

Εισαγωγή στην Τεχνητή Νοημοσύνη

(ΜΠΔ 306)

Εργαστήριο

Εργαστηριακός βοηθός: Κατερίνα Γιαννακάκη

e-mail: agiannakaki1@isc.tuc.gr

Χειμερινό εξάμηνο 2020-2021

Artificial Neural Networks

Τεχνητά Νευρωνικά Δίκτυα

Εργαστήριο 7ο

Μάθηση με ελαχιστοποίηση σφάλματος

ΓΙΑ ΤΝΔ ΧΩΡΙΣ ΚΡΥΦΑ ΕΠΙΠΕΔΑ

υπολογισμός κάθε νέου βάρους:

$$w_i^{(new)} = w_i^{(old)} + \Delta w_i = w_i^{(old)} + d \cdot (y - o) \cdot x_i$$

όπου

x_i : η επιμέρους είσοδος i της οποίας
το βάρος αναπροσαρμόζεται

w_i : το βάρος που αντιστοιχεί στην είσοδο x_i

y : έξοδος-στόχος του νευρώνα

o : πραγματική έξοδος του νευρώνα $o = f\left(\sum_i w_i x_i\right)$

d : ρυθμός μάθησης

Μάθηση με ελαχιστοποίηση σφάλματος

ΓΙΑ ΤΝΔ ΧΩΡΙΣ ΚΡΥΦΑ ΕΠΙΠΕΔΑ

υπολογισμός κάθε νέου βάρους:

$$w_i^{(new)} = w_i^{(old)} + \Delta w_i = w_i^{(old)} + d \cdot (y - o) \cdot x_i$$

όπου

x_i : η επιμέρους είσοδος i της οποίας
το βάρος αναπροσαρμόζεται

w_i : το βάρος που αντιστοιχεί στην είσοδο x_i

y : **έξοδος-στόχος** του νευρώνα

o : πραγματική έξοδος του νευρώνα $o = f\left(\sum_i w_i x_i\right)$

d : ρυθμός μάθησης

δεν γνωρίζουμε την έξοδο στόχο ενός κρυφού νευρώνα

Μάθηση με ελαχιστοποίηση σφάλματος

ΑΝΑΓΩΓΗ ΣΕ ΔΙΚΤΥΟ

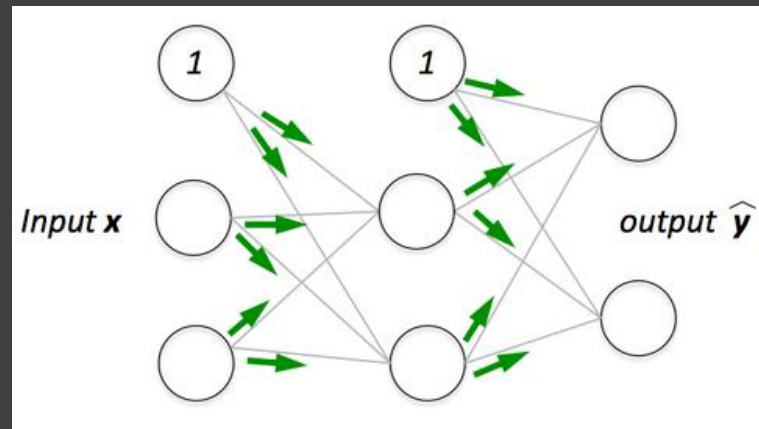
Όταν υπάρχουν κρυφά επίπεδα, δεν γίνεται εφαρμογή της προηγούμενης μεθόδου

Μέθοδος που ακολουθούμε υπό προϋποθέσεις:
Back-Propagation of the Error

Μάθηση με ελαχιστοποίηση σφάλματος

ΓΕΝΙΚΑ για feedforward ΤΝΔ

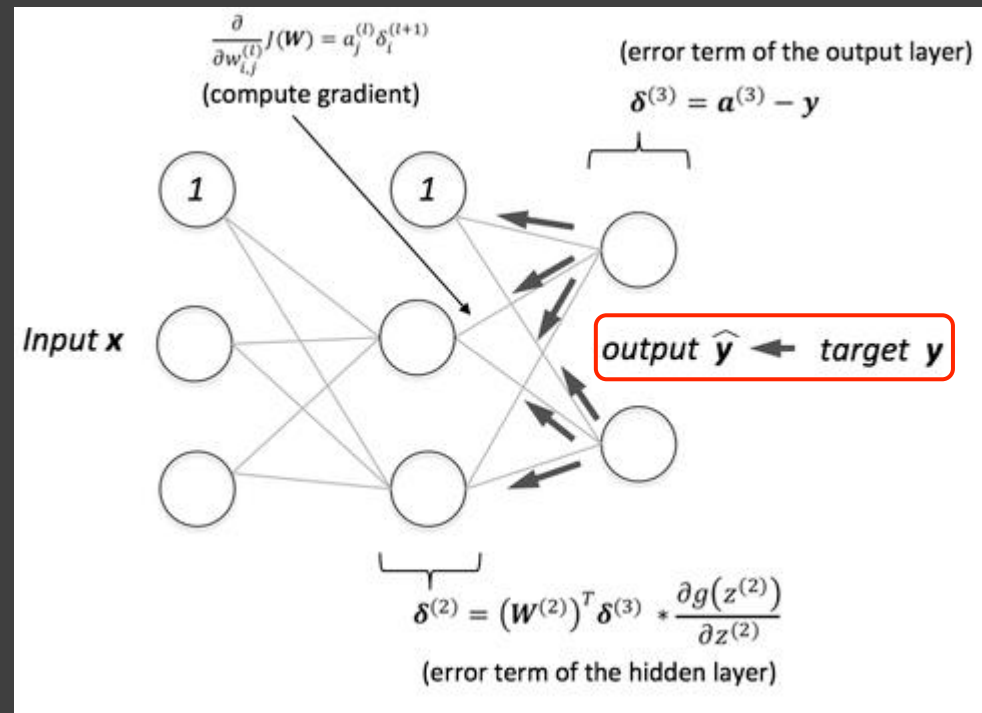
- Υπολογισμός εξόδων όλων των νευρώνων με δεδομένα βάρη, ξεκινώντας από επίπεδο εισόδου και προχωρώντας προς το επίπεδο εξόδου



Μάθηση με ελαχιστοποίηση σφάλματος

ΓΕΝΙΚΑ για feedforward ΤΝΔ

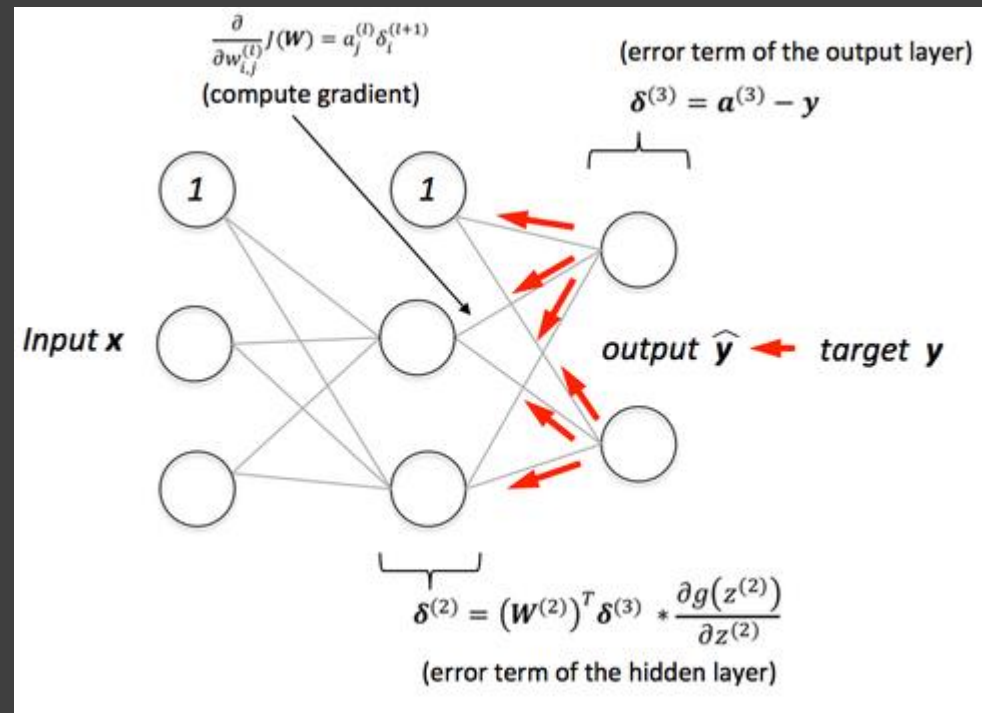
- Υπολογισμός εξόδων όλων των νευρώνων με δεδομένα βάρη, ξεκινώντας από επίπεδο εισόδου και προχωρώντας προς το επίπεδο εξόδου
- Υπολογισμός απόκλισης τιμής εξόδου του δικτύου από **επιθυμητή** τιμή



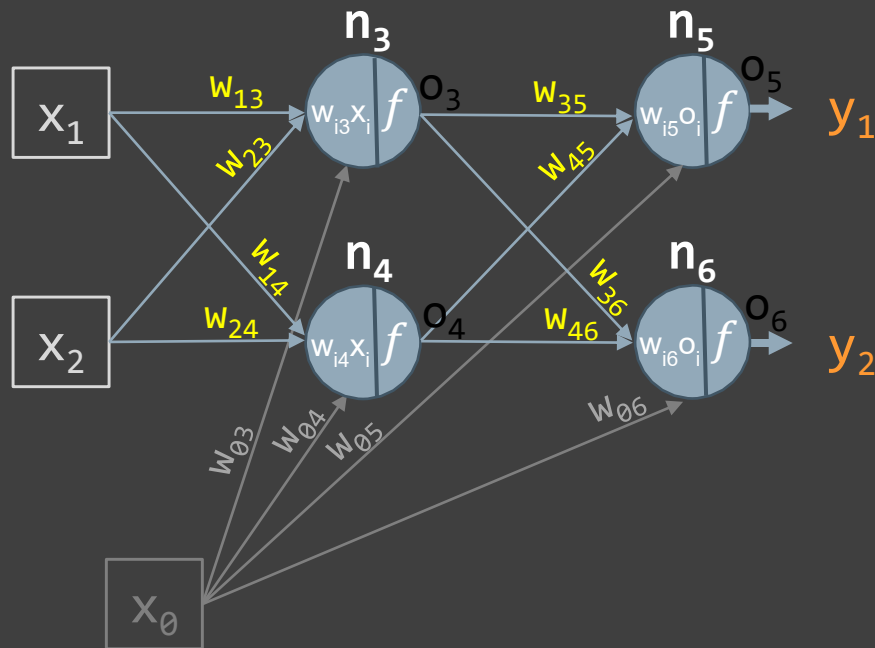
Μάθηση με ελαχιστοποίηση σφάλματος

ΓΕΝΙΚΑ για feedforward ΤΝΔ

- Υπολογισμός εξόδων όλων των νευρώνων με δεδομένα βάρη, ξεκινώντας από επίπεδο εισόδου και προχωρώντας προς το επίπεδο εξόδου
- Υπολογισμός απόκλισης τιμής εξόδου του δικτύου από **επιθυμητή** τιμή
- Αναπροσαρμογή των βαρών ξεκινώντας από επίπεδο εξόδου και προχωρώντας προς επίπεδο εισόδου (back-propagation of the error)



Μάθηση με ελαχιστοποίηση σφάλματος



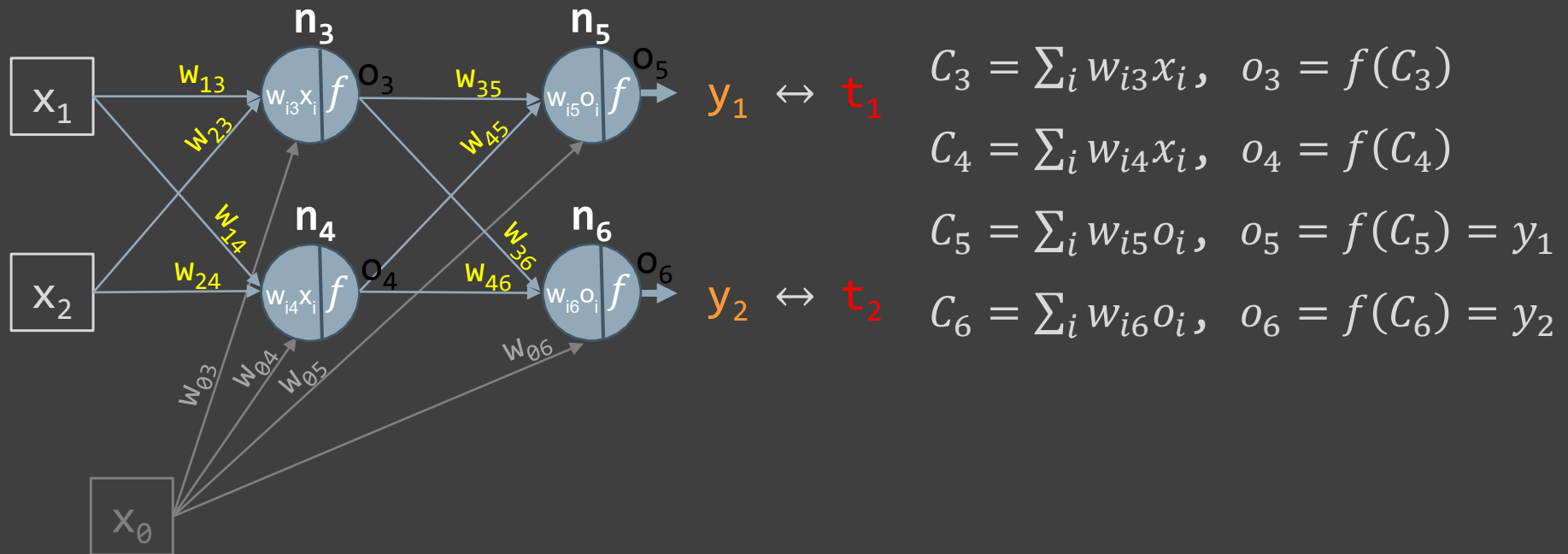
$$C_3 = \sum_i w_{i3}x_i, \quad o_3 = f(C_3)$$

$$C_4 = \sum_i w_{i4}x_i, \quad o_4 = f(C_4)$$

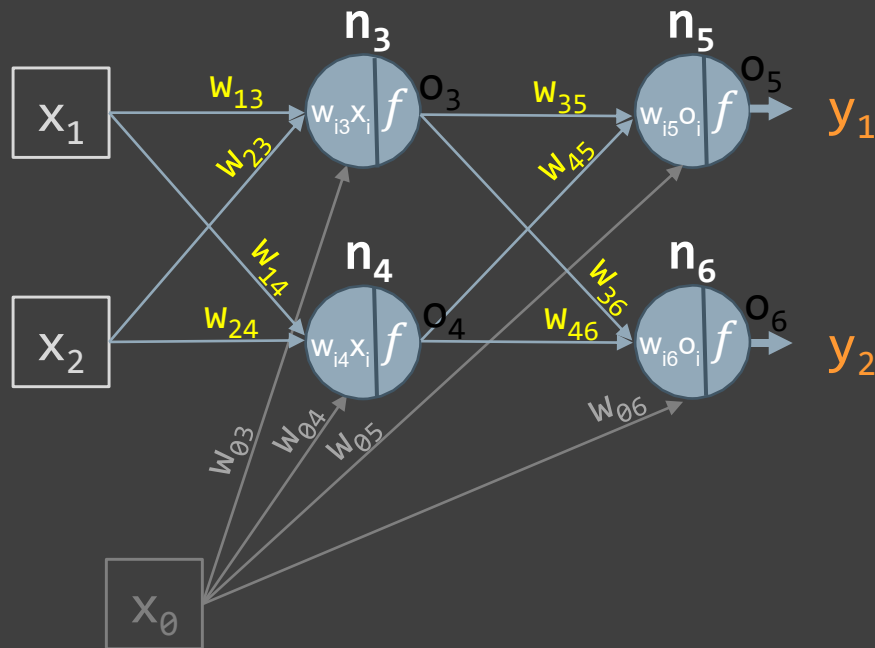
$$C_5 = \sum_i w_{i5}o_i, \quad o_5 = f(C_5) = y_1$$

$$C_6 = \sum_i w_{i6}o_i, \quad o_6 = f(C_6) = y_2$$

Μάθηση με ελαχιστοποίηση σφάλματος



Μάθηση με ελαχιστοποίηση σφάλματος



$$C_3 = \sum_i w_{i3}x_i, \quad o_3 = f(C_3)$$

$$C_4 = \sum_i w_{i4}x_i, \quad o_4 = f(C_4)$$

$$C_5 = \sum_i w_{i5}o_i, \quad o_5 = f(C_5) = y_1$$

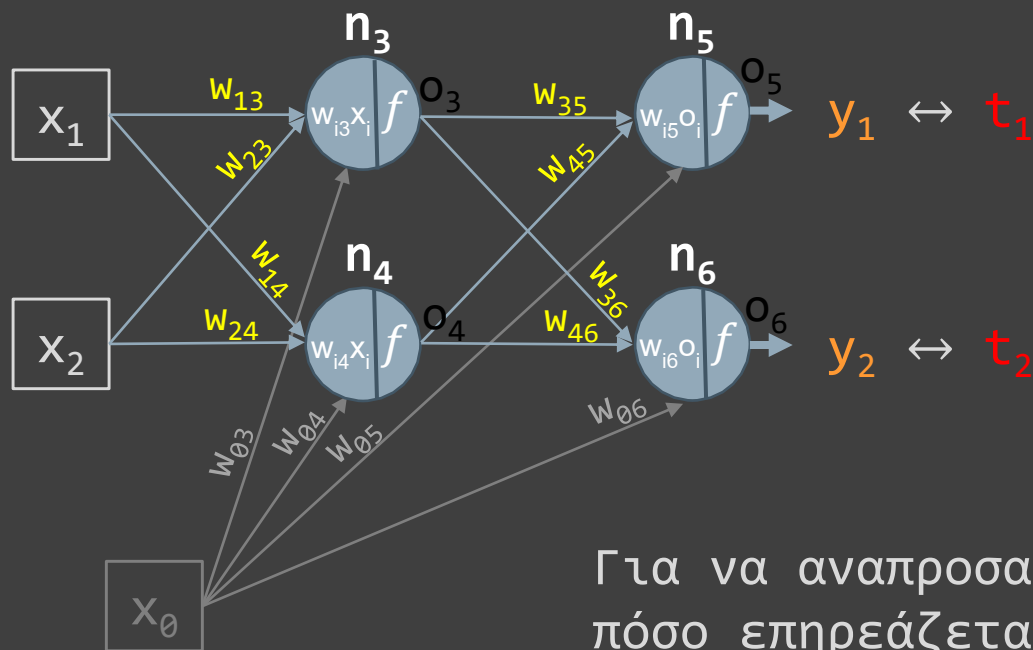
$$C_6 = \sum_i w_{i6}o_i, \quad o_6 = f(C_6) = y_2$$

$$E_{total} = \frac{1}{2} \sum_j (t_j - y_j)^2$$

t_j : έξοδος-στόχος του νευρώνα j

y_j : πραγματική έξοδος του νευρώνα j

Back-propagation of the error

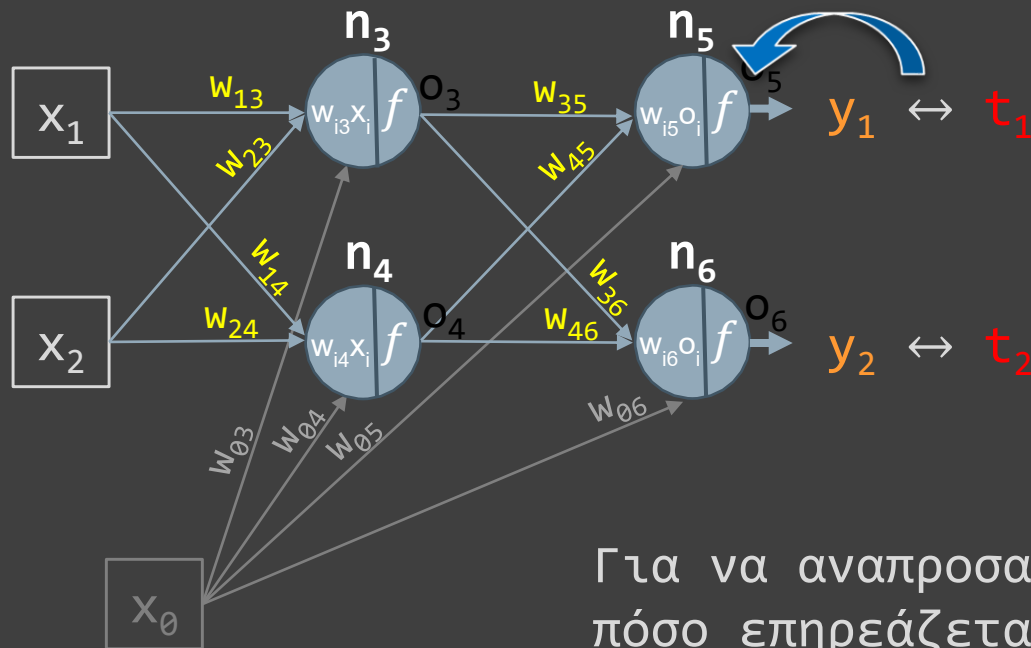


$$E_{total} = \frac{1}{2} \sum_j (t_j - y_j)^2$$

Για να αναπροσαρμόσουμε τα βάρη, βρίσκουμε πόσο επηρεάζεται το σφάλμα από κάθε βάρος:

$$\frac{\partial E_{total}}{\partial w_{ij}}$$

Back-propagation of the error



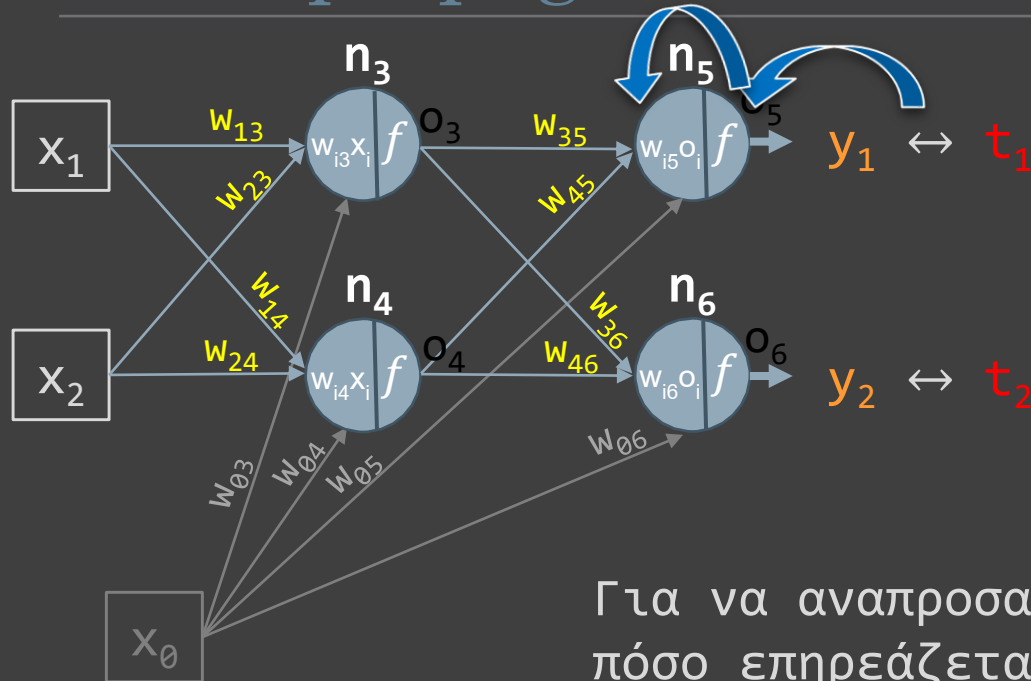
$$E_{total} = \frac{1}{2} \sum_j (t_j - y_j)^2$$

Για να αναπροσαρμόσουμε τα βάρη, βρίσκουμε πόσο επηρεάζεται το σφάλμα από κάθε βάρος:

$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)}$$

(κανόνας αλυσίδας)

Back-propagation of the error



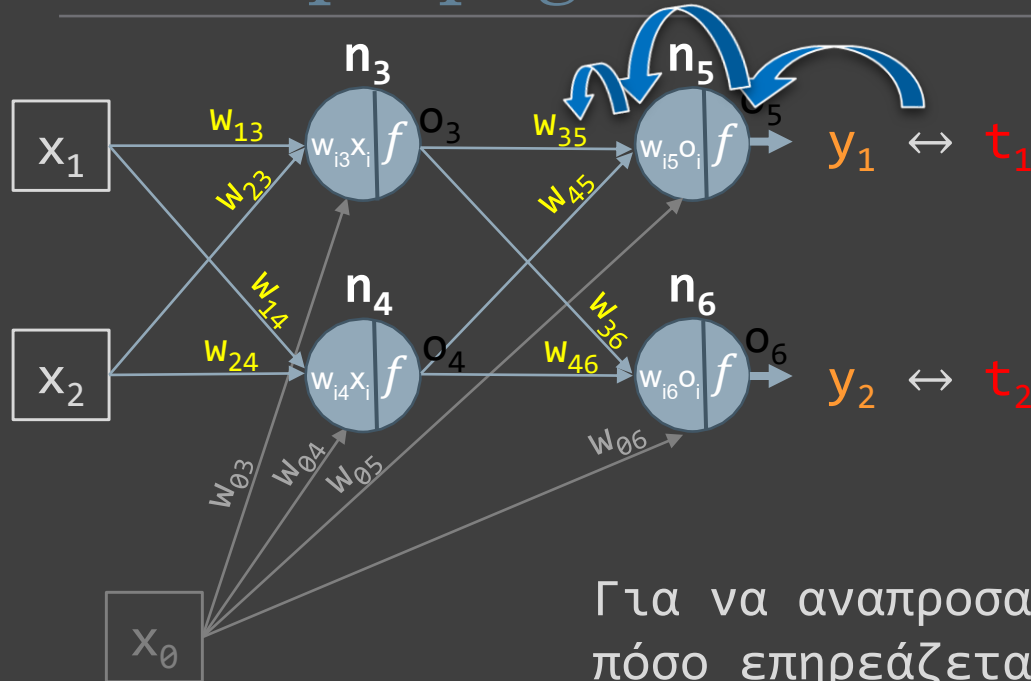
$$E_{total} = \frac{1}{2} \sum_j (t_j - y_j)^2$$

Για να αναπροσαρμόσουμε τα βάρη, βρίσκουμε πόσο επηρεάζεται το σφάλμα από κάθε βάρος:

$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i}$$

(κανόνας αλυσίδας)

Back-propagation of the error



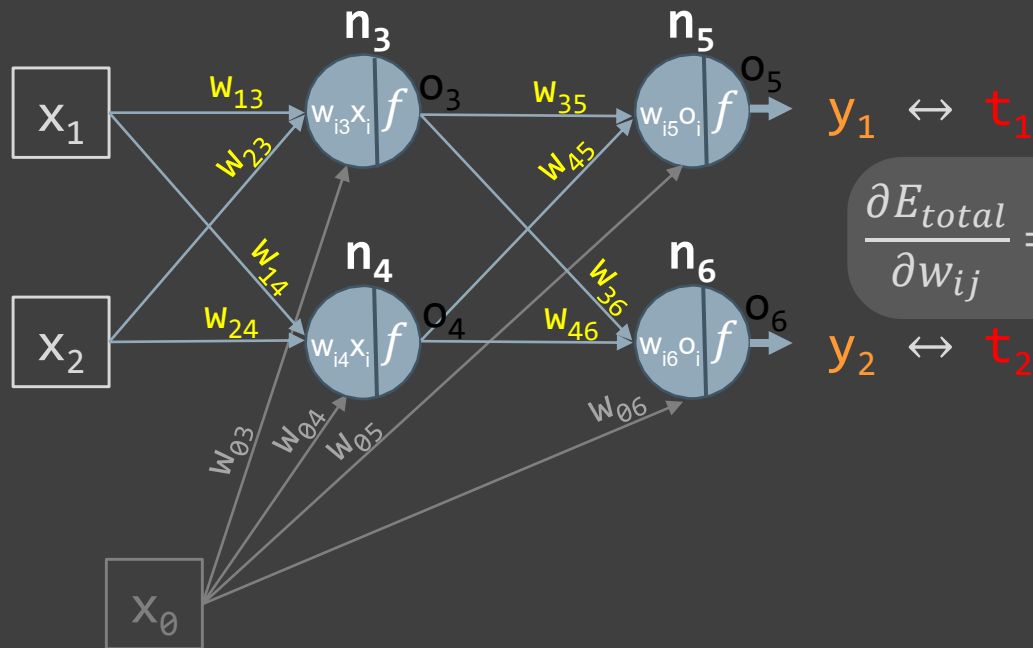
$$E_{total} = \frac{1}{2} \sum_j (t_j - y_j)^2$$

Για να αναπροσαρμόσουμε τα βάρη, βρίσκουμε πόσο επηρεάζεται το σφάλμα από κάθε βάρος:

$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} \cdot \frac{\partial \sum_i w_{ij} x_i}{\partial w_{ij}}$$

(κανόνας αλυσίδας)

Back-propagation of the error

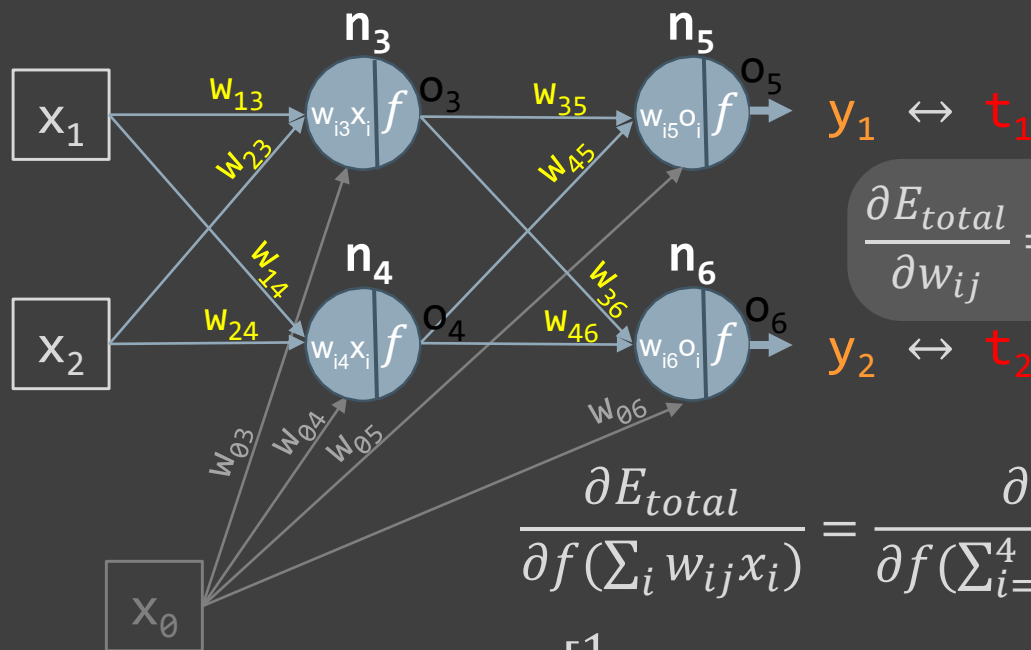


$$y_1 \leftrightarrow t_1$$

$$y_2 \leftrightarrow t_2$$

$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} \cdot \frac{\partial \sum_i w_{ij} x_i}{\partial w_{ij}}$$

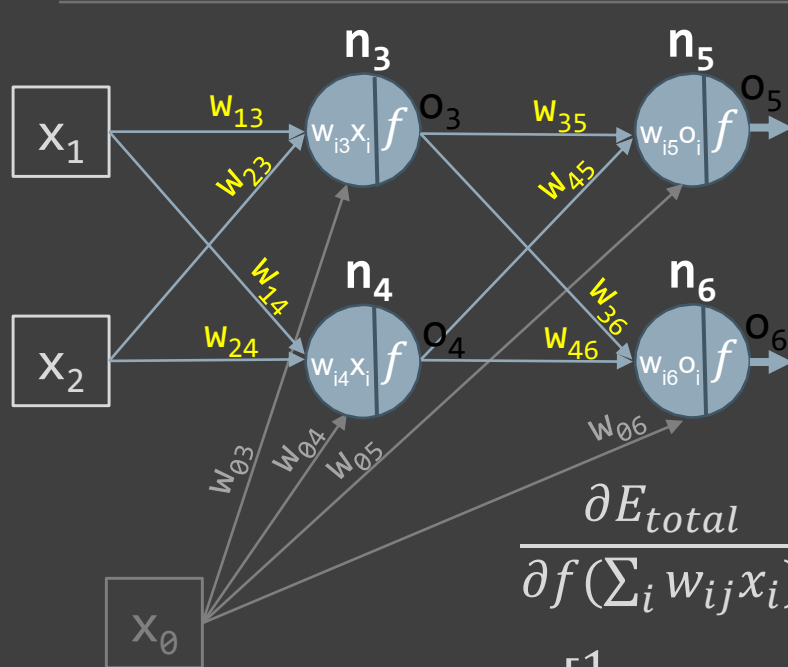
Back-propagation of the error



$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} \cdot \frac{\partial \sum_i w_{ij} x_i}{\partial w_{ij}}$$

$$\begin{aligned} \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} &= \frac{\partial E_1}{\partial f(\sum_{i=3}^4 w_{i5} o_i)} + \frac{\partial E_2}{\partial f(\sum_{i=3}^4 w_{i6} o_i)} = \\ &= \frac{\partial \left[\frac{1}{2} (t_1 - f(\sum_{i=3}^4 w_{i5} o_i))^2 \right]}{\partial f(\sum_{i=3}^4 w_{i5} o_i)} + \frac{\partial \left[\frac{1}{2} (t_2 - f(\sum_{i=3}^4 w_{i6} o_i))^2 \right]}{\partial f(\sum_{i=3}^4 w_{i6} o_i)} \\ &= \frac{1}{2} 2(t_1 - f(\sum_{i=3}^4 w_{i5} o_i))(-1) + \frac{1}{2} 2(t_2 - f(\sum_{i=3}^4 w_{i6} o_i))(-1) \\ &= - \sum_{i,j} \left[t_j - f\left(\sum_{i,j} w_{ij} o_i\right) \right] \end{aligned}$$

Back-propagation of the error



$$-\sum_{i,j} [t_j - f(\vec{w} \vec{x})]$$

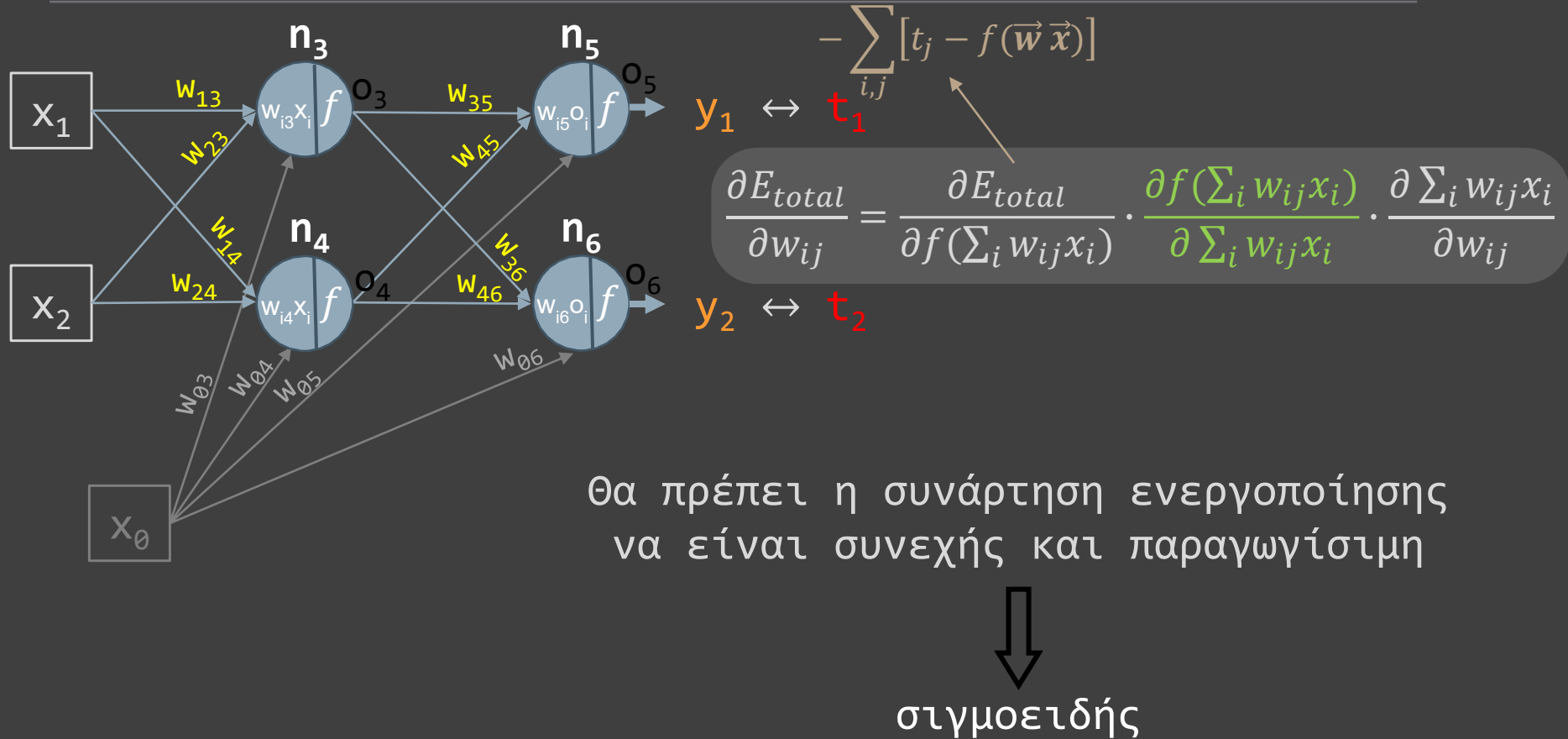
$$y_1 \leftrightarrow t_1$$

$$y_2 \leftrightarrow t_2$$

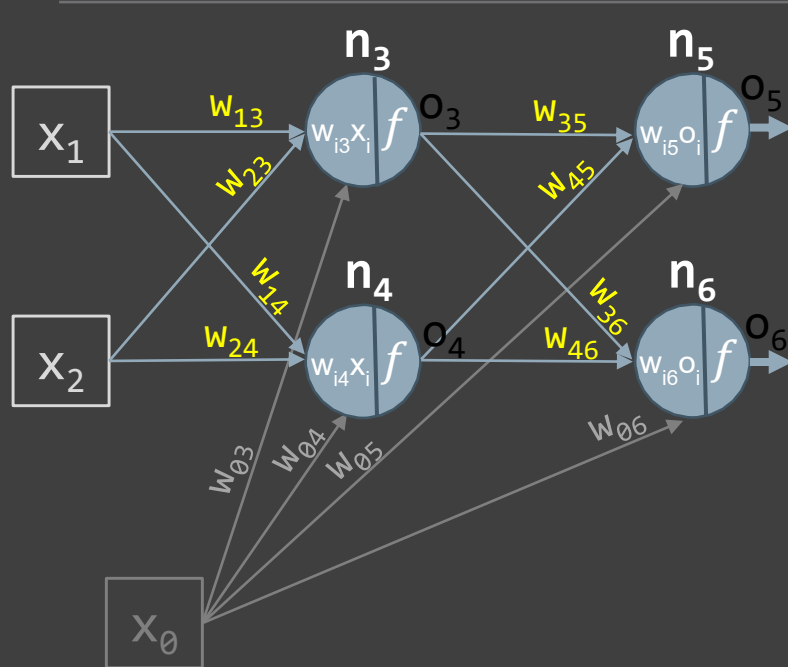
$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} \cdot \frac{\partial \sum_i w_{ij} x_i}{\partial w_{ij}}$$

$$\begin{aligned} \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} &= \frac{\partial E_1}{\partial f(\sum_{i=3}^4 w_{i5} o_i)} + \frac{\partial E_2}{\partial f(\sum_{i=3}^4 w_{i6} o_i)} = \\ &= \frac{\partial \left[\frac{1}{2} (t_1 - f(\sum_{i=3}^4 w_{i5} o_i))^2 \right]}{\partial f(\sum_{i=3}^4 w_{i5} o_i)} + \frac{\partial \left[\frac{1}{2} (t_2 - f(\sum_{i=3}^4 w_{i6} o_i))^2 \right]}{\partial f(\sum_{i=3}^4 w_{i6} o_i)} \\ &= \frac{1}{2} 2(t_1 - f(\sum_{i=3}^4 w_{i5} o_i))(-1) + \frac{1}{2} 2(t_2 - f(\sum_{i=3}^4 w_{i6} o_i))(-1) \\ &= -\sum_{i,j} \left[t_j - f\left(\sum_{i,j} w_{ij} o_i\right) \right] \end{aligned}$$

