



# Τεχνητή Νοημοσύνη

*20η διάλεξη (2023-24)*

Ίων Ανδρουτσόπουλος

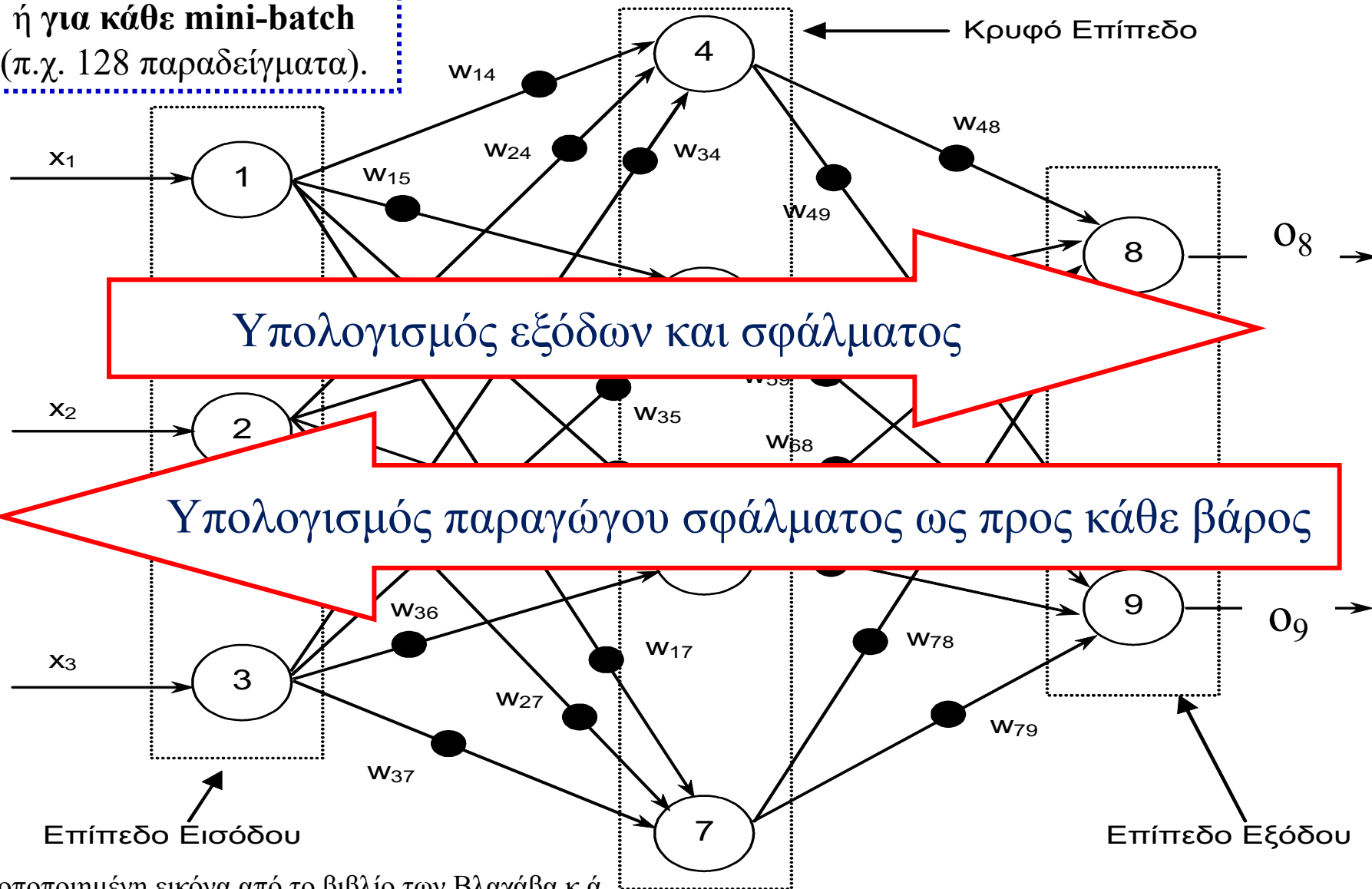
<http://www.aueb.gr/users/ion/>

# Τι θα ακούσετε σήμερα

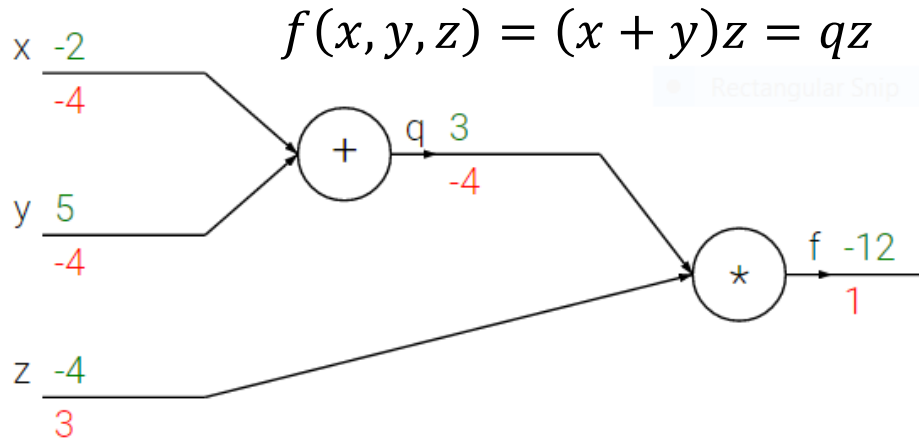
- Περισσότερα για τα πολυ-επίπεδα Perceptron (MLPs) και την ανάστροφη μετάδοση (backpropagation).
- Αυτόματος υπολογισμός παραγώγων κατά την ανάστροφη μετάδοση.
- Διασταυρωμένη εντροπία (cross-entropy).

# Αλγόριθμος ανάστροφης μετάδοσης

Για κάθε ένα παράδειγμα  
ή για κάθε mini-batch  
(π.χ. 128 παραδείγματα).



# Παράδειγμα γράφου υπολογισμού



Παράδειγμα και σχήμα από το μάθημα  
“CNNs for Visual Recognition” (2016,  
F.-F. Li, A. Karpathy, J. Johnson) του  
Πανεπιστημίου Stanford.  
<http://cs231n.github.io/optimization-2/>

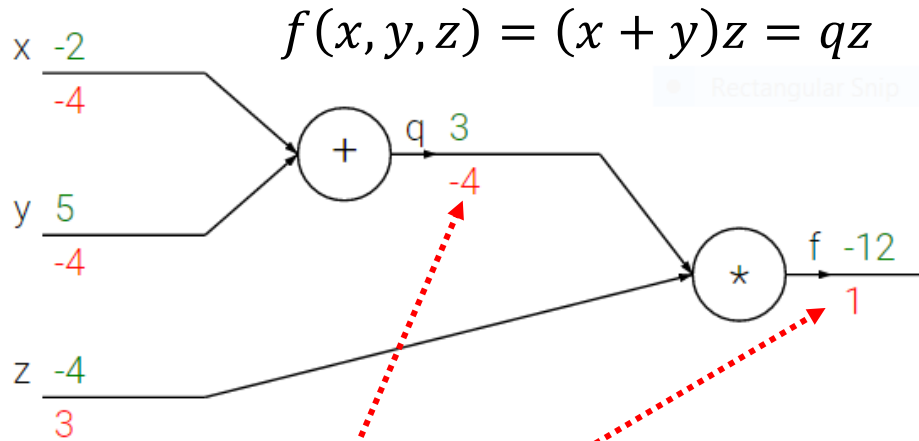
- Προς τα εμπρός:  $\langle x, y, z \rangle = \langle -2, 5, -4 \rangle$ ,  $q = 3$ ,  $f = -12$
- Ας υποθέσουμε ότι θέλουμε να ελαχιστοποιήσουμε την  $f$  χρησιμοποιώντας **στοχαστική κατάβαση κλίσης** (SGD).
  - Σε ένα πιο ρεαλιστικό σενάριο, η  $f$  θα ήταν μια **συνάρτηση σφάλματος** και το  $\langle x, y, z \rangle$  το **διάνυσμα βαρών**.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \leftarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \eta \cdot \nabla f(x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$

Χρειαζόμαστε:

