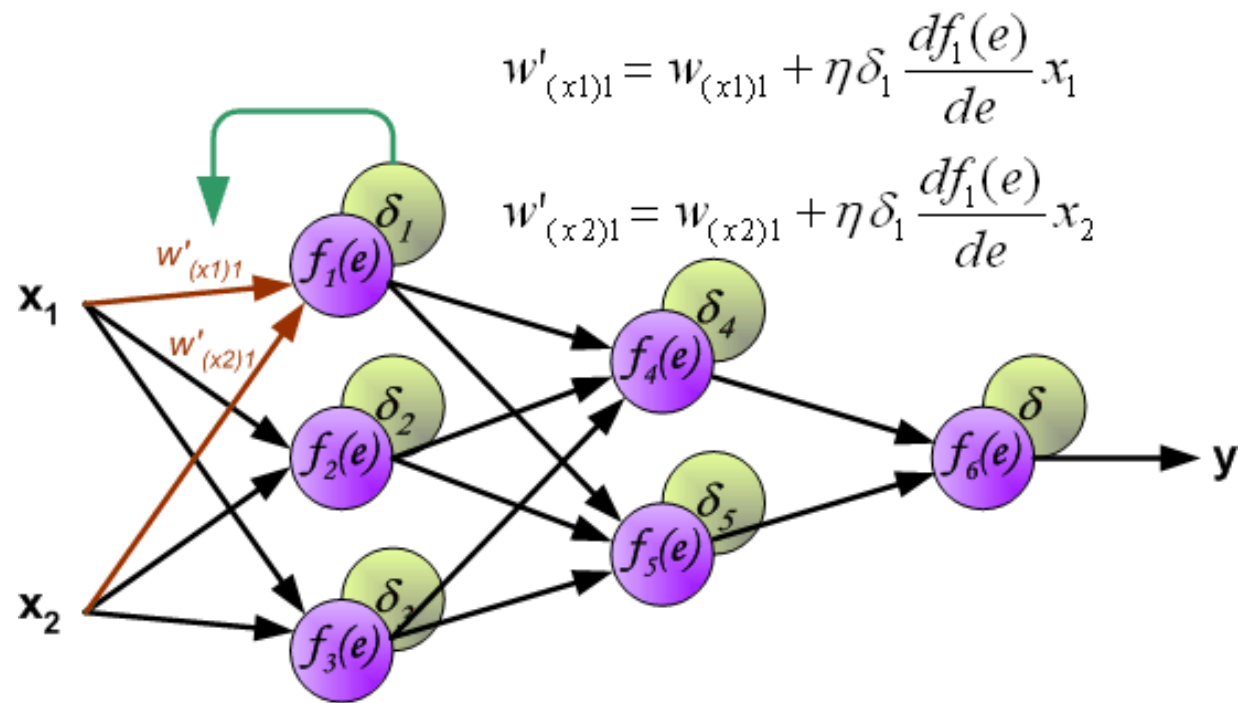
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Erişim Tarihi: 19.11.2020)

...



[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi:19.11.2020)