Back-propagation of the error



Back-propagation of the error



$$y_1 \leftrightarrow t_1 \quad - \sum_{i,j} [t_j - f(\vec{w} \vec{x})]$$

$$\frac{\partial E_{total}}{\partial w_{ij}} = \frac{\partial E_{total}}{\partial f(\sum_i w_{ij} x_i)} \cdot \frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} \cdot \frac{\partial \sum_i w_{ij} x_i}{\partial w_{ij}}$$

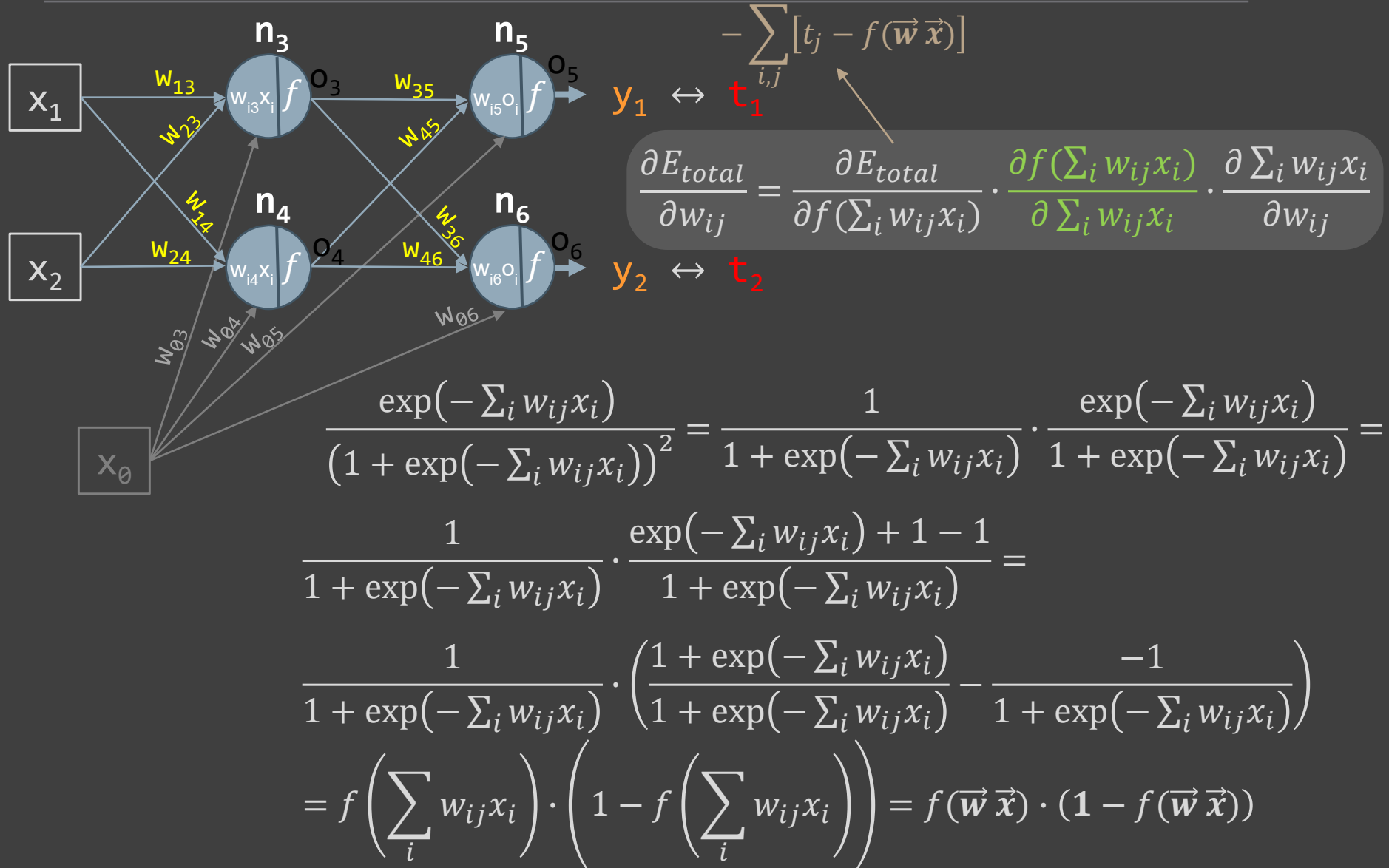
$$y_2 \leftrightarrow t_2$$

$$\text{Έστω } u = 1 + \exp(-\sum_i w_{ij} x_i)$$

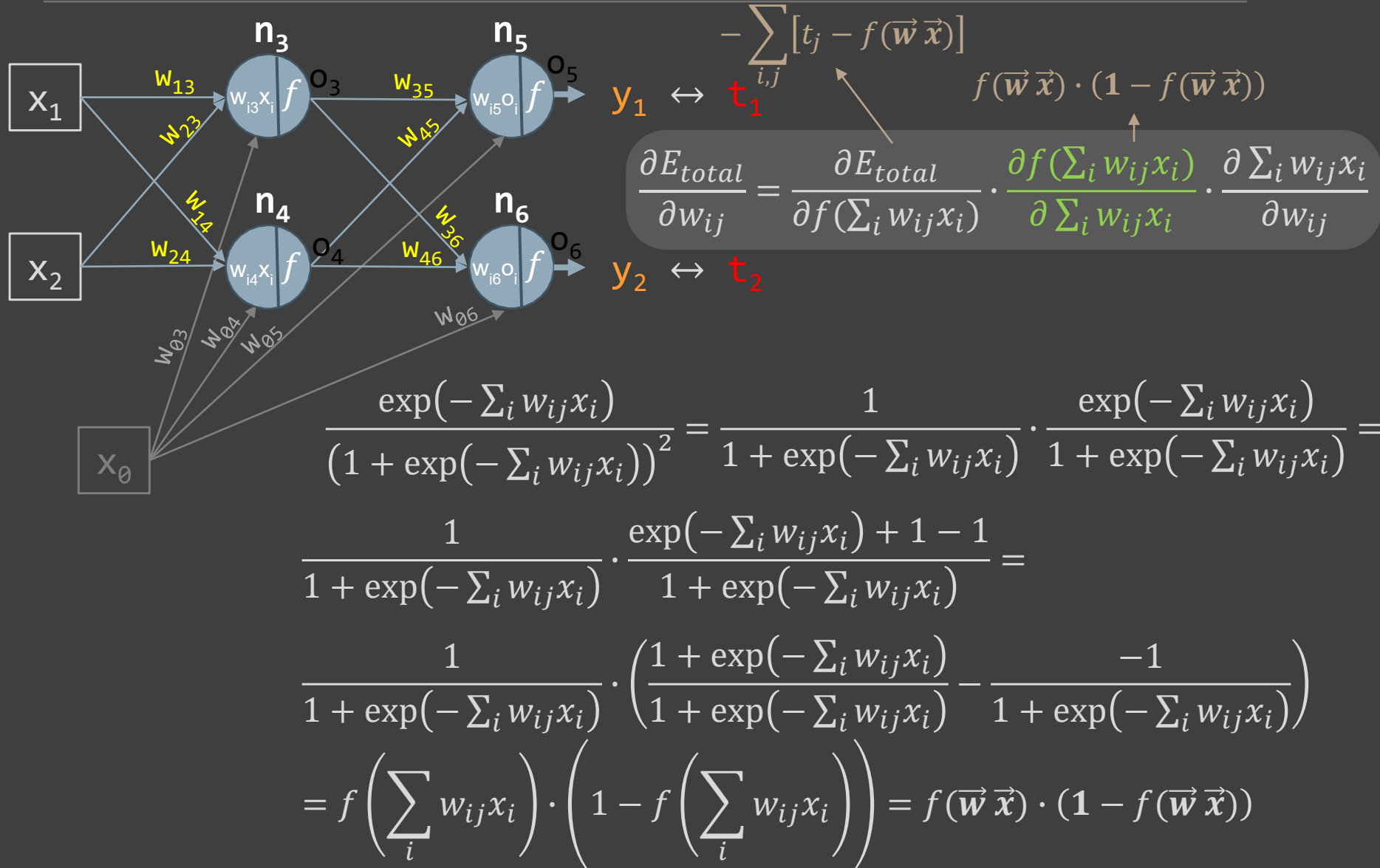
$$\frac{\partial \frac{1}{u}}{\partial u} \cdot \frac{\partial u}{\partial x} = \frac{\exp(-\sum_i w_{ij} x_i)}{(1 + \exp(-\sum_i w_{ij} x_i))^2}$$

$$\frac{\partial f(\sum_i w_{ij} x_i)}{\partial \sum_i w_{ij} x_i} = \frac{\partial}{\partial \sum_i w_{ij} x_i} \cdot \frac{1}{1 + \exp(-\sum_i w_{ij} x_i)} = \frac{\exp(-\sum_i w_{ij} x_i)}{(1 + \exp(-\sum_i w_{ij} x_i))^2}$$

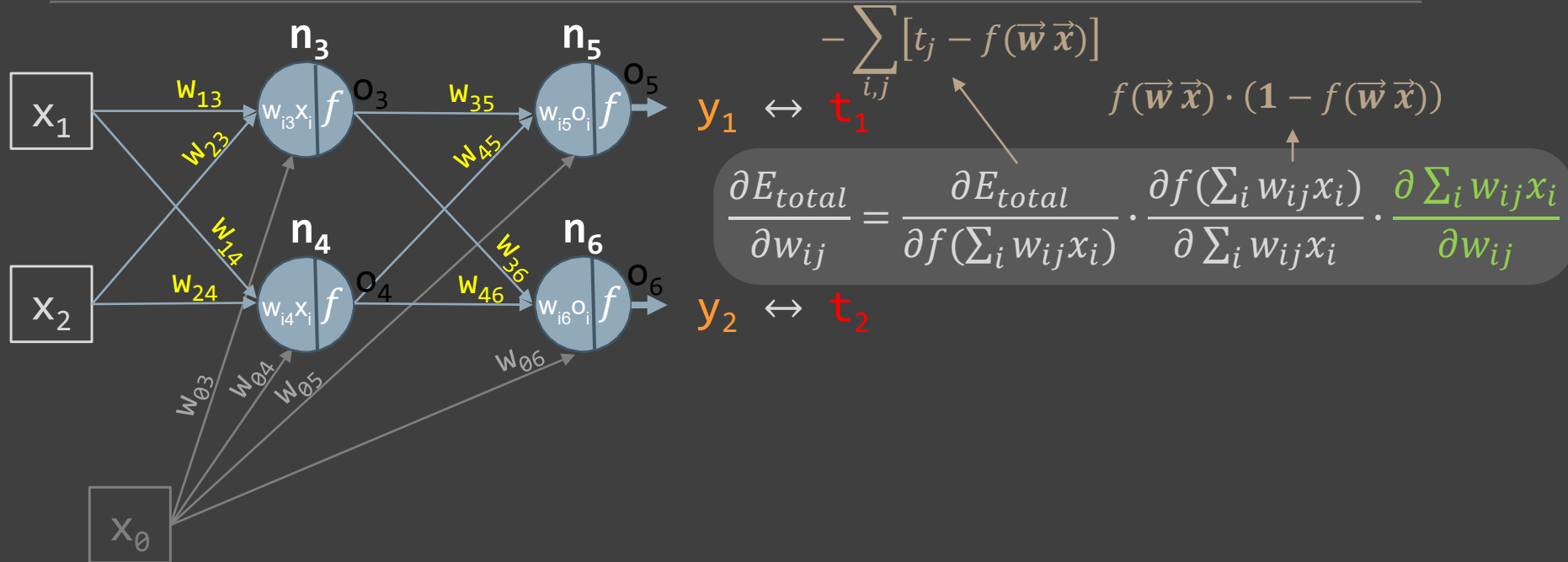
Back-propagation of the error



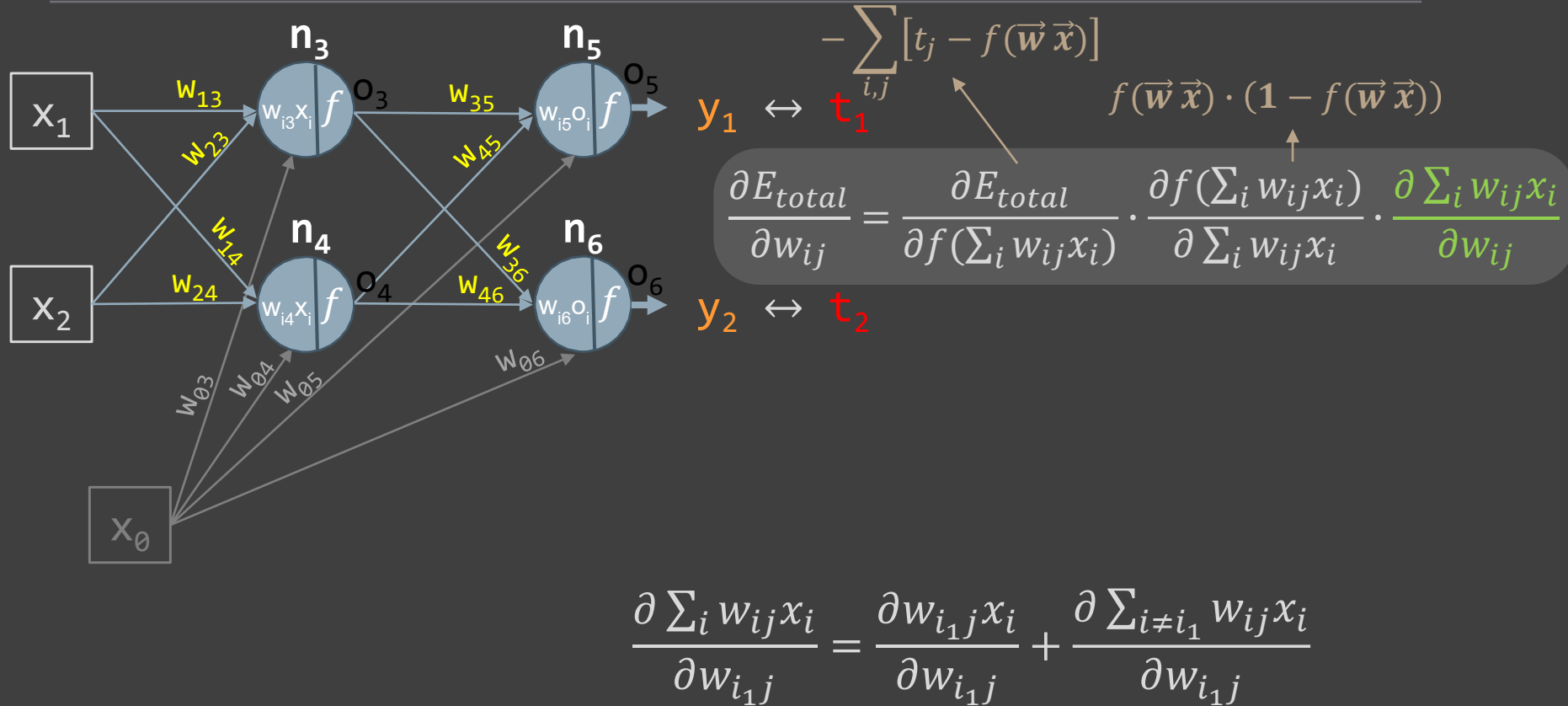
Back-propagation of the error



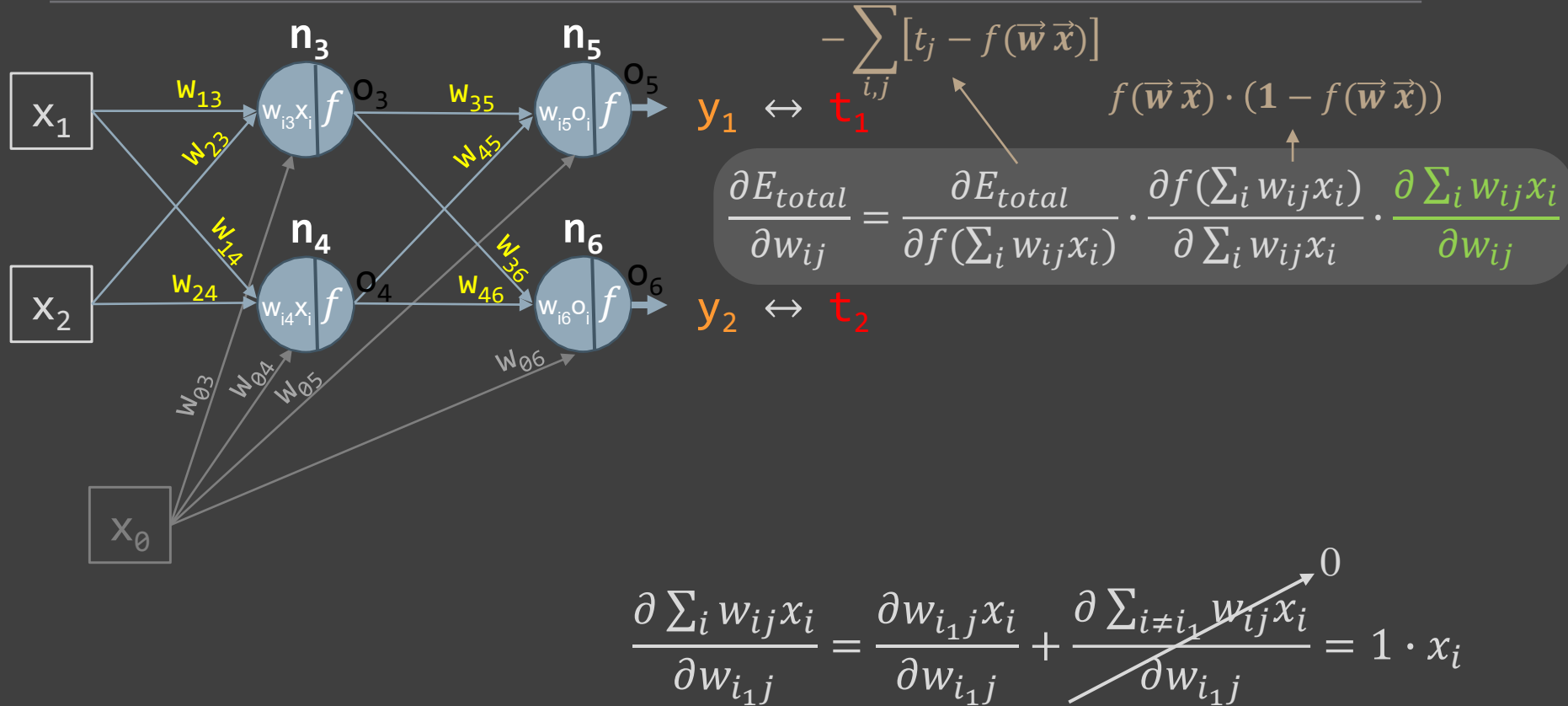
Back-propagation of the error



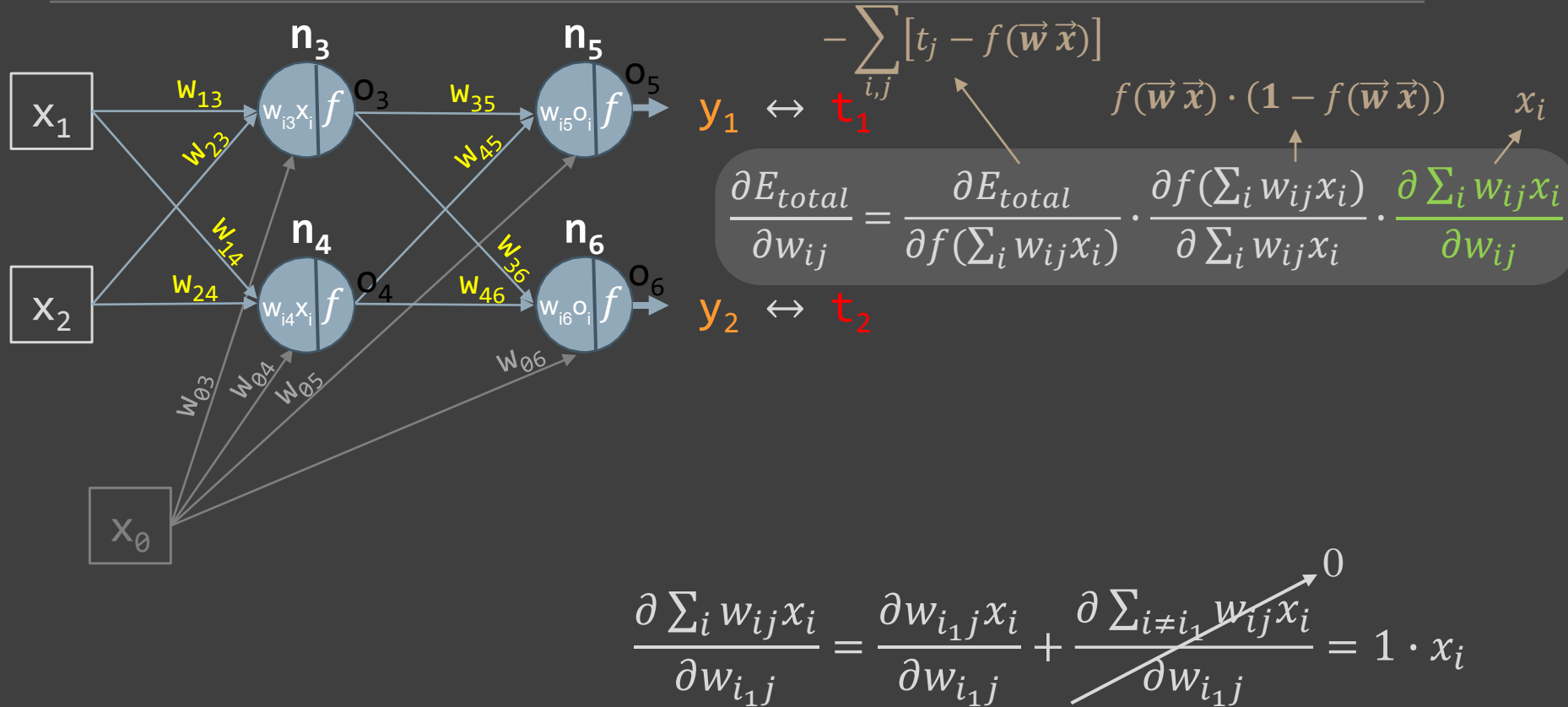
Back-propagation of the error



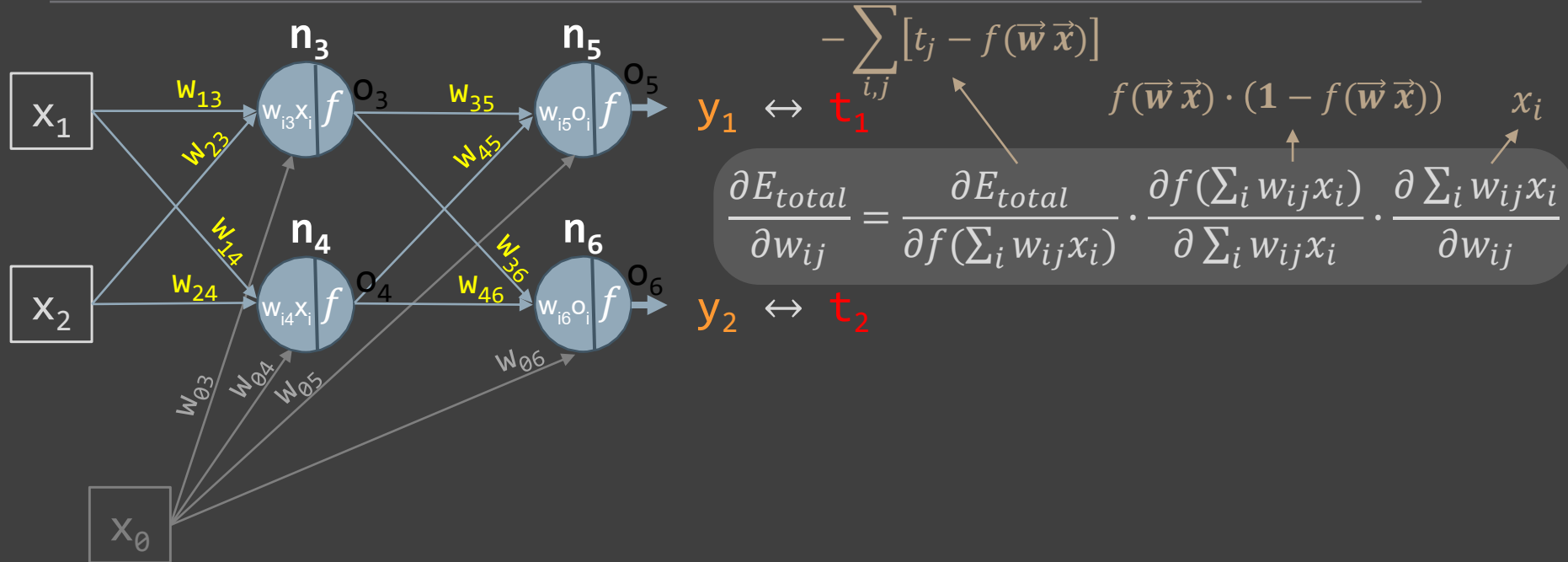
Back-propagation of the error



Back-propagation of the error

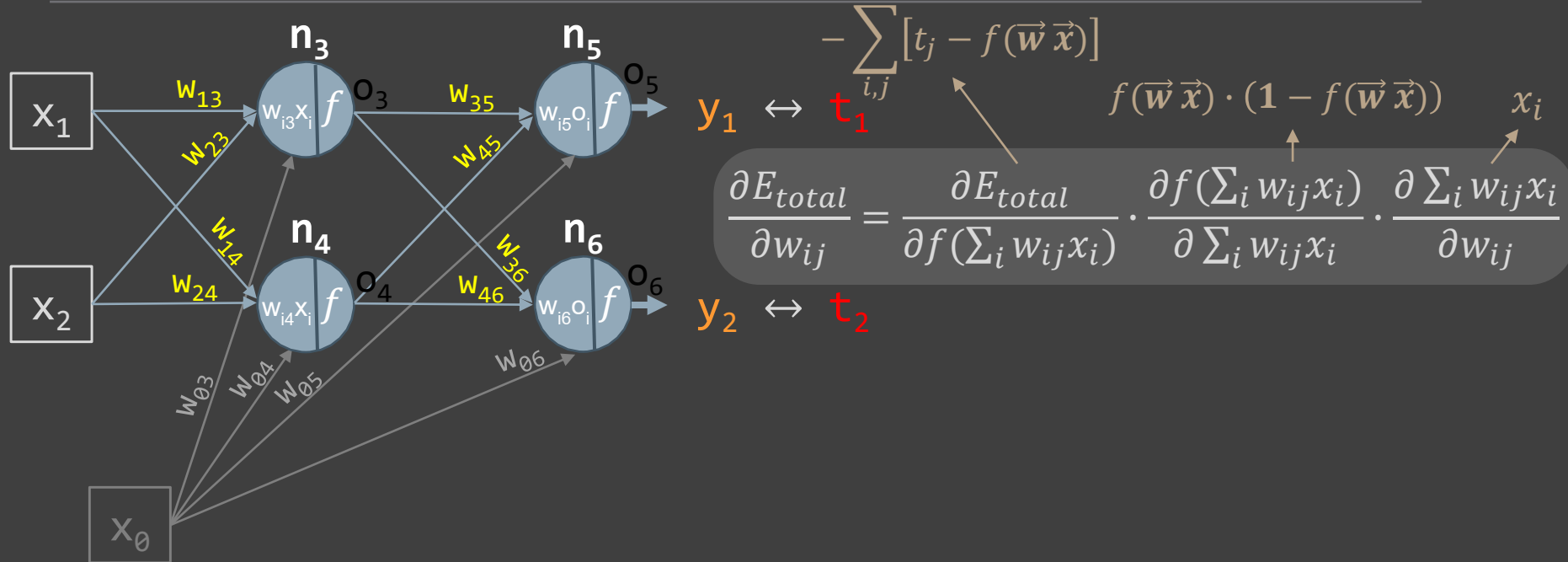


Back-propagation of the error



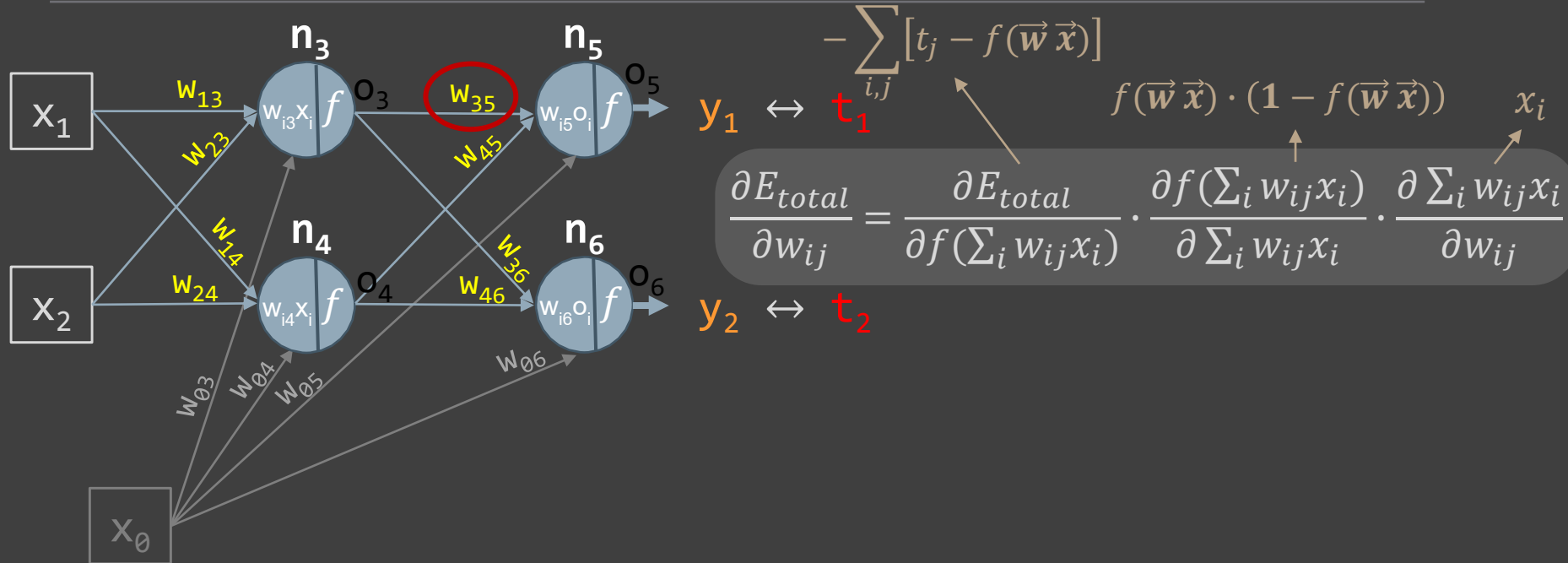
$$\frac{\partial E_{total}}{\partial w_{ij}} = - \sum_{i,j} \left[t_j - f \left(\sum_{i,j} w_{ij} o_i \right) \right] \cdot f \left(\sum_{i,j} w_{ij} o_i \right) \cdot \left(1 - f \left(\sum_{i,j} w_{ij} o_i \right) \right) \cdot x_i$$

Back-propagation of the error



$$\begin{aligned} \frac{\partial E_{total}}{\partial w_{ij}} &= - \sum_{i,j} \left[t_j - f \left(\sum_{i,j} w_{ij} o_i \right) \right] \cdot f \left(\sum_{i,j} w_{ij} o_i \right) \cdot \left(1 - f \left(\sum_{i,j} w_{ij} o_i \right) \right) \cdot x_i \\ &= -\delta_j \cdot x_i \end{aligned}$$

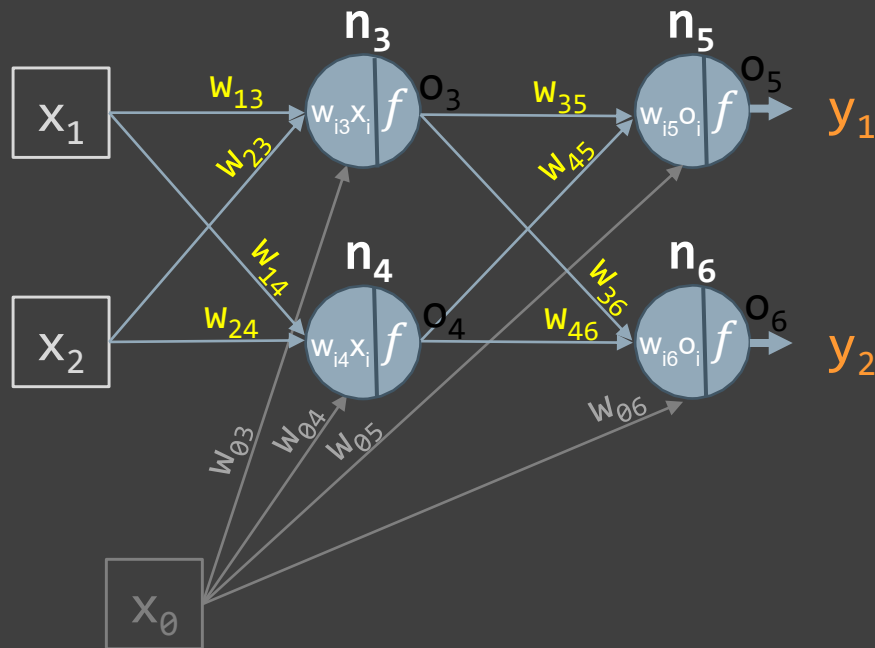
Back-propagation of the error



Έστω $w_{ij} = w_{35}$:

$$\frac{\partial E_{total}}{\partial w_{35}} = -(t_5 - o_5) \cdot o_5 \cdot (1 - o_5) \cdot o_3 = -\delta_5 \cdot o_3$$

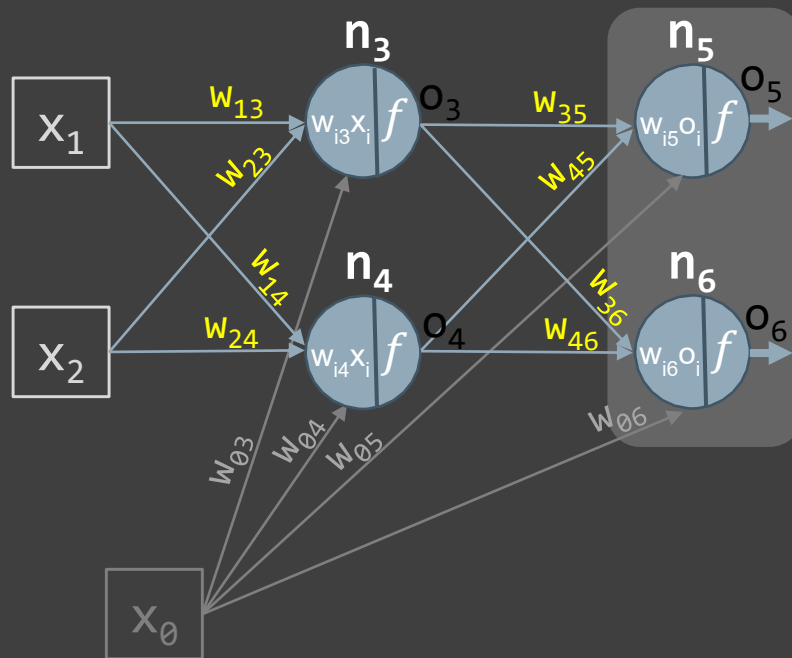
Back-propagation of the error



Έχοντας βρει πόσο συνεισφέρει κάθε βάρος στο τελικό σφάλμα, αναπροσαρμόζουμε τα βάρη, αφαιρώντας αυτή τη “συνεισφορά”

(πολλαπλασιασμένη με τον ρυθμό μάθησης)

Back-propagation of the error



ΓΕΝΙΚΑ

για ΤΝΔ με **ΣΙΓΜΟΕΙΔΗ**
συνάρτηση ενεργοποίησης

Για έναν **νευρώνα εξόδου** k :

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k)$$

$$\Delta w_{jk} = w_{jk}^{(new)} - w_{jk}^{(old)} = a \cdot \delta_k \cdot o_j \Rightarrow w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

t_k : έξοδος-στόχος του νευρώνα k (επιπέδου εξόδου)

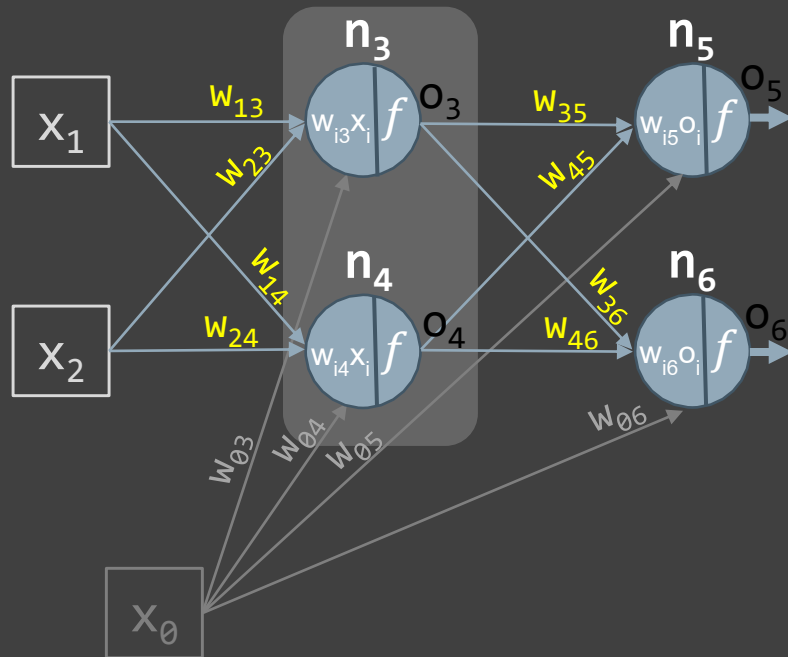
o_k : πραγματική έξοδος του νευρώνα k (επιπέδου εξόδου)

w_{jk} : βάρος σύνδεσης νευρώνα j (προηγούμενου επιπέδου)
με νευρώνα k (επιπέδου εξόδου)

o_j : πραγματική έξοδος του νευρώνα j (προηγούμενου επιπέδου)

a : ρυθμός μάθησης

Back-propagation of the error



ΓΕΝΙΚΑ

για ΤΝΔ με **ΣΙΓΜΟΕΙΔΗ**
συνάρτηση ενεργοποίησης

Για έναν κρυφό νευρώνα j :

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k$$

$$\Delta w_{ij} = w_{ij}^{(new)} - w_{ij}^{(old)} = a \cdot \delta_j \cdot o_i \Rightarrow w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

o_j : πραγματική έξοδος του νευρώνα j

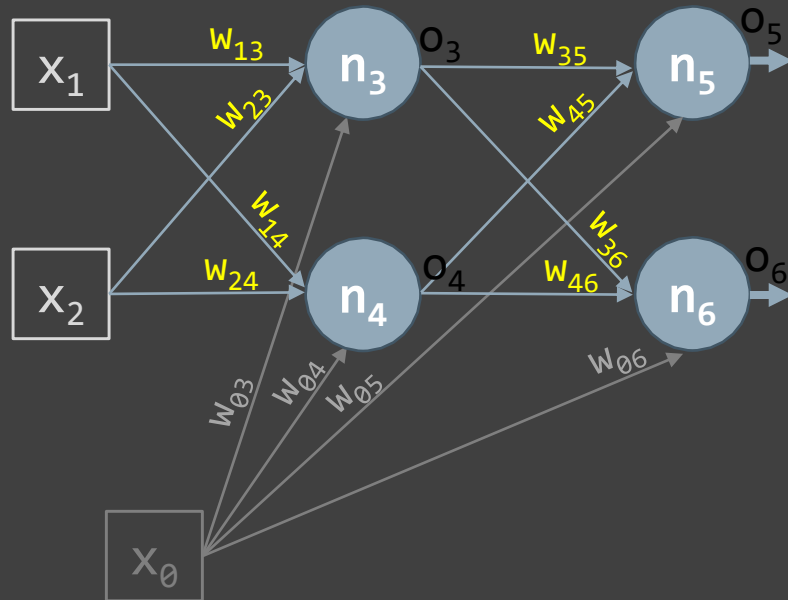
w_{jk} : βάρος σύνδεσης νευρώνα j με νευρώνα k (επόμενου επιπέδου)