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$$

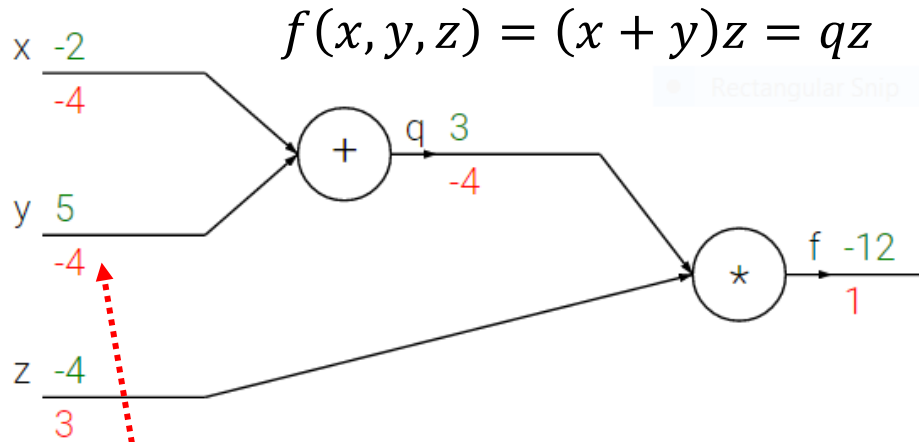
# Ανάστροφη μετάδοση στο γράφο



Παράδειγμα και σχήμα από το μάθημα  
“CNNs for Visual Recognition” (2016,  
F.-F. Li, A. Karpathy, J. Johnson) του  
Πανεπιστημίου Stanford.  
<http://cs231n.github.io/optimization-2/>

- **Ανάστροφη μετάδοση:** Υπολογίζουμε παραγώγους από δεξιά προς αριστερά.
  - $\frac{\partial f}{\partial f} = 1$  εξ ορισμού.
  - $\frac{\partial f}{\partial q} = z$ . Και για το συγκεκριμένο διάνυσμα εισόδου  $\langle x, y, z \rangle$ , έχουμε  $z = -4$ .
  - Κατά τους προς τα εμπρός υπολογισμούς, πρέπει να αποθηκεύουμε τις εισόδους και τις εξόδους όλων των κόμβων (π.χ., εδώ χρειαστήκαμε την τιμή του  $z$ ).

# Ανάστροφη μετάδοση στο γράφο



Παράδειγμα και σχήμα από το μάθημα  
 “CNNs for Visual Recognition” (2016,  
 F.-F. Li, A. Karpathy, J. Johnson) του  
 Πανεπιστημίου Stanford.  
<http://cs231n.github.io/optimization-2/>

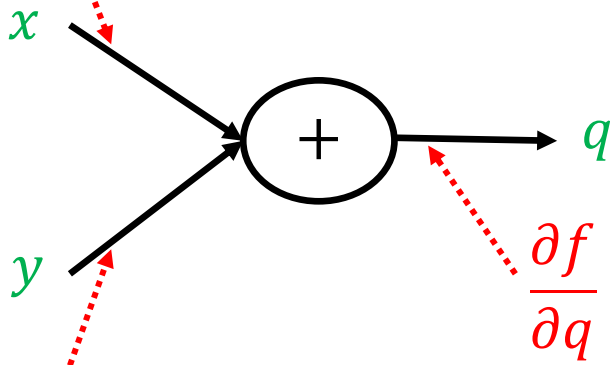
- **Ανάστροφη μετάδοση:**

- $\frac{\partial f}{\partial f} = 1$  εξ ορισμού.
- $\frac{\partial f}{\partial q} = z$ . Και για το συγκεκριμένο  $\langle x, y, z \rangle$ ,  $z = -4$ .
- $\frac{\partial f}{\partial z} = q$ . Και για το συγκεκριμένο  $\langle x, y, z \rangle$ ,  $q = 3$ .
- $\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} = \frac{\partial f}{\partial q} \cdot 1$ . Και εδώ  $\frac{\partial f}{\partial q} = -4$ .
- $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = \frac{\partial f}{\partial q} \cdot 1$ . Και εδώ  $\frac{\partial f}{\partial q} = -4$ .

εισερχόμενη παράγωγος      τοπική παράγωγος της πύλης +

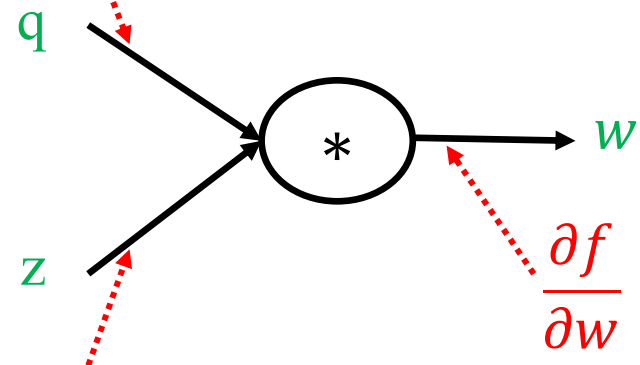
# Υλοποιήσεις πυλών που μπορούμε να συνδυάσουμε

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = \frac{\partial f}{\partial q} \cdot 1$$



$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} = \frac{\partial f}{\partial q} \cdot 1$$

$$\frac{\partial f}{\partial q} = \frac{\partial f}{\partial w} \frac{\partial w}{\partial q} = \frac{\partial f}{\partial w} \cdot z$$



$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial w} \frac{\partial w}{\partial z} = \frac{\partial f}{\partial w} \cdot q$$

εισερχόμενη από  
δεξιά παράγωγος

τοπική παράγωγος  
(μέρος της υλοποίησης)

εισερχόμενη από  
δεξιά παράγωγος

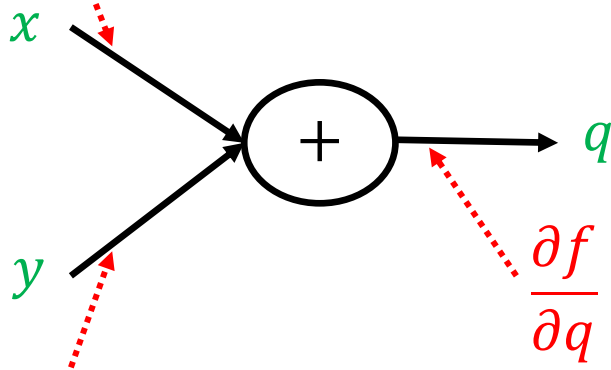
τοπική παράγωγος  
(μέρος της υλοποίησης)

- Μπορούμε να υλοποιήσουμε τις **πύλες** ως **τάξεις** (π.χ., σε Java, C++ ή Python).

- Με **μεθόδους προς τα εμπρός** και **ανάστροφης μετάδοσης**.

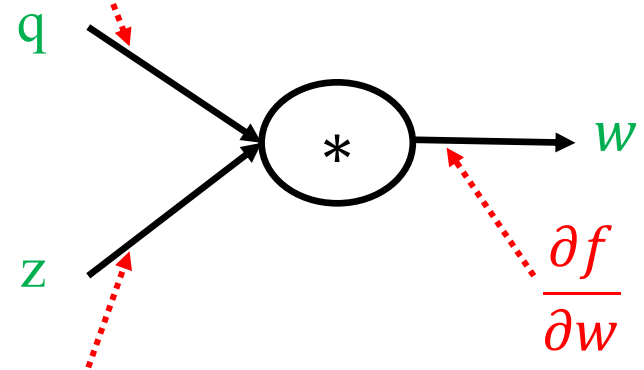
# Plug-and-play gates

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = \frac{\partial f}{\partial q} \cdot 1$$



$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} = \frac{\partial f}{\partial q} \cdot 1$$

$$\frac{\partial f}{\partial q} = \frac{\partial f}{\partial w} \frac{\partial w}{\partial q} = \frac{\partial f}{\partial w} \cdot z$$



$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial w} \frac{\partial w}{\partial z} = \frac{\partial f}{\partial w} \cdot q$$

```
class PlusGate:
```

```
    forward(x, y):
```

```
        return x+y
```

```
    backward( $\frac{\partial f}{\partial q}$ ):
```

```
        return  $\langle \frac{\partial f}{\partial q}, \frac{\partial f}{\partial q} \rangle$ 
```

```
class StarGate:
```

```
    forward(q, z):
```

