Τεχνητή Νοημοσύνη

20η διάλεξη (2024-25)

Ίων Ανδρουτσόπουλος

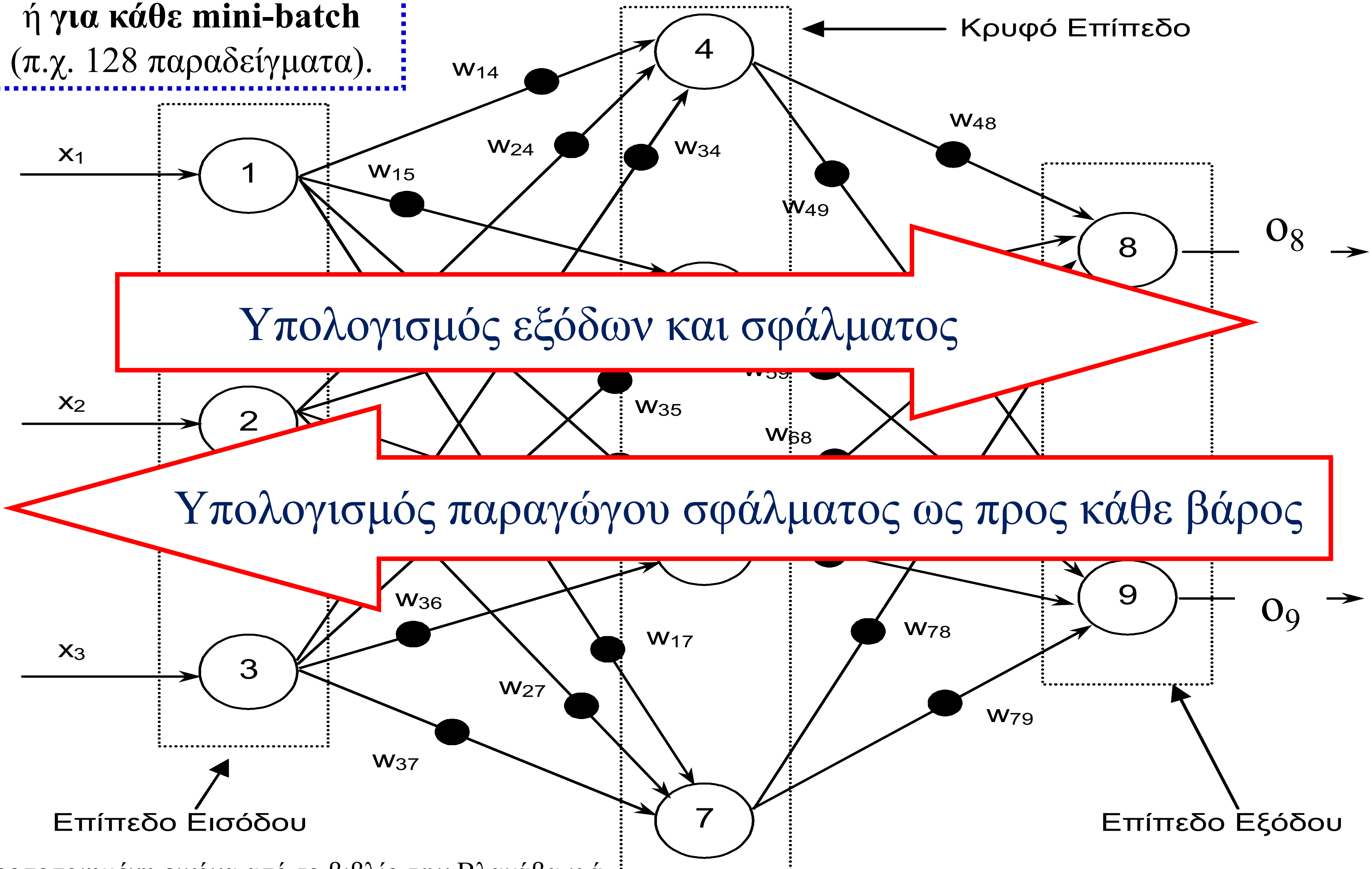
http://www.aueb.gr/users/ion/

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

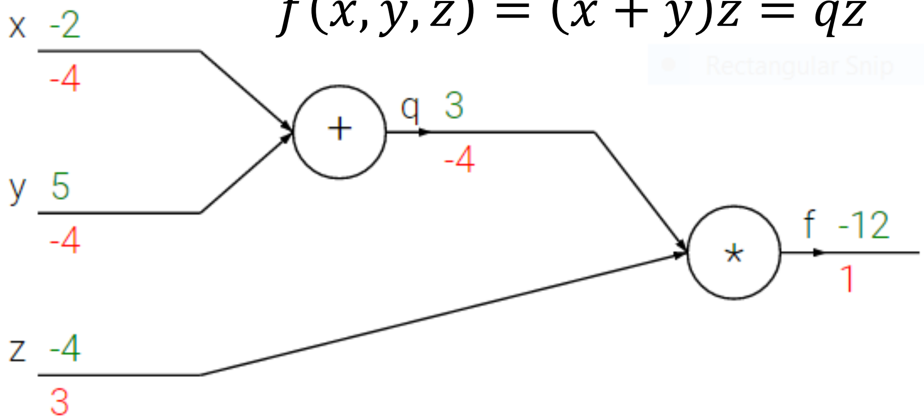
Για κάθε ένα παράδειγμα
ή για κάθε mini-batch
(π.χ. 128 παραδείγματα).



Τροποποιημένη εικόνα από το βιβλίο των Βλαχάβα κ.ά.

Παράδειγμα γράφου υπολογισμού

$$f(x,y,z) = (x+y)z = qz$$



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/

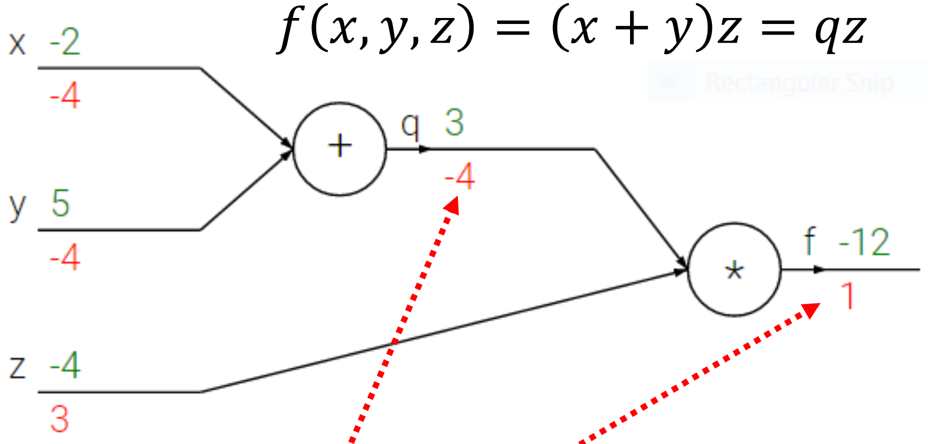
- **Προς τα εμπρός**: $\langle x, y, z\rangle = \langle -2, 5, -4\rangle$, $q = 3$, $f = -12$