δ_k : δ νευρώνων επόμενου επιπέδου από αυτό που βρίσκεται ο νευρώνας j

w_{ij} : βάρος σύνδεσης νευρώνα i (προηγούμενου επιπέδου) με νευρώνα j

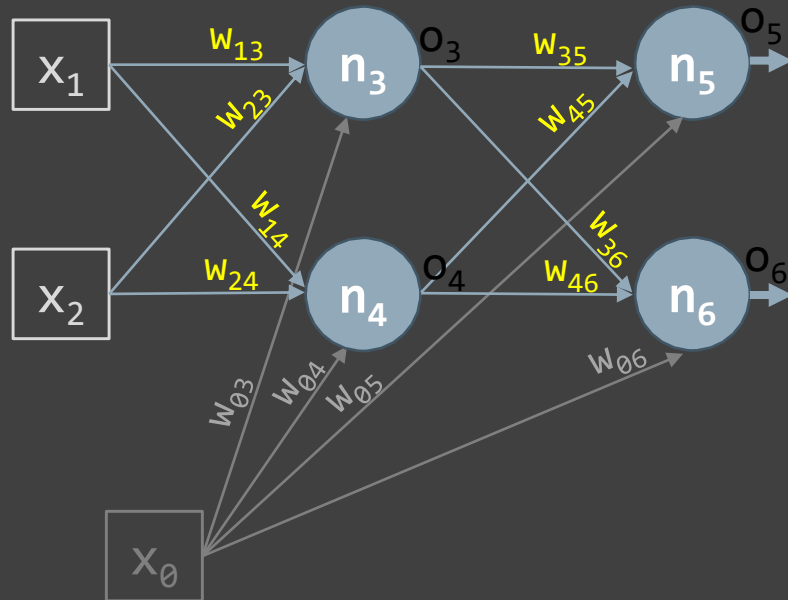
a : ρυθμός μάθησης

Παράδειγμα – Forward pass



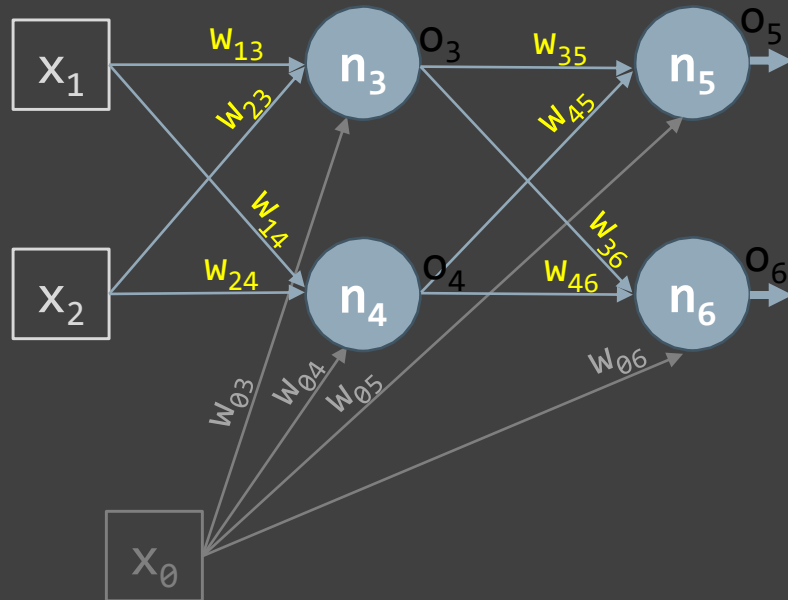
x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5				
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5				
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



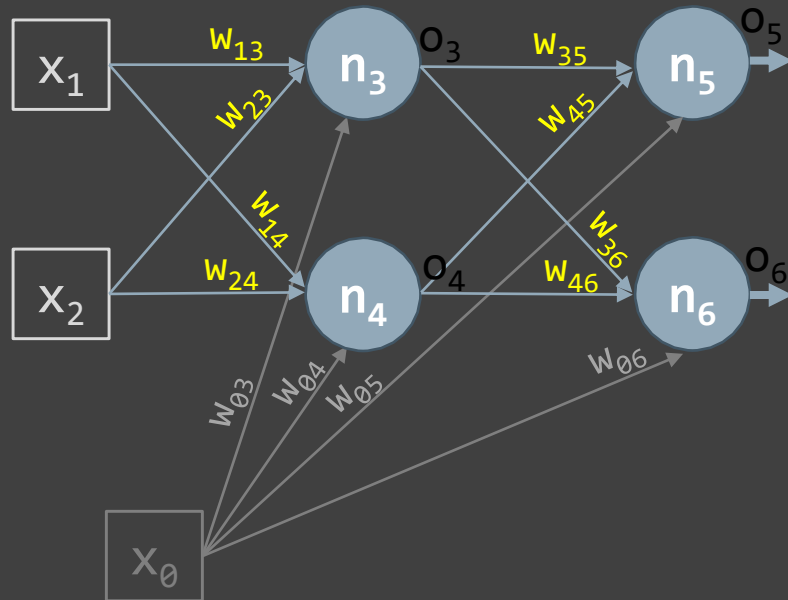
$$C_3 = \sum_i w_{i3} x_i = w_{03} x_0 + w_{13} x_1 + w_{23} x_2$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.05 + 0.5 \cdot 0.10 = 0.575$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.575}} = 0.64$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5				
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



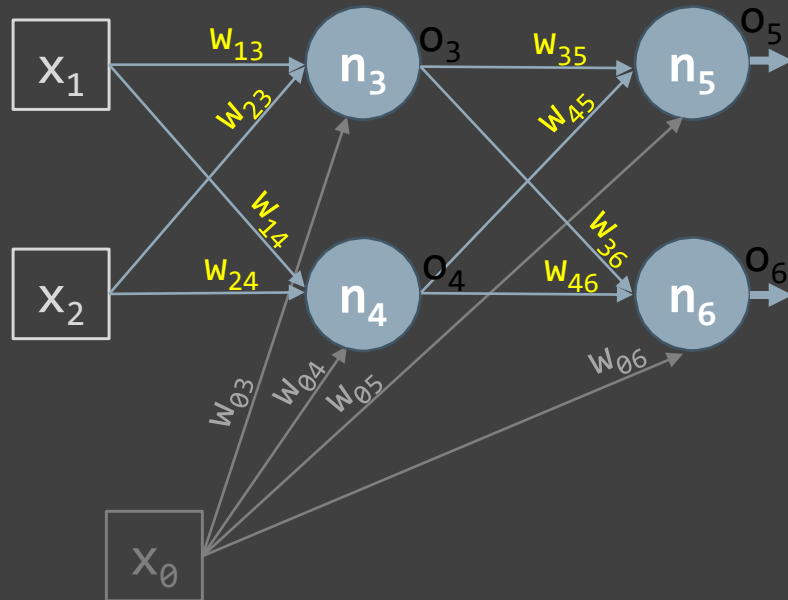
$$C_3 = \sum_i w_{i3} x_i = w_{03} x_0 + w_{13} x_1 + w_{23} x_2$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.05 + 0.5 \cdot 0.10 = 0.575$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.575}} = 0.64$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64			
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



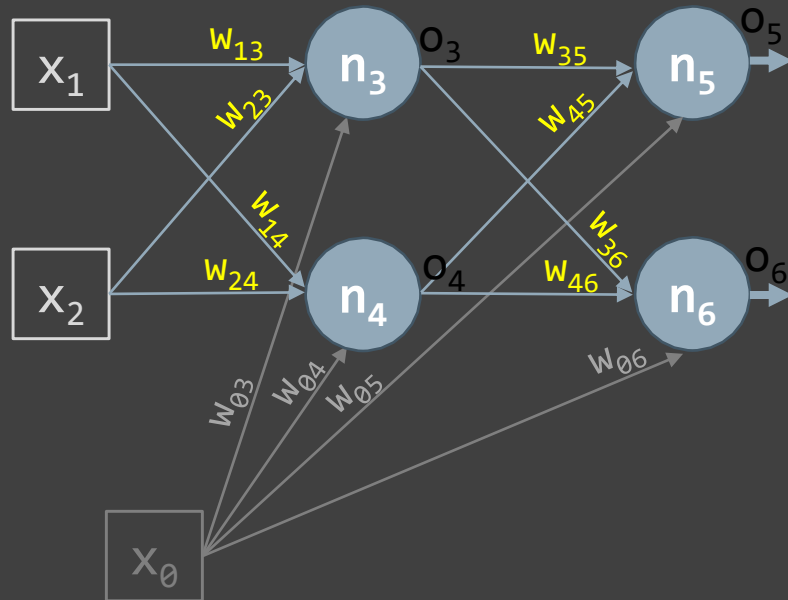
$$C_4 = \sum_i w_{i4} x_i = w_{04} x_0 + w_{14} x_1 + w_{24} x_2$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.05 + 0.5 \cdot 0.10 = 0.575$$

$$o_4 = f(C_4) = \frac{1}{1 + e^{-C_4}} = \frac{1}{1 + e^{-0.575}} = 0.64$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64			
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



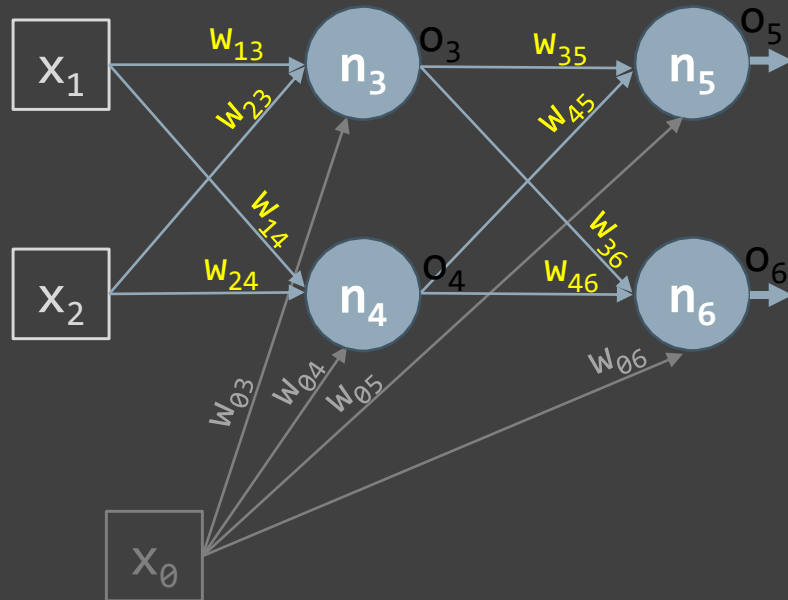
$$C_4 = \sum_i w_{i4} x_i = w_{04} x_0 + w_{14} x_1 + w_{24} x_2$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.05 + 0.5 \cdot 0.10 = 0.575$$

$$o_4 = f(C_4) = \frac{1}{1 + e^{-C_4}} = \frac{1}{1 + e^{-0.575}} = 0.64$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64		
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



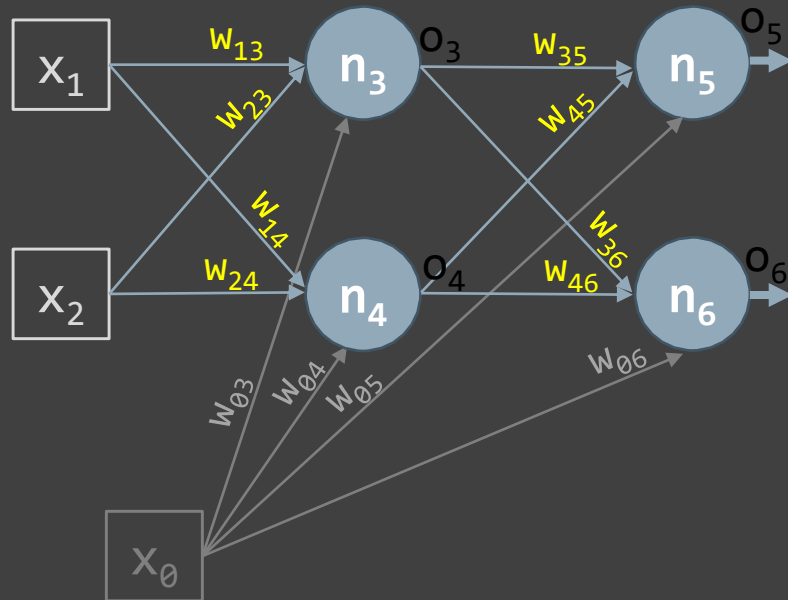
$$C_5 = \sum_j w_{j5} o_j = w_{05} x_0 + w_{35} o_3 + w_{45} o_4$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.64 + 0.5 \cdot 0.64 = 1.14$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.14}} = 0.76$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64		
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



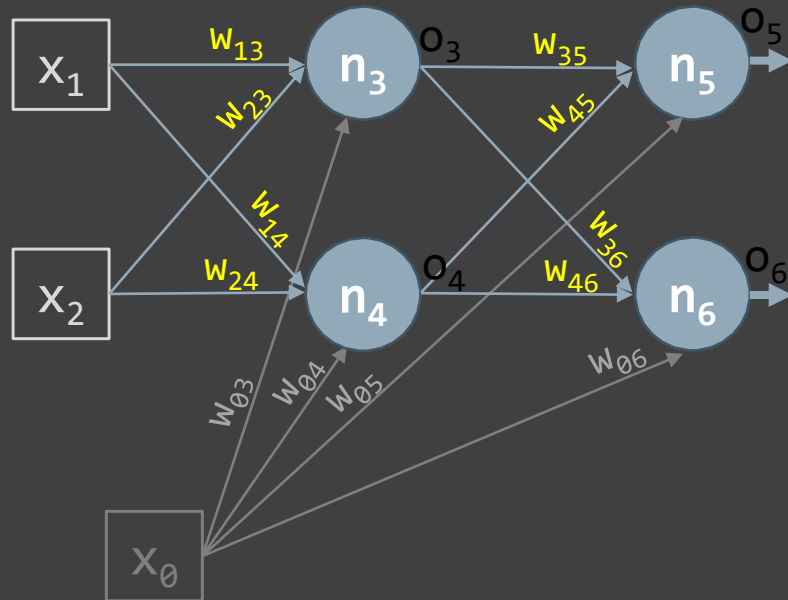
$$C_5 = \sum_j w_{j5} o_j = w_{05} x_0 + w_{35} o_3 + w_{45} o_4$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.64 + 0.5 \cdot 0.64 = 1.14$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.14}} = 0.76$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



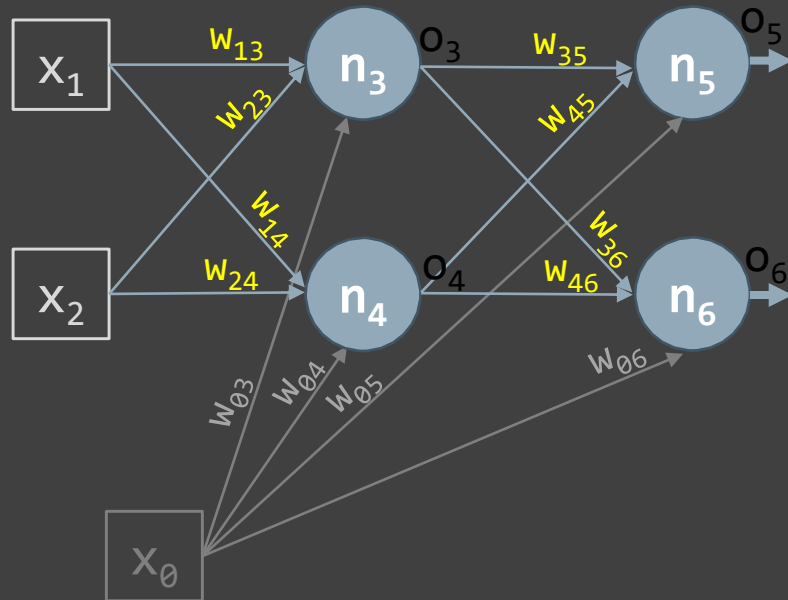
$$C_6 = \sum_j w_{j6} o_j = w_{06} x_0 + w_{36} o_3 + w_{46} o_4$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.64 + 0.5 \cdot 0.64 = 1.14$$

$$o_6 = f(C_6) = \frac{1}{1 + e^{-C_6}} = \frac{1}{1 + e^{-1.14}} = 0.76$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



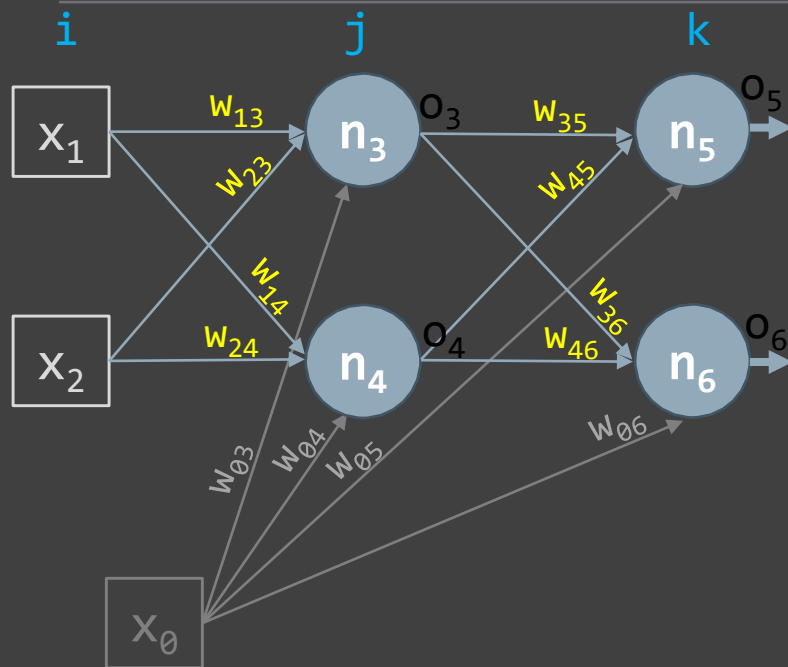
$$C_6 = \sum_j w_{j6} o_j = w_{06} x_0 + w_{36} o_3 + w_{46} o_4$$

$$= 0.5 \cdot 1 + 0.5 \cdot 0.64 + 0.5 \cdot 0.64 = 1.14$$

$$o_6 = f(C_6) = \frac{1}{1 + e^{-C_6}} = \frac{1}{1 + e^{-1.14}} = 0.76$$

x_0	x_1	x_2	t_1	t_2	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

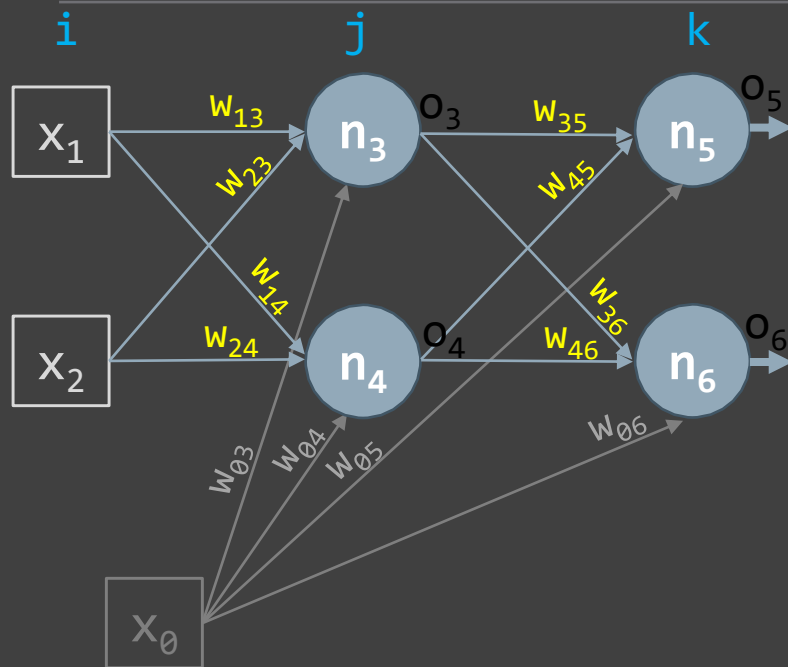
$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

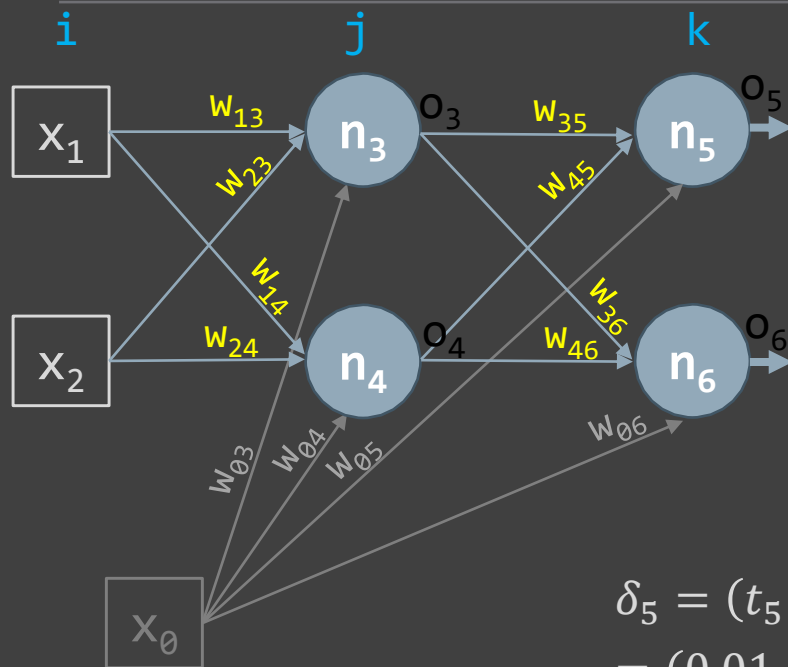
Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

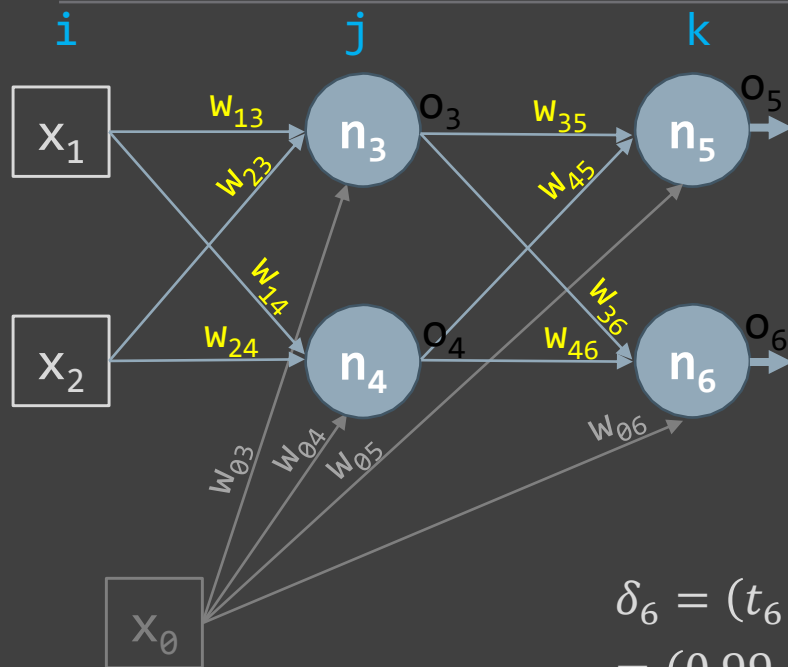
$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

$$\begin{aligned} \delta_5 &= (t_5 - o_5) \cdot o_5 \cdot (1 - o_5) = \\ &= (0.01 - 0.76) \cdot 0.76 \cdot (1 - 0.76) = -0.1368 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

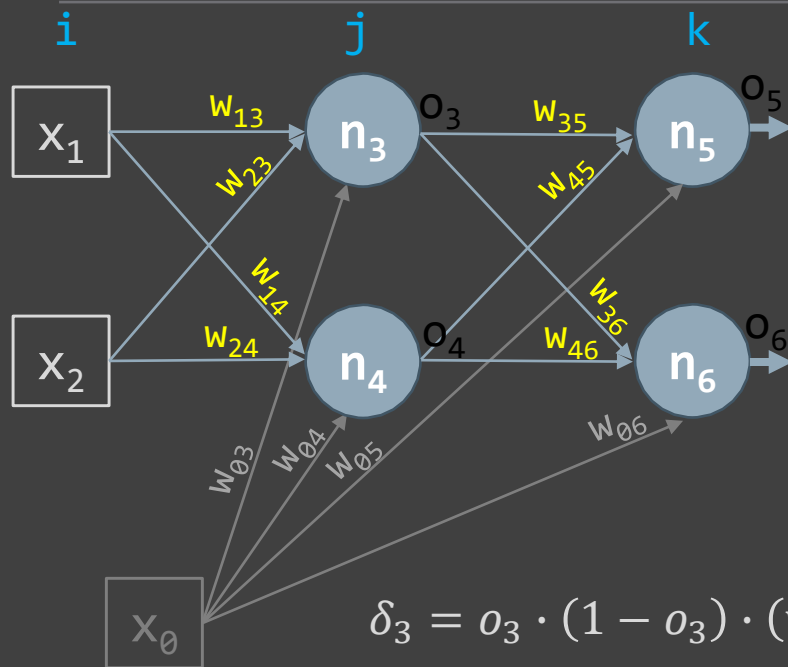
Υπολογίζουμε πρώτα όλα τα δ

$$\delta_5 = -0.1368$$

$$\begin{aligned} \delta_6 &= (t_6 - o_6) \cdot o_6 \cdot (1 - o_6) = \\ &= (0.99 - 0.76) \cdot 0.76 \cdot (1 - 0.76) = -0.042 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

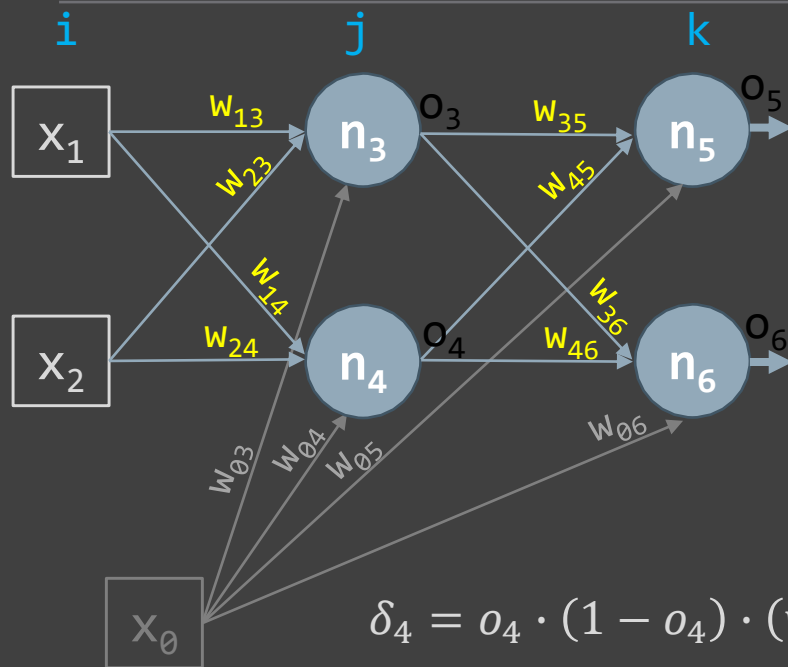
$$\delta_5 = -0.1368$$

$$\delta_6 = -0.0420$$

$$\begin{aligned} \delta_3 &= o_3 \cdot (1 - o_3) \cdot (w_{35} \cdot \delta_5 + w_{36} \cdot \delta_6) = \\ &= 0.64 \cdot (1 - 0.64) \cdot (0.5 \cdot (-0.1368) + 0.5 \cdot (-0.0420)) = -0.0206 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

$$\delta_5 = -0.1368$$

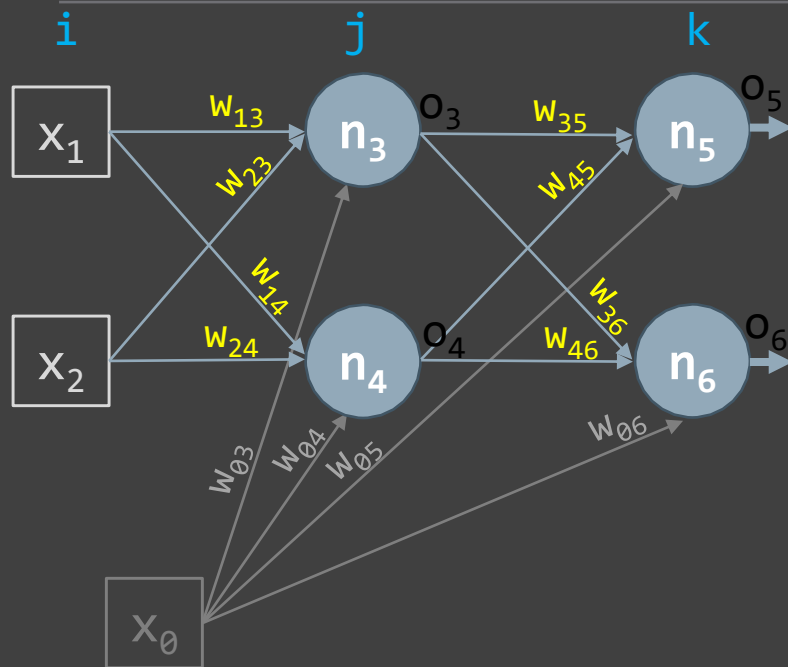
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\begin{aligned} \delta_4 &= o_4 \cdot (1 - o_4) \cdot (w_{45} \cdot \delta_5 + w_{46} \cdot \delta_6) = \\ &= 0.64 \cdot (1 - 0.64) \cdot (0.5 \cdot (-0.1368) + 0.5 \cdot (-0.0420)) = -0.0206 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\delta_5 = -0.1368$$

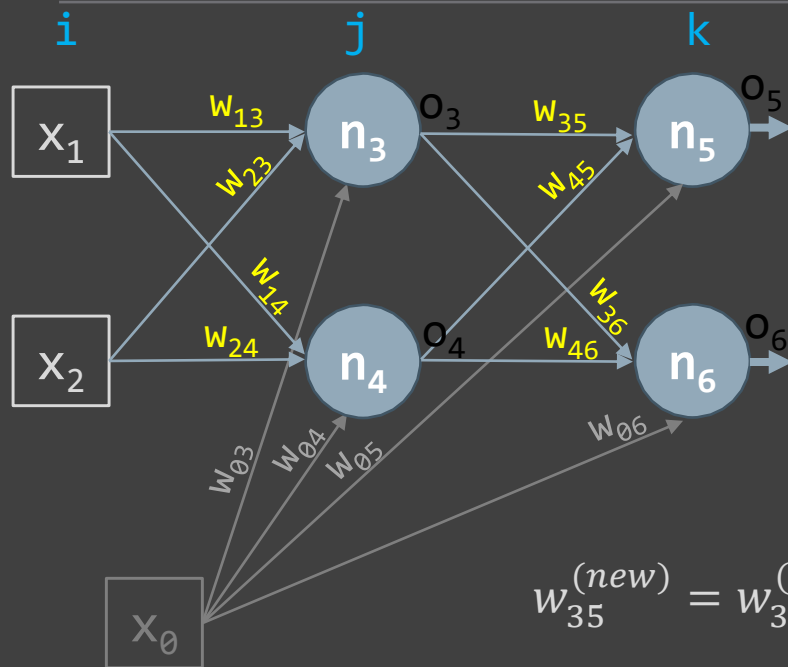
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned}
 w_{35}^{(new)} &= w_{35}^{(old)} + 0.4 \cdot \delta_5 \cdot o_3 = \\
 &= 0.5 + 0.4 \cdot (-0.1368) \cdot 0.64 = 0.465
 \end{aligned}$$

$$\delta_5 = -0.1368$$

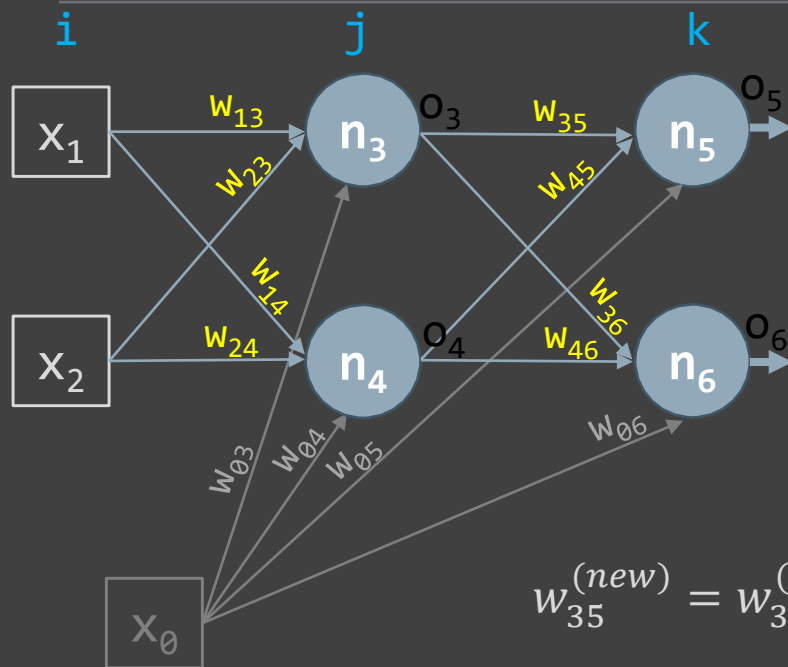
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78																
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{35}^{(new)} &= w_{35}^{(old)} + 0.4 \cdot \delta_5 \cdot o_3 = \\ &= 0.5 + 0.4 \cdot (-0.1368) \cdot 0.64 = 0.465 \end{aligned}$$

$$\delta_5 = -0.1368$$

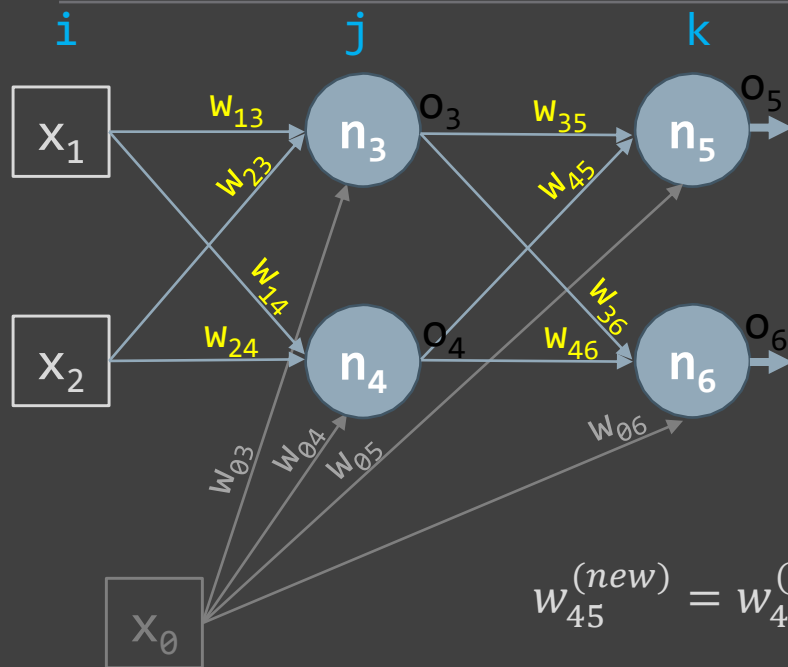
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78									0.465							
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{45}^{(new)} &= w_{45}^{(old)} + 0.4 \cdot \delta_5 \cdot o_4 = \\ &= 0.5 + 0.4 \cdot (-0.1368) \cdot 0.64 = 0.465 \end{aligned}$$

$$\delta_5 = -0.1368$$

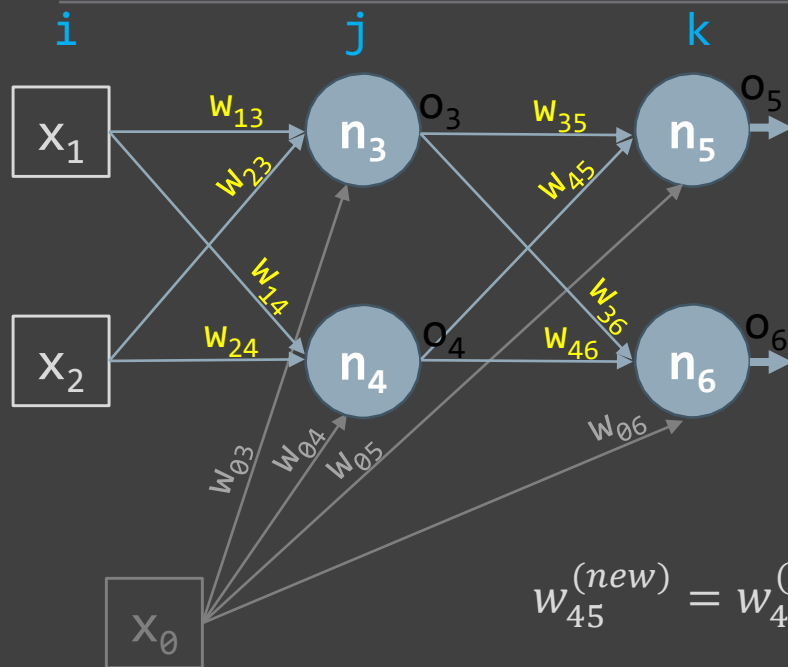
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78									0.465							
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{45}^{(new)} &= w_{45}^{(old)} + 0.4 \cdot \delta_5 \cdot o_4 = \\ &= 0.5 + 0.4 \cdot (-0.1368) \cdot 0.64 = 0.465 \end{aligned}$$

$$\delta_5 = -0.1368$$

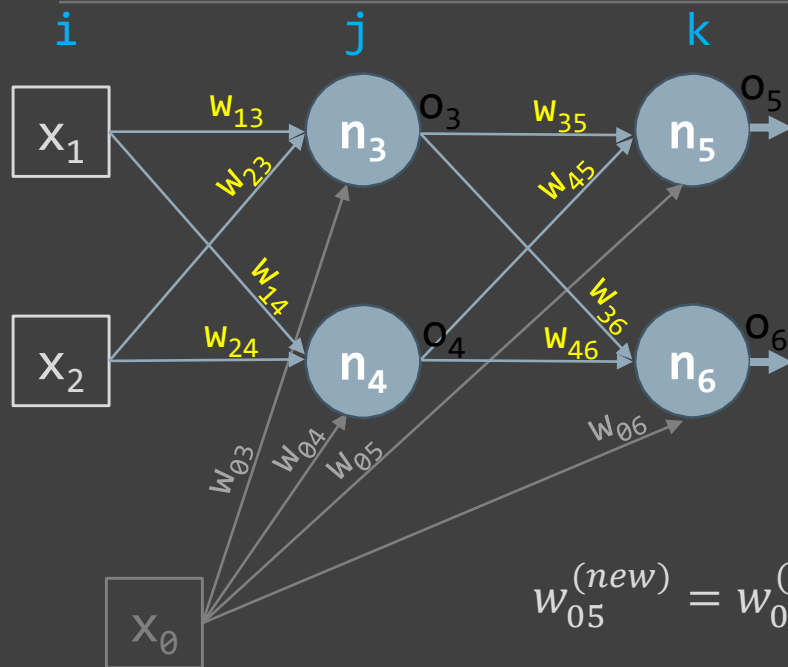
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78									0.465	0.465						
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{05}^{(new)} &= w_{05}^{(old)} + 0.4 \cdot \delta_5 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.1368) \cdot 1 = 0.4453 \end{aligned}$$

$$\delta_5 = -0.1368$$

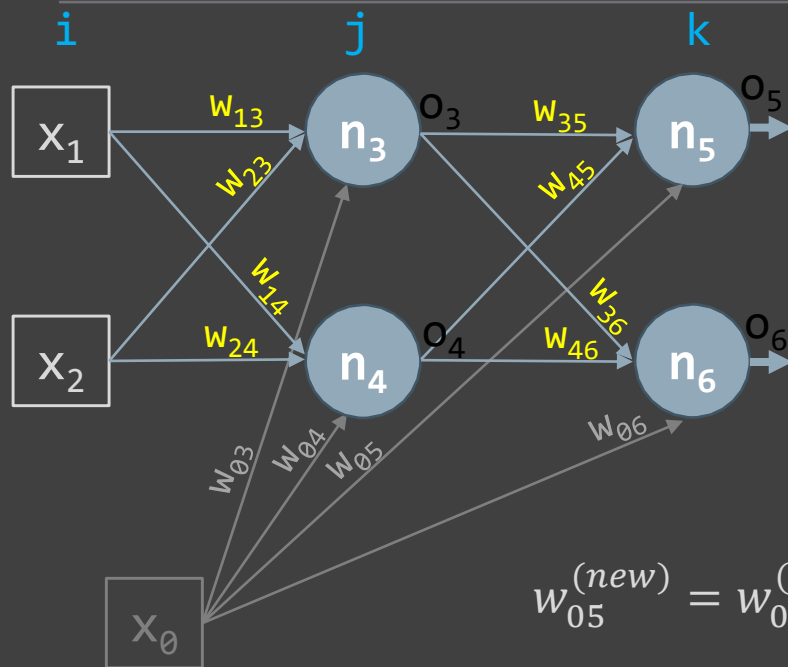
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78									0.465	0.465						
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{05}^{(new)} &= w_{05}^{(old)} + 0.4 \cdot \delta_5 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.1368) \cdot 1 = 0.4453 \end{aligned}$$