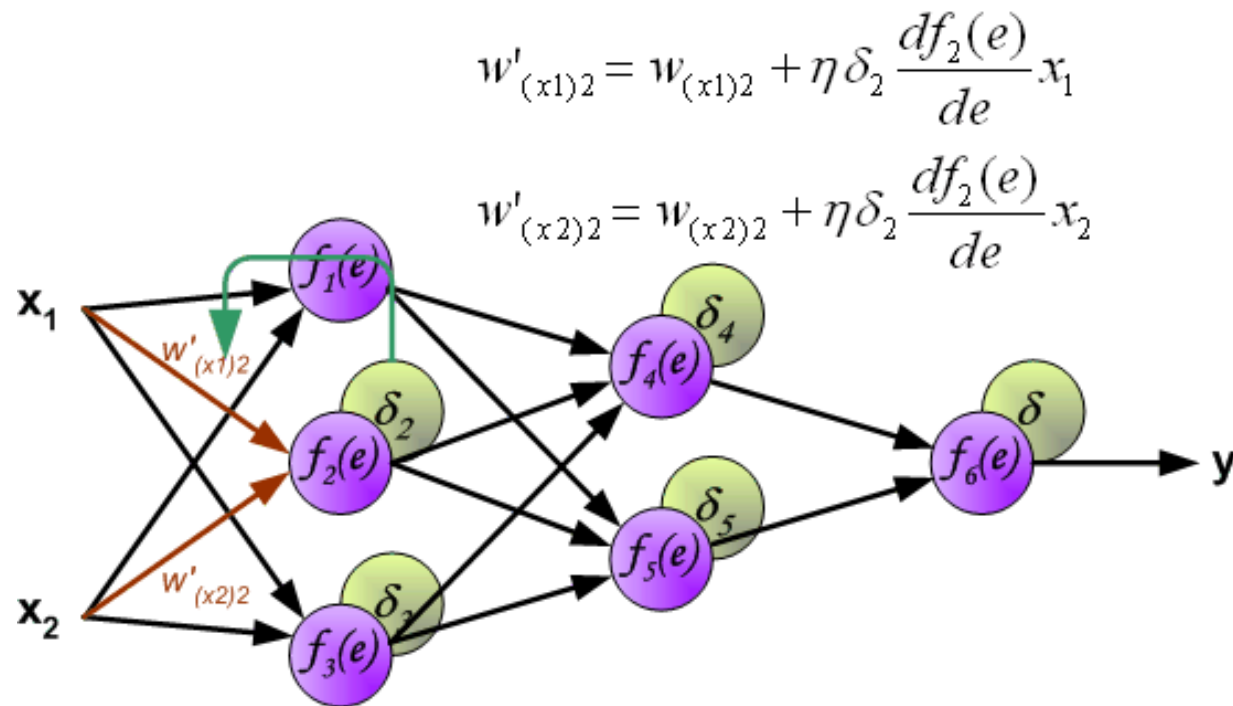
...



[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi:19.11.2020)

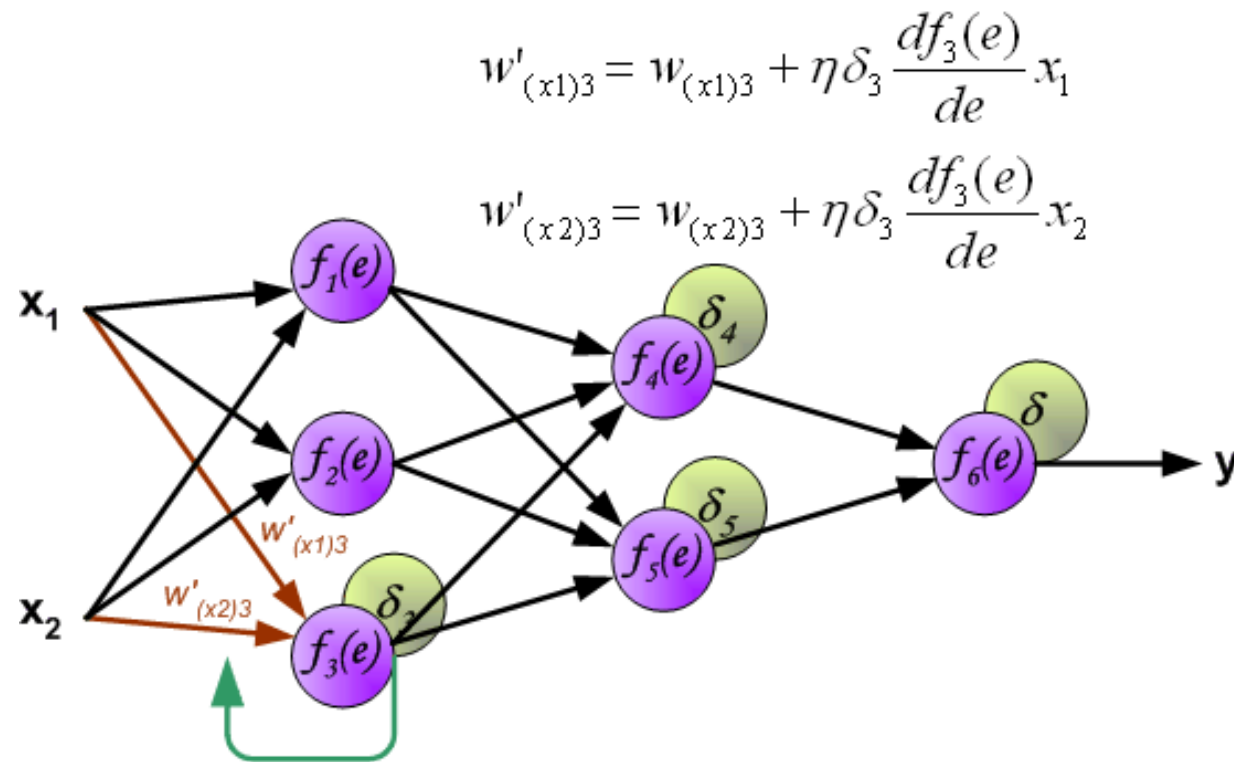


...