- Ας υποθέσουμε ότι θέλουμε να **ελαχιστοποιήσουμε την *f*** χρησιμοποιώντας **στοχαστική κατάβαση κλίσης** (SGD).
  - Σε ένα πιο ρεαλιστικό σενάριο, η ***f*** θα ήταν μια **συνάρτηση σφάλματος** και το $\langle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

$$f(x, y, z) = (x + y)z = qz$$

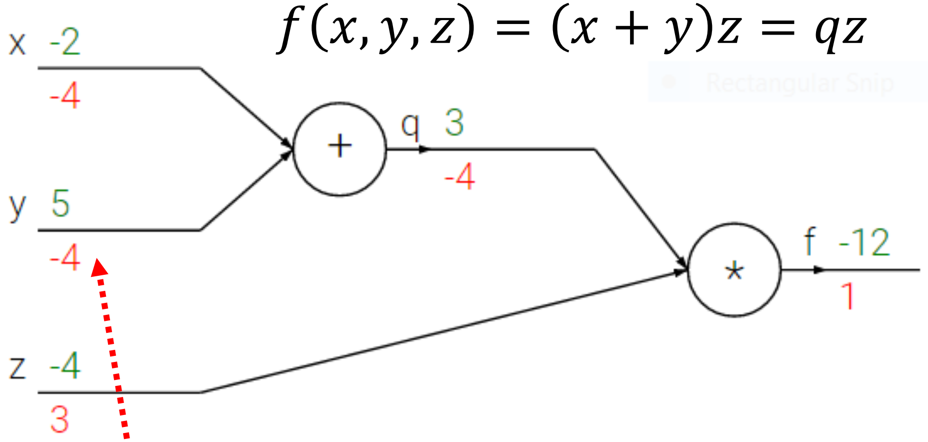


Παράδειγμα και σχήμα από το μάθημα "CNNs for Visual Recognition" (2016, F.-F. Li, A. Karpathy, J. Johnson) του Πανεπιστημίου Stanford. http://cs231n.github.io/optimization-2/

- **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

$f(x, y, z) = (x + y)z = qz$



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 $\frac{\partial f}{\partial q} = -4$.