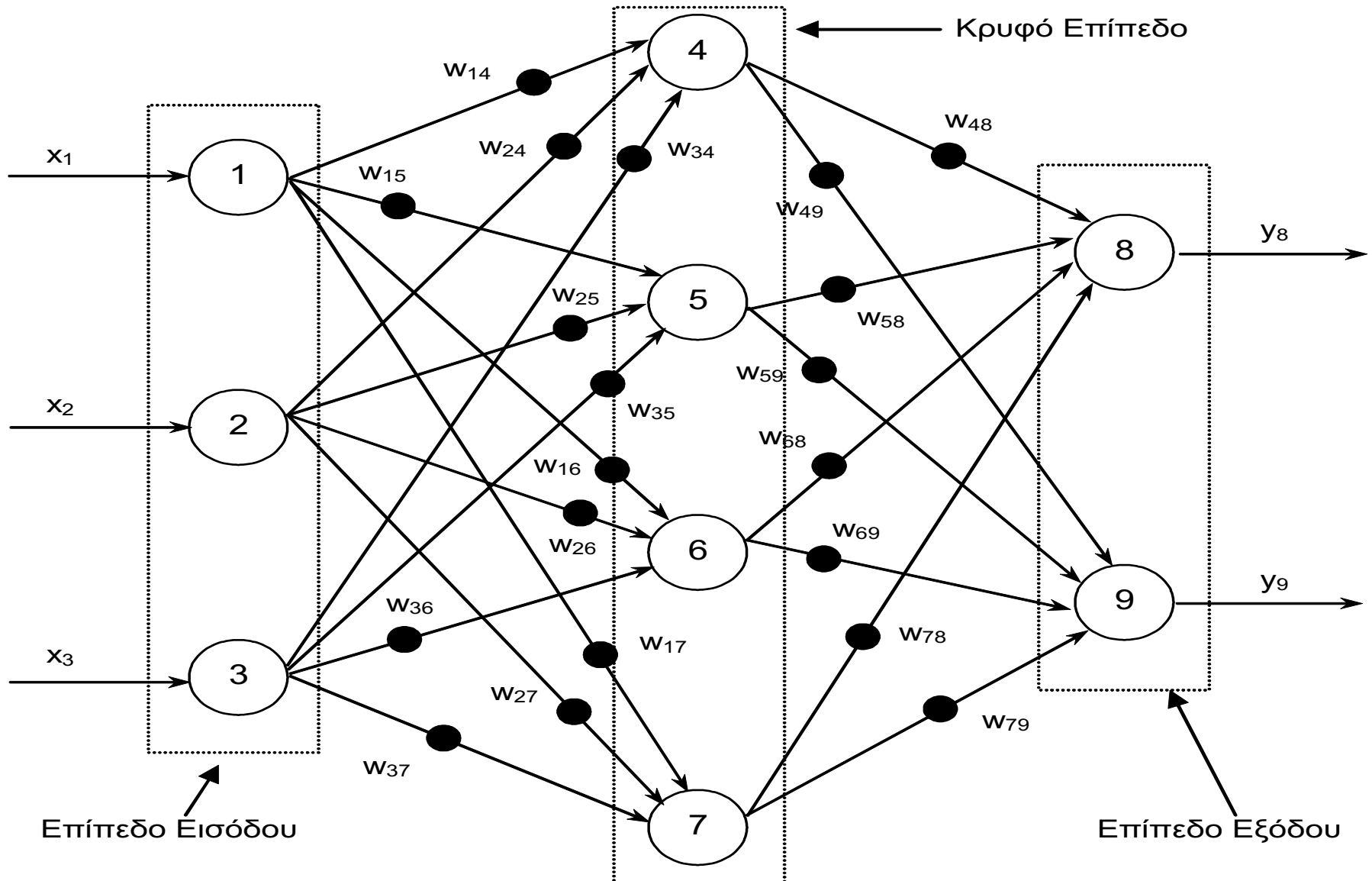
```
        return q * z
```

```
    backward( $\frac{\partial f}{\partial w}$ ):
```

```
        return  $\langle \frac{\partial f}{\partial w} \cdot z, \frac{\partial f}{\partial w} \cdot q \rangle$ 
```



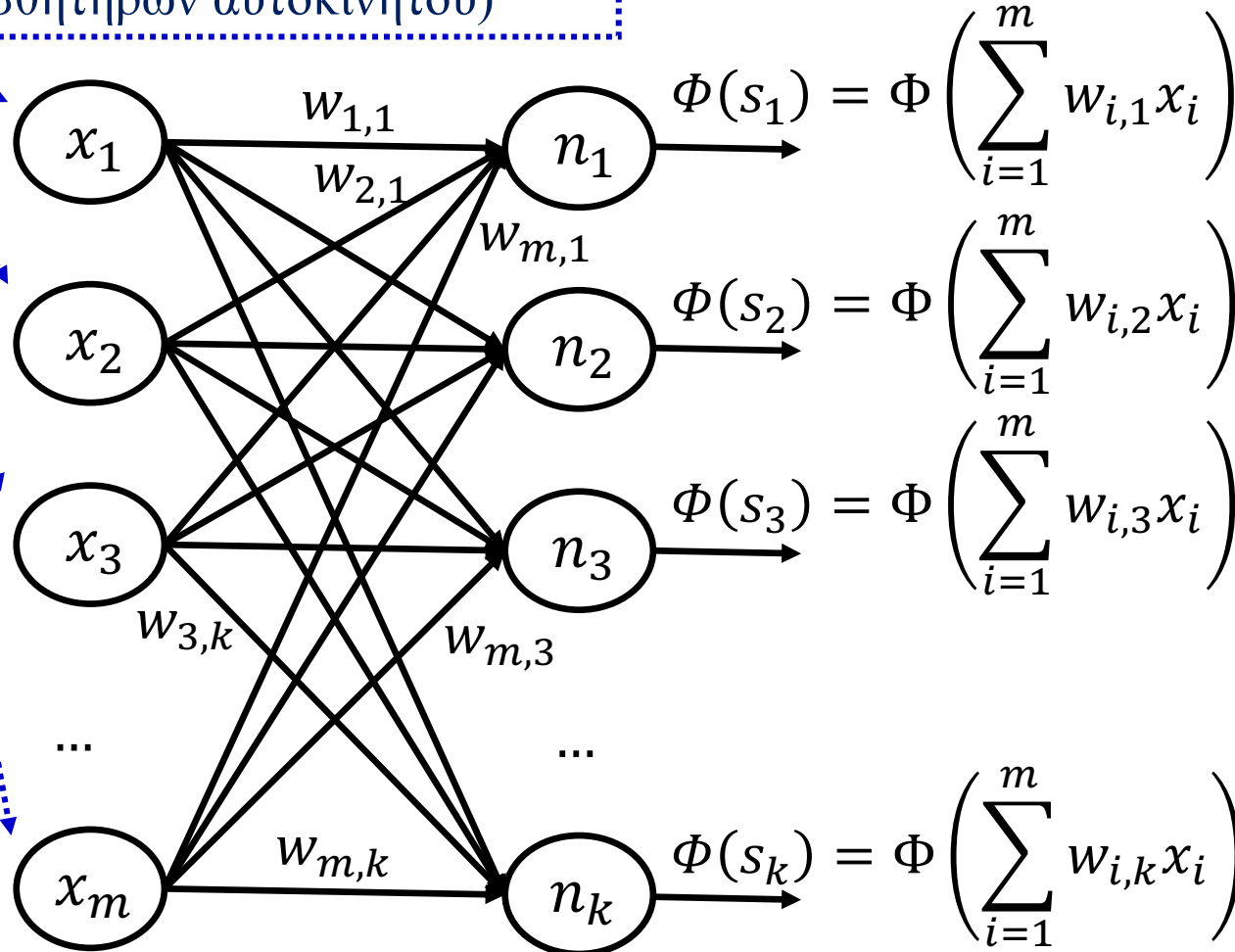
# Πολυ-επίπεδο Perceptron (MLP)