$$\delta_5 = -0.1368$$

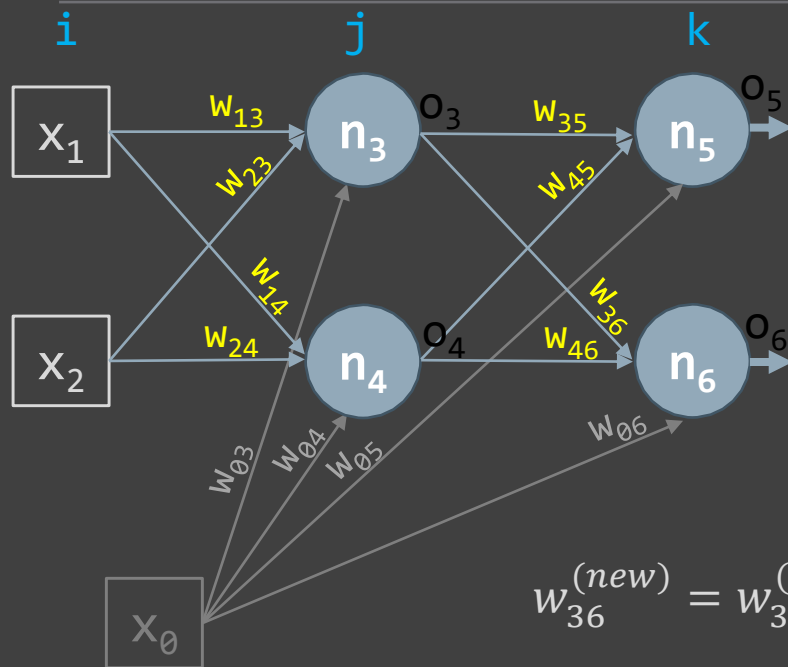
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465						
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{36}^{(new)} &= w_{36}^{(old)} + 0.4 \cdot \delta_6 \cdot o_3 = \\ &= 0.5 + 0.4 \cdot (-0.042) \cdot 0.64 = 0.489 \end{aligned}$$

$$\delta_5 = -0.1368$$

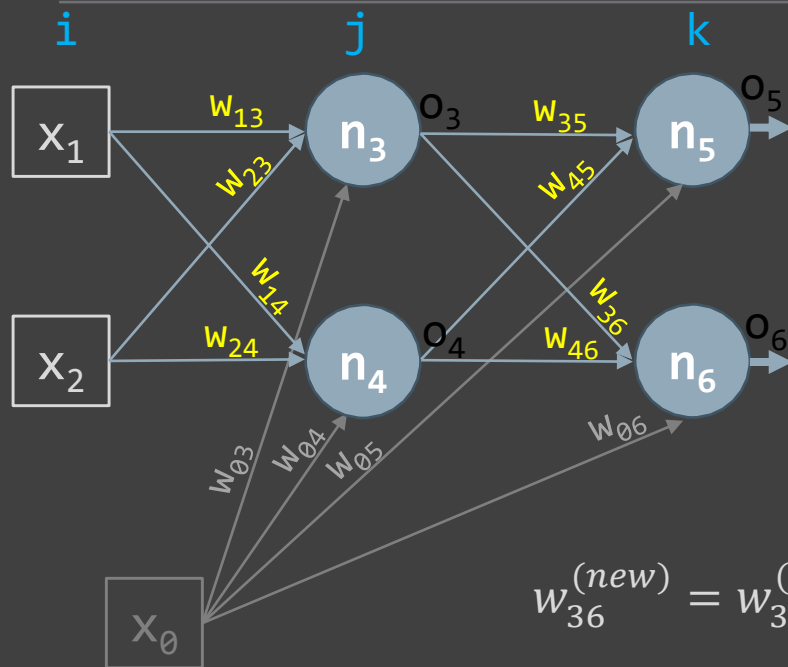
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465						
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{36}^{(new)} = w_{36}^{(old)} + 0.4 \cdot \delta_6 \cdot o_3 =$$

$$= 0.5 + 0.4 \cdot (-0.042) \cdot 0.64 = 0.489$$

$$\delta_5 = -0.1368$$

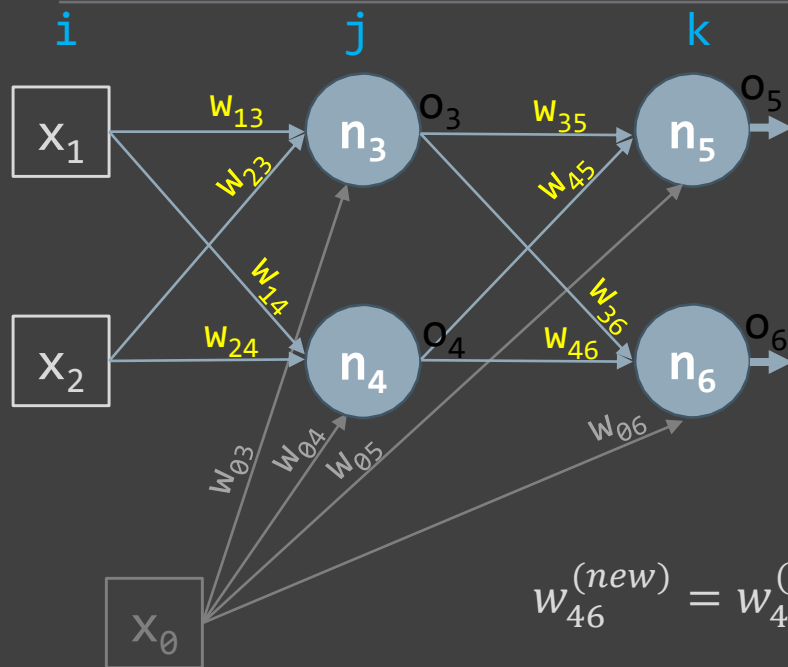
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465	0.489					
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{46}^{(new)} &= w_{46}^{(old)} + 0.4 \cdot \delta_6 \cdot o_4 = \\ &= 0.5 + 0.4 \cdot (-0.042) \cdot 0.64 = 0.489 \end{aligned}$$

$$\delta_5 = -0.1368$$

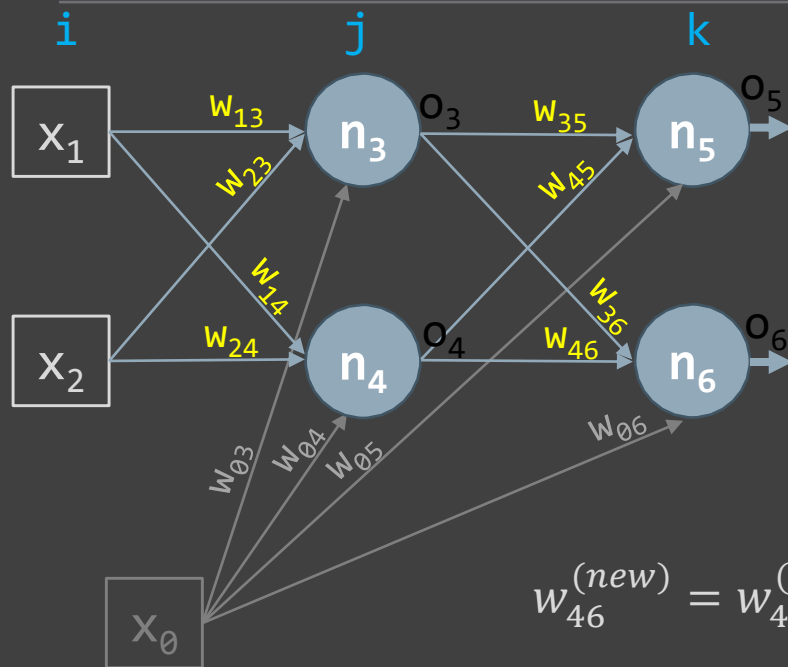
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465	0.489					
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{46}^{(new)} = w_{46}^{(old)} + 0.4 \cdot \delta_6 \cdot o_4 =$$

$$= 0.5 + 0.4 \cdot (-0.042) \cdot 0.64 = 0.489$$

$$\delta_5 = -0.1368$$

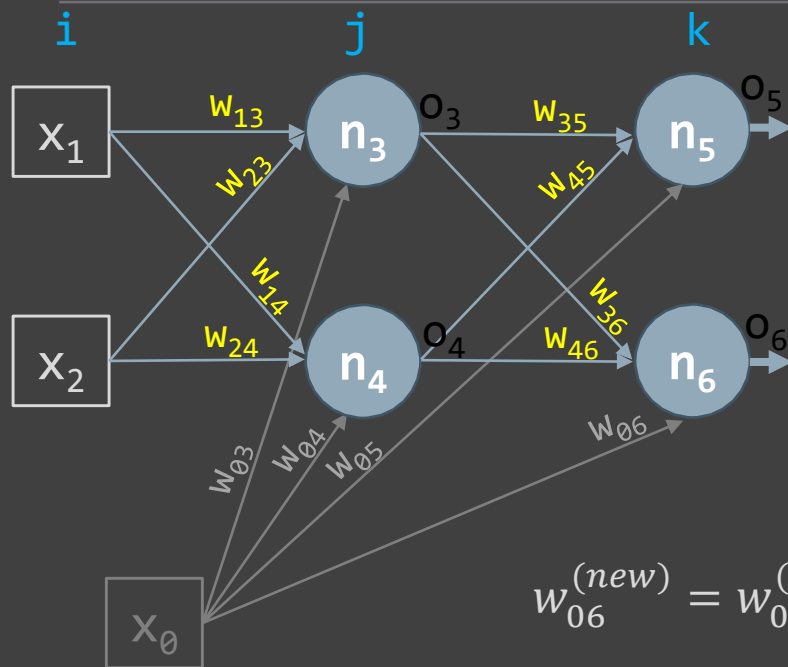
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{06}^{(new)} &= w_{06}^{(old)} + 0.4 \cdot \delta_6 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.042) \cdot 1 = 0.4832 \end{aligned}$$

$$\delta_5 = -0.1368$$

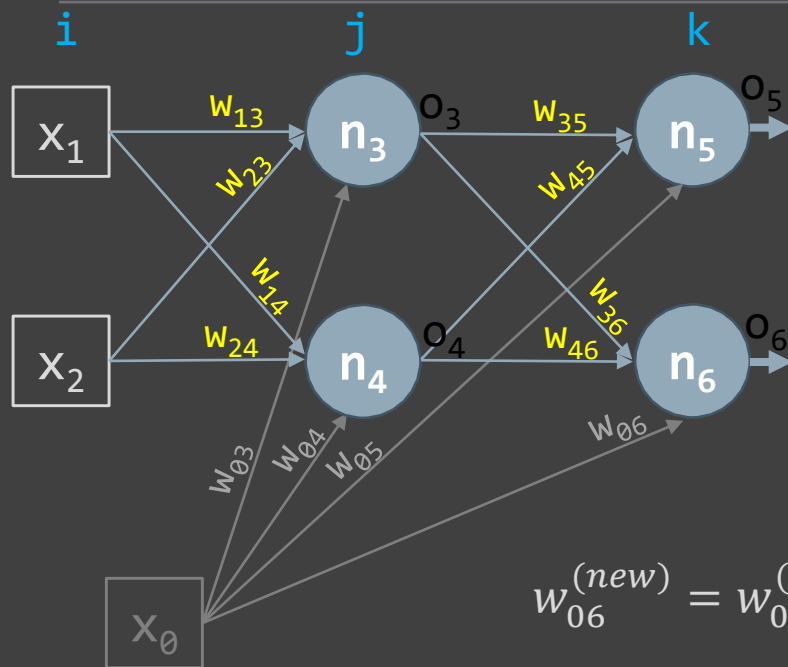
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453						0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{06}^{(new)} = w_{06}^{(old)} + 0.4 \cdot \delta_6 \cdot x_0 = 0.5 + 0.4 \cdot (-0.042) \cdot 1 = 0.4832$$

$$\delta_5 = -0.1368$$

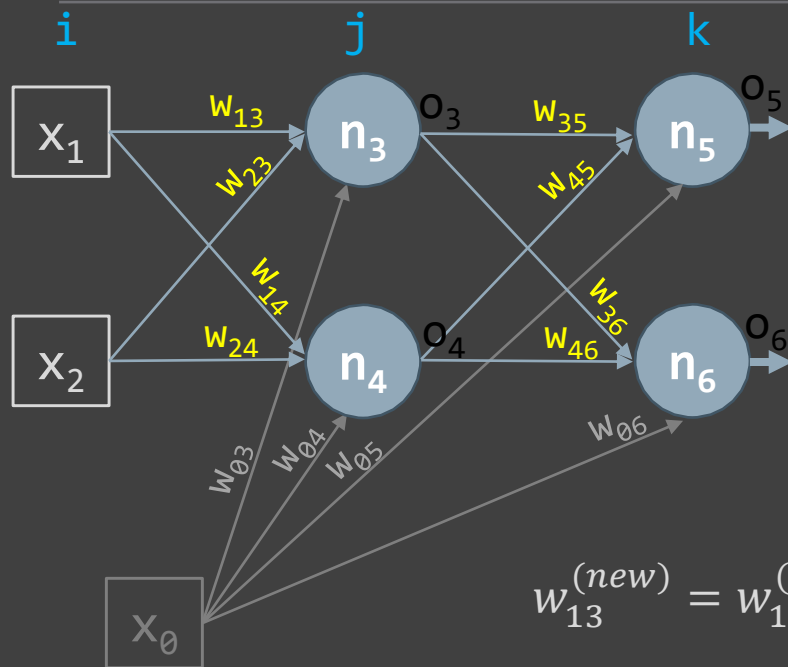
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			.4453	.4832					0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{13}^{(new)} &= w_{13}^{(old)} + 0.4 \cdot \delta_3 \cdot x_1 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.05 = 0.4996 \end{aligned}$$

$$\delta_5 = -0.1368$$

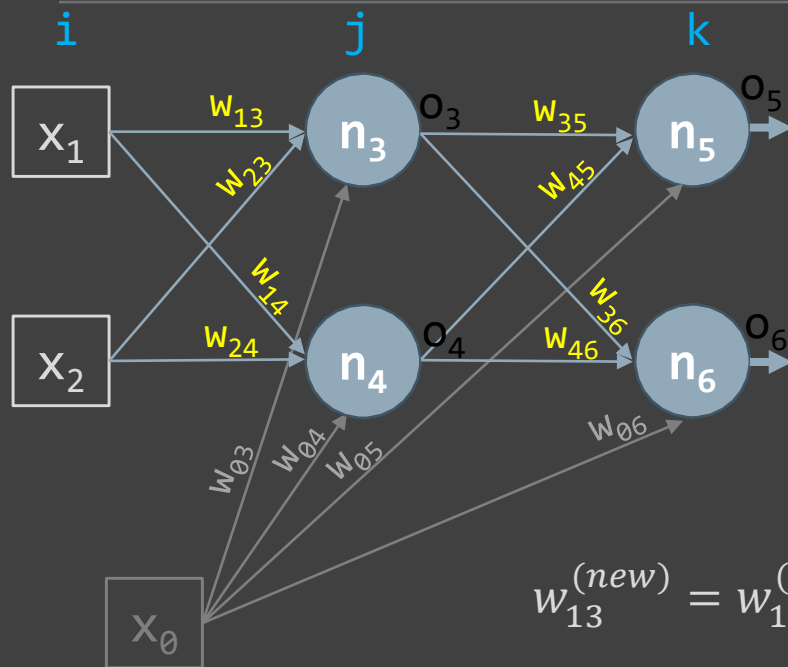
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			.4453	.4832					0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{13}^{(new)} &= w_{13}^{(old)} + 0.4 \cdot \delta_3 \cdot x_1 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.05 = 0.4996 \end{aligned}$$

$$\delta_5 = -0.1368$$

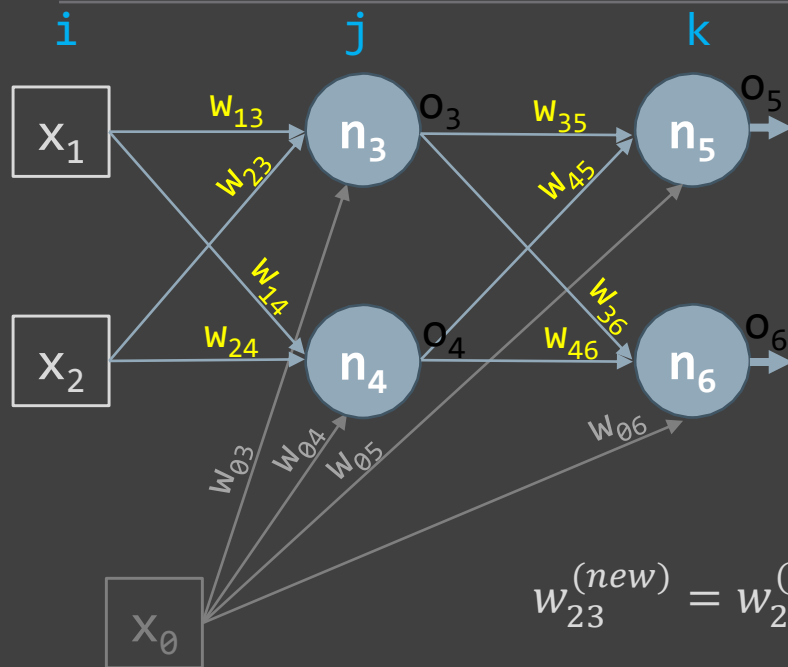
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			.4453	.4832	.4996				0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{23}^{(new)} &= w_{23}^{(old)} + 0.4 \cdot \delta_3 \cdot x_2 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.10 = 0.4992 \end{aligned}$$

$$\delta_5 = -0.1368$$

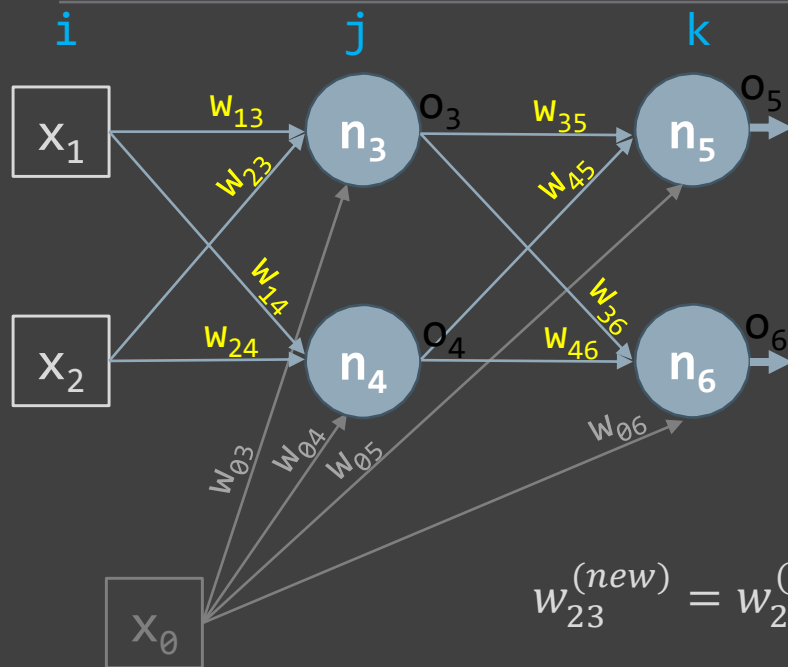
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			.4453	.4832	.4996				0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{23}^{(new)} = w_{23}^{(old)} + 0.4 \cdot \delta_3 \cdot x_2 = 0.5 + 0.4 \cdot (-0.0206) \cdot 0.10 = 0.4992$$

$$\delta_5 = -0.1368$$

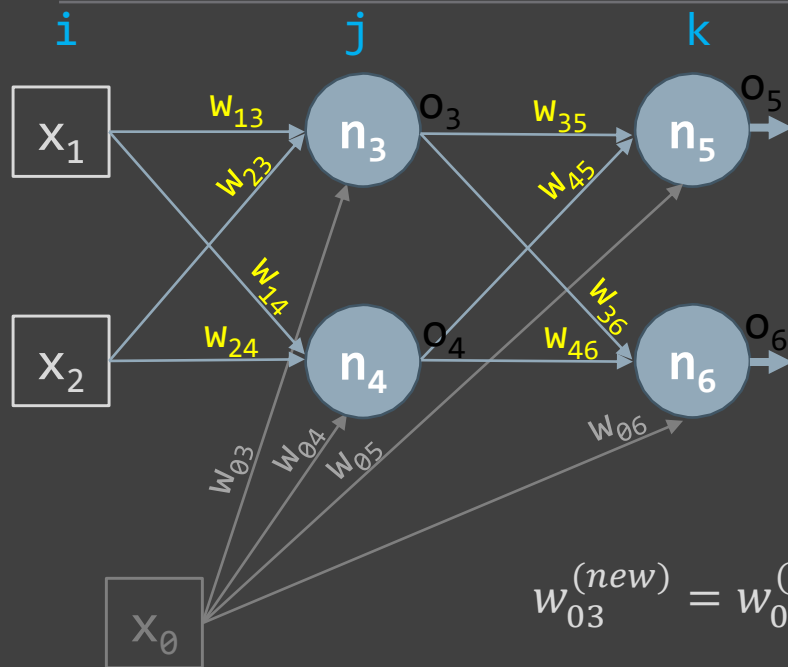
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			0.4453	0.4832	0.4996	0.4992			0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{03}^{(new)} &= w_{03}^{(old)} + 0.4 \cdot \delta_3 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 1 = 0.4918 \end{aligned}$$

$$\delta_5 = -0.1368$$

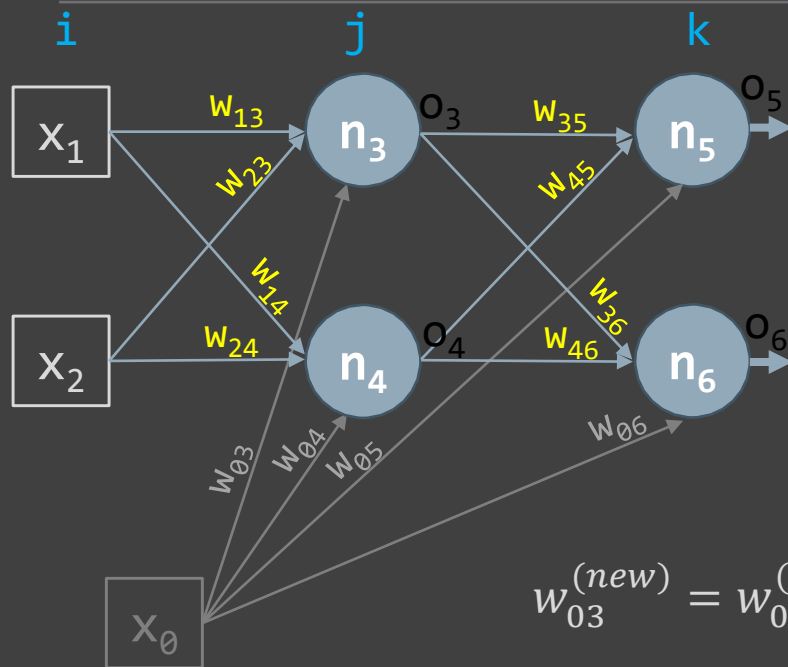
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78			.4453	.4832	.4996	.4992			0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{03}^{(new)} &= w_{03}^{(old)} + 0.4 \cdot \delta_3 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 1 = 0.4918 \end{aligned}$$

$$\delta_5 = -0.1368$$

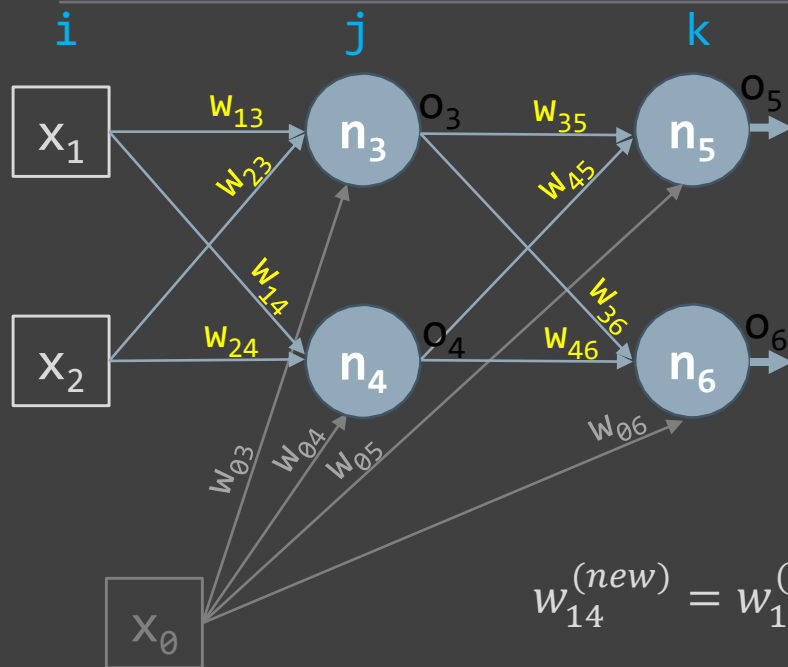
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992			0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned}
 w_{14}^{(new)} &= w_{14}^{(old)} + 0.4 \cdot \delta_4 \cdot x_1 = \\
 &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.05 = 0.4996
 \end{aligned}$$

$$\delta_5 = -0.1368$$

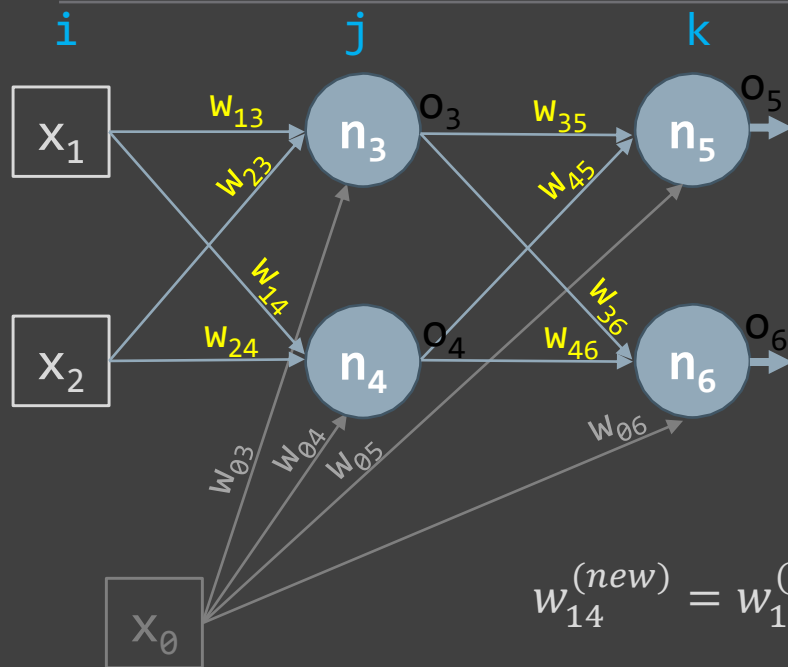
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992			0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{14}^{(new)} &= w_{14}^{(old)} + 0.4 \cdot \delta_4 \cdot x_1 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.05 = 0.4996 \end{aligned}$$

$$\delta_5 = -0.1368$$

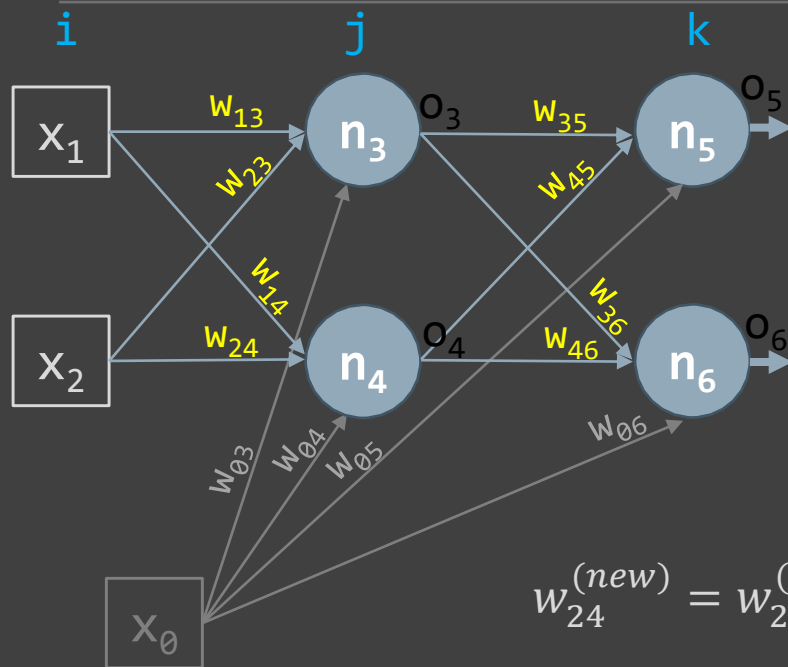
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992	.4996		0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{24}^{(new)} &= w_{24}^{(old)} + 0.4 \cdot \delta_4 \cdot x_2 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.10 = 0.4992 \end{aligned}$$

$$\delta_5 = -0.1368$$

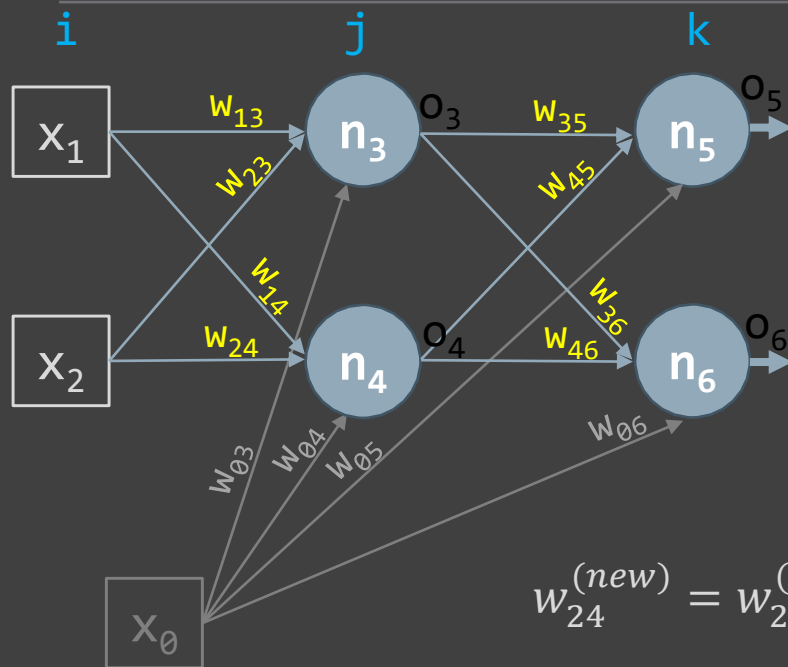
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992	.4996		0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{24}^{(new)} &= w_{24}^{(old)} + 0.4 \cdot \delta_4 \cdot x_2 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 0.10 = 0.4992 \end{aligned}$$

$$\delta_5 = -0.1368$$

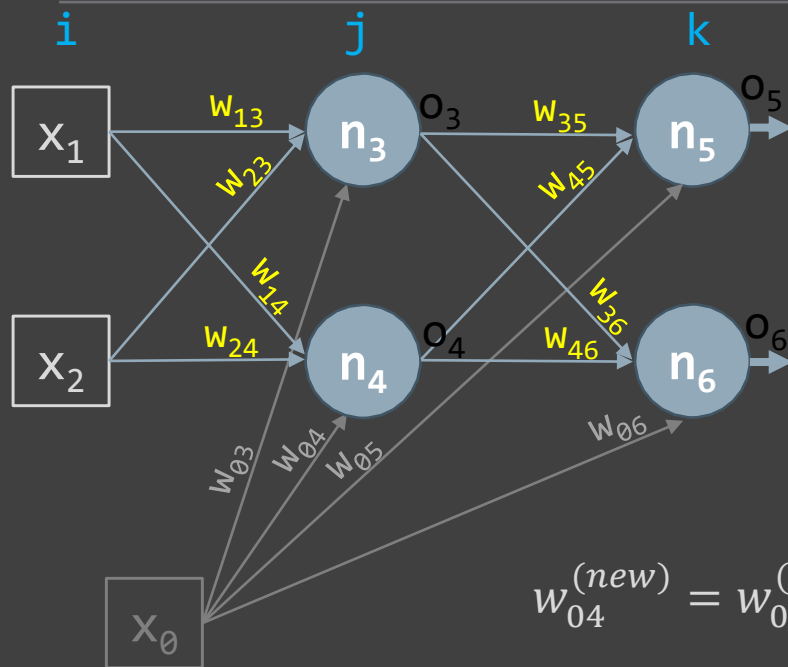
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{04}^{(new)} &= w_{04}^{(old)} + 0.4 \cdot \delta_4 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 1 = 0.4918 \end{aligned}$$

$$\delta_5 = -0.1368$$

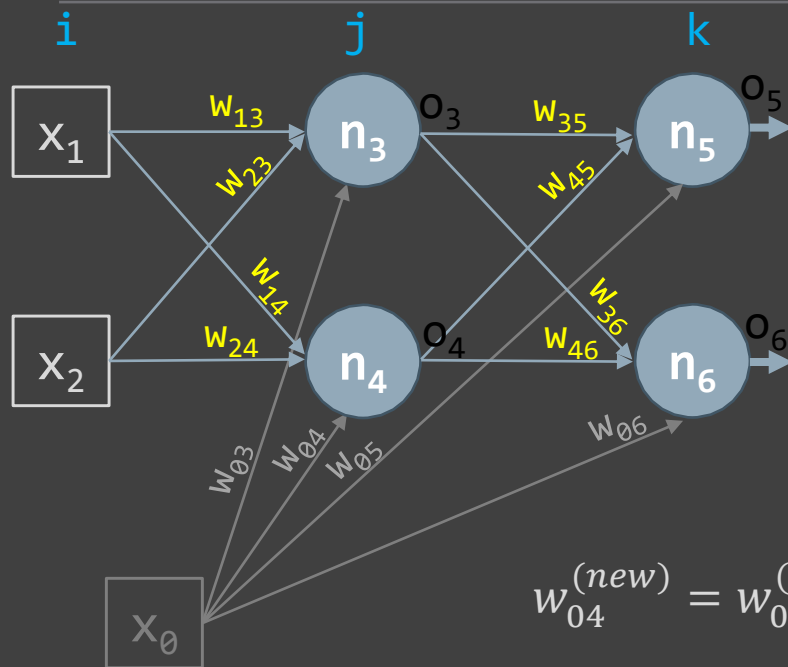
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918		.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{04}^{(new)} &= w_{04}^{(old)} + 0.4 \cdot \delta_4 \cdot x_0 = \\ &= 0.5 + 0.4 \cdot (-0.0206) \cdot 1 = 0.4918 \end{aligned}$$

$$\delta_5 = -0.1368$$

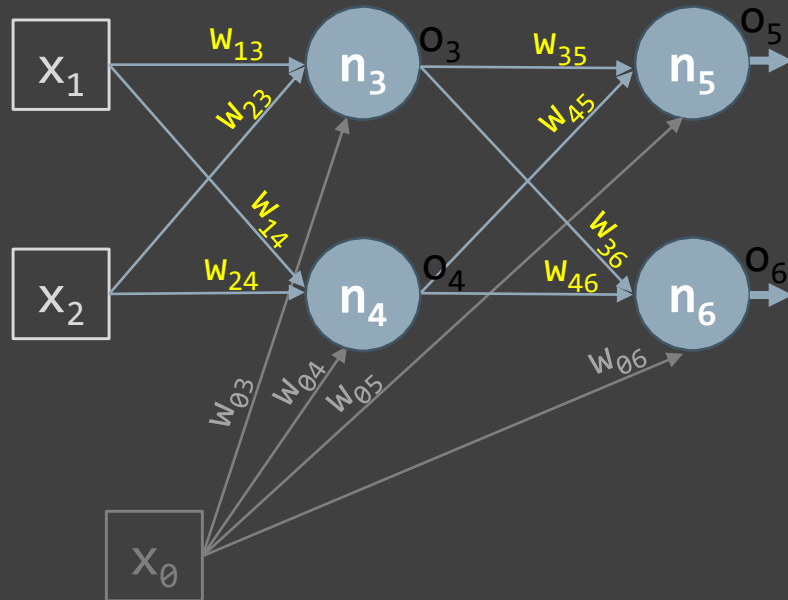
$$\delta_6 = -0.0420$$

$$\delta_3 = -0.0206$$

$$\delta_4 = -0.0206$$

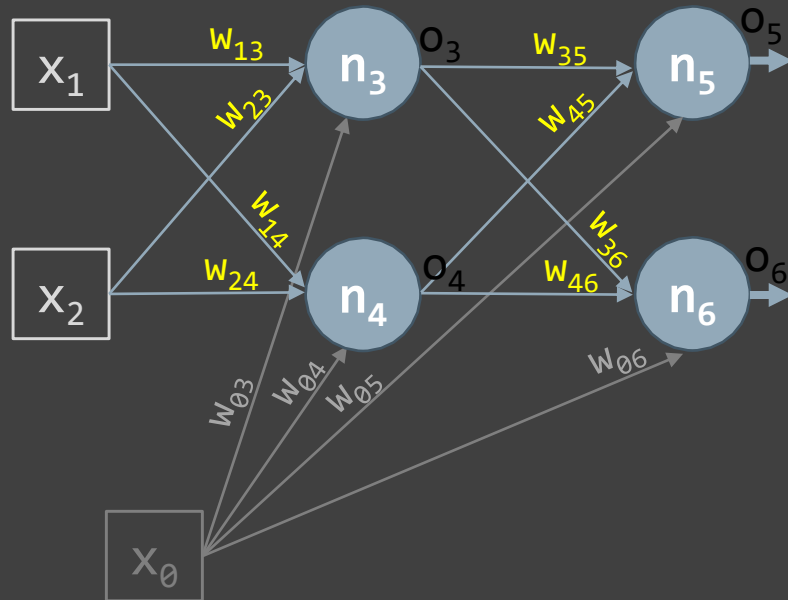
x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_3 = \sum_i w_{i3} x_i = w_{03} x_0 + w_{13} x_1 + w_{23} x_2$$

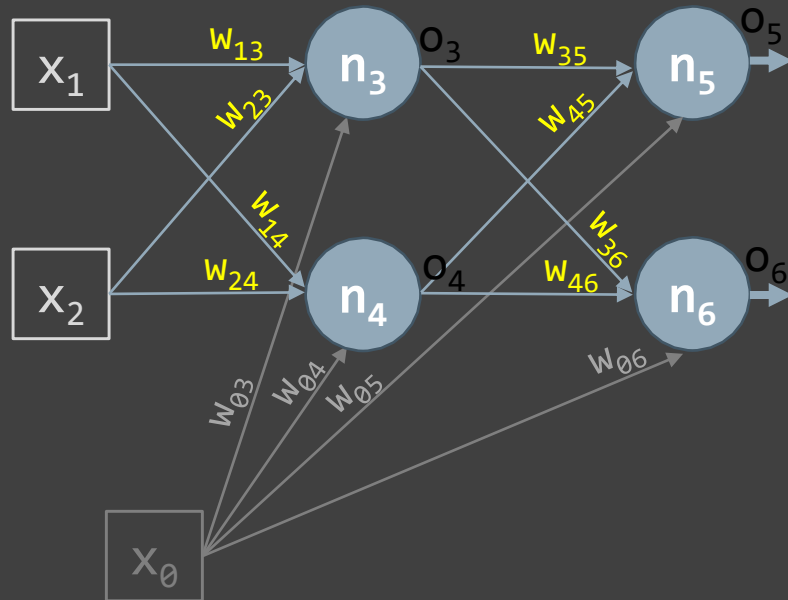
$$= 0.4918 \cdot 1 + 0.4996 \cdot 0.20 + 0.4992 \cdot 0.45$$

$$= 0.81636$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.81636}} = 0.69$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489				
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_3 = \sum_i w_{i3} x_i = w_{03} x_0 + w_{13} x_1 + w_{23} x_2$$

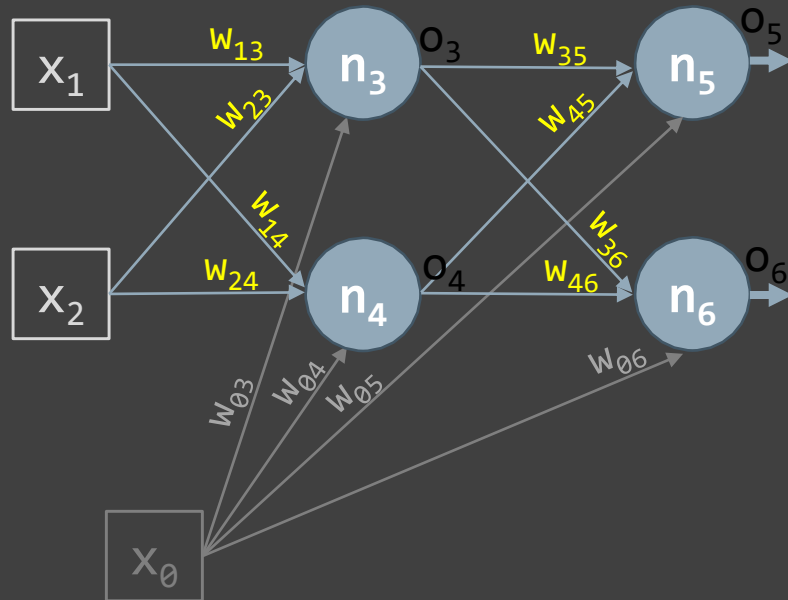
$$= 0.4918 \cdot 1 + 0.4996 \cdot 0.20 + 0.4992 \cdot 0.45$$

$$= 0.81636$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.81636}} = 0.69$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69			
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_4 = \sum_i w_{i4} x_i = w_{04} x_0 + w_{14} x_1 + w_{24} x_2$$