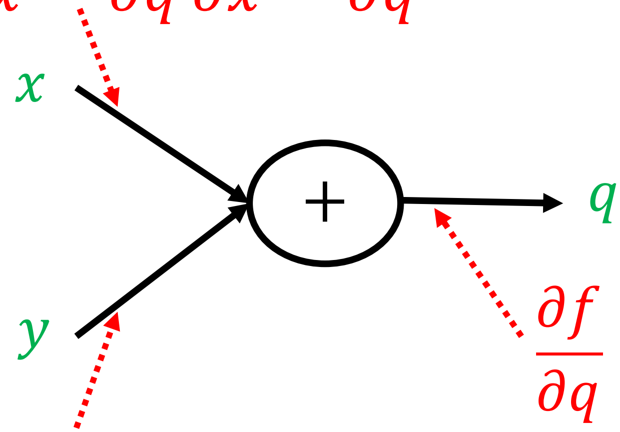
εισερχόμενη παράγωγος

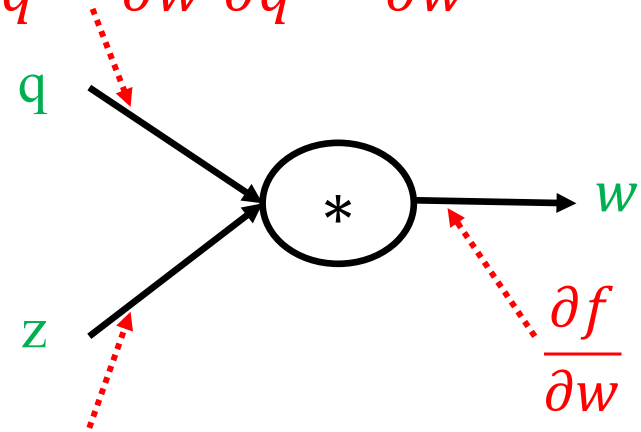
τοπική παράγωγος της πύλης +

Υλοποιήσεις πυλών που μπορούμε να συνδυάσουμε

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

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





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

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

εισερχόμενη από δεξιά παράγωγος

τοπική παράγωγος (μέρος της υλοποίησης)

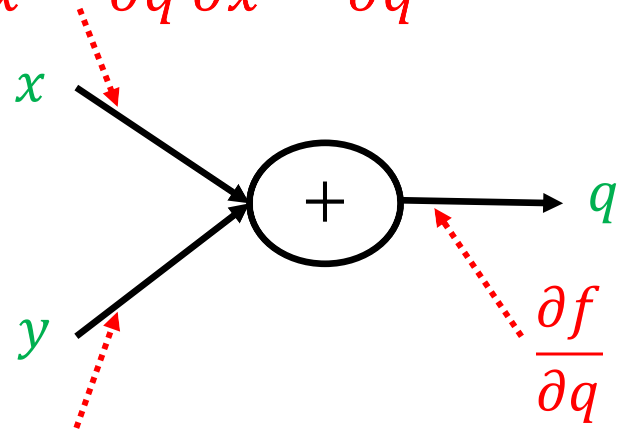
εισερχόμενη από δεξιά παράγωγος

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**.

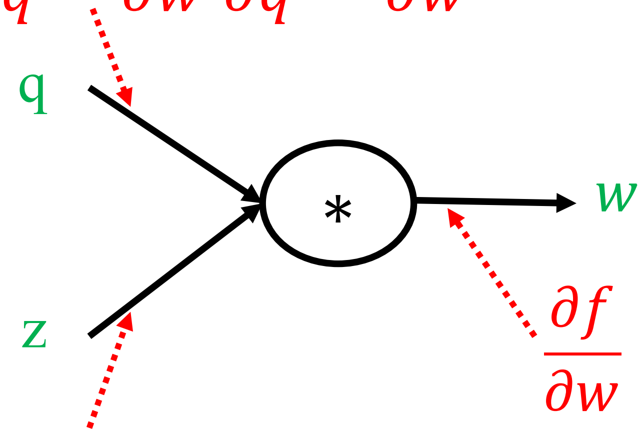
Plug-and-play gates

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



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

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



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

class PlusGate:

forward(x, y):

return x+y

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

return $<\frac{\partial f}{\partial q}, \frac{\partial f}{\partial q}>$

class StarGate:

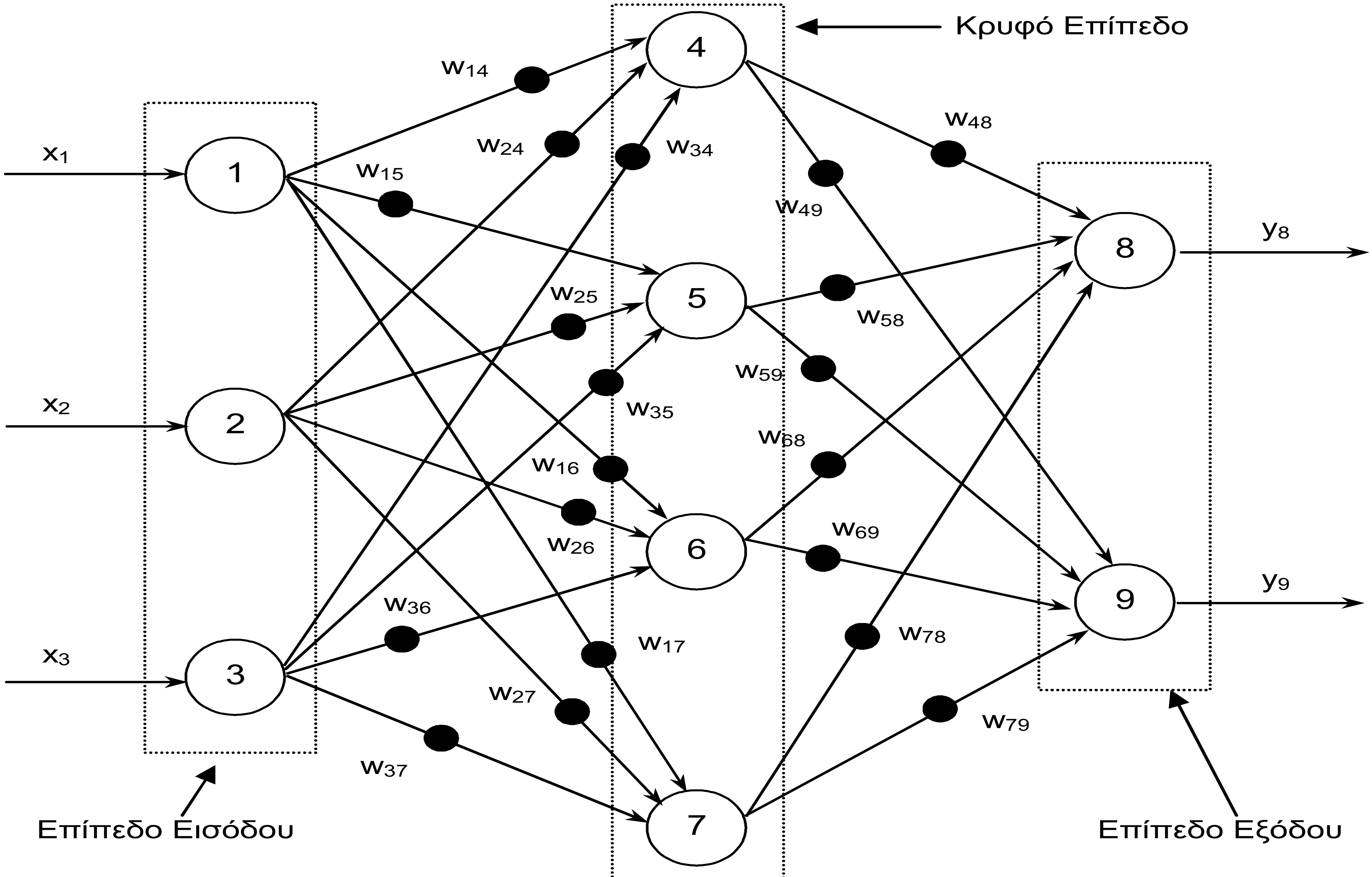
forward(q, z):

return q * z

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

return $<\frac{\partial f}{\partial w} \cdot z, \frac{\partial f}{\partial w} \cdot q>$