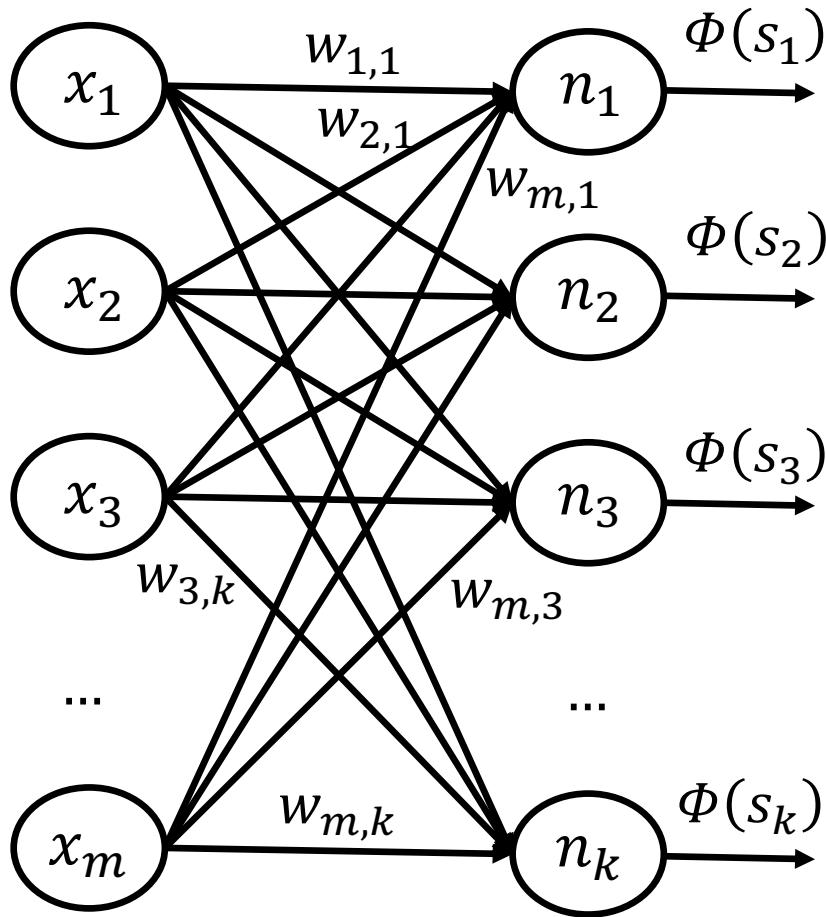
Εικόνα από το βιβλίο των Βλαχάβα κ.ά.

# Πιο συμπαγής συμβολισμός

Διάνυσμα εισόδου (π.χ. ενδείξεις αισθητήρων αυτοκινήτου)



# Πιο συμπαγής συμβολισμός



$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \dots \\ s_k \end{bmatrix} = \begin{bmatrix} w_{1,1}x_1 + w_{2,1}x_2 + \dots + w_{m,1}x_m \\ w_{1,2}x_1 + w_{2,2}x_2 + \dots + w_{m,2}x_m \\ w_{1,3}x_1 + w_{2,3}x_2 + \dots + w_{m,3}x_m \\ \dots \\ w_{1,k}x_1 + w_{2,k}x_2 + \dots + w_{m,k}x_m \end{bmatrix}$$

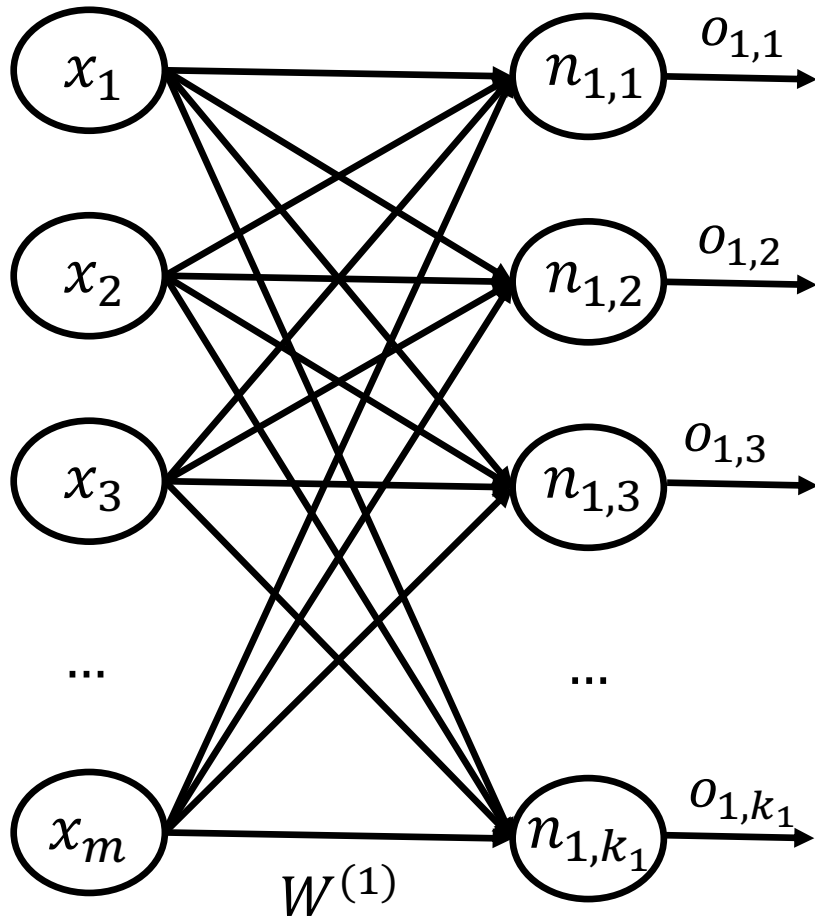
$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \dots \\ s_k \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{m,1} \\ w_{1,2} & w_{2,2} & \dots & w_{m,2} \\ w_{1,3} & w_{2,3} & \dots & w_{m,3} \\ \dots & \dots & \dots & \dots \\ w_{1,k} & w_{2,k} & \dots & w_{m,k} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_m \end{bmatrix}$$

$$\vec{s} = W\vec{x}$$

Μαθαίνουμε τον  $W$   
με ανάστροφη  
μετάδοση.

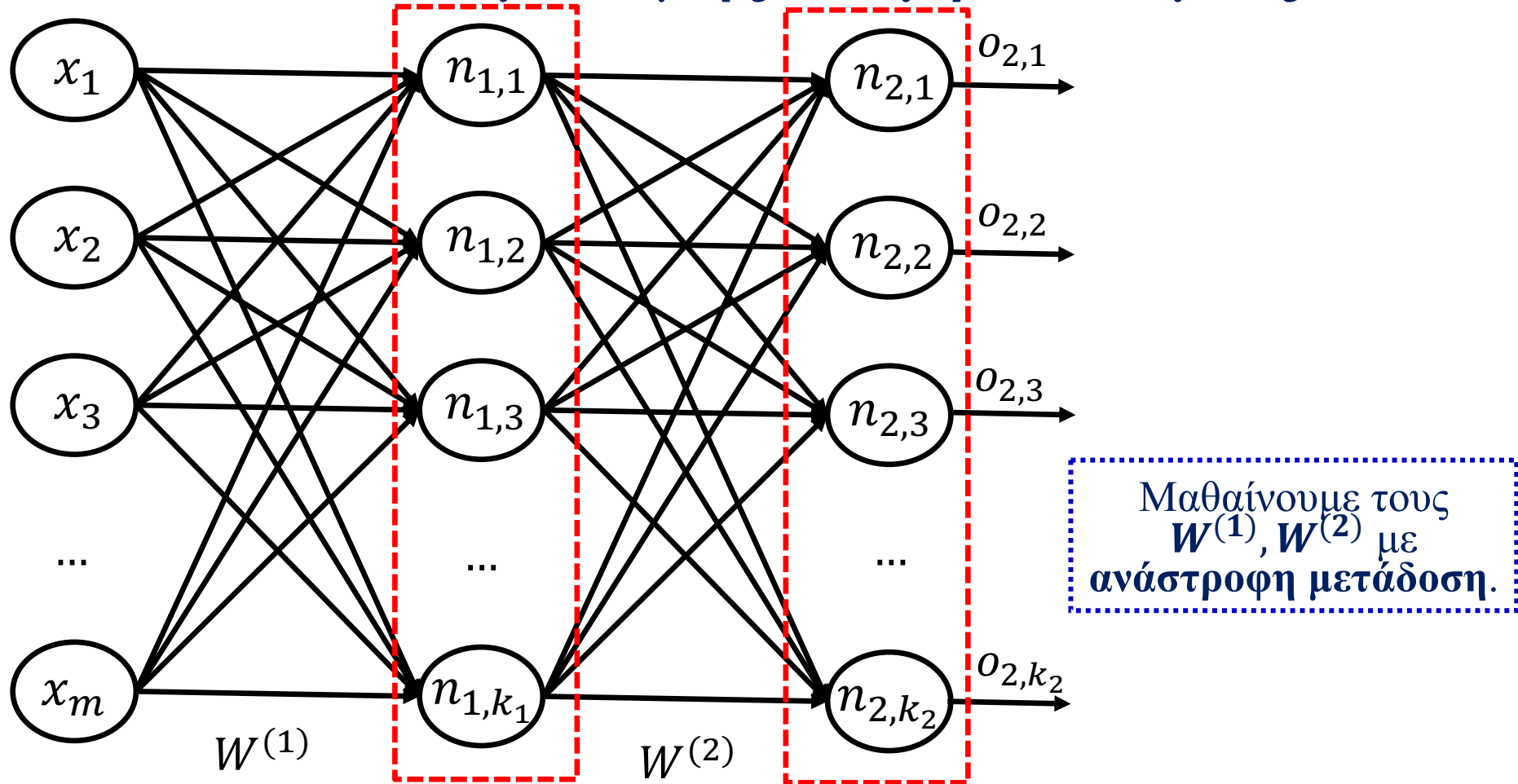
$$\vec{o} = \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ \dots \\ o_k \end{bmatrix} = \begin{bmatrix} \Phi(s_1) \\ \Phi(s_2) \\ \Phi(s_3) \\ \dots \\ \Phi(s_k) \end{bmatrix} = \Phi(\vec{s}) = \Phi(W\vec{x})$$