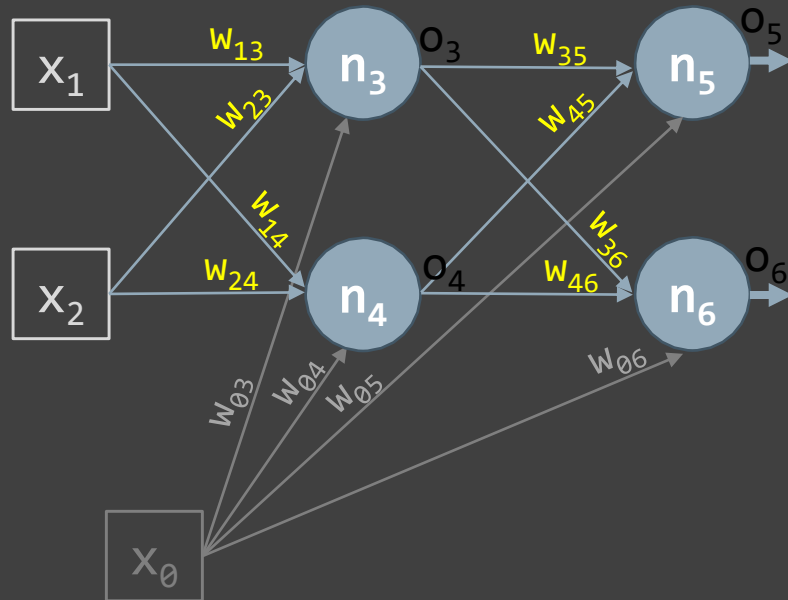
$$= 0.4918 \cdot 1 + 0.4996 \cdot 0.20 + 0.4992 \cdot 0.45$$

$$= 0.81636$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.81636}} = 0.69$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69			
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_4 = \sum_i w_{i4} x_i = w_{04} x_0 + w_{14} x_1 + w_{24} x_2$$

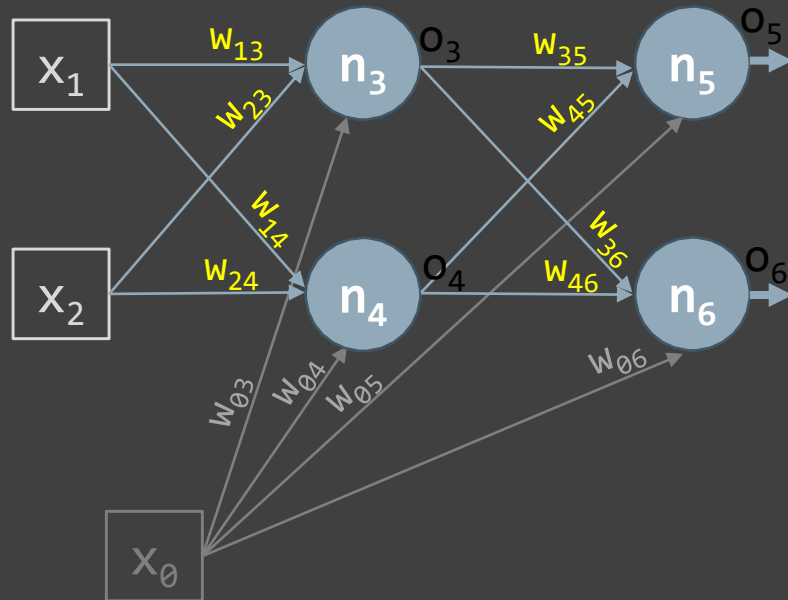
$$= 0.4918 \cdot 1 + 0.4996 \cdot 0.20 + 0.4992 \cdot 0.45$$

$$= 0.81636$$

$$o_3 = f(C_3) = \frac{1}{1 + e^{-C_3}} = \frac{1}{1 + e^{-0.81636}} = 0.69$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69		
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_5 = \sum_j w_{j5} o_j = w_{05} x_0 + w_{35} o_3 + w_{45} o_4$$

$$= 0.4453 \cdot 1 + 0.465 \cdot 0.69 + 0.465 \cdot 0.69$$

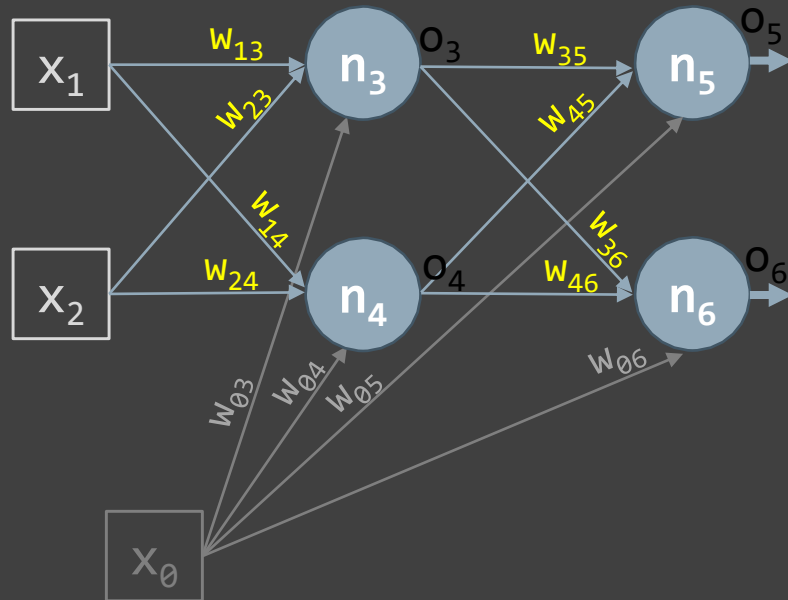
$$= 1.087$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.087}}$$

$$= 0.748$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69		
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_5 = \sum_j w_{j5} o_j = w_{05} x_0 + w_{35} o_3 + w_{45} o_4$$

$$= 0.4453 \cdot 1 + 0.465 \cdot 0.69 + 0.465 \cdot 0.69$$

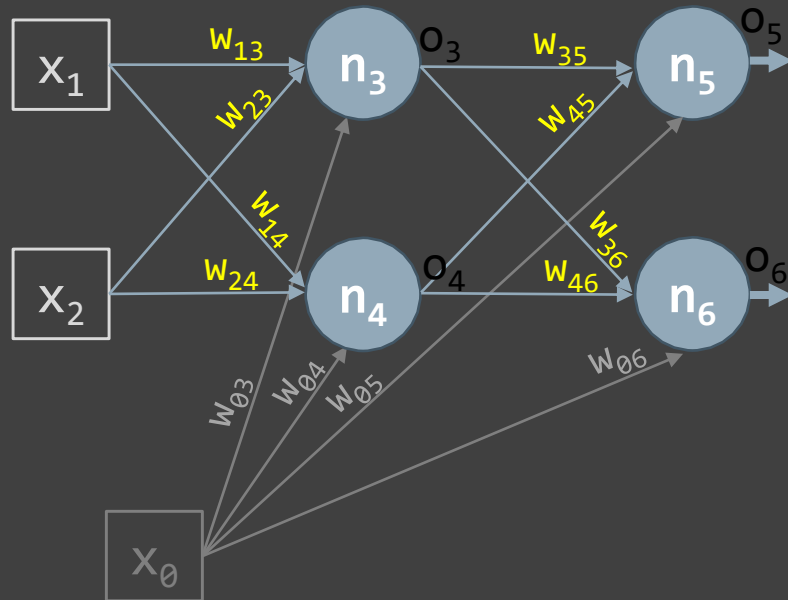
$$= 1.087$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.087}}$$

$$= 0.748$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_6 = \sum_j w_{j6} o_j = w_{06} x_0 + w_{36} o_3 + w_{46} o_4$$

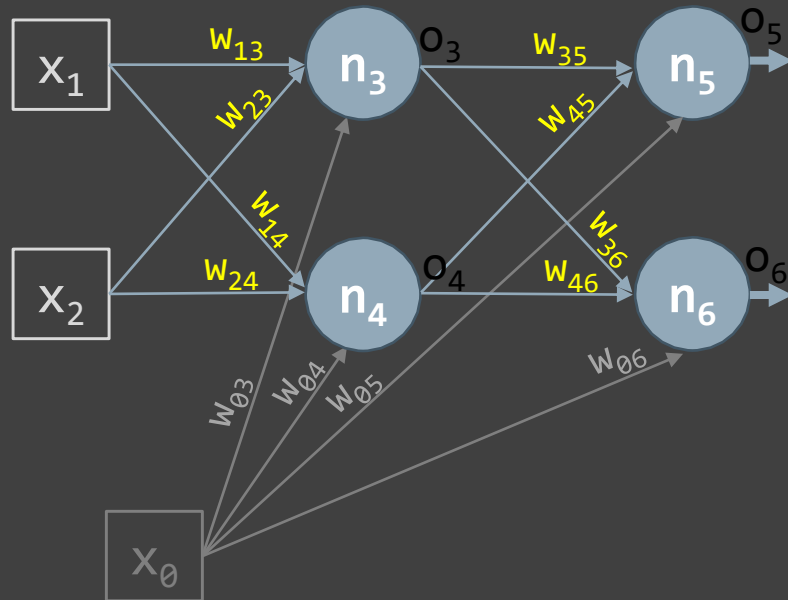
$$= 0.4832 \cdot 1 + 0.489 \cdot 0.69 + 0.489 \cdot 0.69$$

$$= 1.158$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.158}} = 0.76$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



$$C_6 = \sum_j w_{j6} o_j = w_{06} x_0 + w_{36} o_3 + w_{46} o_4$$

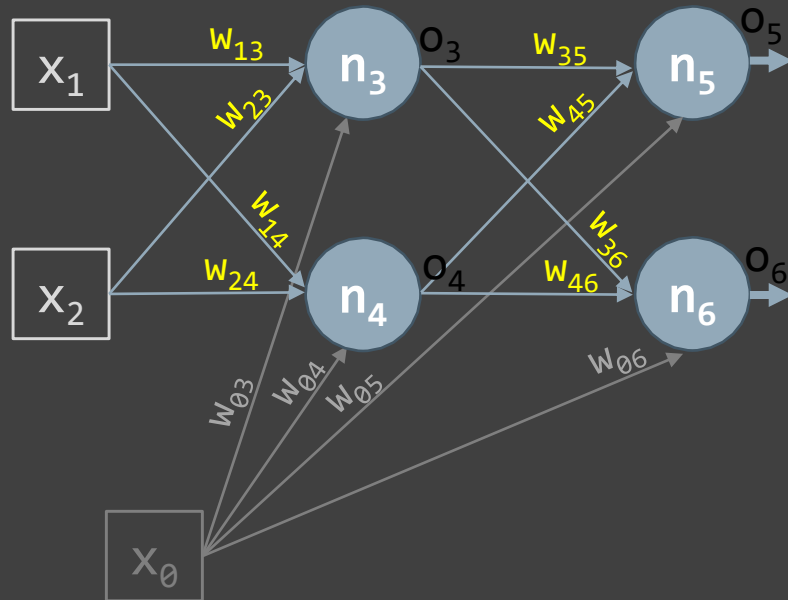
$$= 0.4832 \cdot 1 + 0.489 \cdot 0.69 + 0.489 \cdot 0.69$$

$$= 1.158$$

$$o_5 = f(C_5) = \frac{1}{1 + e^{-C_5}} = \frac{1}{1 + e^{-1.158}} = 0.76$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

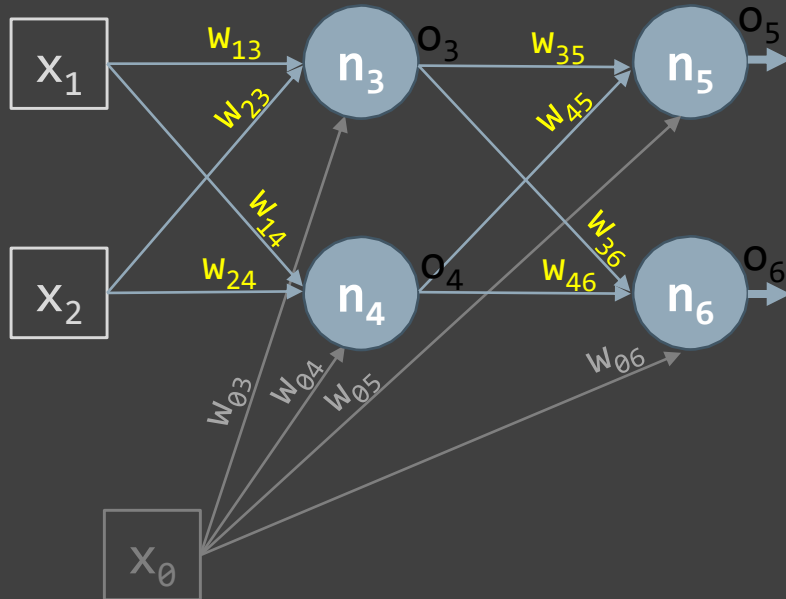
$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

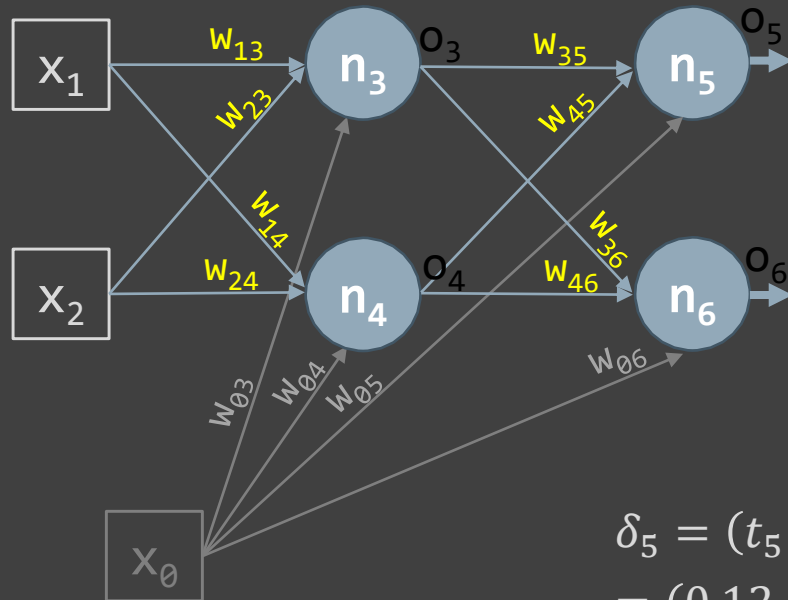
Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

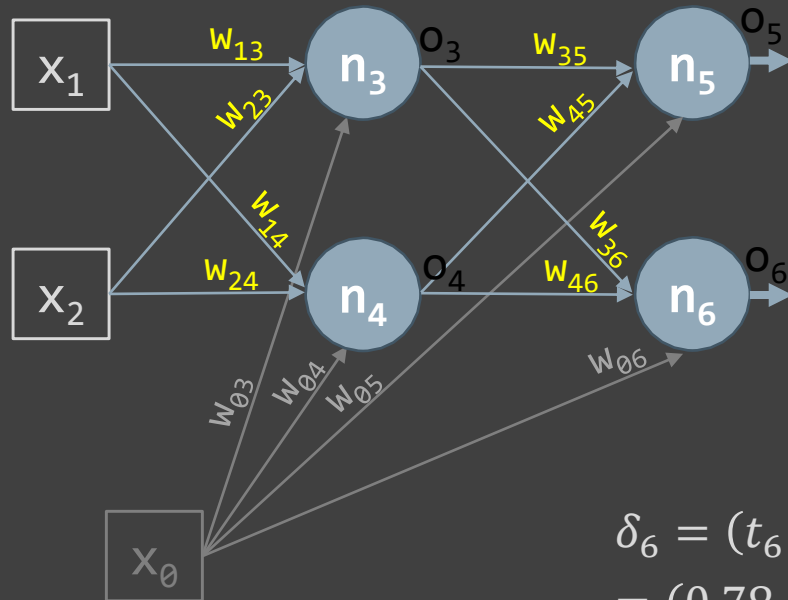
$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

$$\begin{aligned} \delta_5 &= (t_5 - o_5) \cdot o_5 \cdot (1 - o_5) = \\ &= (0.12 - 0.748) \cdot 0.748 \cdot (1 - 0.748) = -0.1184 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

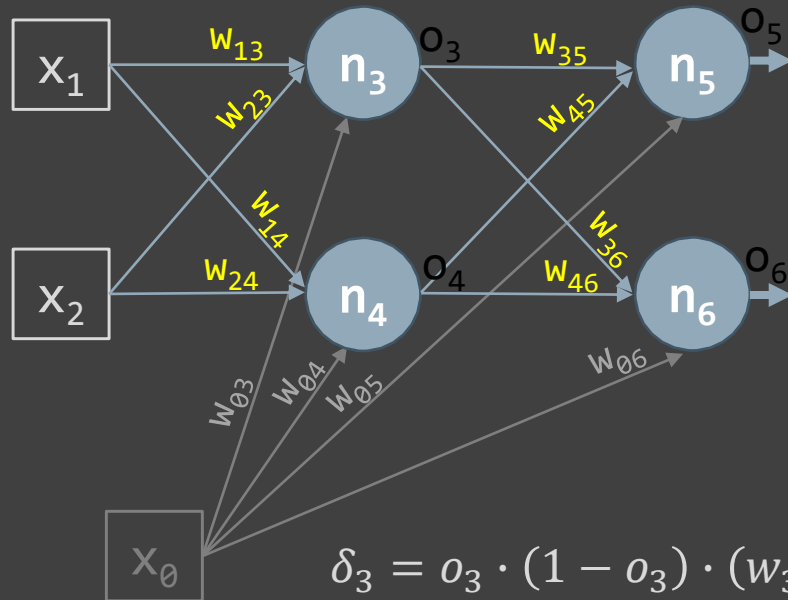
Υπολογίζουμε πρώτα όλα τα δ

$$\delta_5 = -0.1184$$

$$\begin{aligned} \delta_6 &= (t_6 - o_6) \cdot o_6 \cdot (1 - o_6) = \\ &= (0.78 - 0.76) \cdot 0.76 \cdot (1 - 0.76) = -0.0036 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

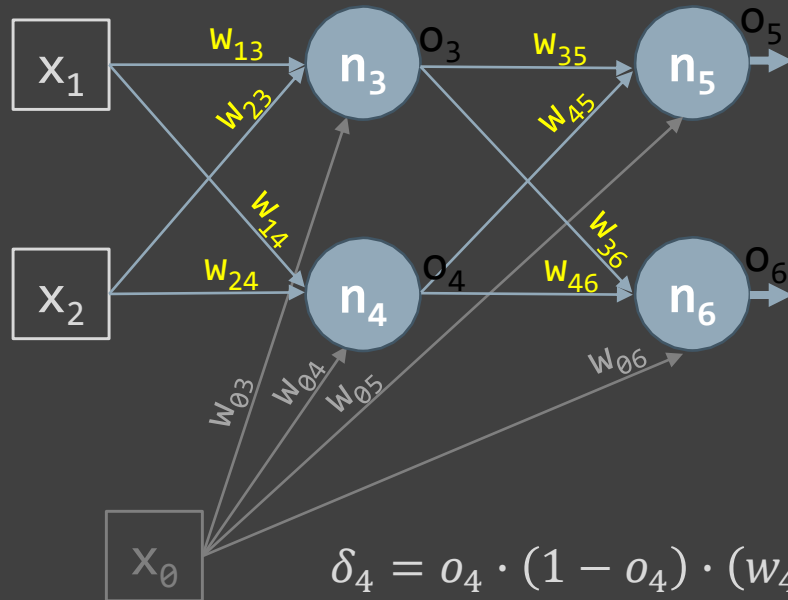
$$\delta_5 = -0.1184$$

$$\delta_6 = -0.0036$$

$$\begin{aligned} \delta_3 &= o_3 \cdot (1 - o_3) \cdot (w_{35} \cdot \delta_5 + w_{36} \cdot \delta_6) = \\ &= 0.69 \cdot (1 - 0.69) \cdot (0.465 \cdot (-0.1184) + 0.489 \cdot (-0.0036)) = -0.0122 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Υπολογίζουμε πρώτα όλα τα δ

$$\delta_5 = -0.1184$$

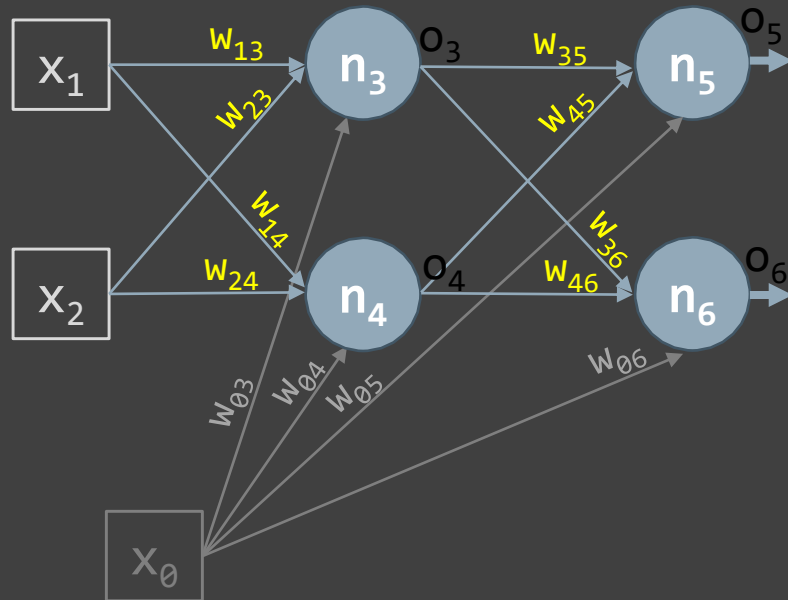
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\begin{aligned} \delta_4 &= o_4 \cdot (1 - o_4) \cdot (w_{45} \cdot \delta_5 + w_{46} \cdot \delta_6) = \\ &= 0.69 \cdot (1 - 0.69) \cdot (0.465 \cdot (-0.1184) + 0.489 \cdot (-0.0036)) = -0.0122 \end{aligned}$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\delta_5 = -0.1184$$

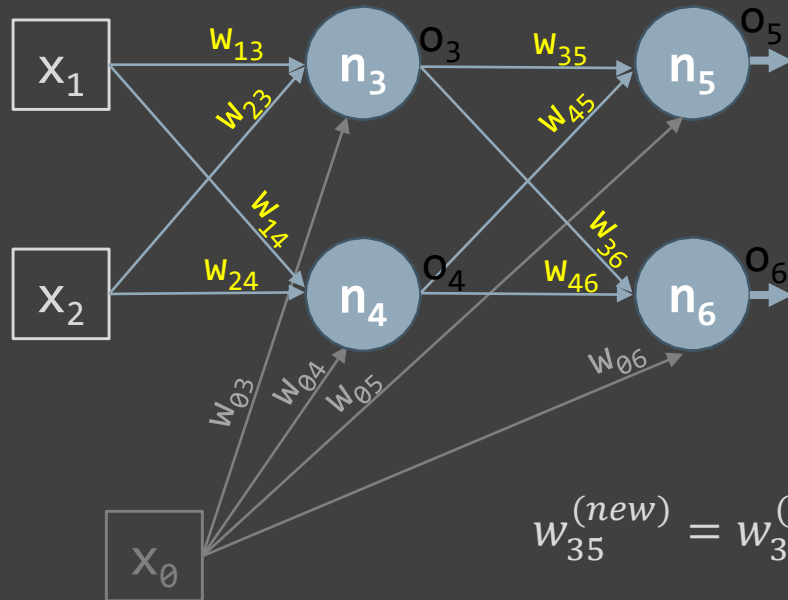
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{35}^{(new)} &= w_{35}^{(old)} + 0.4 \cdot \delta_5 \cdot o_3 = \\ &= 0.465 + 0.4 \cdot (-0.1184) \cdot 0.69 = 0.4323 \end{aligned}$$

$$\delta_5 = -0.1184$$

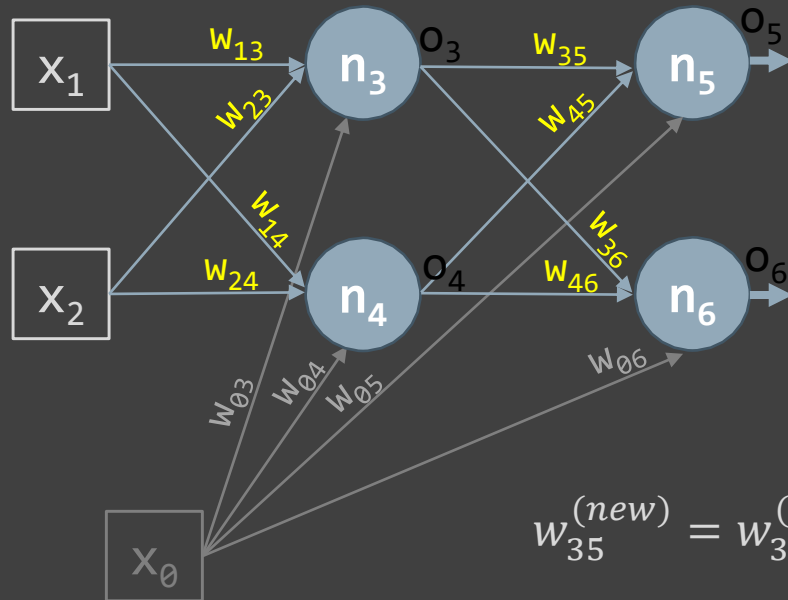
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53																
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{35}^{(new)} &= w_{35}^{(old)} + 0.4 \cdot \delta_5 \cdot o_3 = \\ &= 0.465 + 0.4 \cdot (-0.1184) \cdot 0.69 = 0.4323 \end{aligned}$$

$$\delta_5 = -0.1184$$

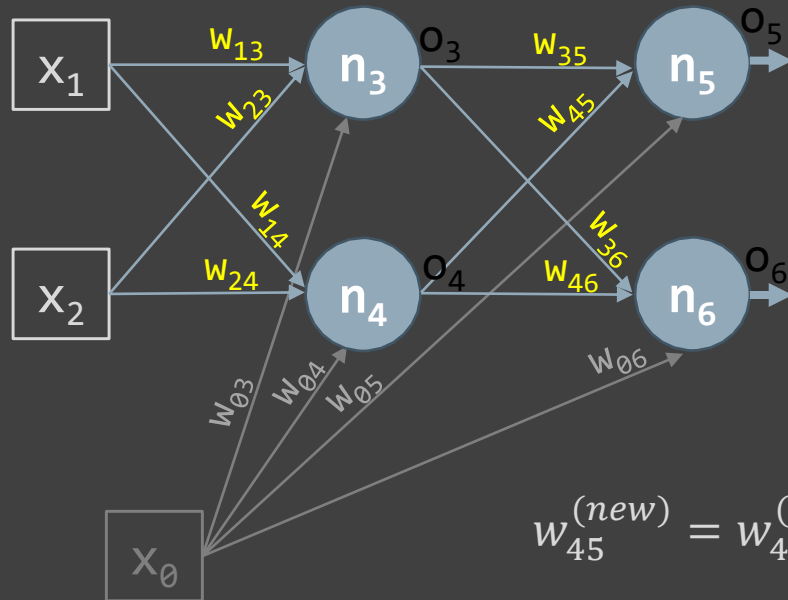
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53									.4323							
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{45}^{(new)} &= w_{45}^{(old)} + 0.4 \cdot \delta_5 \cdot o_4 = \\ &= 0.465 + 0.4 \cdot (-0.1184) \cdot 0.69 = 0.4323 \end{aligned}$$

$$\delta_5 = -0.1184$$

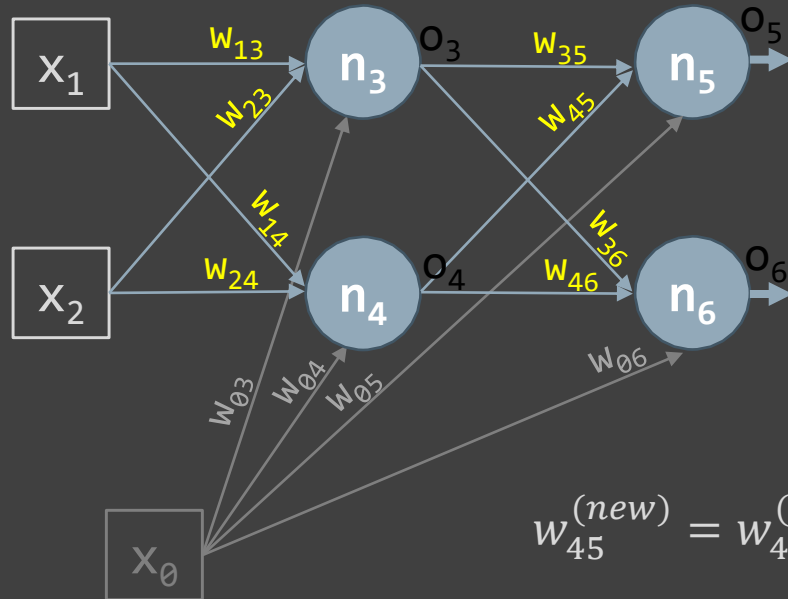
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53									.4323							
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{45}^{(new)} &= w_{45}^{(old)} + 0.4 \cdot \delta_5 \cdot o_4 = \\ &= 0.465 + 0.4 \cdot (-0.1184) \cdot 0.69 = 0.4323 \end{aligned}$$

$$\delta_5 = -0.1184$$

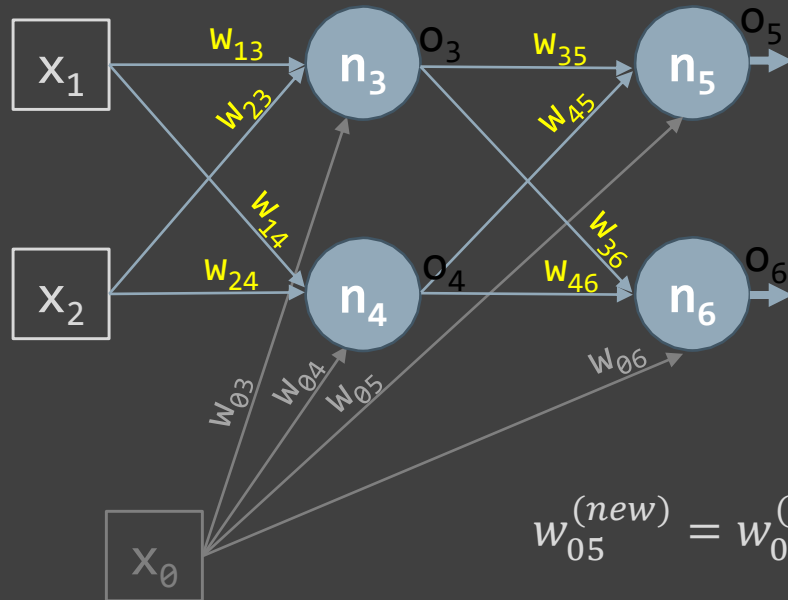
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53									.4323	.4323						
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{05}^{(new)} = w_{05}^{(old)} + 0.4 \cdot \delta_5 \cdot x_0 =$$

$$= 0.4453 + 0.4 \cdot (-0.1184) \cdot 1 = 0.398$$

$$\delta_5 = -0.1184$$

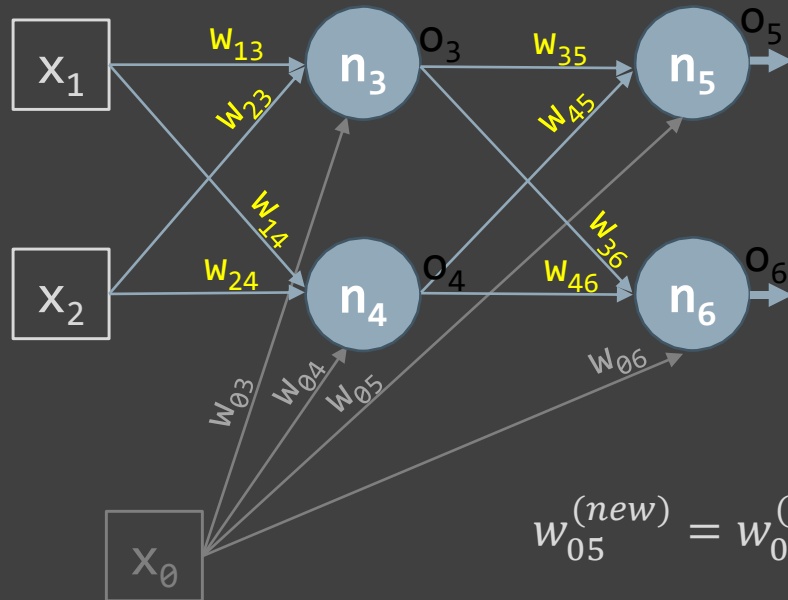
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53									.4323	.4323						
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$w_{05}^{(new)} = w_{05}^{(old)} + 0.4 \cdot \delta_5 \cdot x_0 =$$

$$= 0.4453 + 0.4 \cdot (-0.1184) \cdot 1 = 0.398$$

$$\delta_5 = -0.1184$$

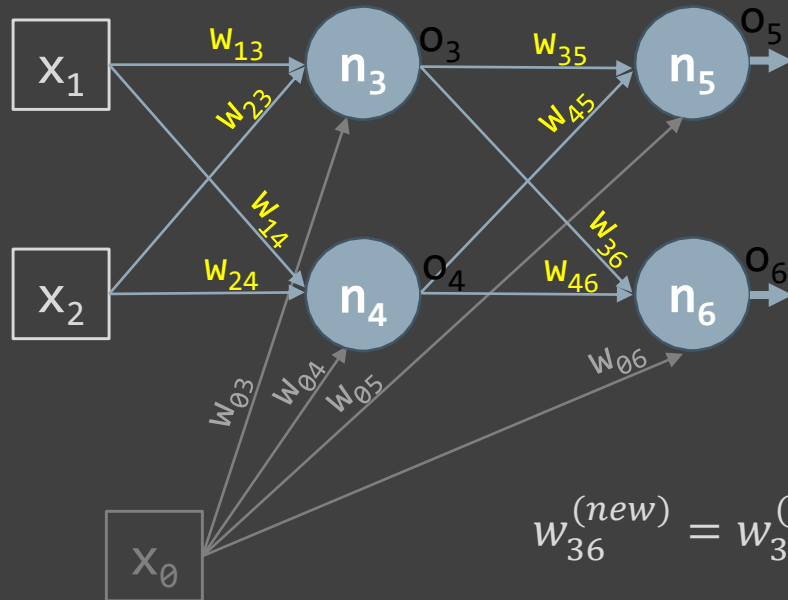
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323						
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{36}^{(new)} &= w_{36}^{(old)} + 0.4 \cdot \delta_6 \cdot o_3 = \\ &= 0.489 + 0.4 \cdot (-0.0032) \cdot 0.69 = 0.488 \end{aligned}$$

$$\delta_5 = -0.1184$$

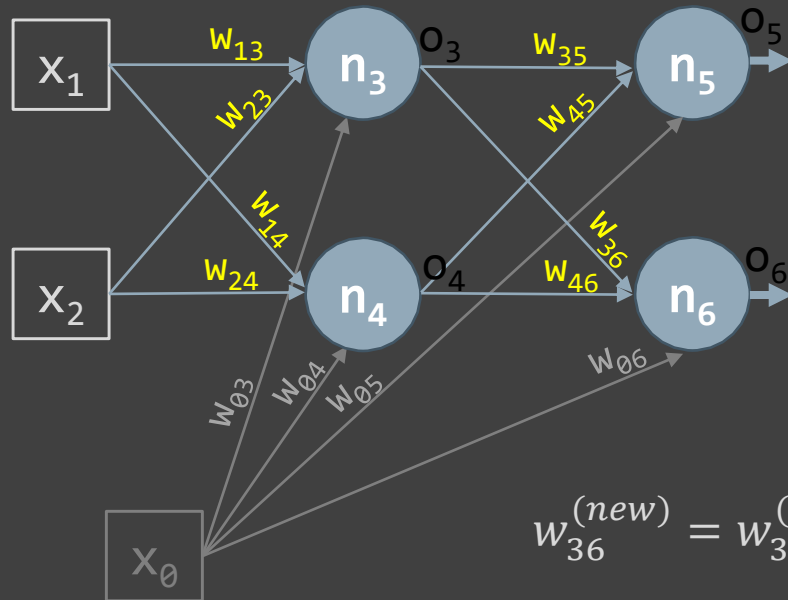
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323						
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{36}^{(new)} &= w_{36}^{(old)} + 0.4 \cdot \delta_6 \cdot o_3 = \\ &= 0.489 + 0.4 \cdot (-0.0032) \cdot 0.69 = 0.488 \end{aligned}$$

$$\delta_5 = -0.1184$$

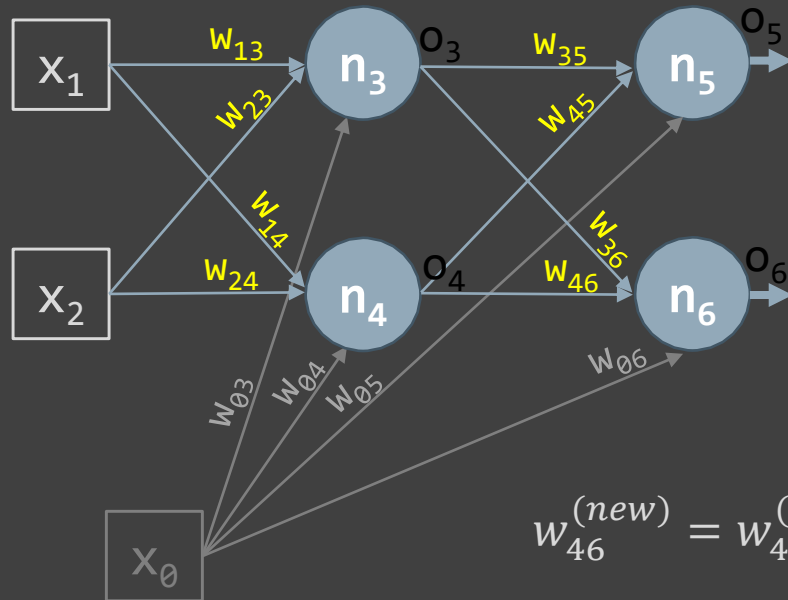
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323	0.488					
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{46}^{(new)} &= w_{46}^{(old)} + 0.4 \cdot \delta_6 \cdot o_4 = \\ &= 0.489 + 0.4 \cdot (-0.0032) \cdot 0.69 = 0.488 \end{aligned}$$

$$\delta_5 = -0.1184$$

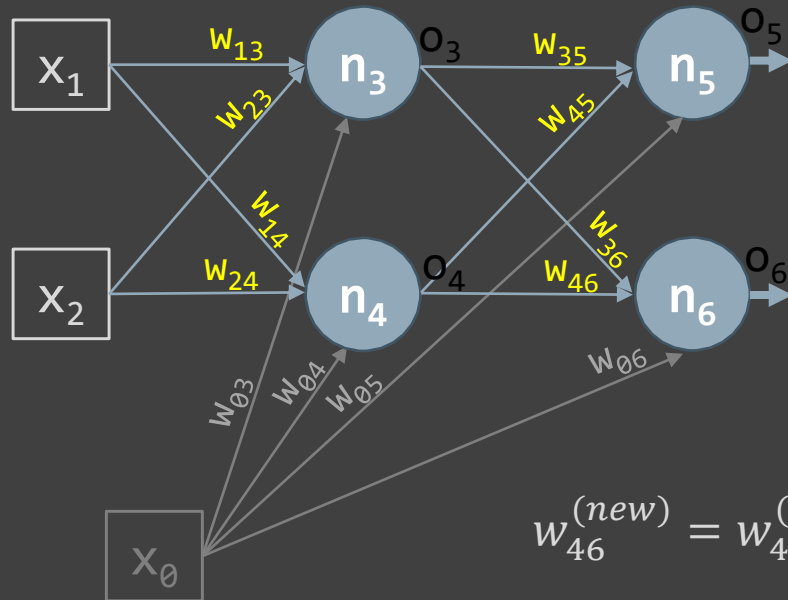
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323	0.488					
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{46}^{(new)} &= w_{46}^{(old)} + 0.4 \cdot \delta_6 \cdot o_4 = \\ &= 0.489 + 0.4 \cdot (-0.0032) \cdot 0.69 = 0.488 \end{aligned}$$