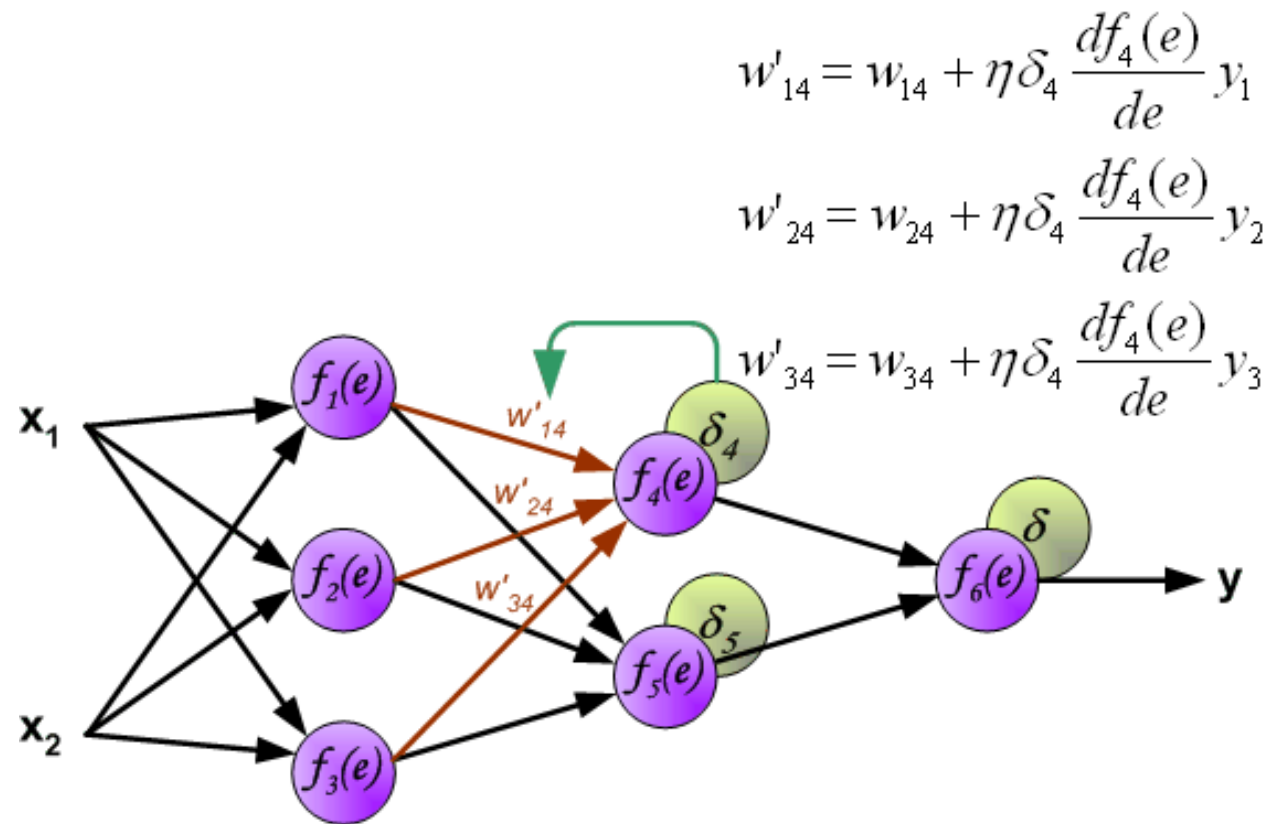
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Erişim Tarihi: 19.11.2020)

...