# Πιο συμπαγής συμβολισμός



$$\vec{o}^{(1)} = \begin{bmatrix} o_{1,1} \\ o_{1,2} \\ \dots \\ o_{1,k_1} \end{bmatrix} = \Phi(\vec{s}^{(1)}) = \Phi(W^{(1)}\vec{x})$$

# Πιο συμπαγής συμβολισμός

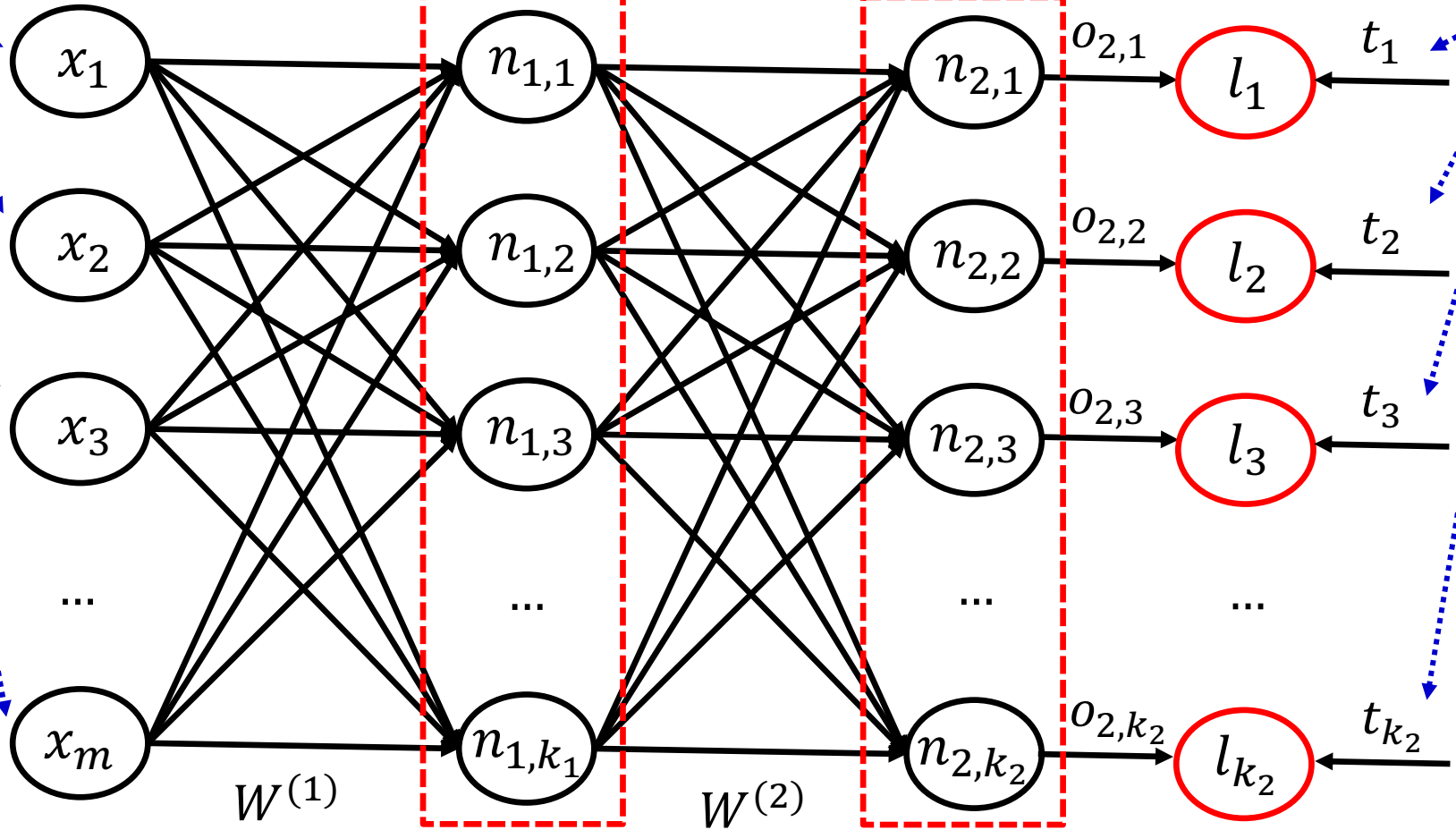


$$\vec{o}^{(1)} = \begin{bmatrix} o_{1,1} \\ o_{1,2} \\ \dots \\ o_{1,k_1} \end{bmatrix} = \Phi(W^{(1)}\vec{x}) \quad \vec{o}^{(2)} = \begin{bmatrix} o_{2,1} \\ o_{2,2} \\ \dots \\ o_{2,k_2} \end{bmatrix} = \Phi(W^{(2)}\vec{o}^{(1)})$$

# Παράδειγμα παλινδρόμησης (regression)

Διάνυσμα εισόδου (π.χ. ενδείξεις αισθητήρων αυτοκινήτου)

Σωστές έξοδοι (π.χ. γωνία τιμονιού, γκάζι)



$$\vec{\delta}^{(1)} = \tanh(W^{(1)}\vec{x})$$

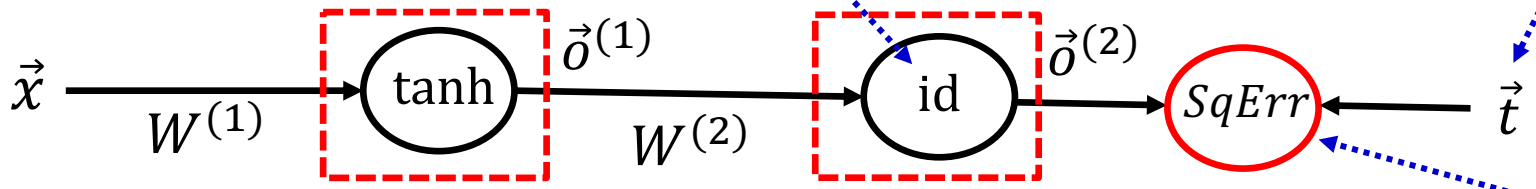
$$\vec{\delta}^{(2)} = W^{(2)}\vec{\delta}^{(1)}$$

Μέσο τετραγωνικό σφάλμα για το τρέχον διάνυσμα εισόδου

$$E = \frac{1}{k_2} \sum_{j=1}^{k_2} l_j^2 = \frac{1}{k_2} \sum_{j=1}^{k_2} (o_{2,j} - t_j)^2$$