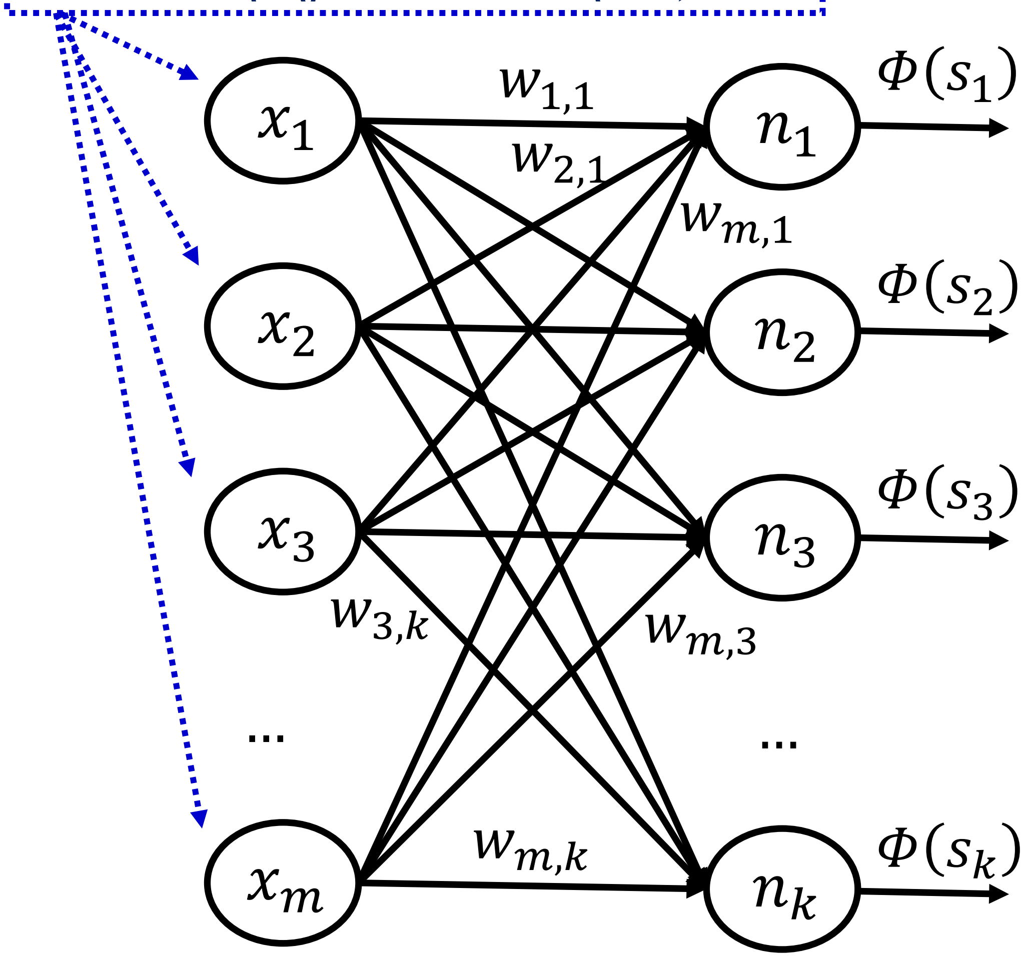
Πολυ-επίπεδο Perceptron (MLP)



Εικόνα από το βιβλίο των Βλαχάβα κ.ά.

Πιο συμπαγής συμβολισμός

Διάνυσμα εισόδου (π.χ. ενδείξεις αισθητήρων αυτοκινήτου)



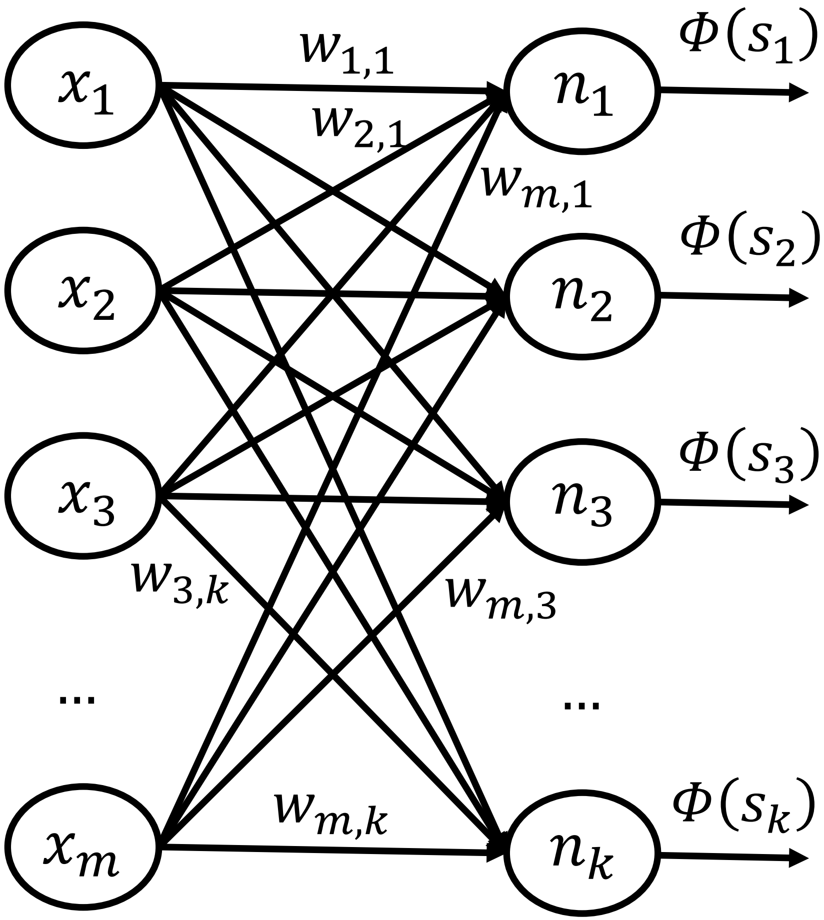
$$\Phi(s_1) = \Phi\left(\sum_{i=1}^{m} w_{i,1} x_i\right)$$