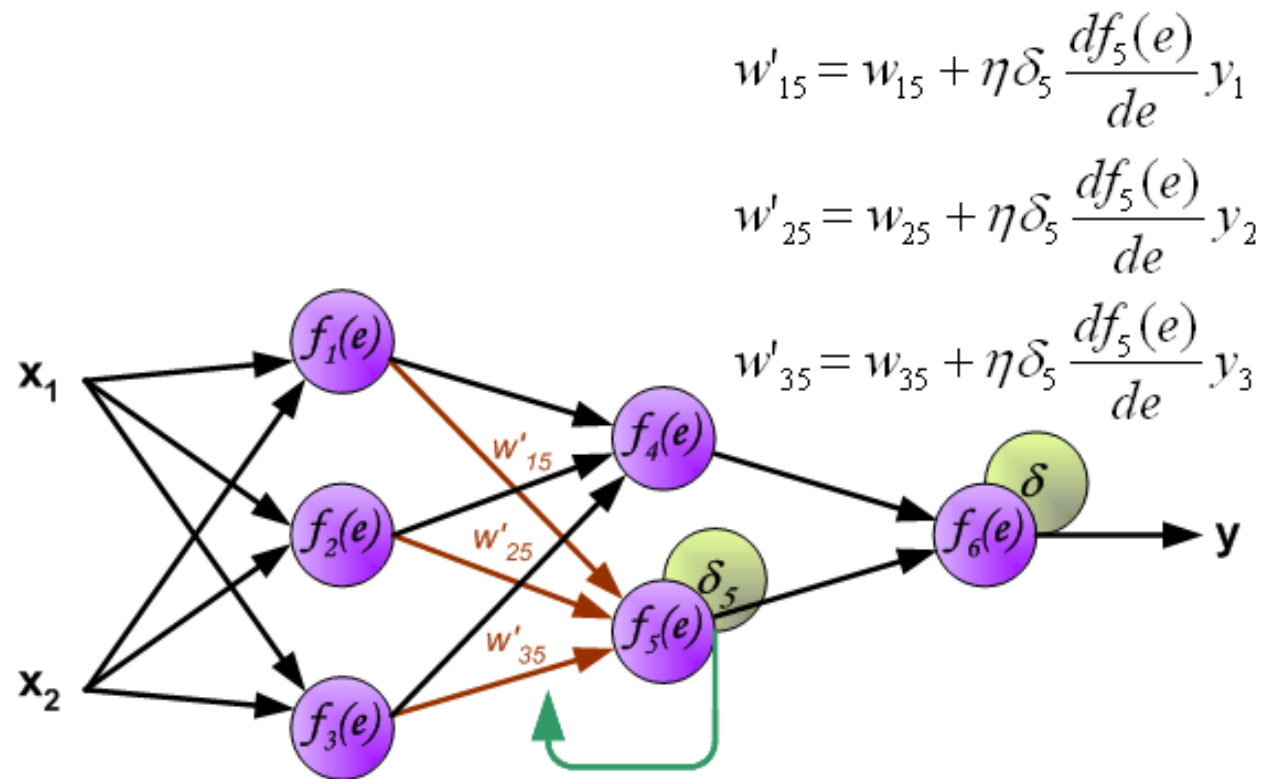
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) (Erişim Tarihi: 19.11.2020)

...



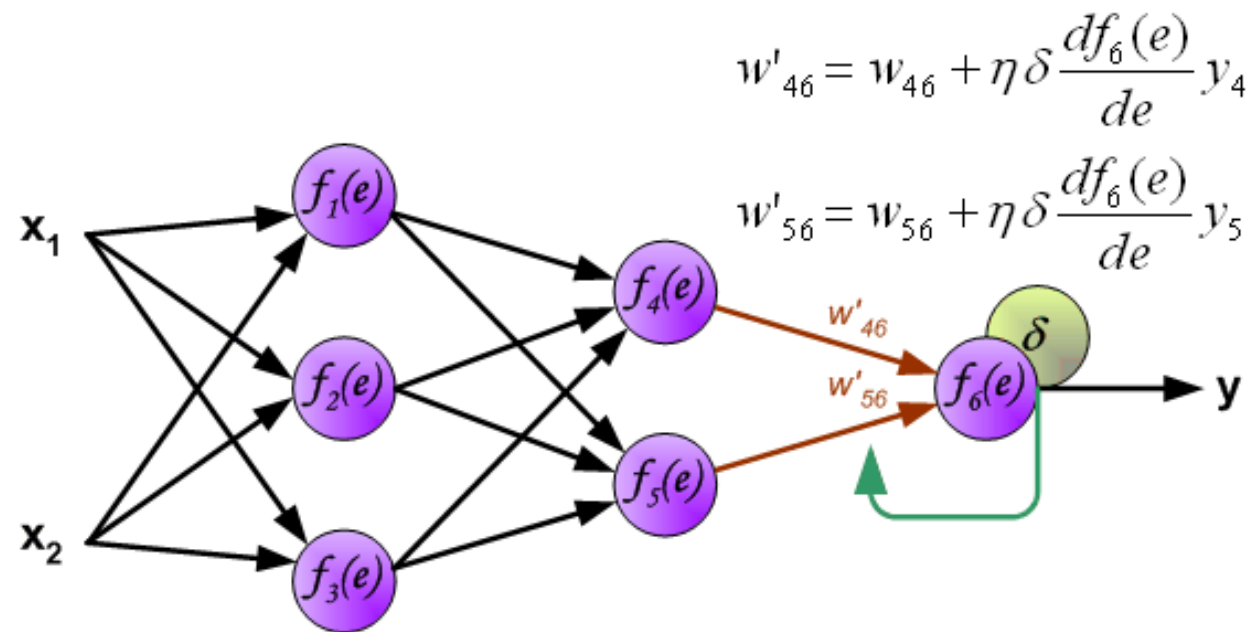
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) (Erişim Tarihi: 19.11.2020)