# Παράδειγμα παλινδρόμησης – πιο συμπαγής συμβολισμός

Διάνυσμα εισόδου (π.χ. ενδείξεις αισθητήρων αυτοκινήτου)       $\Phi(s) = s$  (identity)      Σωστές έξοδοι (π.χ. γωνία τιμονιού, γκάζι)

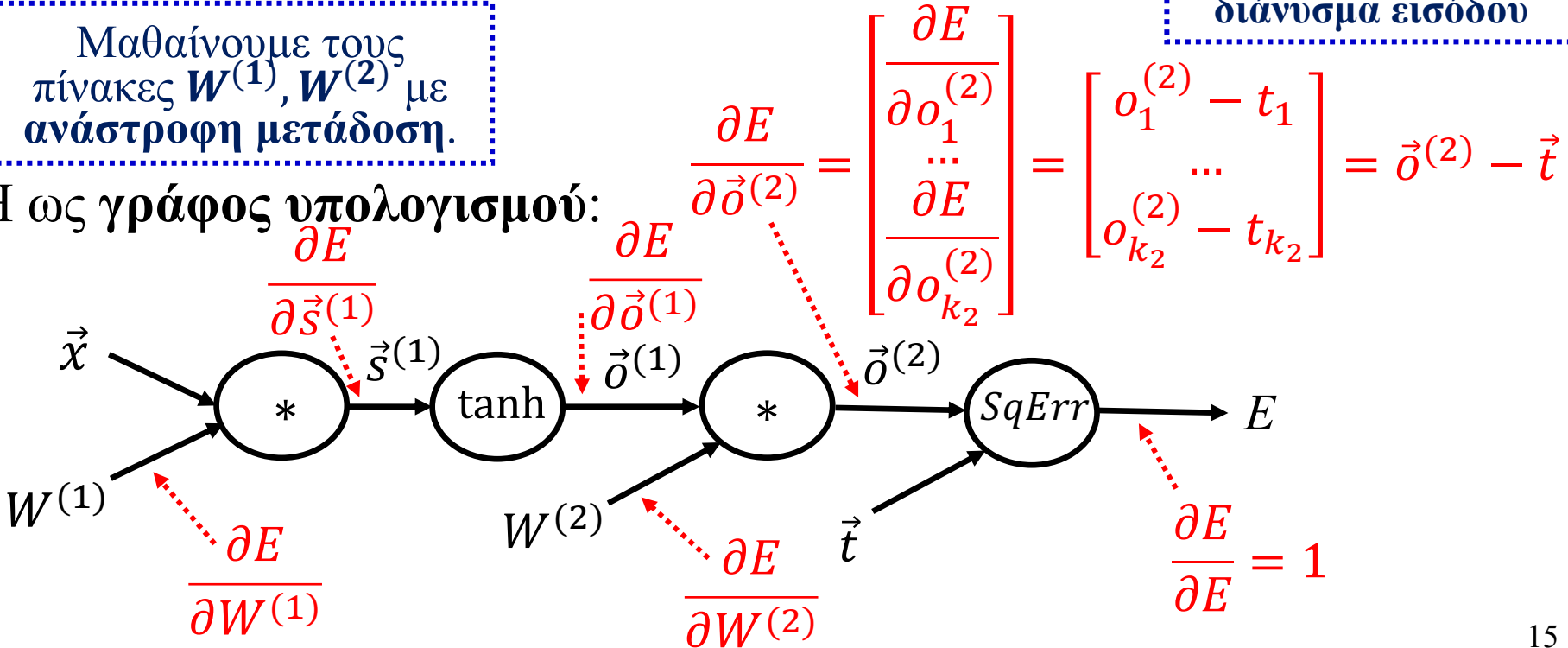


$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x}) \quad \vec{o}^{(2)} = W^{(2)}\vec{o}^{(1)}$$

Μέσο τετραγωνικό σφάλμα για το τρέχον διάνυσμα εισόδου

Μαθαίνουμε τους πίνακες  $W^{(1)}$ ,  $W^{(2)}$  με ανάστροφη μετάδοση.

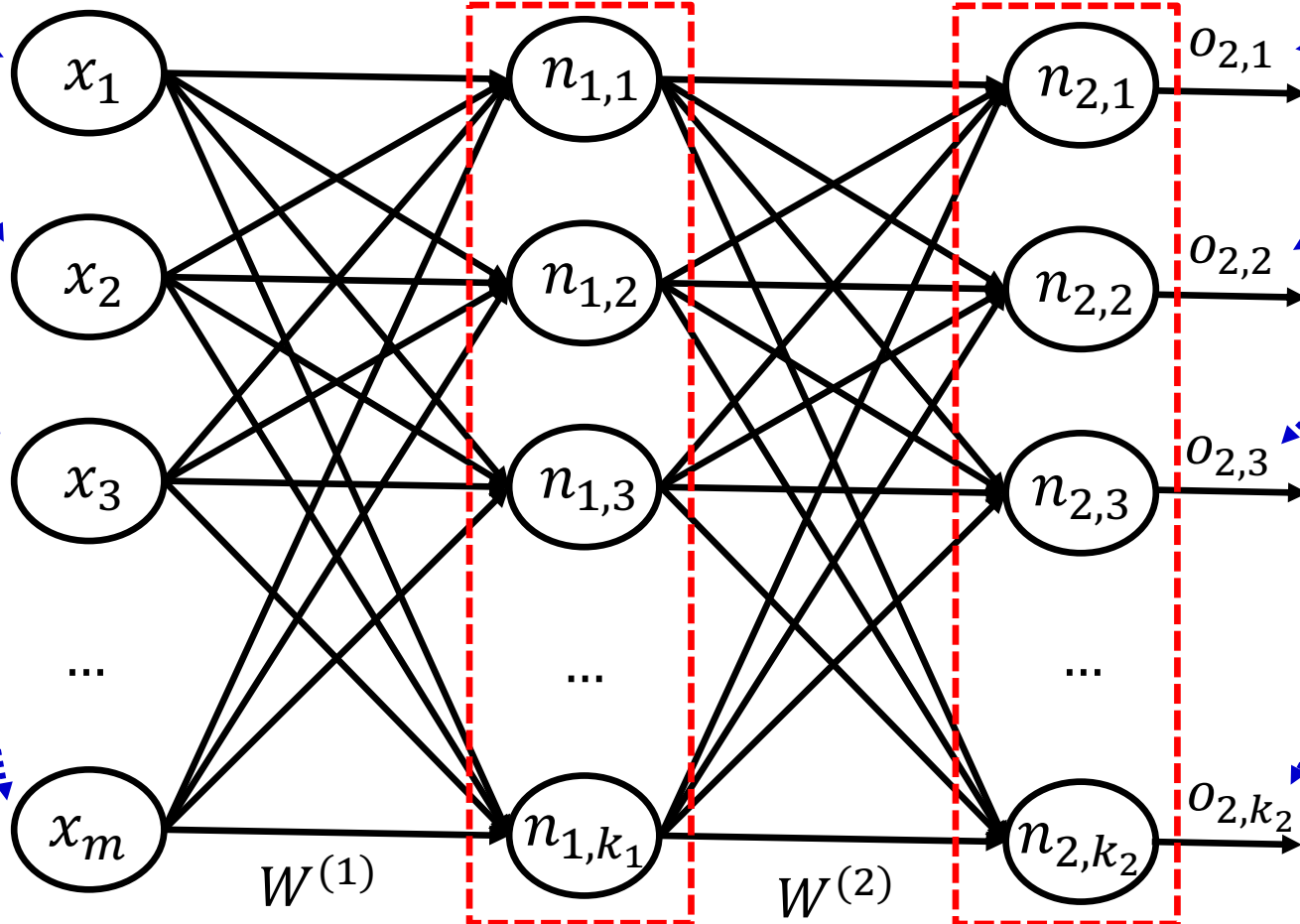
Ή ως γράφος υπολογισμού:



# Παράδειγμα κατηγοριοποίησης

Διάνυσμα εισόδου (π.χ. συχνότητες λέξεων, τιμές pixels)

Πόσο πιθανό θεωρεί το σύστημα να ανήκει η είσοδος σε κάθε μία από τις  $k_2$  κατηγορίες.