$$\Phi(s_2) = \Phi\left(\sum_{i=1}^{m} w_{i,2} x_i\right)$$

$$\Phi(s_3) = \Phi\left(\sum_{i=1}^{m} w_{i,3} x_i\right)$$

$$\Phi(s_k) = \Phi\left(\sum_{i=1}^{m} w_{i,k} x_i\right)$$

Πιο συμπαγής συμβολισμός



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

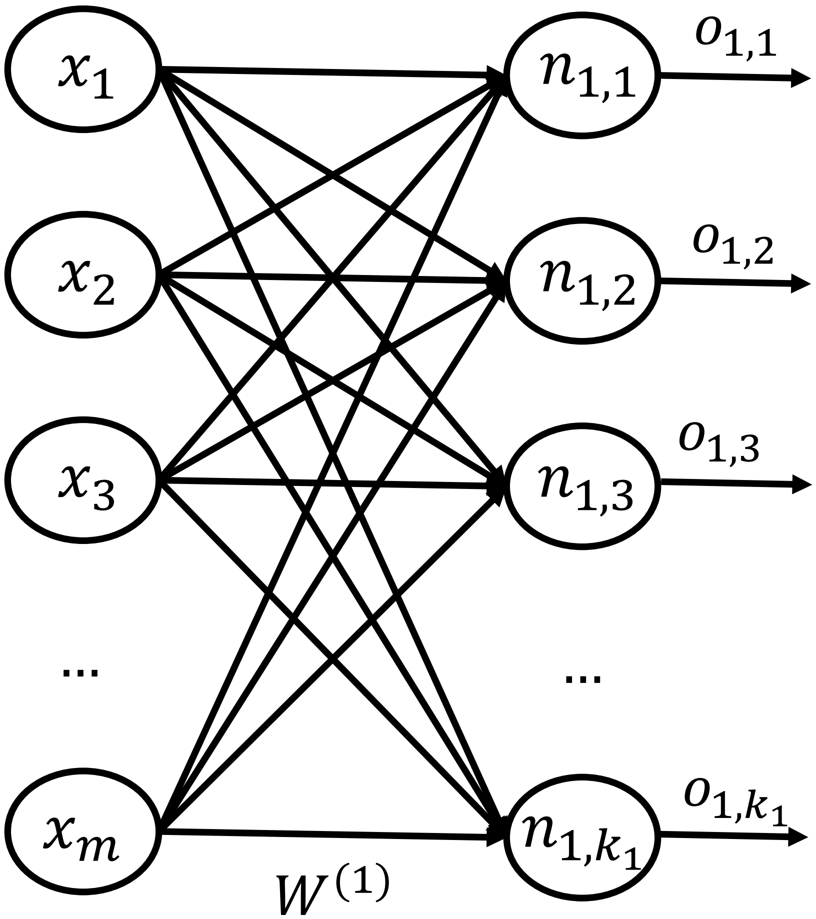
$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \dots \\ s_k \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{m,1} \\ w_{1,2} & w_{2,2} & \dots & w_{m,2} \\ w_{1,3} & w_{2,3} & \dots & w_{m,3} \\ \dots & \dots & \dots & \dots \\ w_{1,k} & w_{2,k} & \dots & w_{m,k} \end{bmatrix} \begin{bmatrix} x_1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

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

Μαθαίνουμε τον ***W*** με **ανάστροφη μετάδοση**.