$$\delta_5 = -0.1184$$

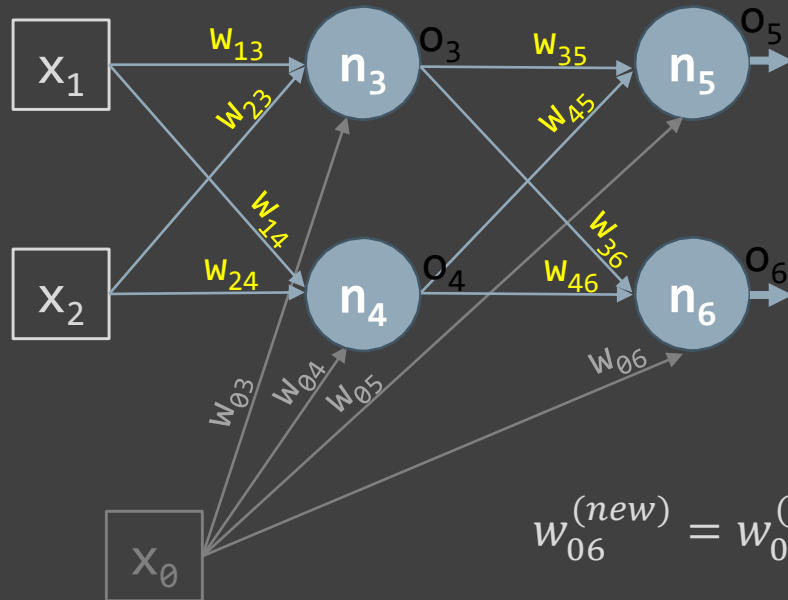
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{06}^{(new)} &= w_{06}^{(old)} + 0.4 \cdot \delta_6 \cdot x_0 = \\ &= 0.4832 + 0.4 \cdot (-0.0032) \cdot 1 = 0.4819 \end{aligned}$$

$$\delta_5 = -0.1184$$

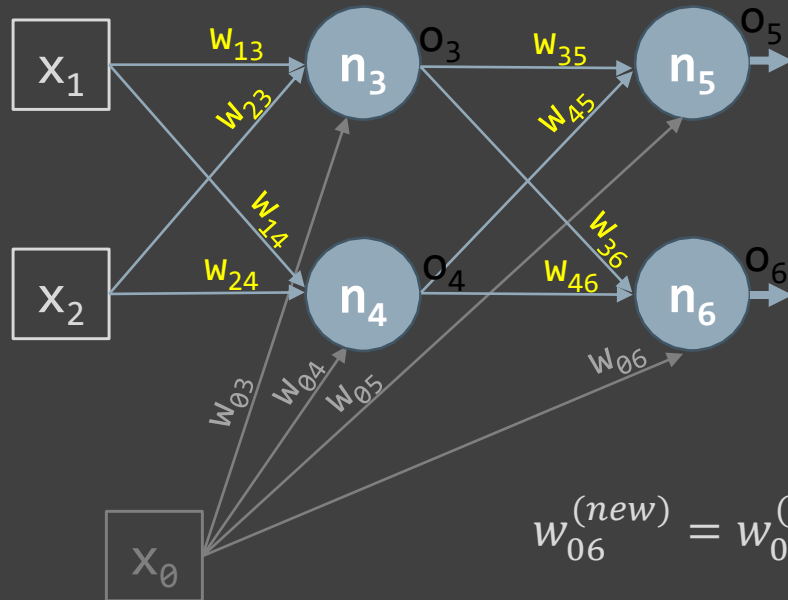
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398						.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{06}^{(new)} &= w_{06}^{(old)} + 0.4 \cdot \delta_6 \cdot x_0 = \\ &= 0.4832 + 0.4 \cdot (-0.0032) \cdot 1 = 0.4819 \end{aligned}$$

$$\delta_5 = -0.1184$$

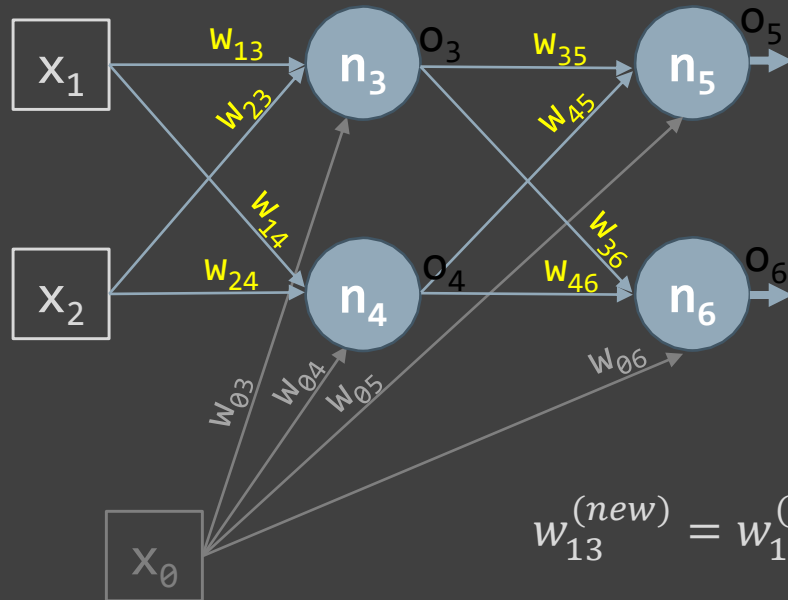
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819					.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{13}^{(new)} &= w_{13}^{(old)} + 0.4 \cdot \delta_3 \cdot x_1 = \\ &= 0.4996 + 0.4 \cdot (-0.0122) \cdot 0.2 = 0.4986 \end{aligned}$$

$$\delta_5 = -0.1184$$

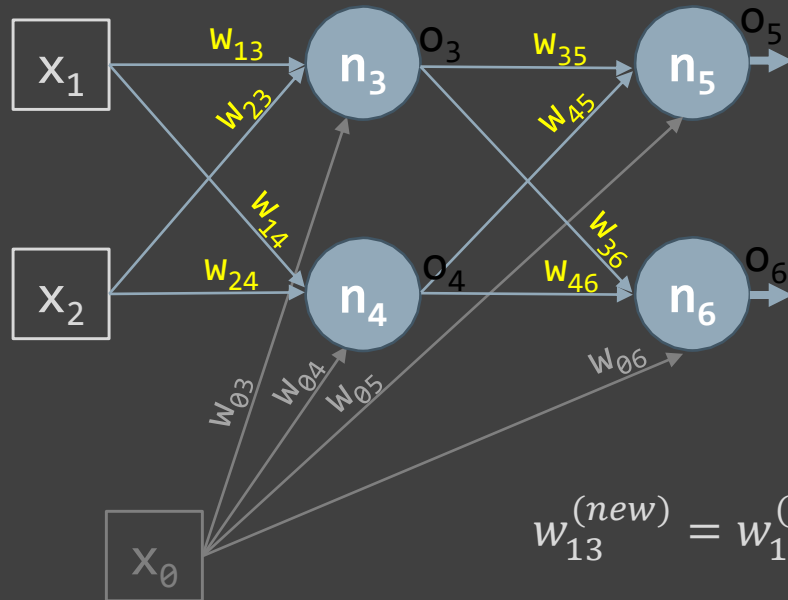
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819					.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{13}^{(new)} &= w_{13}^{(old)} + 0.4 \cdot \delta_3 \cdot x_1 = \\ &= 0.4996 + 0.4 \cdot (-0.0122) \cdot 0.2 = 0.4986 \end{aligned}$$

$$\delta_5 = -0.1184$$

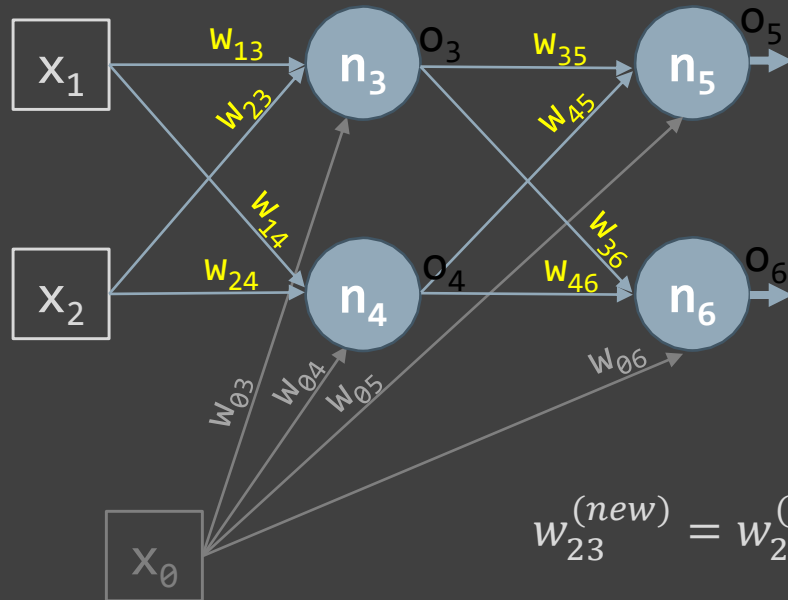
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819	.4986				.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{23}^{(new)} &= w_{23}^{(old)} + 0.4 \cdot \delta_3 \cdot x_2 = \\ &= 0.4992 + 0.4 \cdot (-0.0122) \cdot 0.45 = 0.497 \end{aligned}$$

$$\delta_5 = -0.1184$$

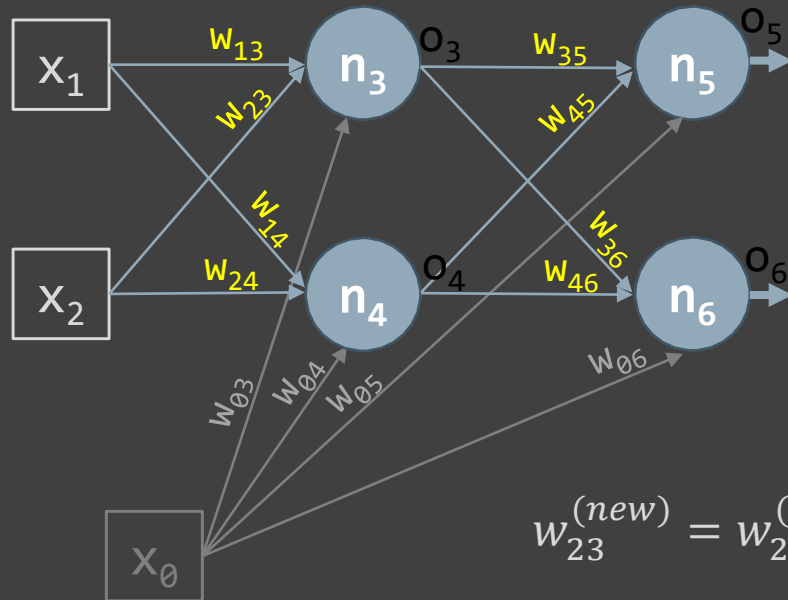
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819	.4986				.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{23}^{(new)} &= w_{23}^{(old)} + 0.4 \cdot \delta_3 \cdot x_2 = \\ &= 0.4992 + 0.4 \cdot (-0.0122) \cdot 0.45 = 0.497 \end{aligned}$$

$$\delta_5 = -0.1184$$

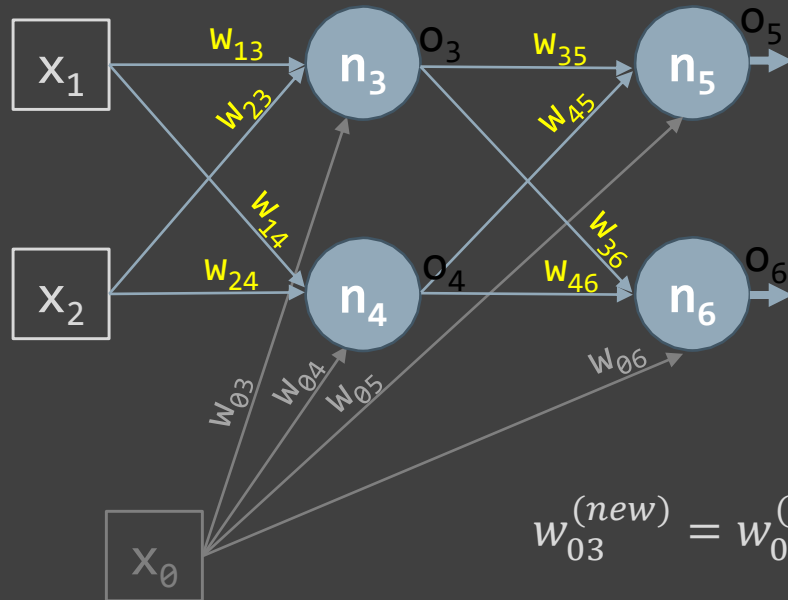
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819	.4986	0.497			.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{03}^{(new)} &= w_{03}^{(old)} + 0.4 \cdot \delta_3 \cdot x_0 = \\ &= 0.4918 + 0.4 \cdot (-0.0122) \cdot 1 = 0.4869 \end{aligned}$$

$$\delta_5 = -0.1184$$

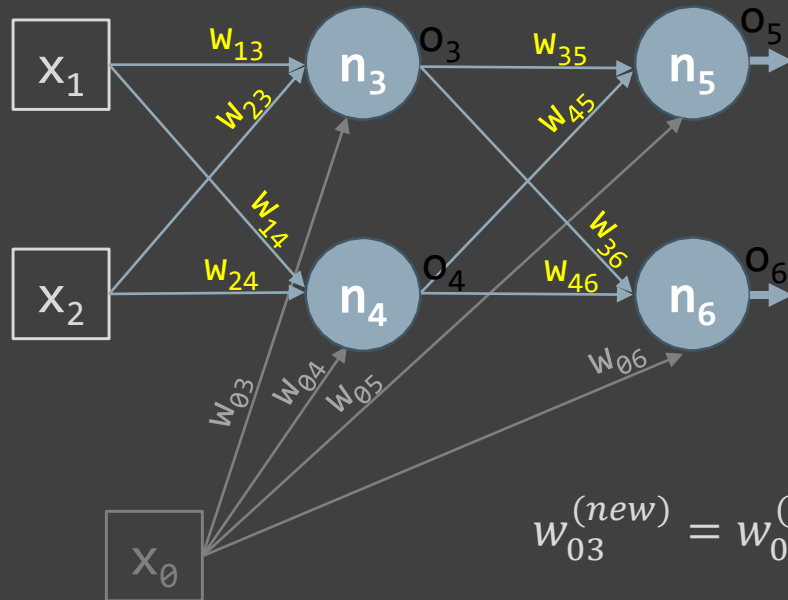
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53			0.398	.4819	.4986	0.497			.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{03}^{(new)} &= w_{03}^{(old)} + 0.4 \cdot \delta_3 \cdot x_0 = \\ &= 0.4918 + 0.4 \cdot (-0.0122) \cdot 1 = 0.4869 \end{aligned}$$

$$\delta_5 = -0.1184$$

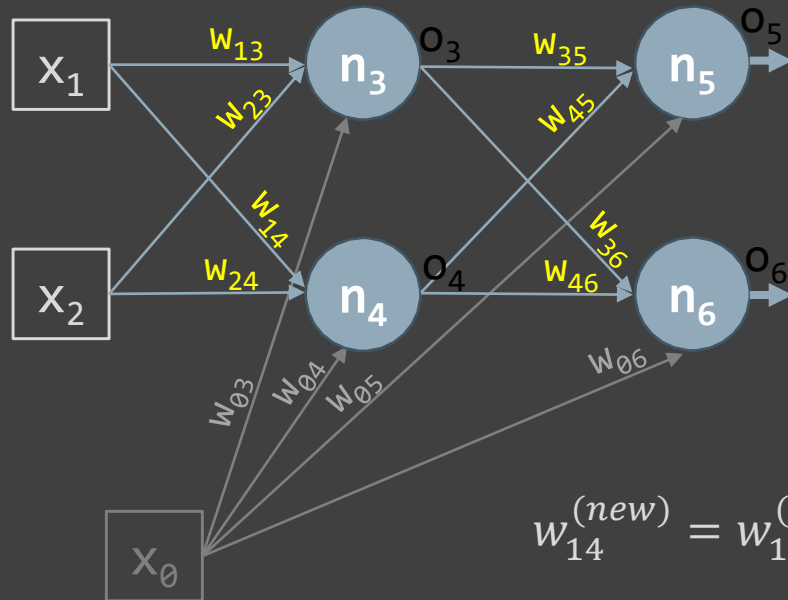
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497			.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{14}^{(new)} &= w_{14}^{(old)} + 0.4 \cdot \delta_4 \cdot x_1 = \\ &= 0.4996 + 0.4 \cdot (-0.0122) \cdot 0.2 = 0.4986 \end{aligned}$$

$$\delta_5 = -0.1184$$

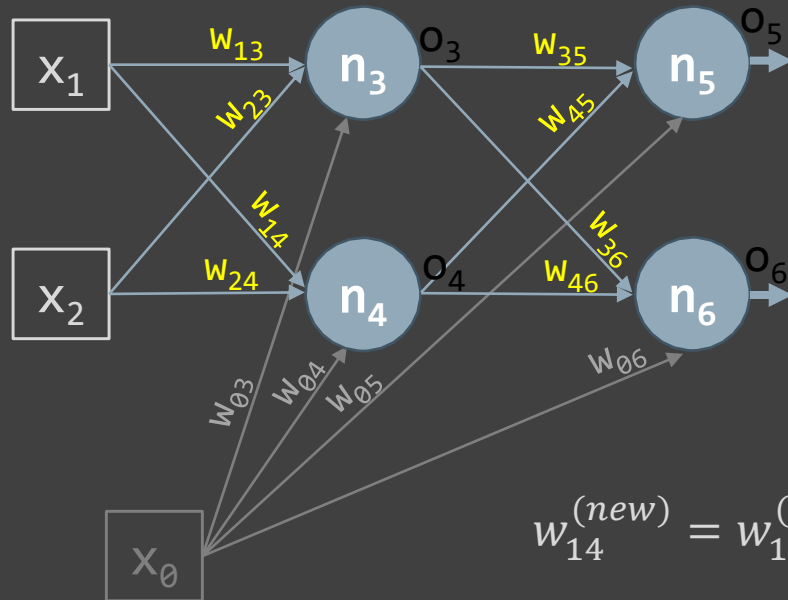
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497			.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{14}^{(new)} &= w_{14}^{(old)} + 0.4 \cdot \delta_4 \cdot x_1 = \\ &= 0.4996 + 0.4 \cdot (-0.0122) \cdot 0.2 = 0.4986 \end{aligned}$$

$$\delta_5 = -0.1184$$

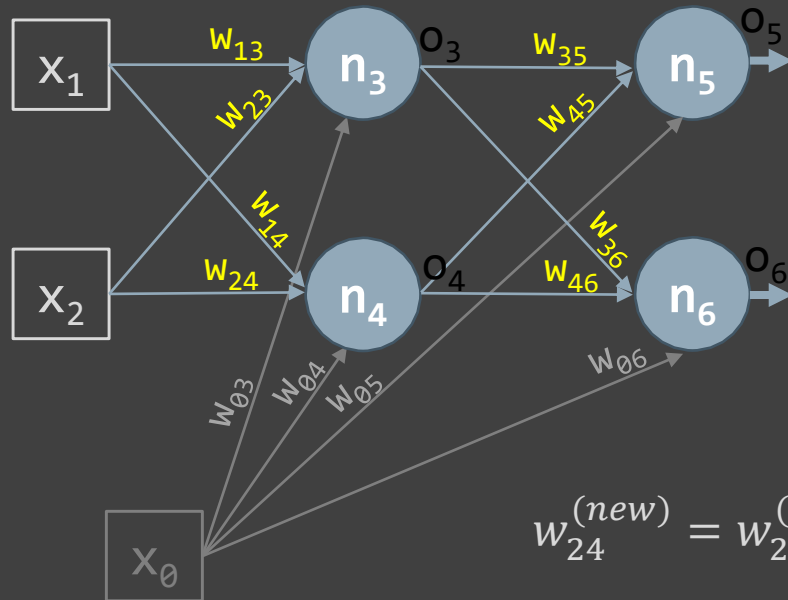
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497	.4986		.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{24}^{(new)} &= w_{24}^{(old)} + 0.4 \cdot \delta_4 \cdot x_2 = \\ &= 0.4992 + 0.4 \cdot (-0.0122) \cdot 0.45 = 0.497 \end{aligned}$$

$$\delta_5 = -0.1184$$

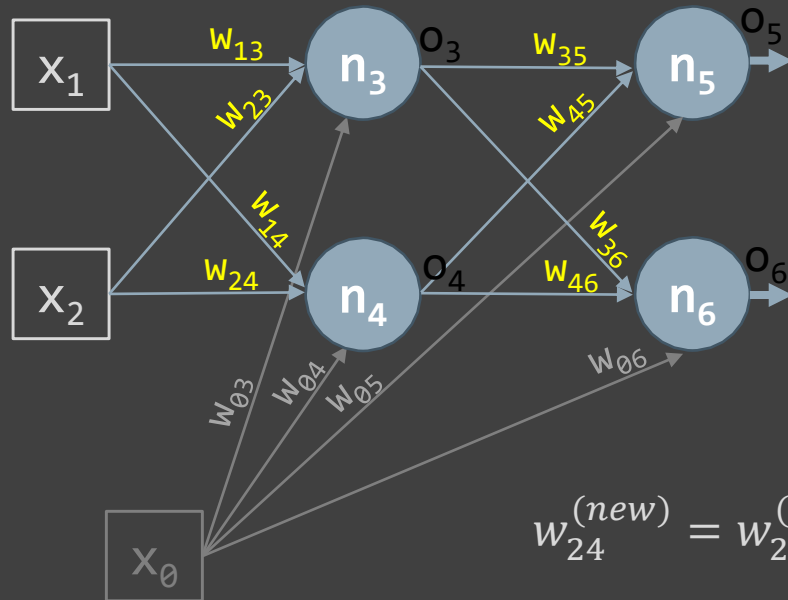
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497	.4986		.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{24}^{(new)} &= w_{24}^{(old)} + 0.4 \cdot \delta_4 \cdot x_2 = \\ &= 0.4992 + 0.4 \cdot (-0.0122) \cdot 0.45 = 0.497 \end{aligned}$$

$$\delta_5 = -0.1184$$

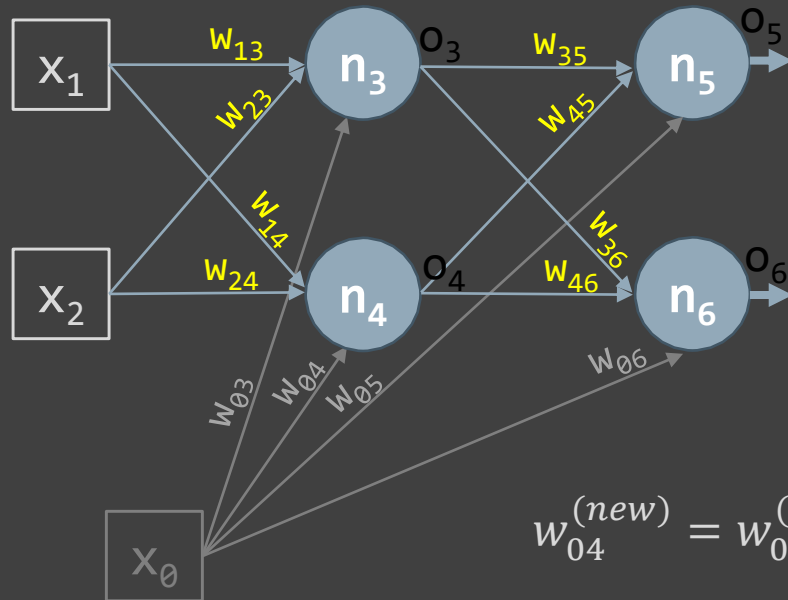
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497	.4986	0.497	.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{04}^{(new)} &= w_{04}^{(old)} + 0.4 \cdot \delta_4 \cdot x_0 = \\ &= 0.4918 + 0.4 \cdot (-0.0122) \cdot 1 = 0.4869 \end{aligned}$$

$$\delta_5 = -0.1184$$

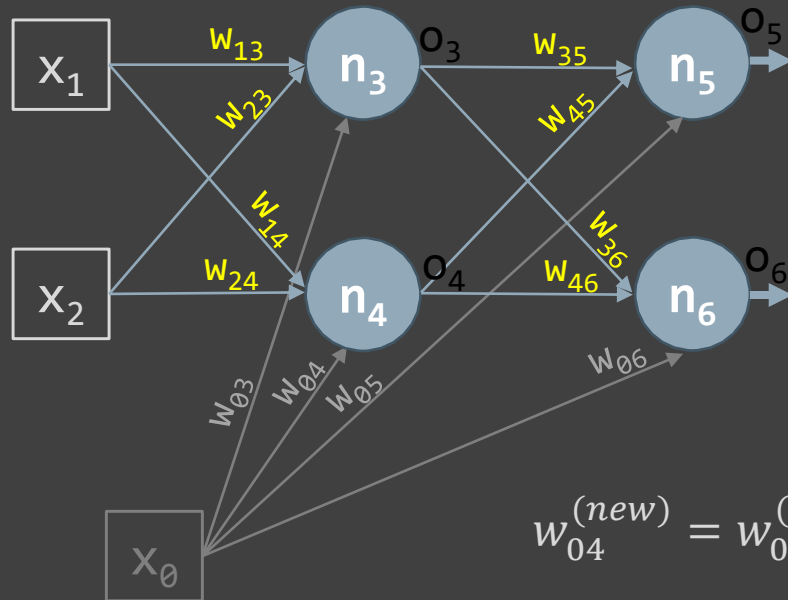
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869		0.398	.4819	.4986	0.497	.4986	0.497	.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Back-propagation



Για τους νευρώνες εξόδου:

$$\delta_k = (t_k - o_k) \cdot o_k \cdot (1 - o_k), \quad w_{jk}^{(new)} = w_{jk}^{(old)} + a \cdot \delta_k \cdot o_j$$

Για τους κρυφούς νευρώνες:

$$\delta_j = o_j \cdot (1 - o_j) \cdot \sum_{j,k} w_{jk} \cdot \delta_k, \quad w_{ij}^{(new)} = w_{ij}^{(old)} + a \cdot \delta_j \cdot x_i$$

Ανανεώνουμε τα βάρη

$$\begin{aligned} w_{04}^{(new)} &= w_{04}^{(old)} + 0.4 \cdot \delta_4 \cdot x_0 = \\ &= 0.4918 + 0.4 \cdot (-0.0122) \cdot 1 = 0.4869 \end{aligned}$$

$$\delta_5 = -0.1184$$

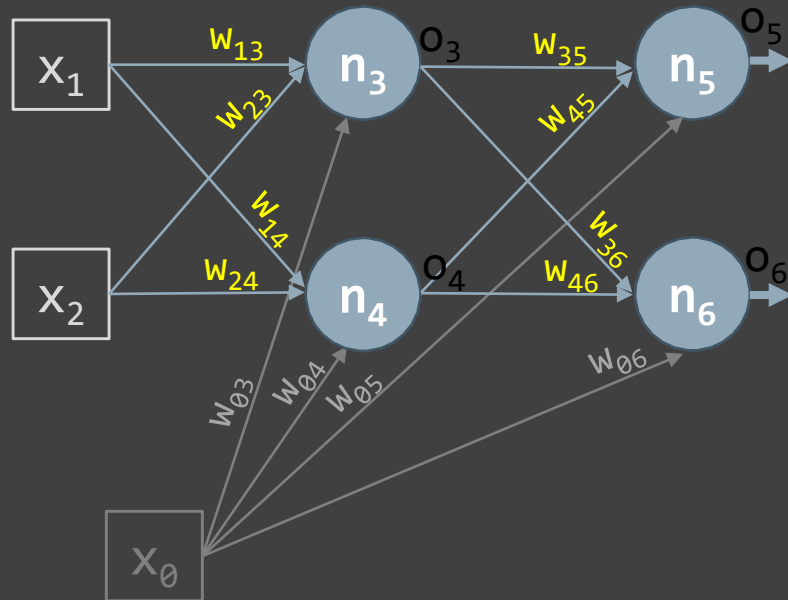
$$\delta_6 = -0.0036$$

$$\delta_3 = -0.0122$$

$$\delta_4 = -0.0122$$

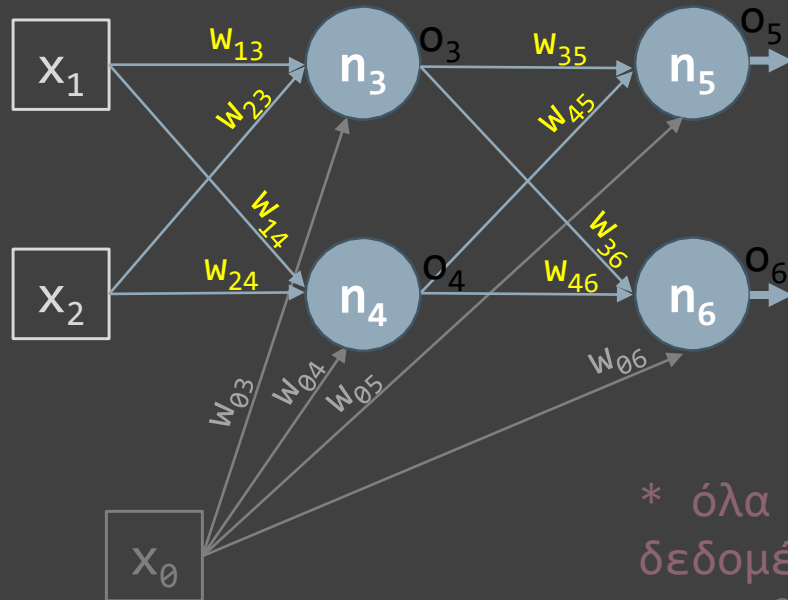
x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869	.4869	0.398	.4819	.4986	0.497	.4986	0.497	.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα – Forward pass



x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869	.4869	0.398	.4819	.4986	0.497	.4986	0.497	.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Παράδειγμα



Τα **epochs*** συνεχίζονται μέχρι να ικανοποιηθεί κάποιο κριτήριο τερματισμού

π.χ. η μεταβολή των βαρών να είναι πάρα πολύ μικρή ή το σφάλμα να πέσει κάτω από ένα κατώφλι

* όλα τα παραδείγματα εκπαίδευσης του πίνακα δεδομένων έχουν “περάσει” μια φορά (forward pass & back-propagation) μέσα στο ΤΝΔ

x_0	x_1	x_2	t_5	t_6	w_{03}	w_{04}	w_{05}	w_{06}	w_{13}	w_{23}	w_{14}	w_{24}	w_{35}	w_{45}	w_{36}	w_{46}	o_3	o_4	o_5	o_6
1	0.05	0.10	0.01	0.99	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.64	0.64	0.76	0.76
1	0.20	0.45	0.12	0.78	.4918	.4918	.4453	.4832	.4996	.4992	.4996	.4992	0.465	0.465	0.489	0.489	0.69	0.69	0.748	0.76
1	0.54	0.22	0.46	0.53	.4869	.4869	0.398	.4819	.4986	0.497	.4986	0.497	.4323	.4323	0.488	0.488				
1	0.68	0.79	0.24	0.08																

Εκπαίδευση

- Το ΤΝΔ **τροφοδοτείται** τόσο με διανύσματα εισόδου όσο και με διανύσματα εξόδου
- Το δίκτυο υπολογίζει ένα προσωρινό αποτέλεσμα εξόδου, το **συγκρίνει** με το επιθυμητό και **υπολογίζει** μέσω μιας συνάρτησης το **σφάλμα/απόκλιση**
- Το λάθος “στέλνει σήμα” στους νευρώνες εξόδου και αυτοί με τη σειρά τους στους υπόλοιπους για **τροποποίηση των βαρών** ώστε να ελαττωθεί το σφάλμα
- Αν το σφάλμα δεν ελαττωθεί κάτω από ένα συγκεκριμένο **κατώφλι** τότε τροποποιούνται εκ νέου τα βάρη

Παράμετροι εκπαίδευσης

- Αριθμός κρυφών επιπέδων
- Αριθμός νευρώνων ανά επίπεδο
Εμπειρικός κανόνας για προβλήματα κατηγοριοποίησης: αριθμός νευρώνων στα κρυφά επίπεδα < αριθμό παραδειγμάτων εκπαίδευσης
- Αριθμός κύκλων εκπαίδευσης (epochs)
- Μέγεθος συνόλου εκπαίδευσης (training set)
- Μέγεθος συνόλου δοκιμής (test set)
- Ρυθμός εκπαίδευσης

Στην πράξη οι παραπάνω τιμές δίνονται αρχικά τυχαία ή εμπειρικά και ανάλογα με τη φύση του προς επίλυση προβλήματος.

Παρακολουθώντας την εξέλιξη του δικτύου γίνονται αναπροσαρμογές όπου χρειάζεται, π.χ. αν το δίκτυο καθυστερεί πολύ να τερματίσει αλλάζουμε τον ρυθμό εκπαίδευσης

Εκπαίδευση & Αξιολόγηση

- Πρόβλεψη πιθανότητας βροχόπτωσης
- Φίλτρο σε τηλεπικοινωνιακές γραμμές που “κόβει” θόρυβο/παράσιτα
- Συσκευή αναγνώρισης αντικειμένων που περιέχουν εκρηκτικά σε αεροδρόμια
- Αναγνώριση εικόνων, κειμένων και γενικά προτύπων (pattern recognition)
- Υπολογισμός παράγοντα ρίσκου σε μια αίτηση για δάνειο

Εκπαίδευση & Αξιολόγηση

- Υπολογισμός παράγοντα ρίσκου σε μια αίτηση για δάνειο

X01	Κυκλοφορούν ενεργητικό / Βραχυπρόθεσμες υποχρεώσεις (Γενική ρευστότητα)
X02	(Διαθέσιμα + Βραχυπρόθεσμες επενδύσεις + Καθαρές αποδοχές) / Βραχυπρόθεσμες Υποχρεώσεις
X03	(Έσοδα συνεχούς εκμετάλλευσης + Αποσβέσεις + Πληρωμές) / Βραχυπρόθεσμες Υποχρεώσεις + Μακροπρόθεσμες Υποχρεώσεις
X04	(Βραχυπρόθεσμες Υποχρεώσεις + Μακροπρόθεσμες Υποχρεώσεις) / Σύνολο Ενεργητικού
X05	(Κυκλοφορούν Ενεργητικό – Βραχυπρόθεσμες υποχρεώσεις) / Σύνολο Ενεργητικού
X06	Έσοδα συνεχούς εκμετάλλευσης / Σύνολο Ενεργητικού
X07	(Έσοδα συνεχούς εκμετάλλευσης + Φόροι εισοδήματος + Τόκοι) / Σύνολο Ενεργητικού
X08	Καθαρές Πωλήσεις / Σύνολο Ενεργητικού
X09	Κέρδη προς διάθεση (ΚΚΜΦ) / Σύνολο Ενεργητικού
X10	Κυκλοφορούν Ενεργητικό / Καθαρές Πωλήσεις
X11	(Κυκλοφορούν Ενεργητικό – Βραχυπρόθεσμες Υποχρεώσεις) / Καθαρές Πωλήσεις
X12	Κυκλοφορούν Ενεργητικό / Σύνολο Ενεργητικού
X13	(Διαθέσιμα + Βραχυπρόθεσμες επενδύσεις) / Σύνολο Ενεργητικού

Εκπαίδευση & Αξιολόγηση

- Υπολογισμός παράγοντα ρίσκου σε μια αίτηση για δάνειο

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

Εκπαίδευση & Αξιολόγηση

- Υπολογισμός παράγοντα ρίσκου σε μια αίτηση για δάνειο

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

κάθε γραμμή : ένα παράδειγμα εκπαίδευσης

Εκπαίδευση

Δίνουμε αρχικές τιμές στα βάρη (τυχαίες) και στο ρυθμό μάθησης

Επαναληπτικά:

- εισάγουμε ένα παράδειγμα (π.χ. τιμές μιας γραμμής του προηγούμενου πίνακα για τις στήλες X01-X13)
- εκτελούμε ένα forward pass, υπολογίζοντας τις εξόδους των νευρώνων
- βρίσκουμε την απόκλιση της εξόδου που βρήκαμε από την επιθυμητή (π.χ. τιμή της στήλης “Ρίσκο” στον προηγούμενο πίνακα)
- εκτελούμε back-propagation, αναπροσαρμόζοντας τα βάρη ανάλογα με το σφάλμα

Μέχρι να ικανοποιηθεί ένα κριτήριο τερματισμού

Το ΤΝΔ έχει εκπαιδευτεί και το **μοντέλο** που έχει παραχθεί από την εκπαίδευση είναι οι **τελικές τιμές των βαρών**

Εκπαίδευση & Αξιολόγηση

Πόσο καλά εκπαιδεύτηκε το ΤΝΔ;

Αν εισάγω ένα νέο παράδειγμα, που δεν ανήκει σε αυτά με τα οποία εκπαιδεύτηκε, θα κάνει σωστή πρόβλεψη;

Εκπαίδευση & Αξιολόγηση

Πόσο καλά εκπαιδεύτηκε το ΤΝΔ;

Αν εισάγω ένα νέο παράδειγμα, που δεν ανήκει σε αυτά με τα οποία εκπαιδεύτηκε, θα κάνει σωστή πρόβλεψη;

ΑΝΑΓΚΗ ΓΙΑ ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΜΟΝΤΕΛΟΥ που έχει προκύψει από την εκπαίδευση

Hold-Out

1 Subject	2 Session	3 Trial	4 delta	5 theta	6 alpha	7 alpha1_8_10Hz	8 alpha2_10_13Hz	9 beta	10 beta1_13_15Hz	11 beta2_15_18Hz
1	1	1	9.2405	87.863	120.64	47.242	73.396	78.639	16.217	26.284
1	1	2	4.4156	77.265	100.72	28.315	72.404	65.048	14.608	13.234
1	1	3	10.516	127.28	107.26	74.041	33.216	80.559	22.065	22.884
1	1	4	8.6905	109.11	78.061	37.607	40.454	65.683	20.999	14.118
1	1	5	5.2061	85.999	73.512	36.317	37.195	47.936	12.243	9.3750
1	1	6	6.9557	98.582	70.616	37.517	33.100	53.771	10.428	12.539
1	1	7	3.7817	100.33	152.09	63.402	88.689	59.638	20.998	13.357
1	1	8	4.4857	78.957	78.146	33.295	44.851	51.265	12.579	14.243
1	1	9	4.0515	94.767	95.752	38.281	57.472	50.503	13.815	14.956
1	1	10	6.1090	118.15	106.27	54.117	52.157	57.645	14.473	12.001
1	1	11	9.5728	85.211	80.422	32.514	47.908	43.608	11.989	11.628
1	1	12	5.5940	111.16	98.540	47.729	50.811	56.750	18.950	12.970
1	1	13	35.494	175.16	117.45	53.993	63.461	64.876	15.839	19.311
1	1	14	1.6630	80.411	130.22	60.91	63.858	64.009	24.377	14.328
1	1	15	9.4938	116.08	195.73	105.27	86.457	64.370	19.568	18.995
1	1	16	4.0153	88.418	103.32	59.556	43.761	45.068	11.871	10.195
1	1	17	4.5975	212.95	202.94	110.83	92.110	58.284	19.797	13.975
1	1	18	4.0817	99.701	159.86	71.728	88.128	56.615	24.216	13.173
1	1	19	7.9554	144.87	168.38	72.107	96.273	48.664	16.111	10.284
1	1	20	12.903	177.77	129.81	66.273	63.539	65.953	22.897	18.240
1	1	21	5.8212	100.86	137.72	60.331	77.385	60.947	20.803	11.543
1	1	22	9.7441	87.479	94.710	36.373	58.337	54.280	18.727	13.868
1	1	23	15.470	131.28	208.72	132.81	75.912	54.350	10.780	15.563
1	1	24	2.2123	60.108	80.410	36.168	44.242	51.106	17.397	9.8900
1	1	25	6.3903	151.45	208.83	102.97	105.85	68.573	22.383	16.458
1	1	26	11.530	134.04	116.44	58.737	57.702	86.027	46.500	16.158
1	1	27	7.2757	110.95	86.631	35.004	51.627	47.114	16.119	9.2700
1	1	28	8.0154	106.11	105.77	38.270	67.501	64.864	24.998	15.169
1	1	29	12.156	99.962	87.327	33.747	53.580	54.523	18.955	13.774
1	1	30	12.034	110.83	103.68	41.985	61.698	55.238	17.943	14.350
1	2	1	625.68	31.847	37.503	25.930	15.727	14.326	4.5460	3.1413
1	2	2	443.52	20.686	18.156	7.9930	10.163	14.601	3.7480	4.4535
1	2	3	360.58	36.204	56.661	34.293	22.368	16.458	4.6946	5.7907
1	2	4	245.69	28.087	31.291	14.803	16.488	12.958	3.6699	4.0717

Training set

Test set



Train the
model on the
training set



Evaluate its
accuracy on the
test set

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

ΑΛΛΑ:

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

ΑΛΛΑ:

Προσοχή στην κατανομή της κλάσης!

Αντιπροσωπευτικό δείγμα παραδειγμάτων σε κάθε set

π.χ.

100 παραδείγματα: 60 -> κλάση 1, 40 -> κλάση 2

training set: 70%

test set: 30%

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

AAA:

Προσοχή στην κατανομή της κλάσης!