$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

$$\vec{o}^{(2)} = \text{softmax}(W^{(2)}\vec{o}^{(1)})$$

Θερούμε εδώ ότι κάθε αντικείμενο προς κατάταξη (π.χ. κείμενο, εικόνα) ανήκει σε **ακριβώς μία κατηγορία.**



# Softmax

$$W^{(2)}\vec{o}^{(1)} = \vec{s}^{(2)} = \begin{bmatrix} s_{2,1} \\ s_{2,2} \\ \dots \\ s_{2,k_2} \end{bmatrix}$$

Έξοδοι τελευταίου επιπέδου χωρίς συνάρτηση ενεργοποίησης. **Βαθμοί βεβαιότητας** (πραγματικοί αριθμοί) για κάθε κατηγορία. Θέλουμε να τους μετατρέψουμε σε **πιθανότητες με άθροισμα 1**.

$$\text{softmax}(W^{(2)}\vec{o}^{(1)}) = \text{softmax}(\vec{s}^{(2)}) = \text{softmax}\left(\begin{bmatrix} s_{2,1} \\ s_{2,2} \\ \dots \\ s_{2,k_2} \end{bmatrix}\right)$$

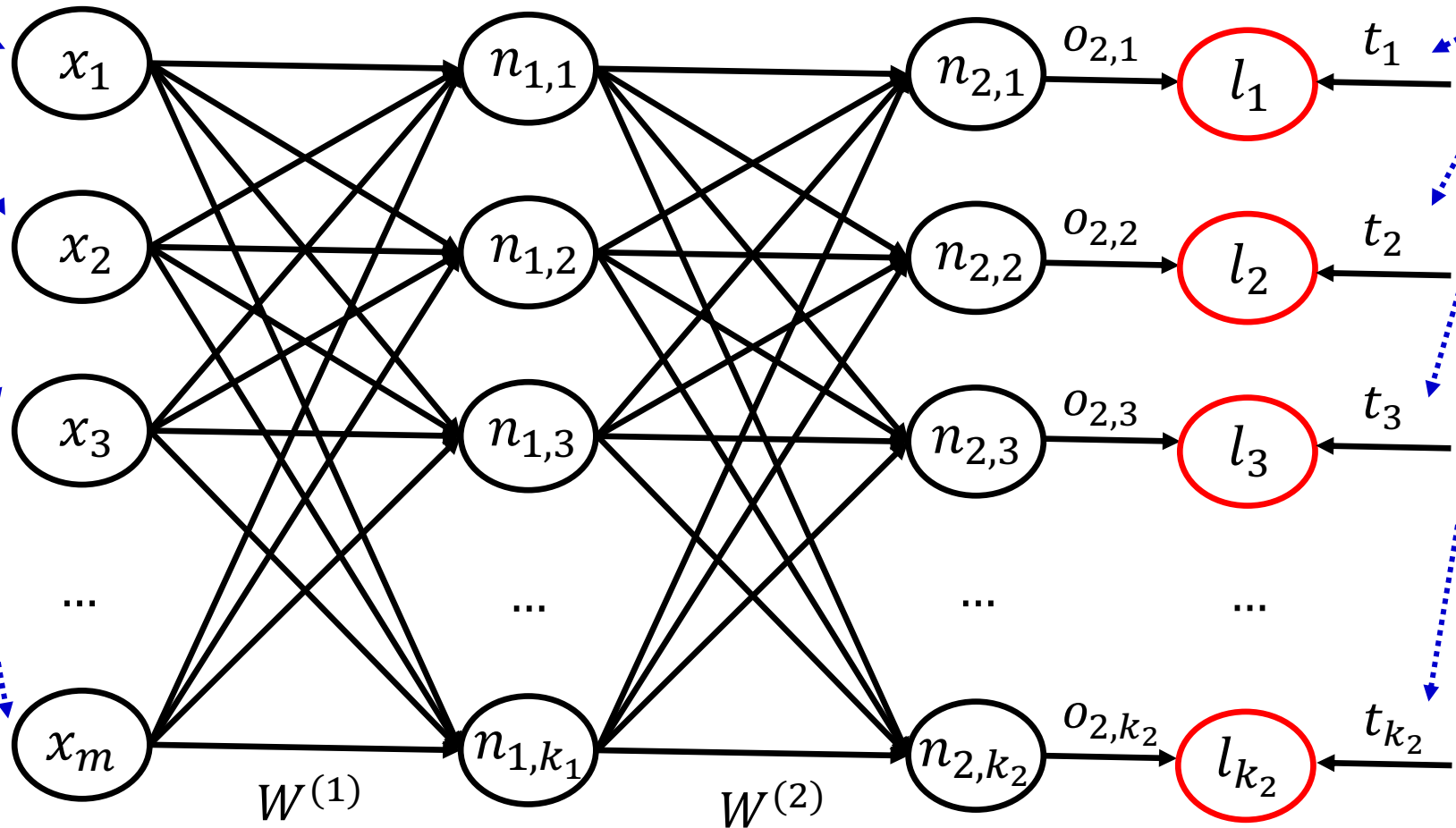
$$= \begin{bmatrix} \frac{\exp(s_{2,1})}{\sum_{j=1}^{k_2} \exp(s_{2,j})} \\ \frac{\exp(s_{2,2})}{\sum_{j=1}^{k_2} \exp(s_{2,j})} \\ \dots \\ \frac{\exp(s_{2,k_2})}{\sum_{j=1}^{k_2} \exp(s_{2,j})} \end{bmatrix}$$

Η softmax μετακινεί επίσης τις **μεγαλύτερες** από τις εισόδους της **προς το 1**, ενώ τις υπόλοιπες προς το **0**.  
Διαισθητικά **soft argmax!**

# Single-label multi-class classification example

Input instance (e.g., word frequencies or pixel values)

Correct output ( $t_j = 1$  means the single correct class is the  $j$ -th one)



Cross entropy loss at the current training instance. See slides below.

$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

$$\vec{o}^{(2)} = \text{softmax}(W^{(2)}\vec{o}^{(1)})$$

$$l = - \sum_{j=1}^{k_2} t_j \log(o_{2,j})$$

# Διασταυρωμένη εντροπία

$$\vec{o}^{(2)} = \begin{bmatrix} P_m(C = c_1) \\ P_m(C = c_2) \\ P_m(C = c_3) \\ \dots \\ P_m(C = c_k) \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.12 \\ 0.08 \\ \dots \\ 0.14 \end{bmatrix}$$

Οι εκτιμήσεις πιθανοτήτων του ταξινομητή για τις κατηγορίες.

Οι σωστές «πιθανότητες» για τις κατηγορίες. Διάνυσμα **1-hot**.

$$\vec{t} = \begin{bmatrix} P(C = c_1) \\ P(C = c_2) \\ P(C = c_3) \\ \dots \\ P(C = c_k) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

Η λογαριθμική πιθανοφάνεια της σωστής κατηγορίας σύμφωνα με τον ταξινομητή (αλλά με αρνητικό πρόσημο).

$$H_{P_m}(C) = - \sum_{i=1}^k P(C = c_i) \log_2 P_m(C = c_i) = - \log_2 P_m(C = c_2)$$

Ελαχιστοποιούμε τη διασταυρωμένη εντροπία ή ισοδύναμα μεγιστοποιούμε τη λογαριθμική πιθανοφάνεια (την πιθανότητα που δίνει το μοντέλο στη σωστή απάντηση).