...

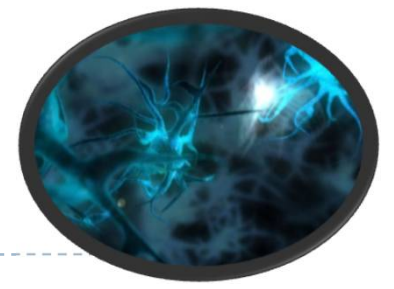


[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) (Erişim Tarihi: 19.11.2020)

...



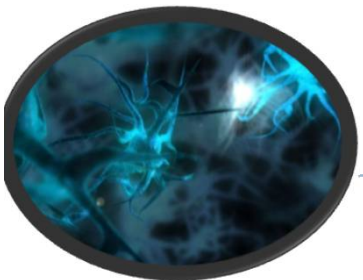
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Eriřim Tarihi: 19.11.2020)



# ÇKA Ağının Çalışma Prosedürü

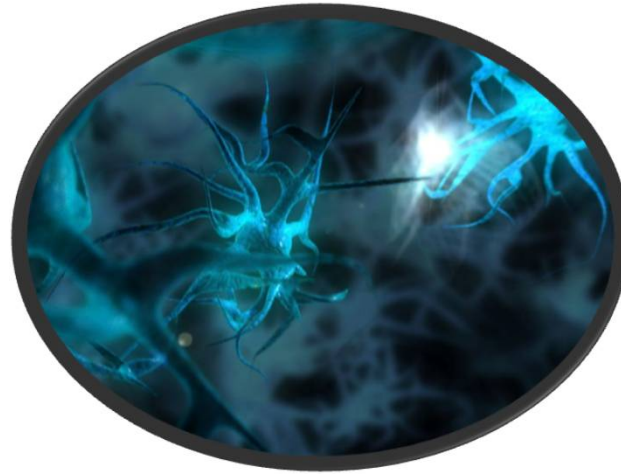
---

- ▶ 1. Örneklerin toplanması
- ▶ 2. Ağın topolojik yapısının belirlenmesi
- ▶ 3. Öğrenme parametrelerinin belirlenmesi
- ▶ 4. Ağırlıkların başlangıç değerlerinin atanması
- ▶ 5. Öğrenme setinden örneklerin seçilmesi ve ağa gösterilmesi
- ▶ 6. Öğrenme sırasında ileri hesaplamaların yapılması
- ▶ 7. Gerçekleşen çıktının beklenen çıktı ile karşılaştırılması
- ▶ 8. Ağırlıkların değiştirilmesi
- ▶ 9. Öğrenmenin tamamlanması