Αντιπροσωπευτικό δείγμα παραδειγμάτων σε κάθε set

π.χ.

100 παραδείγματα: 60 -> κλάση 1, 40 -> κλάση 2

```
training set: 70%    -> 70 παραδείγματα
                        60 από κλάση 1
                        10 από κλάση 2
```

```
test set: 30%      -> 30 παραδείγματα
                    όλα από κλάση 2
```

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

AAA:

Προσοχή στην κατανομή της κλάσης!

Αντιπροσωπευτικό δείγμα παραδειγμάτων σε κάθε set

π.χ.

100 παραδείγματα: 60 -> κλάση 1, 40 -> κλάση 2

training set: 70% -> 70 παραδείγματα

60 από κλάση 1

10 από κλάση 2

```
test set: 30%      -> 30 παραδείγματα
                     όλα από κλάση 2
```

όλα από κλάση 2

Μπορώ να βρω καλύτερο τρόπο?

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

ΑΛΛΑ:

Προσοχή στην κατανομή της κλάσης!

Αντιπροσωπευτικό δείγμα παραδειγμάτων σε κάθε set

π.χ.

100 παραδείγματα: 60 -> κλάση 1, 40 -> κλάση 2 —→

κατανομή
κλάσης:
60% κλάση 1
40% κλάση 2

training set: 70% -> 70 παραδείγματα

test set: 30% -> 30 παραδείγματα

Hold-Out

Η επιλογή των παραδειγμάτων που θα αποτελέσουν το training set και το test set είναι τυχαία

ΑΛΛΑ:

Προσοχή στην κατανομή της κλάσης!

Αντιπροσωπευτικό δείγμα παραδειγμάτων σε κάθε set

π.χ.

100 παραδείγματα: 60 -> κλάση 1, 40 -> κλάση 2

κατανομή
κλάσης:
60% κλάση 1
40% κλάση 2

training set: 70% -> 70 παραδείγματα

42 από κλάση 1 (60% x 70)

28 από κλάση 2 (40% x 70)

test set: 30% -> 30 παραδείγματα

18 από κλάση 1 (60% x 30)

12 από κλάση 2 (40% x 30)

Hold-Out

μεταβλητές (variables)

μεταβλητή κλάσης
(class variable)

παραδείγματα

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

Hold-Out

μεταβλητές (variables)

μεταβλητή κλάσης
(class variable)

παραδείγματα

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

Training set

Test set

Hold-Out

μεταβλητές (variables)

μεταβλητή κλάσης
(class variable)

X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-------

Εταιρία 1

Εταιρία 2

Εταιρία 3

Εταιρία 4

Εταιρία 5

Εταιρία 6

Εταιρία 7

Εταιρία 8

Εταιρία 9

Εταιρία 10

Εταιρία 11

Εταιρία 12

Εταιρία 13

Εταιρία 14

Εταιρία 15

Εταιρία 16

training set:
γνωστό στο ΤΝΔ κατά
την εκπαίδευση

Training set

Test set

παραδείγματα

Hold-Out

μεταβλητές (variables)

μεταβλητή κλάσης
(class variable)

παραδείγματα

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

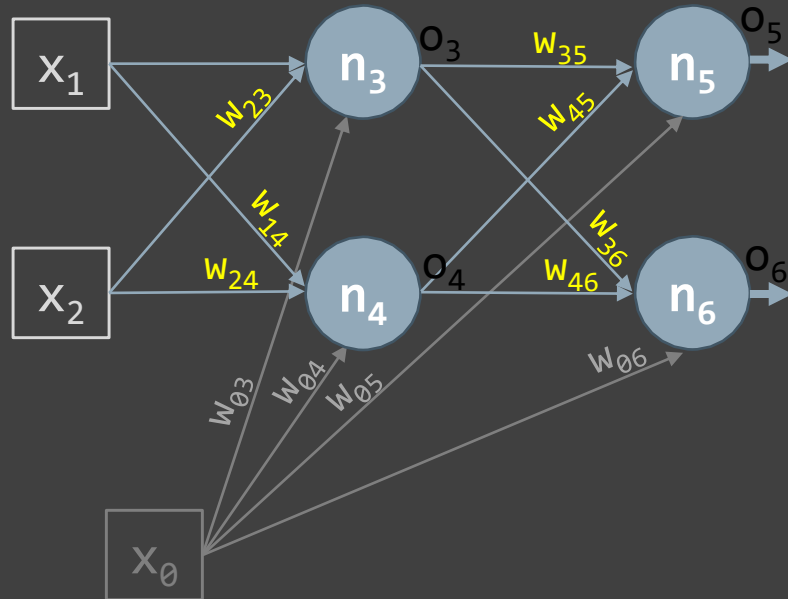
Training set

training set:
γνωστό στο ΤΝΔ κατά
την εκπαίδευση

Test set

test set:
κρυφό στο ΤΝΔ κατά
την εκπαίδευση

Αξιολόγηση



Αφού έχει εκπαιδευτεί με το training set $\rightarrow$ τιμές βαρών

ΑΞΙΟΛΟΓΗΣΗ

Με τις παραπάνω τιμές βαρών σταθερές, κάνω ένα πρόσθιο πέρασμα (forward pass) με καθένα από τα παραδείγματα του test set και βρίσκω μια έξοδο για κάθε παράδειγμα

Αξιολογώ τις εξόδους που βρήκα: απόκλιση από την πραγματική έξοδο (μεταβλητή κλάσης του test set)

Αξιολόγηση: Confusion Matrix

μας δίνει τη δυνατότητα να αξιολογήσουμε ένα μοντέλο κατηγοριοποίησης

περιέχει πληροφορία για την κατηγοριοποίηση που κάνει το μοντέλο σε κάποια κλάση και τις αντίστοιχες πραγματικές κλάσεις

Αξιολόγηση: Confusion Matrix

Έστω 2 κλάσεις: 0 και 1

Μετά το “πέρασμα” του **test set** από το ΤΝΔ, μετράμε:

	Actual class 1	Actual class 0
Predicted class 1	Αριθμός παραδειγμάτων που το ΤΝΔ τα κατέταξε στην κλάση 1 και όντως ανήκαν στην κλάση 1	Αριθμός παραδειγμάτων που το ΤΝΔ τα κατέταξε στην κλάση 1 ενώ ανήκαν στην κλάση 0
Predicted class 0	Αριθμός παραδειγμάτων που το ΤΝΔ τα κατέταξε στην κλάση 0 ενώ ανήκαν στην κλάση 1	Αριθμός παραδειγμάτων που το ΤΝΔ τα κατέταξε στην κλάση 0 και όντως ανήκαν στην κλάση 0

Αξιολόγηση: Confusion Matrix

Έστω 2 κλάσεις: 0 και 1

Μετά το “πέρασμα” του **test set** από το ΤΝΔ, μετράμε:

	Actual class 1	Actual class 0
Predicted class 1	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
Predicted class 0	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)

Αξιολόγηση: Confusion Matrix

Έστω 2 κλάσεις: 0 και 1

Μετά το “πέρασμα” του **test set** από το ΤΝΔ, μετράμε:

	Actual class 1	Actual class 0
Predicted class 1	TRUE POSITIVE (TP)	FALSE POSITIVE (FP) Type I Error
Predicted class 0	FALSE NEGATIVE (FN) Type II Error	TRUE NEGATIVE (TN)

Αξιολόγηση: Confusion Matrix

Type I Error: “false alarm” → π.χ. ενεργοποίηση ανιχνευτή καπνού ενώ δεν υπάρχει φωτιά

Type II Error: “miss” → π.χ. μη ενεργοποίηση ανιχνευτή καπνού ενώ υπάρχει φωτιά

	Actual class	
	1	0
Predicted class 1	TRUE POSITIVE (TP)	FALSE POSITIVE (FP) Type I Error
Predicted class 0	FALSE NEGATIVE (FN) Type II Error	TRUE NEGATIVE (TN)

Αξιολόγηση: Confusion Matrix

	Actual class 1	Actual class 0
Predicted class 1	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
Predicted class 0	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

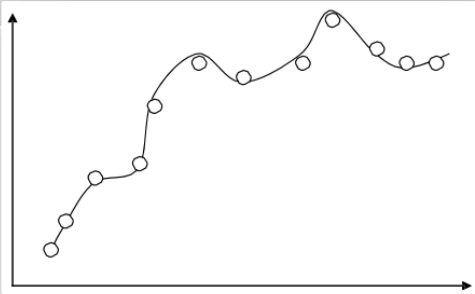
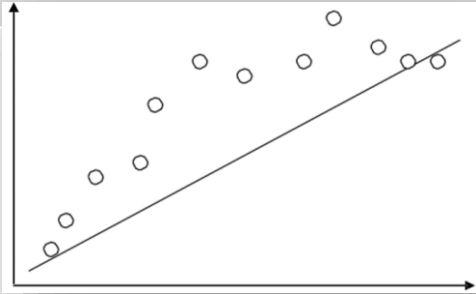
απόδοση εκπαίδευσης: πόσο καλά έχει εκπαιδευτεί το ΤΝΔ ή τι ποσοστό παραδειγμάτων του test set ταξινομεί σωστά

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο		
μέγεθος του training set		
μέγεθος του test set		
ρυθμός εκπαίδευσης		
Αριθμός κύκλων εκπαίδευσης (epochs)		

Chirdaris, D (2018). *Applications of Artificial Neural Networks in Bioinformatics Mathematical Modelling*. Retrieved from apothesis.eap.gr/handle/repo/40409

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο	υπερ-προσαρμογή (over-fitting)	υπο-προσαρμογή (under-fitting)
μέγεθος του training set		
μέγεθος του test set		
ρυθμός εκπαίδευσης		
Αριθμός κύκλων εκπαίδευσης (epochs)		

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο	υπερ-προσαρμογή (over-fitting)	υπο-προσαρμογή (under-fitting)
μέγεθος του training set	καλή εκπαίδευση και ικανότητα γενίκευσης	αδυναμία πλήρους κατανόησης του προβλήματος και μέτρια ικανότητα γενίκευσης
μέγεθος του test set		
ρυθμός εκπαίδευσης		
Αριθμός κύκλων εκπαίδευσης (epochs)		

Chirdaris, D (2018). *Applications of Artificial Neural Networks in Bioinformatics Mathematical Modelling*. Retrieved from apothesis.eap.gr/handle/repo/40409

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο	υπερ-προσαρμογή (over-fitting)	υπο-προσαρμογή (under-fitting)
μέγεθος του training set	καλή εκπαίδευση και ικανότητα γενίκευσης	αδυναμία πλήρους κατανόησης του προβλήματος και μέτρια ικανότητα γενίκευσης
μέγεθος του test set	αύξηση αξιοπιστίας αξιολόγησης	μείωση αξιοπιστίας αξιολόγησης
ρυθμός εκπαίδευσης		
Αριθμός κύκλων εκπαίδευσης (epochs)		

Chirdaris, D (2018). *Applications of Artificial Neural Networks in Bioinformatics Mathematical Modelling*. Retrieved from apothesis.eap.gr/handle/repo/40409

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο	υπερ-προσαρμογή (over-fitting)	υπο-προσαρμογή (under-fitting)
μέγεθος του training set	καλή εκπαίδευση και ικανότητα γενίκευσης	αδυναμία πλήρους κατανόησης του προβλήματος και μέτρια ικανότητα γενίκευσης
μέγεθος του test set	αύξηση αξιοπιστίας αξιολόγησης	μείωση αξιοπιστίας αξιολόγησης
ρυθμός εκπαίδευσης	γρήγορη εκπαίδευση του δικτύου, παλινδρόμηση	αργή εκπαίδευση του δικτύου, αποφυγή παλινδρόμησης
Αριθμός κύκλων εκπαίδευσης (epochs)		

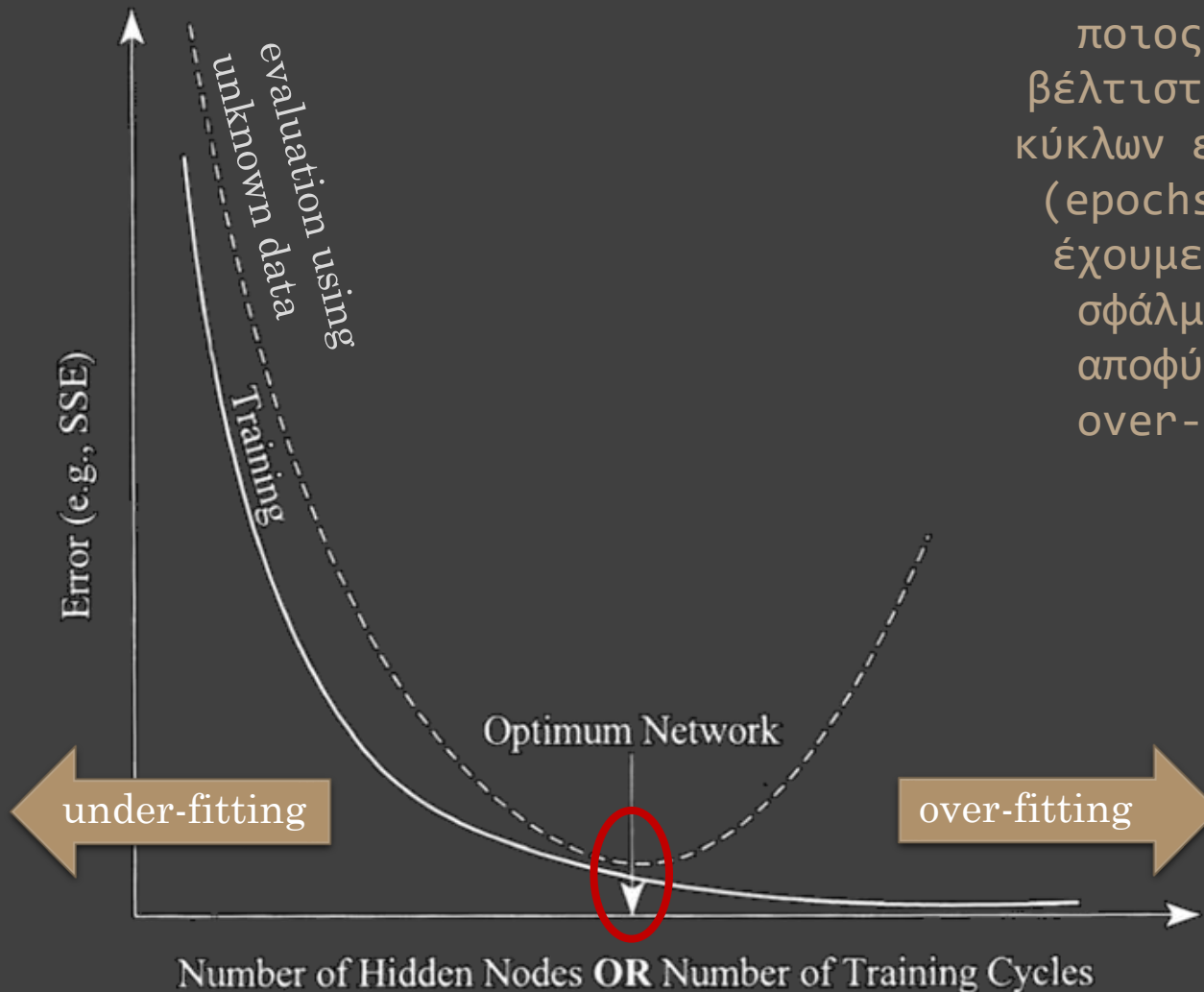
Chirdaris, D (2018). *Applications of Artificial Neural Networks in Bioinformatics Mathematical Modelling*. Retrieved from apothesis.eap.gr/handle/repo/40409

Παράμετροι εκπαίδευσης

ΠΑΡΑΜΕΤΡΟΣ	▲	▼
# κρυφών επιπέδων & # νευρώνων ανά επίπεδο	υπερ-προσαρμογή (over-fitting)	υπο-προσαρμογή (under-fitting)
μέγεθος του training set	καλή εκπαίδευση και ικανότητα γενίκευσης	αδυναμία πλήρους κατανόησης του προβλήματος και μέτρια ικανότητα γενίκευσης
μέγεθος του test set	αύξηση αξιοπιστίας αξιολόγησης	μείωση αξιοπιστίας αξιολόγησης
ρυθμός εκπαίδευσης	γρήγορη εκπαίδευση του δικτύου, παλινδρόμηση	αργή εκπαίδευση του δικτύου, αποφυγή παλινδρόμησης
Αριθμός κύκλων εκπαίδευσης (epochs)	κακή ικανότητα γενίκευσης	υπό-προσαρμογή (under-fitting)

Chirdaris, D (2018). *Applications of Artificial Neural Networks in Bioinformatics Mathematical Modelling*. Retrieved from apothesis.eap.gr/handle/repo/40409

Παράμετροι εκπαίδευσης



ποιος είναι ο βέλτιστος αριθμός κύκλων εκπαίδευσης (epochs) ώστε να έχουμε ελάχιστο σφάλμα και να αποφύγουμε το over-fitting;

Hold-Out

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

Training set

Test set

Hold-Out

	X01	X02	X03	X04	X05	X06	X07	X08	X09	X10	X11	X12	X13	Ρίσκο
Εταιρία 1														
Εταιρία 2														
Εταιρία 3														
Εταιρία 4														
Εταιρία 5														
Εταιρία 6														
Εταιρία 7														
Εταιρία 8														
Εταιρία 9														
Εταιρία 10														
Εταιρία 11														
Εταιρία 12														
Εταιρία 13														
Εταιρία 14														
Εταιρία 15														
Εταιρία 16														

Training set

Validation set

Test set

MATLAB: Neural Pattern Recognition tool

Αρχικά πρέπει να φορτωθούν τα δεδομένα στο Workspace του MATLAB και να γίνει όποια προ-επεξεργασία απαιτείται.

Τελικά θα πρέπει να υπάρχει μια μεταβλητή-πίνακας με τις στήλες των μεταβλητών του προβλήματος και μια μεταβλητή-διάνυσμα ή πίνακας με τη στήλη/στήλες της μεταβλητής κλάσης (ή των μεταβλητών κλάσης)

	var1	var2	var3	var4	var5	Έξοδος
Παρά-δειγμα 1						
Παρά-δειγμα 2						
Παρά-δειγμα 3						
Παρά-δειγμα 4						

	var1	var2	var3	var4	var5	Έξοδος 1	Έξοδος 2
Παρά-δειγμα 1							
Παρά-δειγμα 2							
Παρά-δειγμα 3							
Παρά-δειγμα 4							

MATLAB: Neural Pattern Recognition tool

Αρχικά πρέπει να φορτωθούν τα δεδομένα στο Workspace του MATLAB και να γίνει όποια προ-επεξεργασία απαιτείται.

Τελικά θα πρέπει να υπάρχει μια μεταβλητή-πίνακας με τις στήλες των μεταβλητών του προβλήματος και μια μεταβλητή-διάνυσμα ή πίνακας με τη στήλη/στήλες της μεταβλητής κλάσης (ή των μεταβλητών κλάσης)

	var1	var2	var3	var4	var5	Έξοδος
Παρά-δειγμα 1						
Παρά-δειγμα 2						
Παρά-δειγμα 3						
Παρά-δειγμα 4						

	var1	var2	var3	var4	var5	Έξοδος 1	Έξοδος 2
Παρά-δειγμα 1							
Παρά-δειγμα 2							
Παρά-δειγμα 3							
Παρά-δειγμα 4							

MATLAB: Neural Pattern Recognition tool

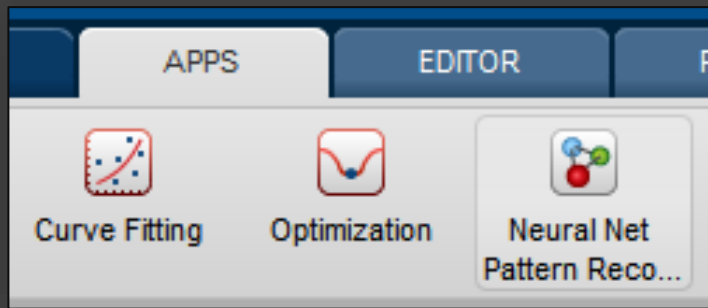
Αρχικά πρέπει να φορτωθούν τα δεδομένα στο Workspace του MATLAB και να γίνει όποια προ-επεξεργασία απαιτείται.

Τελικά θα πρέπει να υπάρχει μια μεταβλητή-πίνακας με τις στήλες των μεταβλητών του προβλήματος και μια μεταβλητή-διάνυσμα ή πίνακας με τη στήλη/στήλες της μεταβλητής κλάσης (ή των μεταβλητών κλάσης)

	var1	var2	var3	var4	var5	Έξοδος		var1	var2	var3	var4	var5	Έξοδος 1	Έξοδος 2
Παρά-δειγμα 1							Παρά-δειγμα 1							
Παρά-δειγμα 2							Παρά-δειγμα 2							
Παρά-δειγμα 3							Παρά-δειγμα 3							
Παρά-δειγμα 4							Παρά-δειγμα 4							

MATLAB: Neural Pattern Recognition tool

Έναρξη του Toolbox:



Menu ► Apps ► Neural Net
Pattern
Recognition

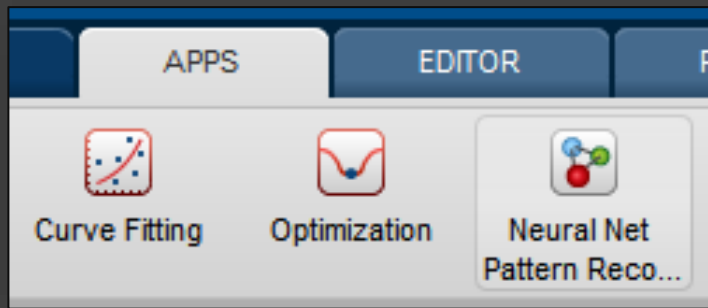
ή

στο Workspace:

```
>> nprtool
```

MATLAB: Neural Pattern Recognition tool

Έναρξη του Toolbox:



Menu ► Apps ► Neural Net
Pattern
Recognition

feed-forward ΤΝΔ με 1 κρυφό επίπεδο

MATLAB: Neural Pattern Recognition tool

 Neural Pattern Recognition (nprtool)



Welcome to the Neural Network Pattern Recognition app.

Solve a pattern-recognition problem with a two-layer feed-forward network.

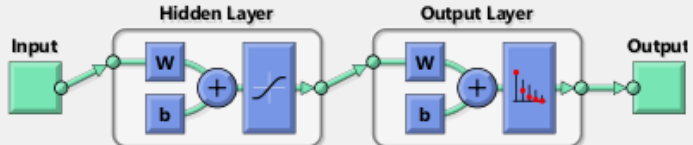
Introduction

In pattern recognition problems, you want a neural network to classify inputs into a set of target categories.

For example, recognize the vineyard that a particular bottle of wine came from, based on chemical analysis (`wine_dataset`); or classify a tumor as benign or malignant, based on uniformity of cell size, clump thickness, mitosis (`cancer_dataset`).

The Neural Pattern Recognition app will help you select data, create and train a network, and evaluate its performance using cross-entropy and confusion matrices.

Neural Network



A two-layer feed-forward network, with sigmoid hidden and softmax output neurons (`patternnet`), can classify vectors arbitrarily well, given enough neurons in its hidden layer.

The network will be trained with scaled conjugate gradient backpropagation (`trainscg`).

 To continue, click [Next].

 Neural Network Start

 Welcome

 Back

 Next

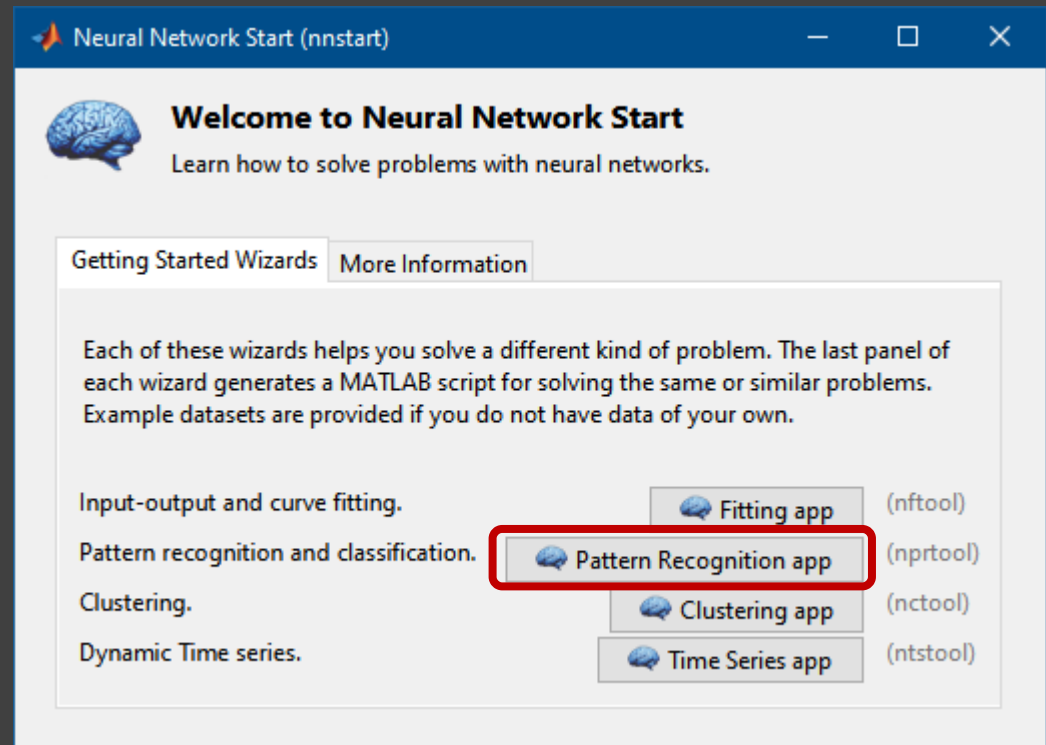
 Cancel

MATLAB: Neural Pattern Recognition tool

Εναλλακτικά:

στο Workspace:

```
>> nnstart
```



MATLAB: Neural Pattern Recognition tool

 Neural Pattern Recognition (nprtool)



Welcome to the Neural Network Pattern Recognition app.

Solve a pattern-recognition problem with a two-layer feed-forward network.

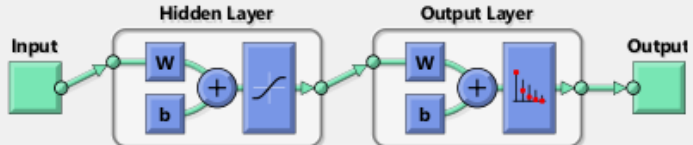
Introduction

In pattern recognition problems, you want a neural network to classify inputs into a set of target categories.

For example, recognize the vineyard that a particular bottle of wine came from, based on chemical analysis (`wine_dataset`); or classify a tumor as benign or malignant, based on uniformity of cell size, clump thickness, mitosis (`cancer_dataset`).

The Neural Pattern Recognition app will help you select data, create and train a network, and evaluate its performance using cross-entropy and confusion matrices.

Neural Network



A two-layer feed-forward network, with sigmoid hidden and softmax output neurons (`patternnet`), can classify vectors arbitrarily well, given enough neurons in its hidden layer.

The network will be trained with scaled conjugate gradient backpropagation (`trainscg`).

 To continue, click [Next].

 Neural Network Start

 Welcome

 Back

 Next

 Cancel

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

Select Data
What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: (none) ...

Target data defining desired network output.

Targets: (none) ...

Samples are: ☒ Matrix columns ☐ Matrix rows

Want to try out this tool with an example data set?

Load Example Data Set

Summary

No inputs selected.

No targets selected.

επιλογή πίνακα δεδομένων εισόδου (επιλογή από μεταβλητές που υπάρχουν στο Workspace ή φόρτωση αρχείου δεδομένων)

Select inputs and targets, then click [Next].

Neural Network Start Welcome Back Next Cancel

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

 **Select Data**
What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: (none) ...

Target data defining desired network output.

Targets: (none) ...

Samples are: ☒ Matrix columns ☐ Matrix rows

Summary

No inputs selected.

No targets selected.

Want to try out this tool with an example data set?

[Load Example Data Set](#)

 Select inputs and targets, then click [Next].

[Neural Network Start](#) [Welcome](#) [Back](#) [Next](#) [Cancel](#)

επιλογή πίνακα δεδομένων εισόδου (επιλογή από μεταβλητές που υπάρχουν στο Workspace ή φόρτωση αρχείου δεδομένων)

επιλογή διανύσματος/πίνακα δεδομένων εξόδου (επιλογή από μεταβλητές που υπάρχουν στο Workspace ή φόρτωση αρχείου δεδομένων)

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

 **Select Data**
What inputs and targets define your pattern recognition problem?

Get Data from Workspace
Input data to present to the network.

Inputs: (none) ...

Target data defining desired network output.

Targets: (none) ...

Samples are: ☒ Matrix columns ☐ Matrix rows

Summary
No inputs selected.
No targets selected.

Want to try out this tool with an example data set?
[Load Example Data Set](#)

! Select inputs and targets, then click [Next].

Neural Network Start Welcome Back Next Cancel

επιλογή πίνακα δεδομένων εισόδου (επιλογή από μεταβλητές που υπάρχουν στο Workspace ή φόρτωση αρχείου δεδομένων)

επιλογή διανύσματος/πίνακα δεδομένων εξόδου (επιλογή από μεταβλητές που υπάρχουν στο Workspace ή φόρτωση αρχείου δεδομένων)

επιλογή για το αν τα παραδείγματα είναι οργανωμένα σε γραμμές ή σε στήλες

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

Select Data

What inputs and targets define your pattern recognition problem?

Get Data from Workspace

Input data to present to the network.

Inputs: data ...

Target data defining desired network output.

Targets: class ...

Samples are: ☐ Matrix columns ☒ Matrix rows

Want to try out this tool with an example data set?

[Load Example Data Set](#)

Summary

Inputs 'data' is a 699x9 matrix, representing static data: 699 samples of 9 elements.

Targets 'class' is a 699x2 matrix, representing static data: 699 samples of 2 elements.

To continue, click [Next].

[Neural Network Start](#) [Welcome](#) [Back](#) [Next](#) [Cancel](#)

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages

Randomly divide up the 699 samples:

Training:	70%	489 samples
Validation:	15% ▾	105 samples
Testing:	15% ▾	105 samples

Restore Defaults

Explanation

Three Kinds of Samples:

Training:
These are presented to the network during training, and the network is adjusted according to its error.

Validation:
These are used to measure network generalization, and to halt training when generalization stops improving.

Testing:
These have no effect on training and so provide an independent measure of network performance during and after training.

➡ Change percentages if desired, then click [Next] to continue.

Neural Network Start | Welcome | Back | Next | Cancel

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

Network Architecture

Set the number of neurons in the pattern recognition network's hidden layer.

Hidden Layer

Define a pattern recognition neural network. (patternnet)

Number of Hidden Neurons:

Restore Defaults

Recommendation

Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

The diagram illustrates a neural network architecture. It starts with an **Input** layer with 9 neurons, represented by a green box with the number 9. This connects to a **Hidden Layer** with 10 neurons, represented by a blue box with the number 10. The hidden layer contains a sub-diagram showing weights (W) and biases (b) being added (+) and then passed through an activation function. The hidden layer connects to an **Output Layer** with 2 neurons, represented by a blue box with the number 2. The output layer also shows a sub-diagram with weights (W), biases (b), and an addition (+) operation, followed by a linear plot. Finally, the output layer connects to an **Output** layer with 2 neurons, represented by a green box with the number 2.

νευρώνων εισόδου = # αριθμός μεταβλητών προβλήματος

νευρώνων εξόδου = # αριθμός μεταβλητών κλάσης προβλήματος

Change settings if desired, then click [Next] to continue.

Neural Network Start Welcome Back Next Cancel

MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)

Train Network

Train the network to classify the inputs according to the targets.

Train Network

Train using scaled conjugate gradient backpropagation. (trainscg)

Train

Training automatically stops when generalization stops improving, as indicated by an increase in the cross-entropy error of the validation samples.

Results

	Samples	CE	%E
Training:	16	-	-
Validation:	4	-	-
Testing:	4	-	-

Plot Confusion Plot ROC

Notes

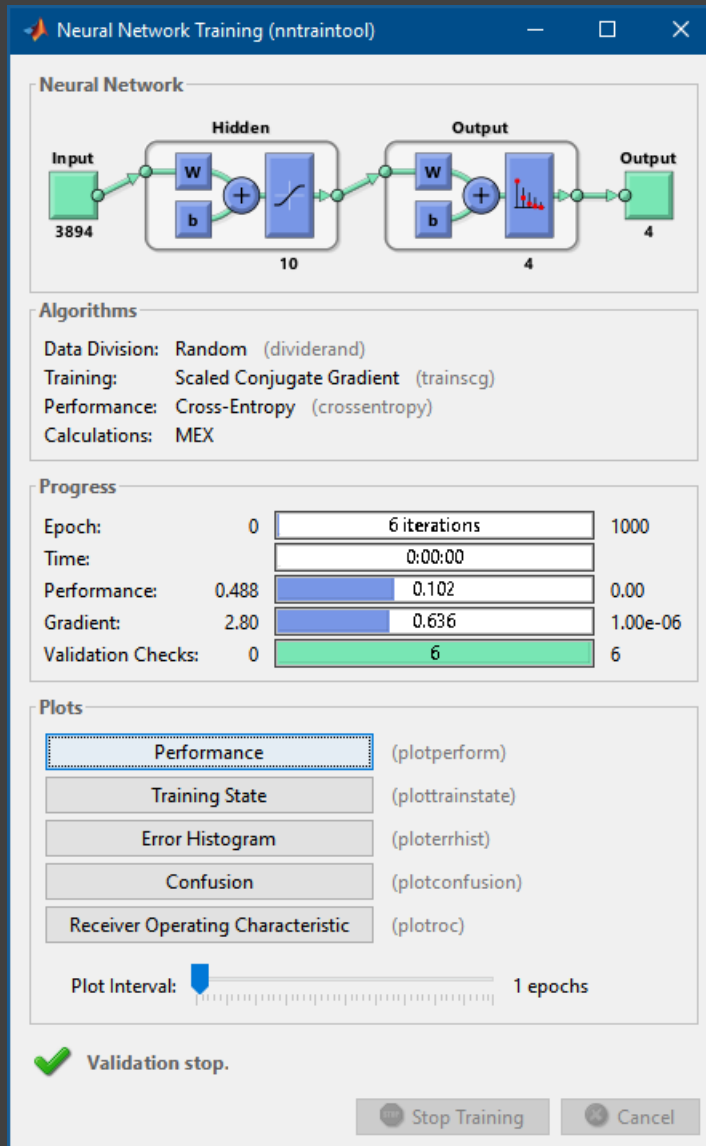
- Training multiple times will generate different results due to different initial conditions and sampling.
- Minimizing Cross-Entropy results in good classification. Lower values are better. Zero means no error.
- Percent Error indicates the fraction of samples which are misclassified. A value of 0 means no misclassifications, 100 indicates maximum misclassifications.

Train network, then click [Next].