# Διασταυρωμένη εντροπία (cross-entropy)

- Η εντροπία μιας τυχαίας μεταβλητής  $C$  δείχνει πόσο αβέβαιοι είμαστε για την τιμή της.

$$H(C) = - \sum_{c_i} P(C = c_i) \cdot \log_2 P(C = c_i)$$

- Πόσα bits (αναμενόμενη τιμή) χρειάζεται να μεταδώσουμε με ιδανική κωδικοποίηση για να μεταδώσουμε την τιμή της.
- Χρησιμοποιούμε  $-\log_2 P(c_i)$  bits για κάθε δυνατή τιμή  $c_i$ .
- Αν χρησιμοποιούμε κωδικοποίηση βασισμένη σε ανακριβείς εκτιμήσεις πιθανοτήτων  $P_m(c_i)$ :

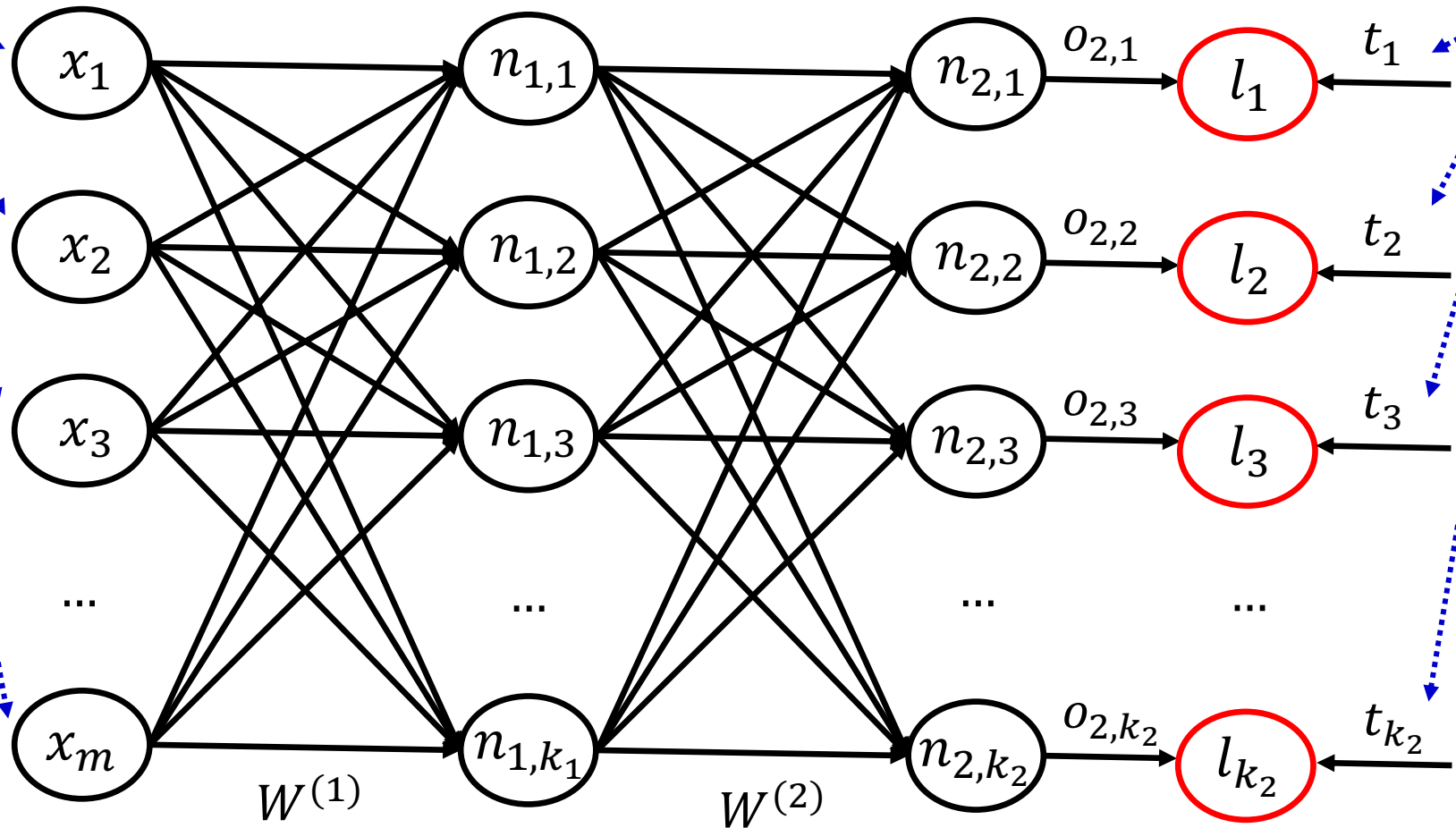
$$H_{P_m}(C) = - \sum_{c_i} P(C = c_i) \cdot \log_2 P_m(C = c_i) \geq H(C)$$

- Η διασταυρωμένη εντροπία μάς λέει πόσα bits μεταδίδουμε.
- Όσο πιο λανθασμένες είναι οι εκτιμήσεις  $P_m(c_i)$ , τόσο πιο πολλά bits μεταδίδουμε.

# Single-label multi-class classification example

Input instance (e.g., word frequencies or pixel values)

Correct output ( $t_j = 1$  means the single correct class is the  $j$ -th one)



Cross entropy loss at the current training instance.

$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

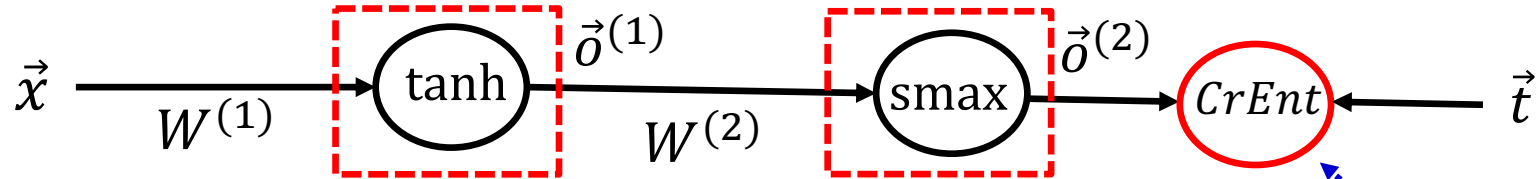
$$\vec{o}^{(2)} = \text{softmax}(W^{(2)}\vec{o}^{(1)})$$

$$l = - \sum_{j=1}^{k_2} t_j \log(o_{2,j})$$

# Single-label classification – more compact

**Input instance** (e.g., word frequencies or pixel values)

**Correct output** (correct class prediction, 1-hot vector)

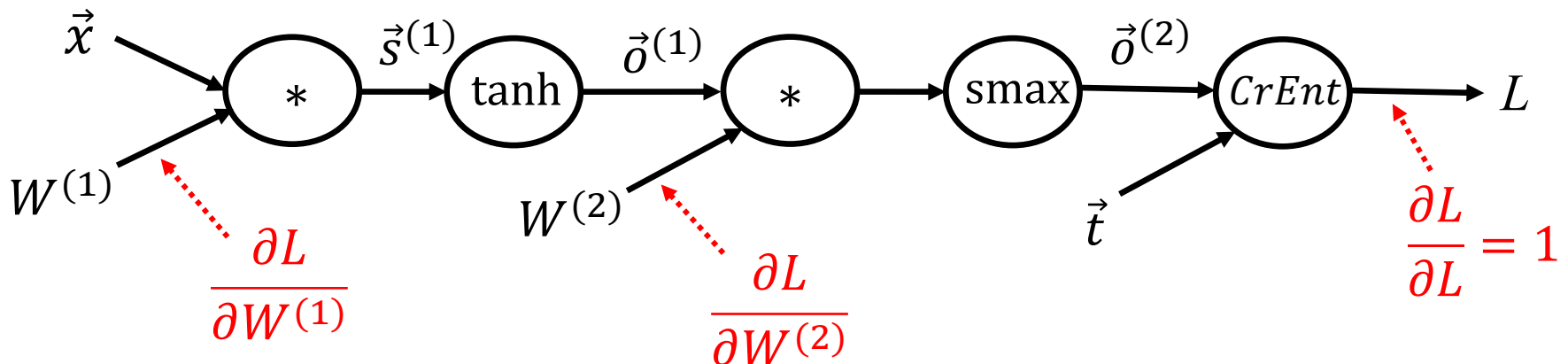


$$\vec{o}^{(1)} = \tanh(W^{(1)}\vec{x})$$

$$\vec{o}^{(2)} = \text{softmax}(W^{(2)}\vec{o}^{(1)})$$

**Cross-entropy loss** during training

Or as a **computation graph**:



# Βιβλιογραφία

- Russel & Norvig (4<sup>η</sup> έκδοση, ελληνική μετάφραση): ενότητα 21.4.
  - Την υπο-ενότητα 21.4.2 (batch normalization) θα την καλύψουμε στην επόμενη διάλεξη.
- Βλαχάβας κ.ά.: ίδιες ενότητες με την προηγούμενη διάλεξη.
- Δείτε και την πρόσθετη προτεινόμενη βιβλιογραφία της 19ης διάλεξης.