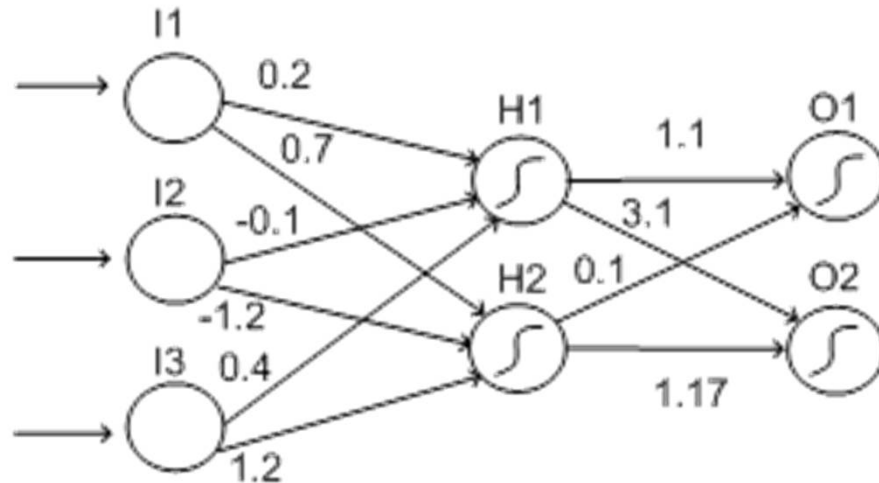
---

# ÖRNEK





# Örnekleri ileri sürmek



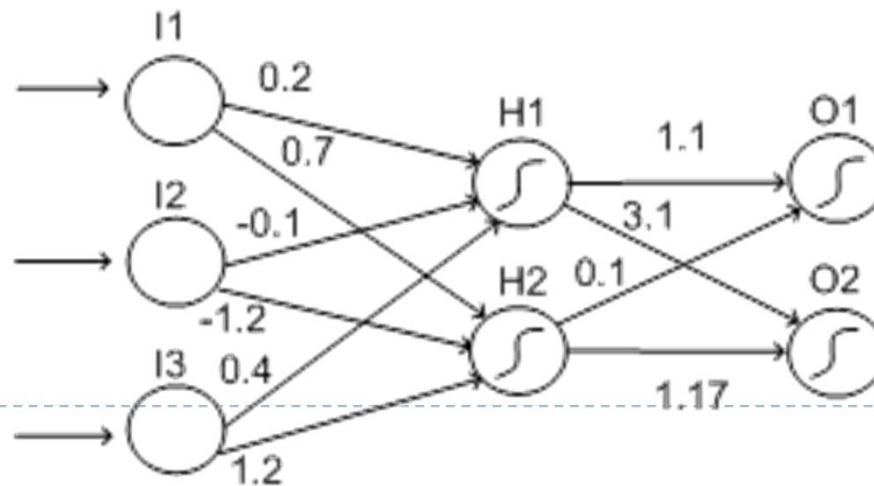
<http://www.sfu.ca/iat813/lectures/lecture8.html>

Ağa sunulan örnekler:  $E(10,30,20)$   
Beklenen Çıkış:  $t1(E)=1$   $t2(E)=0$

- ▶ Gizli katmanlarda ağırlıklı toplamalar:
  - ▶  $NET_{H1} = (0.2*10) + (-0.1*30) + (0.4*20) = 2-3+8 = 7$
  - ▶  $NET_{H2} = (0.7*10) + (-1.2*30) + (1.2*20) = 7-36+24 = -5$
- ▶ Bir sonraki adımda gizli katman çıkışları hesaplanır:
  - ▶  $\sigma(NET) = 1/(1 + e^{-S})$
  - ▶  $\sigma(NET_{H1}) = 1/(1 + e^{-7}) = 1/(1+0.000912) = 0.999$
  - ▶  $\sigma(NET_{H2}) = 1/(1 + e^5) = 1/(1+148.4) = 0.0067$

...

- ▶ Ağırlıklı toplamaları çıkış katmanında hesapla:
  - ▶  $NET_{O1} = (1.1 * 0.999) + (0.1 * 0.0067) = 1.0996$
  - ▶  $NET_{O2} = (3.1 * 0.999) + (1.17 * 0.0067) = 3.1047$
- ▶ Son olarak ANN çıkışını hesapla:
  - ▶  $\sigma(NET_{O1}) = 1/(1+e^{-1.0996}) = 1/(1+0.333) = 0.750$
  - ▶  $\sigma(NET_{O2}) = 1/(1+e^{-3.1047}) = 1/(1+0.045) = 0.957$