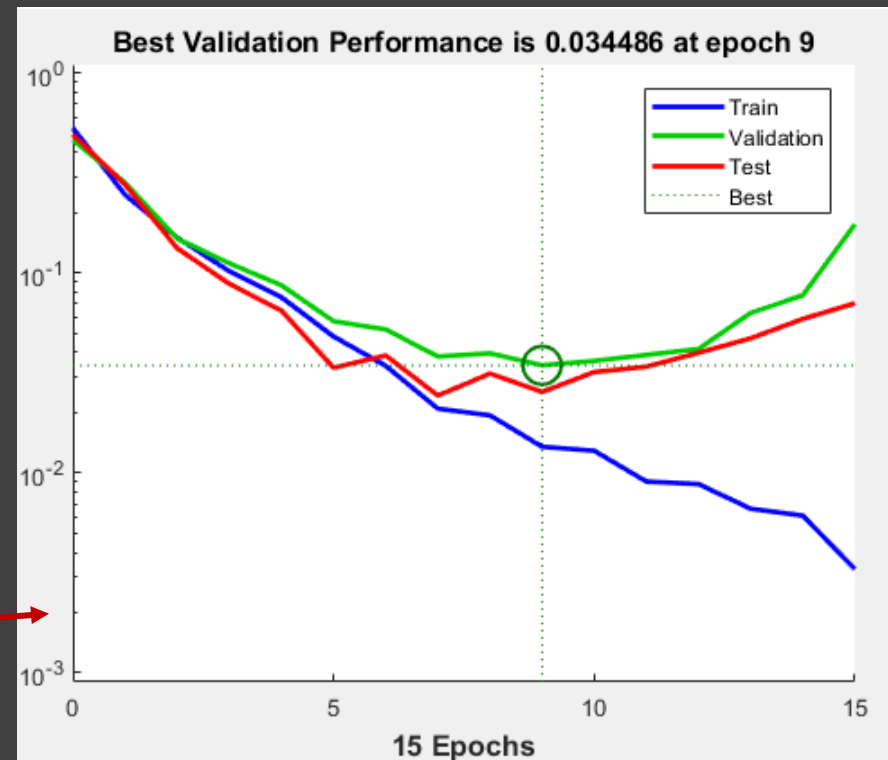
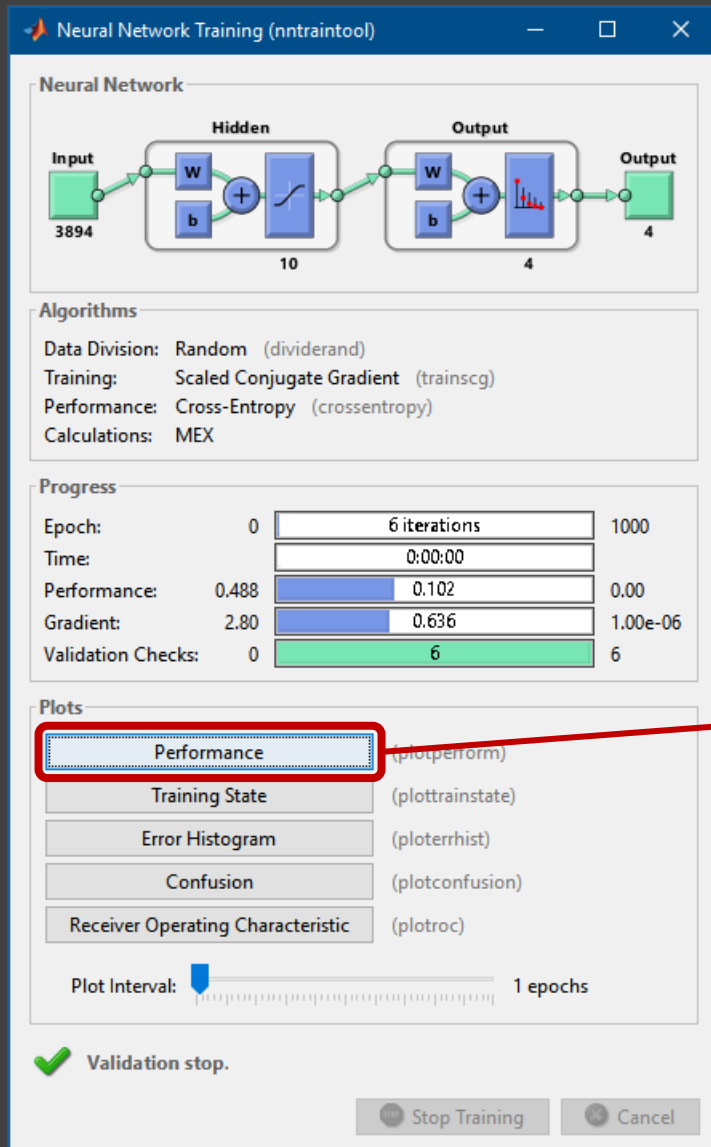
Neural Network Start Welcome Back Next Cancel

εκπαίδευση του νευρωνικού δικτύου

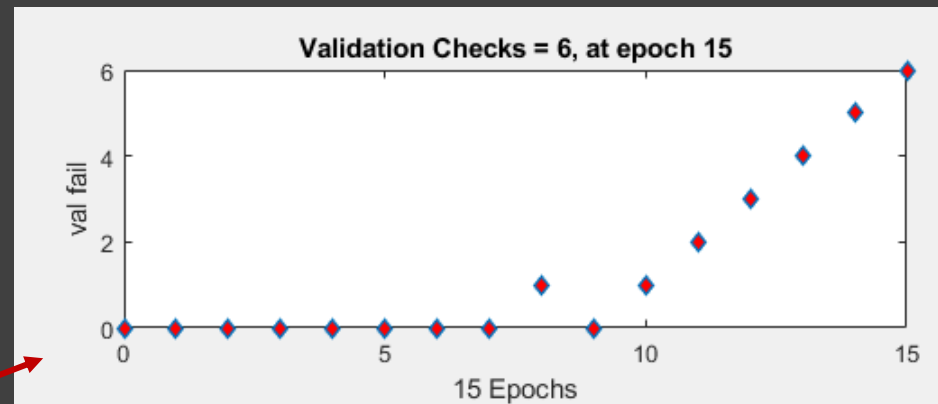
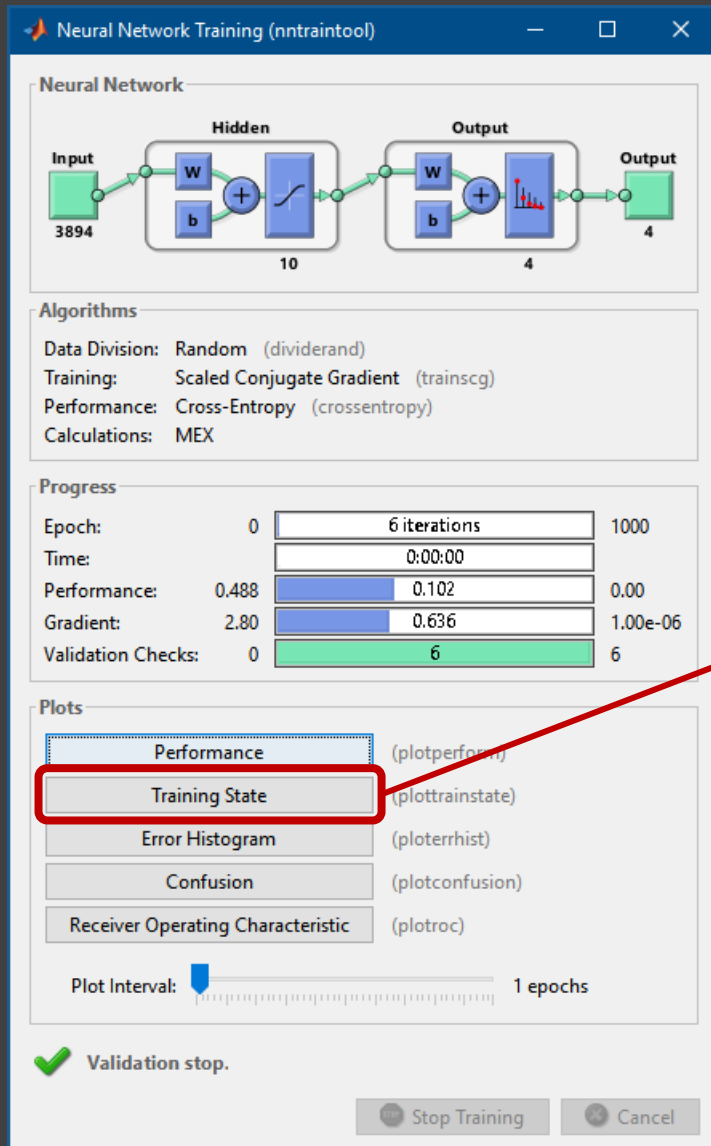
MATLAB: Neural Pattern Recognition tool



MATLAB: Neural Pattern Recognition tool

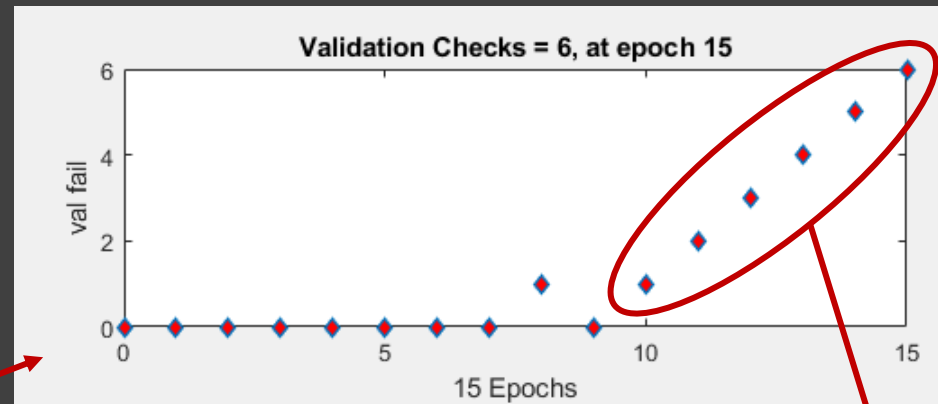
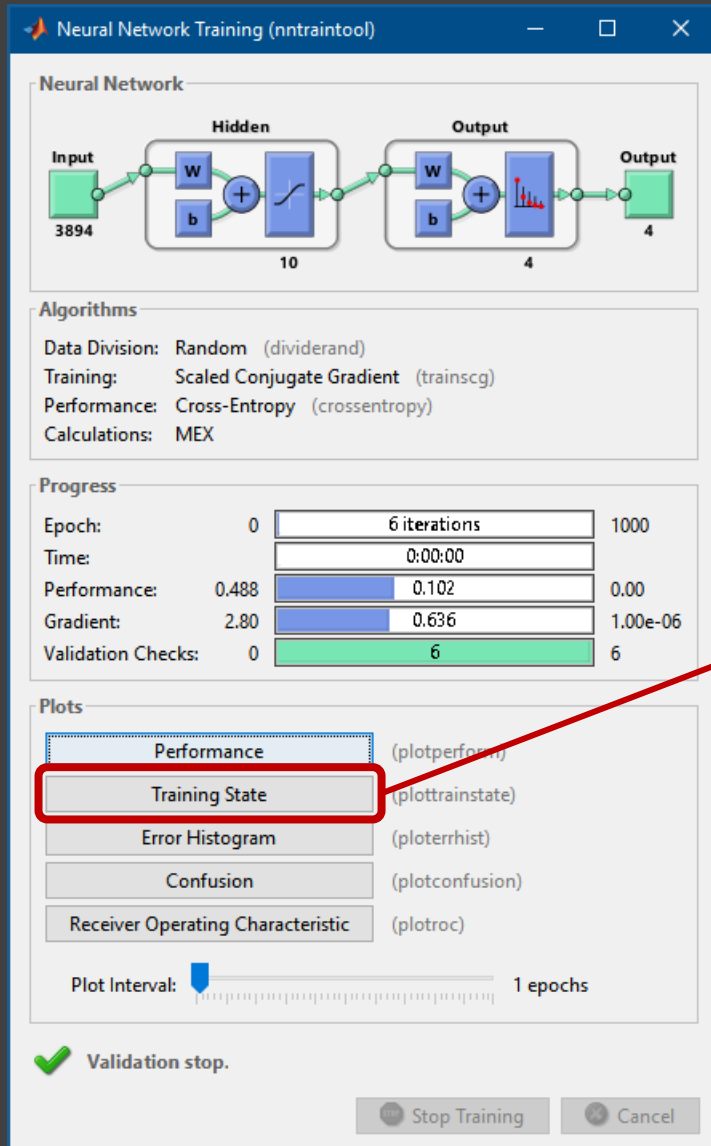


MATLAB: Neural Pattern Recognition tool



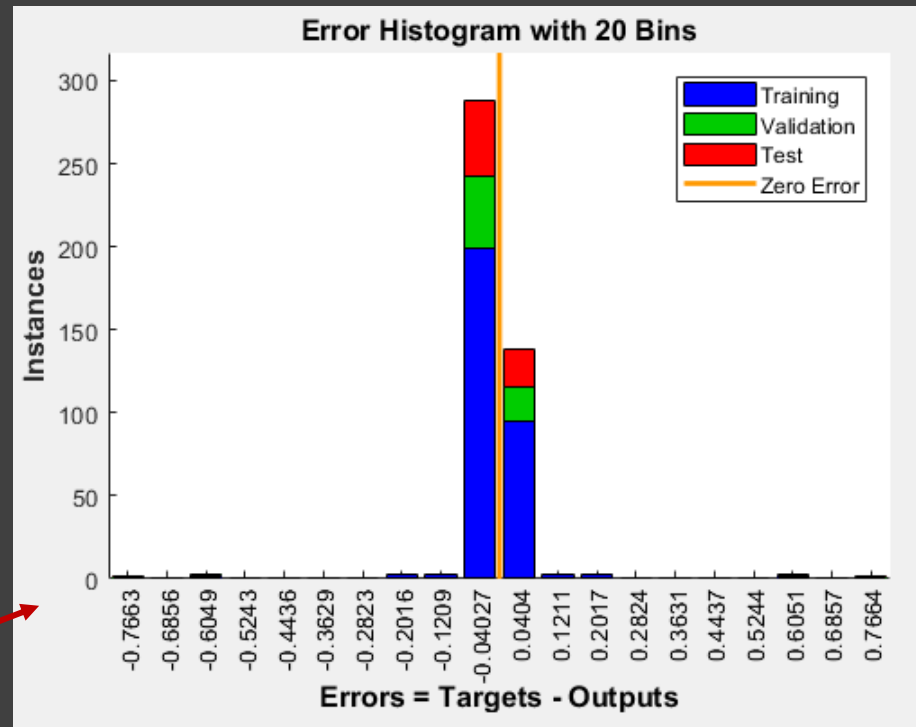
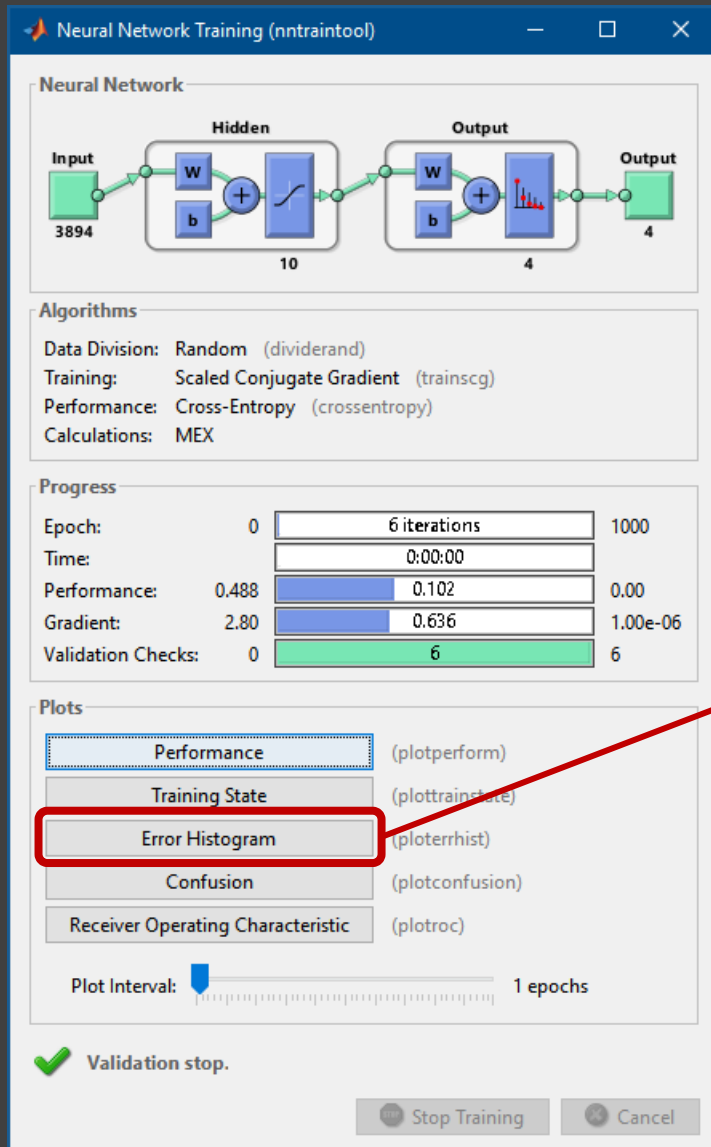
Απεικόνιση της χειροτέρευσης του validation error για διαδοχικές εποχές

MATLAB: Neural Pattern Recognition tool



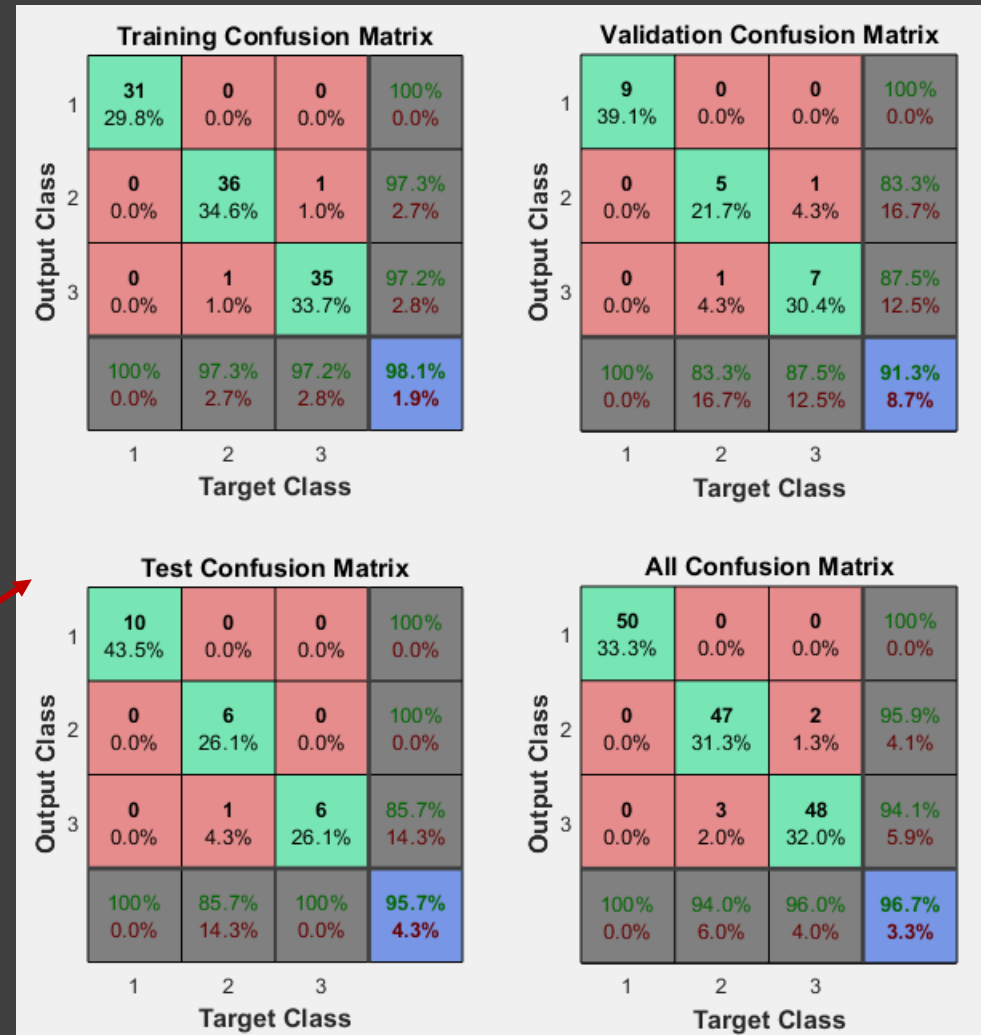
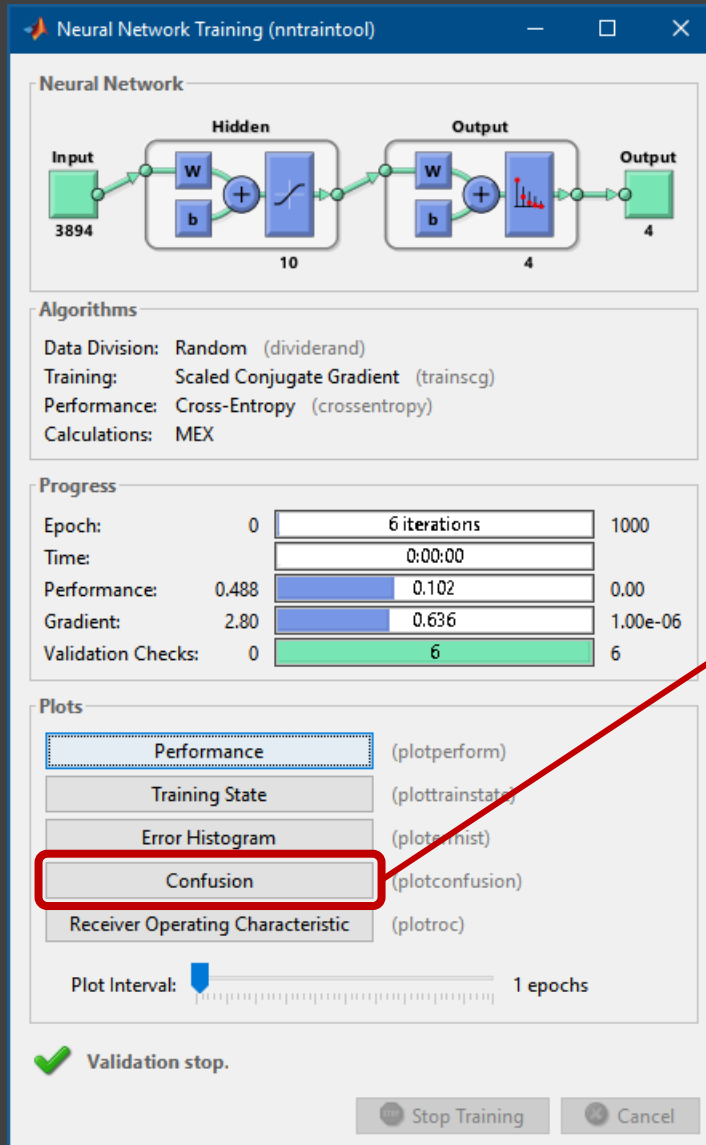
Απεικόνιση της χειροτέρευσης του validation error για διαδοχικές εποχές στο συγκεκριμένο παράδειγμα: για 6 διαδοχικές εποχές υπήρχε χειροτέρευση του validation, άρα για 6 διαδοχικές εποχές δεν υπήρξε βελτίωση

MATLAB: Neural Pattern Recognition tool

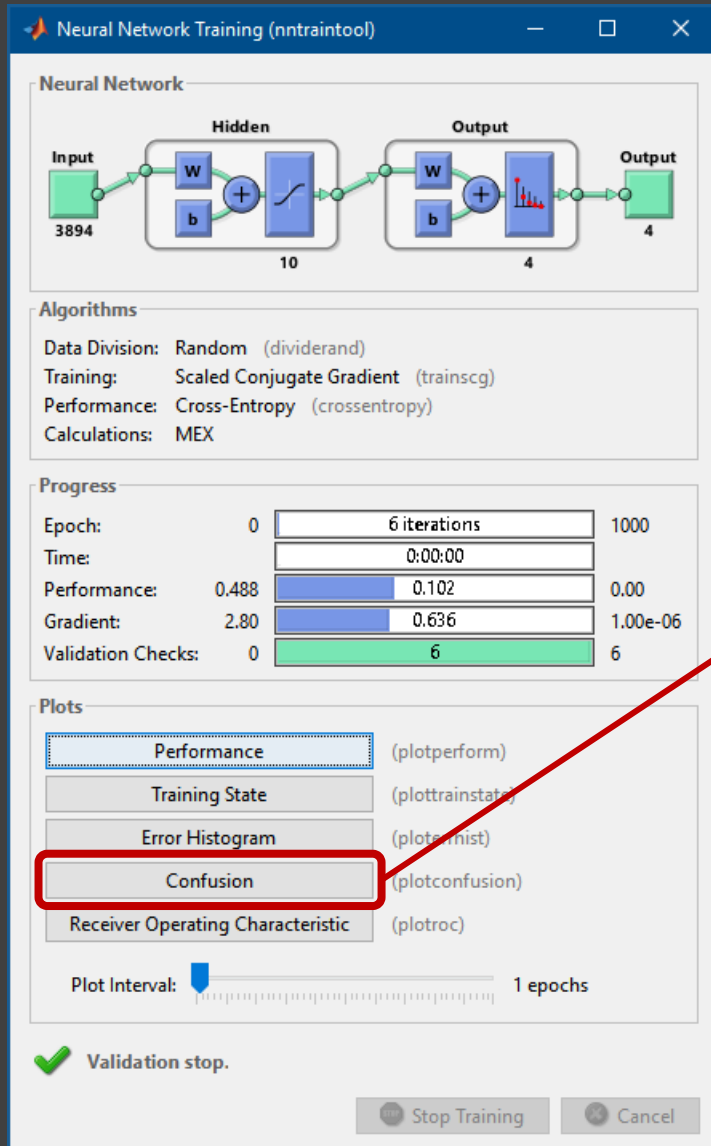


Ιστόγραμμα σφαλμάτων:
κάθε τιμή σφάλματος σε πόσα
παράδειγματα αντιστοιχεί

MATLAB: Neural Pattern Recognition tool



MATLAB: Neural Pattern Recognition tool



MATLAB: Neural Pattern Recognition tool

Neural Pattern Recognition (nprtool)



Save Results
Generate MATLAB scripts, save results and generate diagrams.

Generate Scripts

Recommended >> Use these scripts to reproduce results and solve similar problems.

Generate a script to train and test a neural network as you just did with this tool:

Generate a script with additional options and example code:

 Simple Script

 Advanced Script

Save Data to Workspace

☒ Save network to MATLAB network object named:

☒ Save performance and data set information to MATLAB struct named:

☒ Save outputs to MATLAB matrix named:

☒ Save errors to MATLAB matrix named:

☐ Save inputs to MATLAB matrix named:

☐ Save targets to MATLAB matrix named:

☐ Save ALL selected values above to MATLAB struct named:

Restore Defaults

Save Results

 Save results and click [Finish].

Neural Network Start

Welcome

Back

Next

 Finish

2020-2021 | Πολυτεχνείο Κρήτης | Σχολή Μηχανικών Παραγωγής & Διοίκησης | Εισαγωγή στην Τεχνητή Νοημοσύνη (ΜΠΔ306) | Εργαστήριο 7ο

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y)
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

← συναρτήσεις
βελτιστοποίησης

π.χ.

'trainscg': χρησιμοποιεί τη μέθοδο των συζυγών παραγώγων (Conjugate Gradient) για την ανανέωση των βαρών

'trainlm': χρησιμοποιεί τη μέθοδο Levenberg-Marquardt για την ανανέωση των βαρών

γράφοντας στο Command Window

>> help nntrain

βλέπουμε όλες τις δυνατές επιλογές

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

hiddenLayerSize: διάνυσμα

- # στηλών = # κρυφών επιπέδων
- τιμή στην θέση X = # νευρώνων στο κρυφό επίπεδο X

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

hiddenLayerSize = [10];
1 κρυφό επίπεδο με 10 νευρώνες

hiddenLayerSize = [10 4 6];
3 κρυφά επίπεδα με
10 νευρώνες στο 1^ο από τα 3
4 νευρώνες στο 2^ο από τα 3
6 νευρώνες στο 3^ο από τα 3

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

patternnet(): δημιουργεί ένα feedforward δίκτυο που θα εκπαιδευτεί για να κάνει κατηγοριοποίηση των δεδομένων σε κλάσεις (κατηγορίες)

<https://www.mathworks.com/help/deeplearning/ref/patternnet.html>

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

η έξοδος της `patternnet()`
είναι μεταβλητή τύπου `struct`

MATLAB: struct array

struct

Structure array

Description

A *structure array* is a data type that groups related data using data containers called *fields*. Each field can contain any type of data. Access data in a field using dot notation of the form `structName.fieldName`.

Creation

When you have data to put into a new structure, create the structure using dot notation to name its fields one at a time:

```
s.a = 1;  
s.b = {'A', 'B', 'C'}
```

```
s = struct with fields:  
  a: 1  
  b: {'A' 'B' 'C'}
```

You also can create a structure array using the `struct` function, described below. You can specify many fields simultaneously, or create a nonscalar structure array.

Syntax

```
s = struct  
s = struct(field,value)  
s = struct(field1,value1,...,fieldN,valueN)  
s = struct([])
```

Μια μεταβλητή τύπου `struct` ομαδοποιεί συναφή δεδομένα χρησιμοποιώντας θέσεις δεδομένων που ονομάζονται *πεδία* (*fields*).

Κάθε πεδίο μπορεί να περιέχει οποιοδήποτε τύπο δεδομένων.

Για να έχουμε πρόσβαση σε δεδομένα που βρίσκονται σε ένα πεδίο γράφουμε:
`structName.fieldName`

όπου `structName` είναι το όνομα της μεταβλητής τύπου `struct` και `fieldName` είναι το όνομα του πεδίου

MATLAB: struct array

struct

Structure array

Description

A *structure array* is a data type that groups related data using fields. Each field can contain any type of data. Access data in a field using `structName.fieldName`.

Creation

When you have data to put into a new structure, create the structure and name its fields one at a time:

```
s.a = 1;  
s.b = {'A', 'B', 'C'}
```

```
s = struct with fields:  
  a: 1  
  b: {'A' 'B' 'C'}
```

You also can create a structure array using the `struct` function, described below. You can specify many fields simultaneously, or create a nonscalar structure array.

Syntax

```
s = struct  
s = struct(field,value)  
s = struct(field1,value1,...,fieldN,valueN)  
s = struct([])
```



MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y)
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

net : μεταβλητή τύπου struct με
όνομα "net"

περιέχει όλες τις παραμέτρους
του δικτύου που δημιούργησε η
συνάρτηση patternnet()

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

net : μεταβλητή τύπου struct με
όνομα "net"

divideParam : πεδίο (field)
της μεταβλητής net που περιέχει
τις τιμές των παραμέτρων για το
διαχωρισμό των δεδομένων

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

εκπαίδευση
του δικτύου

ορίσματα της `train()`:
`net`: περιέχει τις παραμέτρους
του δικτύου που δημιούργησε η
συνάρτηση `patternnet()`
`x`: πίνακας μεταβλητών
`t`: πίνακας μεταβλητών κλάσης

<https://www.mathworks.com/help/deeplearning/ref/train.html>

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

εκπαίδευση
του δικτύου

δοκιμή του
δικτύου

<https://www.mathworks.com/help/deeplearning/ref/perform.html>

`perform()`: κάνει αξιολόγηση της εκπαίδευσης του δικτύου χρησιμοποιώντας το training set
Δίνει ως έξοδο σφάλμα

MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

εκπαίδευση
του δικτύου

δοκιμή του
δικτύου

<https://www.mathworks.com/help/deeplearning/ref/perform.html>

ορίσματα της `perform()`:

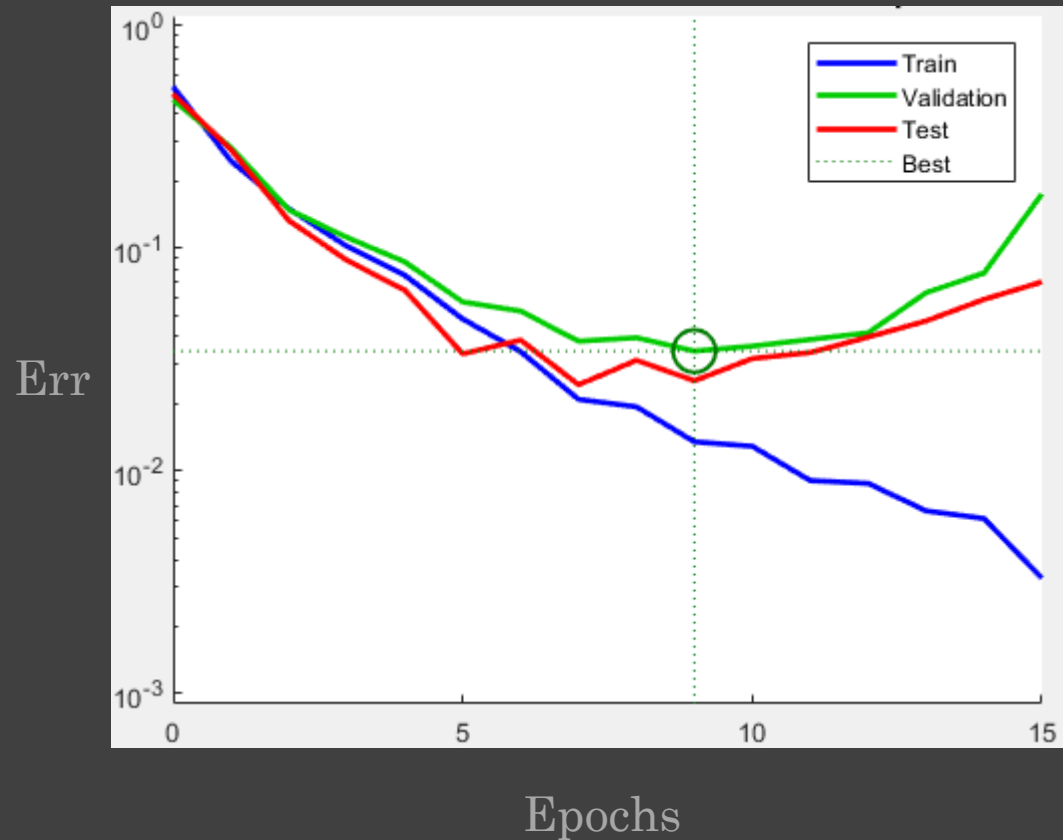
`net`: περιέχει τις παραμέτρους του δικτύου

`t`: πίνακας μεταβλητών κλάσης του training set

`y`: έξοδος δικτύου όταν ως είσοδο έχουμε βάλει το training set

Παράμετροι εκπαίδευσης

Όμως δεν μας είναι πολύ χρήσιμο το training error, αλλά το test error



MATLAB: Neural Pattern Recognition tool

```
1 % Solve a Pattern Recognition Problem with a Neural Network
2 % Script generated by Neural Pattern Recognition app
3 % Created 11-Dec-2019 01:49:45
4 %
5 % This script assumes these variables are defined:
6 %
7 % irisInputs - input data.
8 % irisTargets - target data.
9
10 x = irisInputs;
11 t = irisTargets;
12
13 % Choose a Training Function
14 % For a list of all training functions type: help nntrain
15 % 'trainlm' is usually fastest.
16 % 'trainbr' takes longer but may be better for challenging problems.
17 % 'trainscg' uses less memory. Suitable in low memory situations.
18 trainFcn = 'trainscg';
19
20 % Create a Pattern Recognition Network
21 hiddenLayerSize = 10;
22 net = patternnet(hiddenLayerSize, trainFcn);
23
24 % Setup Division of Data for Training, Validation, Testing
25 net.divideParam.trainRatio = 70/100;
26 net.divideParam.valRatio = 15/100;
27 net.divideParam.testRatio = 15/100;
28
29 % Train the Network
30 [net,tr] = train(net,x,t);
31
32 % Test the Network
33 y = net(x);
34 e = gsubtract(t,y);
35 performance = perform(net,t,y);
36 tind = vec2ind(t);
37 yind = vec2ind(y);
38 percentErrors = sum(tind ~= yind)/numel(tind);
39
40 % View the Network
41 view(net)
```

δεδομένα

επιλογή συνάρτησης
εκπαίδευσης

διαμόρφωση
αρχιτεκτονικής δικτύου

επιλογή ποσοστών για training,
validation, test sets

εκπαίδευση
του δικτύου

δοκιμή του
δικτύου

<https://www.mathworks.com/help/deeplearning/ref/perform.html>

για αξιολόγηση με το test set:

t: πίνακας μεταβλητών κλάσης
του test set

y: έξοδος δικτύου όταν ως
είσοδο έχουμε βάλει το test set

Αξιολόγηση Εκπαίδευσης

	var1	var2	var3	var4	var5	Έξοδος 1	Έξοδος 2
Παρά- δειγμα 12							
Παρά- δειγμα 13							
Παρά- δειγμα 14							
Παρά- δειγμα 15							

testVars

πίνακας διαστάσεων 4x5

στον κώδικα: `y`

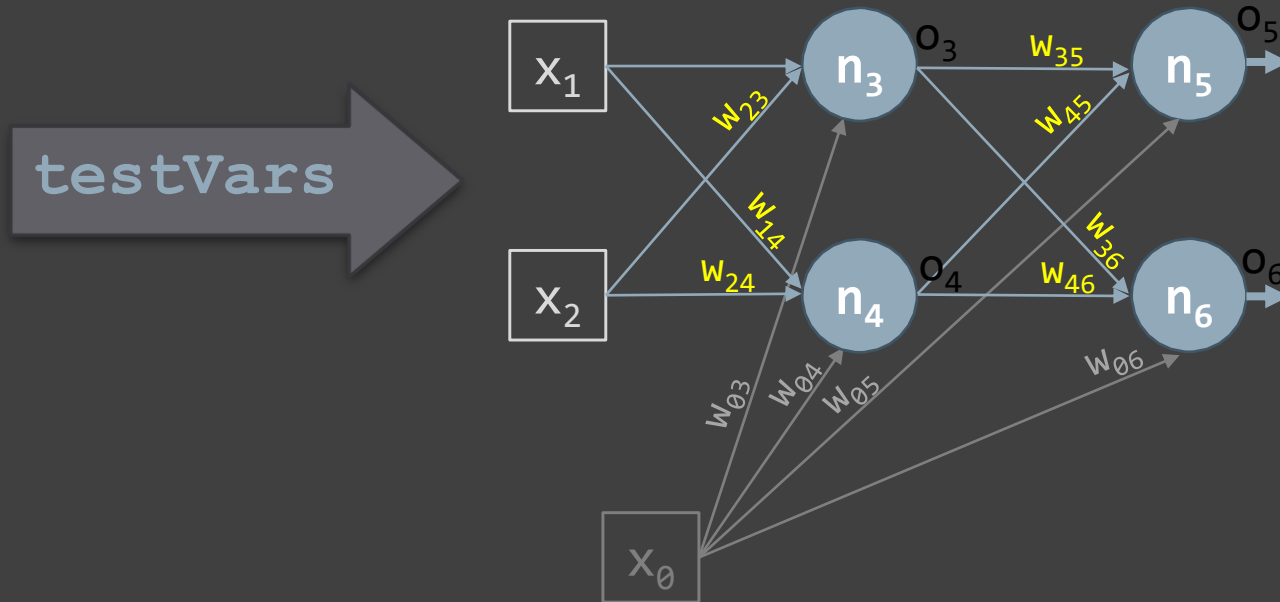
testTargets

πίνακας
διαστάσεων 4x2

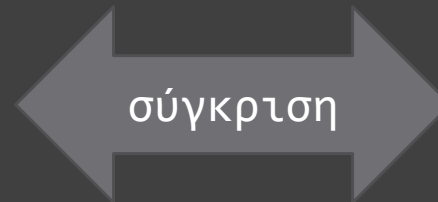
στον κώδικα: `t`

Αξιολόγηση Εκπαίδευσης

ΕΚΠΑΙΔΕΥΜΕΝΟ ΤΝΔ



testTargets
πίνακας
διαστάσεων 4x2



testOutput
πίνακας
διαστάσεων 4x2

MATLAB: Neural Pattern Recognition tool

```
% Test the Network  
y = net(testVars);
```

δοκιμή του
δικτύου

```
tind = vec2ind(t);  
yind = vec2ind(y);  
percentErrors = sum(tind ~= yind)/numel(tind);
```

ποσοστό εξόδων που
δεν είναι ίσες με
τις εξόδους-στόχους

testTargets: πίνακας μεταβλητών κλάσης του test set

y: έξοδος δικτύου όταν ως είσοδο έχουμεβάλει το test set (testVars)

vec2ind(): μετατρέπει τα δυαδικά διανύσματα σε δεκαδικούς αριθμούς

numel(): επιστρέφει τον αριθμό στοιχείων του ορίσματος

MATLAB: επιπλέον παραμετροποίηση

ΜΕΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ `patternnet()` ΚΑΙ ΠΡΙΝ ΤΗ `train()`

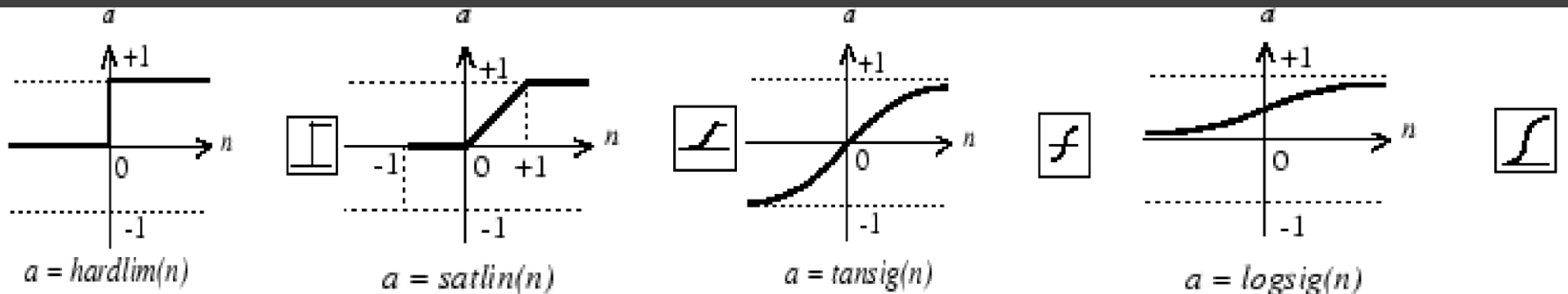
`net.layers{1}` : βλέπουμε τις παραμέτρους του 1^{ου} επιπέδου

`net.layers{2}` : βλέπουμε τις παραμέτρους του 2^{ου} επιπέδου

`net.layers{1}.transferFcn` : βλέπουμε τη συνάρτηση ενεργοποίησης των νευρώνων του 1^{ου} επιπέδου και μπορούμε να την αλλάξουμε

π.χ. `net.layers{1}.transferFcn = 'tansig';`

επιλογές: 'hardlim', 'tansig', 'logsig', 'satlin'



Hard-Limit Transfer Function Satlin Transfer Function Tan-Sigmoid Transfer Function Log-Sigmoid Transfer Function

MATLAB: επιπλέον παραμετροποίηση

ΜΕΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ `patternnet()` ΚΑΙ ΠΡΙΝ ΤΗ `train()`

`net.performFcn` : μέθοδος υπολογισμού σφάλματος δικτύου

επιλογές:

- 'mae' - Mean absolute error performance function
- 'mse' - Mean squared error performance function
- 'sae' - Sum absolute error performance function
- 'sse' - Sum squared error performance function
- 'crossentropy' - Cross-entropy performance
- 'msespase' - Mean squared error performance function with L2 weight and sparsity regularizers

στο Command Window: `help nnperformance`

MATLAB: επιπλέον παραμετροποίηση

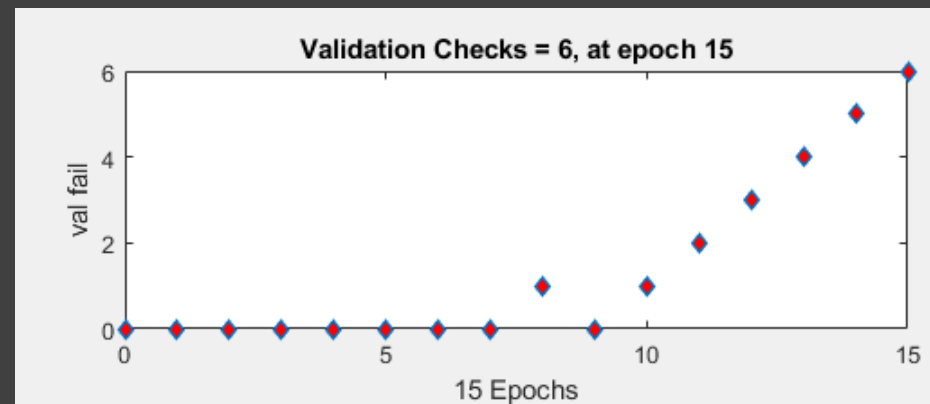
ΜΕΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ `patternnet()` ΚΑΙ ΠΡΙΝ ΤΗ `train()`

`net.trainParam.epochs` : πόσοι κύκλοι εκπαίδευσης (epochs) θα πραγματοποιηθούν (μέγιστο)

`net.trainParam.goal` : αποδεκτή απόκλιση εξόδου του δικτύου από την έξοδο-στόχο

`net.trainParam.time` : ο χρόνος που θα εκπαιδεύεται το δίκτυο πριν σταματήσει (σε sec)

`net.trainParam.max_fail` : αποδεκτός αριθμός διαδοχικών epochs που δεν παρατηρείται βελτίωση κατά το validation



MATLAB: επιπλέον παραμετροποίηση

ΜΕΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ `patternnet()` ΚΑΙ ΠΡΙΝ ΤΗ `train()`

`net.divideParam.trainRatio` : τι ποσοστό από τα δεδομένα θα αποτελέσει το `training set`, π.χ. 0.7

`net.divideParam.valRatio` : τι ποσοστό από τα δεδομένα θα αποτελέσει το `validation set`, π.χ. 0.15

`net.divideParam.testRatio` : τι ποσοστό από τα δεδομένα θα αποτελέσει το `test set`, π.χ. 0.15

ΠΡΟΣΟΧΗ!

οι τρεις παραπάνω τιμές πρέπει να αθροίζουν στο 1
($0.7+0.15+0.15=1$)

MATLAB: επιπλέον παραμετροποίηση

ΜΕΤΑ ΤΗΝ ΕΦΑΡΜΟΓΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ `patternnet()` ΚΑΙ ΠΡΙΝ ΤΗ `train()`

`net.trainParam.lr` : ρυθμός μάθησης (learning rate)

MATLAB: Neural Pattern Recognition tool

Πληροφορίες για όλες τις παραμέτρους του net :

<https://www.mathworks.com/help/deeplearning/ug/neural-network-object-properties.html>