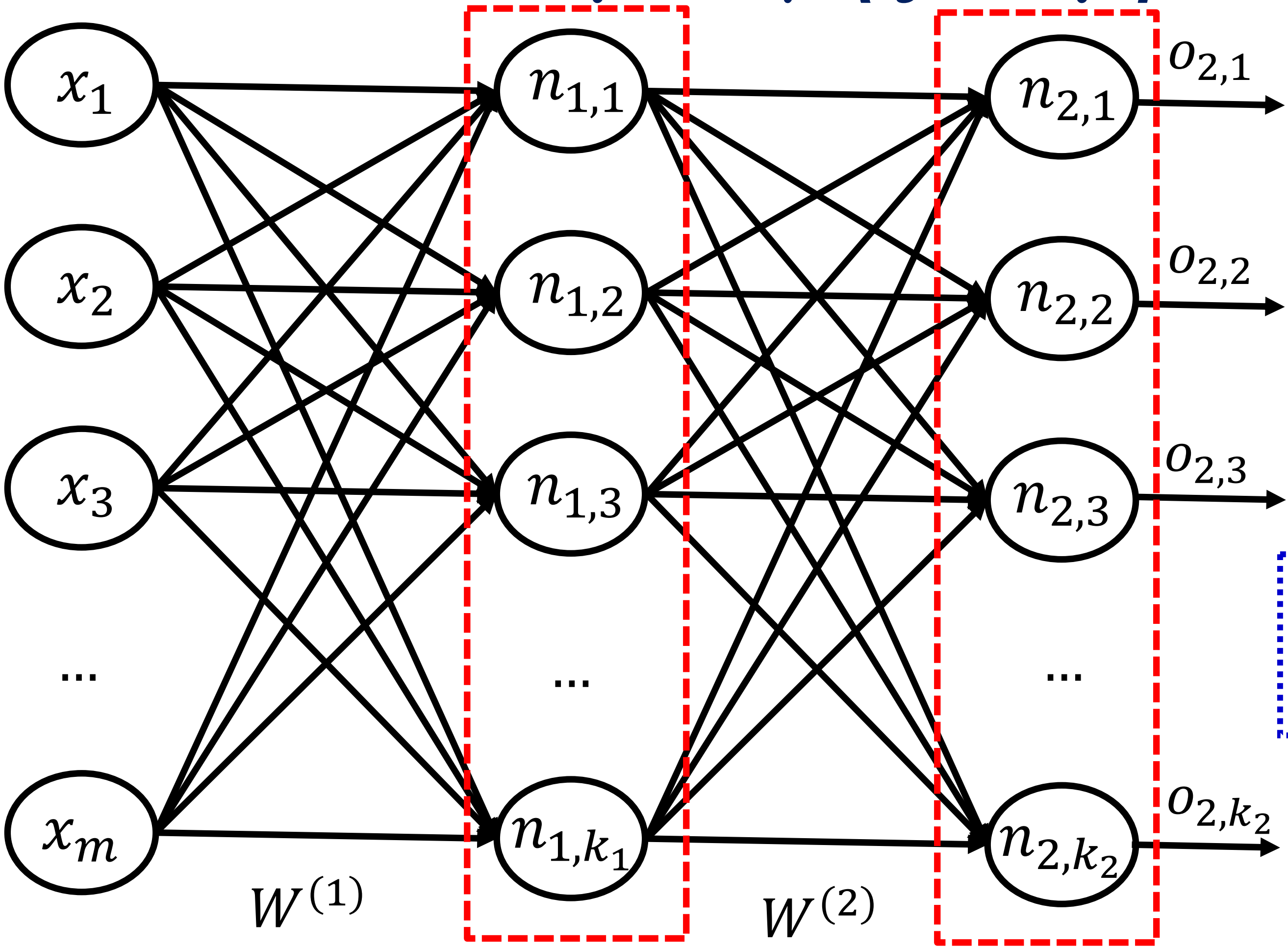
$$\vec{o} = \begin{bmatrix} o_1 \\ o_2 \\ o_3 \\ \dots \\ o_k \end{bmatrix} = \begin{bmatrix} \Phi(s_1) \\ \Phi(s_2) \\ \Phi(s_3) \\ \dots \\ \Phi(s_\kappa) \end{bmatrix} = \Phi(\vec{s}) = \Phi(W\vec{x})$$