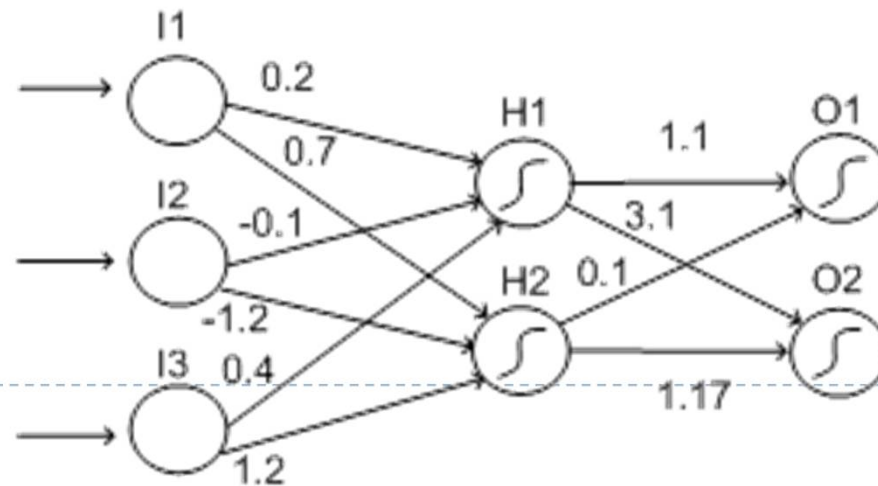


# Örnekleri ileri doğru sürdüğümüzde elde edilen hesaplamalar

Ağa sunulan örnekler E(10,30,20)

$$\eta = 0.1$$

Input units		Hidden units			Output units		
Unit	Output	Unit	Weighted Sum Input	Output	Unit	Weighted Sum Input	Output
I1	10	H1	7	0.999	O1	1.0996	0.750
I2	30	H2	-5	0.0067	O2	3.1047	0.957
I3	20						



# Çıkış Birimlerinin Hata Değerleri

---

- ▶  $o_1(E)=0.750$ ,  $o_2(E)=0.957$
- ▶ Beklenen Çıkış:  $t_1(E)=1$   $t_2(E)=0$

Ağdaki her  $k$  çıkış birimi için, bu çıkış biriminin  $\delta_k$  hata terimini hesapla.

$$\circ \quad \delta_k = o_k(1 - o_k)(t_k - o_k)$$

$$\delta_{O1} = o_1(E)(1 - o_1(E))(t_1(E) - o_1(E)) = 0.750(1-0.750)(1-0.750) = 0.0469$$

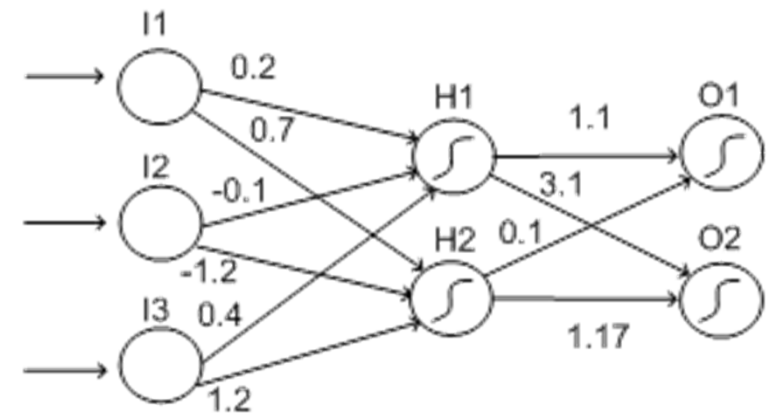
$$\delta_{O2} = o_2(E)(1 - o_2(E))(t_2(E) - o_2(E)) = 0.957(1-0.957)(0-0.957) = -0.0394$$



# Gizli Birimler için Hata Değerleri

ağıdaki her h gizli katmanı için, gizli birim  $\delta_k$  hatasını hesapla

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} w_{kh} \delta_k$$



▶  $\delta_{O1} = 0.0469$  ve  $\delta_{O2} = -0.0394$

▶  $h_1(E) = 0.999$  ve  $h_2(E) = 0.0067$

▶ H1 için:

▶  $(w_{11} * \delta_{O1}) + (w_{12} * \delta_{O2}) = (1.1 * 0.0469) + (3.1 * -0.0394) = -0.0706$

▶ Ve bu değeri,  $h_1(E)(1-h_1(E))$  ile çarpalım, sonuç:

▶  $(0.999) * (1-0.999) * -0.0706 = -0.0000705 = \delta_{H1}$

▶ H2 için:

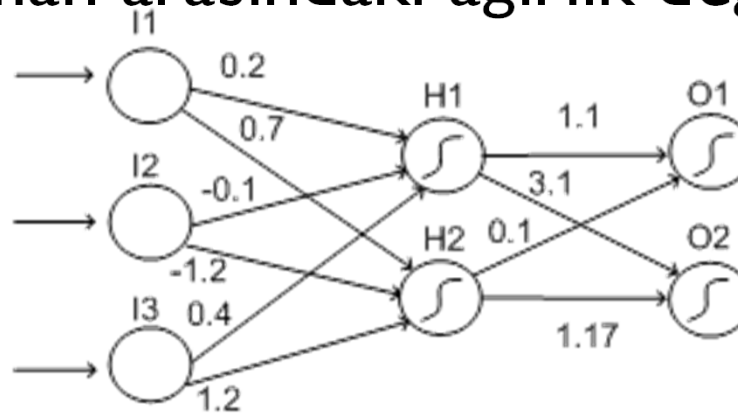
▶  $(w_{21} * \delta_{O1}) + (w_{22} * \delta_{O2}) = (0.1 * 0.0469) + (1.17 * -0.0394) = -0.0414$

▶ Ve bu değeri,  $h_2(E)(1-h_2(E))$  ile çarpalım, sonuç:

▶  $0.0067 * (1-0.0067) * -0.0414 = -0.0002755 = \delta_{H2}$

# Ağırlık güncellemeleri için hesaplamalar

- Giriş ve gizli katman arasındaki ağırlık değerleri:

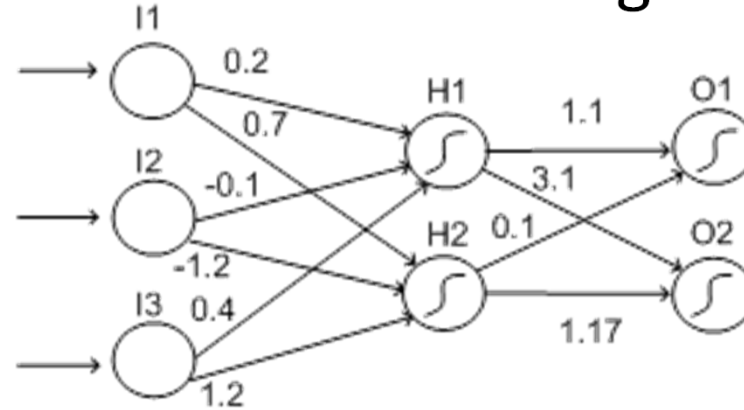


Input unit	Hidden unit	$\eta$	$\delta_H$	$x_i$	$\Delta = \eta * \delta_H * x_i$	Old weight	New weight
I1	H1	0.1	-0.0000705	10	-0.0000705	0.2	0.1999295
I1	H2	0.1	-0.0002755	10	-0.0002755	0.7	0.6997245
I2	H1	0.1	-0.0000705	30	-0.0002115	-0.1	-0.1002115
I2	H2	0.1	-0.0002755	30	-0.0008265	-1.2	-1.2008265
I3	H1	0.1	-0.0000705	20	-0.000141	0.4	0.39999
I3	H2	0.1	-0.0002755	20	-0.000551	1.2	1.199449

$$W_{yeni} = W_{eski} + \eta \delta X$$

# Ağırlık güncellemeleri için hesaplamalar

- Gizli ve çıkış katmanı arasındaki ağırlık değerleri:



Hidden unit	Output unit	$\eta$	$\delta_O$	$h_i(E)$	$\Delta = \eta * \delta_O * h_i(E)$	Old weight	New weight
H1	O1	0.1	0.0469	0.999	0.000469	1.1	1.100469
H1	O2	0.1	-0.0394	0.999	-0.00394	3.1	3.0961
H2	O1	0.1	0.0469	0.0067	0.00314	0.1	0.10314
H2	O2	0.1	-0.0394	0.0067	-0.0000264	1.17	1.16998

$$W_{yeni} = W_{eski} + \eta \delta X$$



# KAYNAKLAR

---

- ▶ **Öztemel, E.**, 2003. *Yapay Sinir Ağları*, Papatya Yayıncılık, İstanbul.
- ▶ [http://www.ibrahimcayiroglu.com/Dokumanlar/Goruntusleme/Goruntu\\_Isleme\\_Ders\\_Notlari-II.Hafta.pdf](http://www.ibrahimcayiroglu.com/Dokumanlar/Goruntusleme/Goruntu_Isleme_Ders_Notlari-II.Hafta.pdf)
- ▶ Öğücü, M. Orkun , 2006.Yapay Sinir Ağları ile Sistem Tanıma, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- ▶ [http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html) ( Erişim Tarihi:19.11.2020)
- ▶ <http://www.sfu.ca/iat813/lectures/lecture8.html> [Erişim Tarihi: 13.12.2020]