Πιο συμπαγής συμβολισμός



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

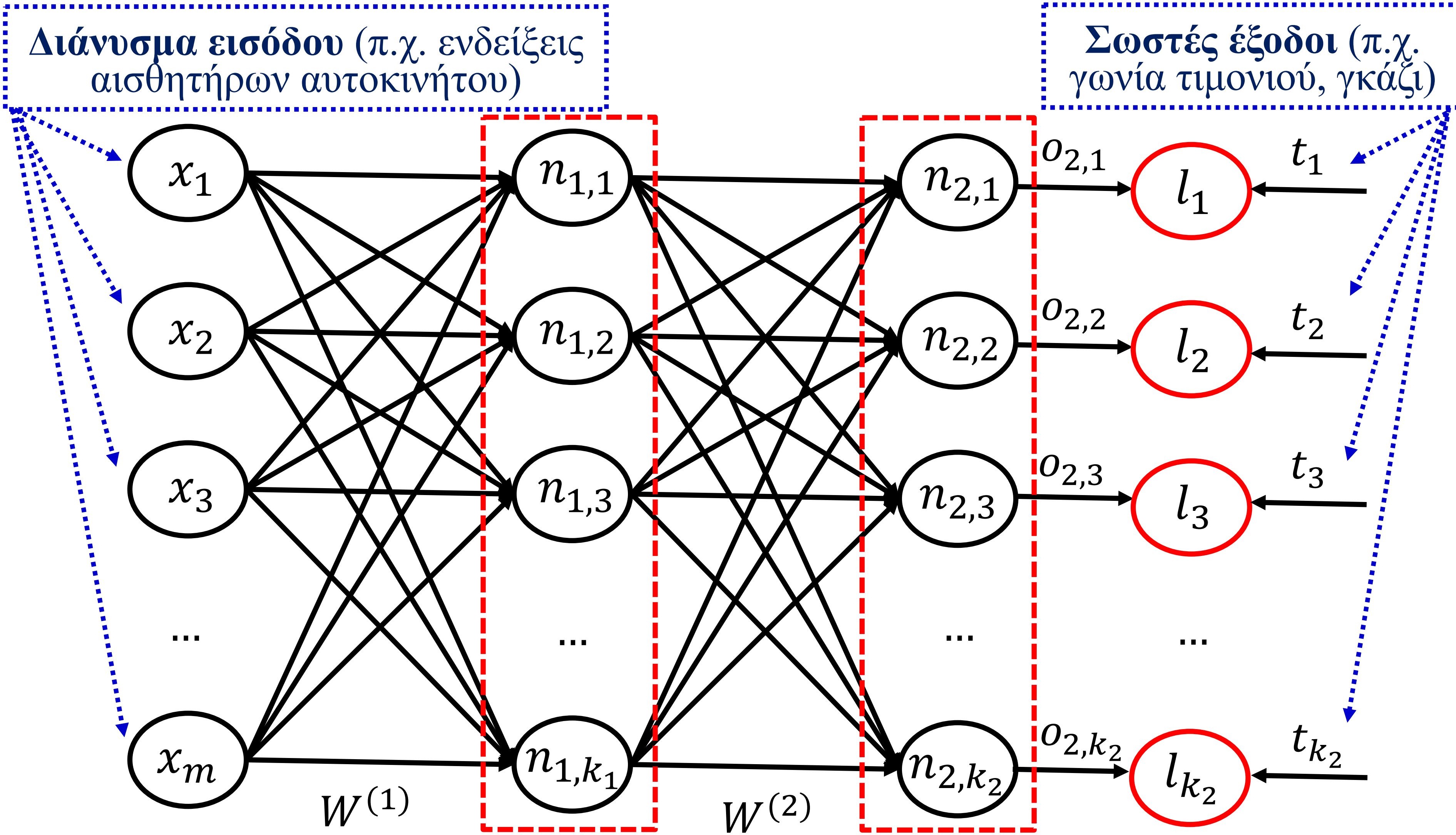
Πιο συμπαγής συμβολισμός



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

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

Παράδειγμα παλινδρόμησης (regression)



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

Μέσο τετραγωνικό σφάλμα για το **τρέχον διάνυσμα εισόδου**

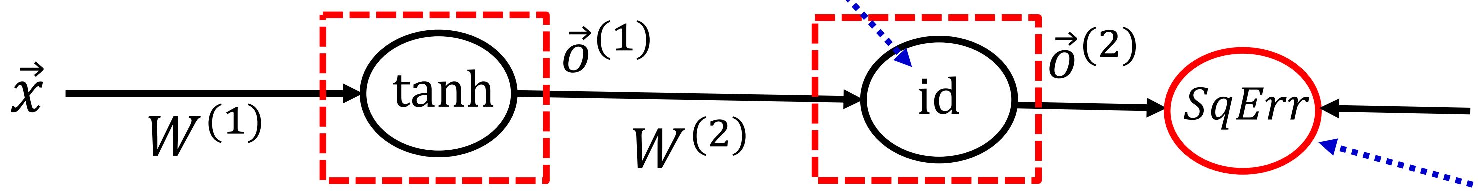
$$E = \frac{1}{k_2}\sum_{j=1}^{k_2} l_j^2 = \frac{1}{k_2}\sum_{j=1}^{k_2}\left(o_{2,j} - t_j\right)^2$$

Παράδειγμα παλινδρόμησης – πιο συμπαγής συμβολισμός

Διάνυσμα εισόδου (π.χ. ενδείξεις αισθητήρων αυτοκινήτου)

$\Phi(s) = s$ (identity)

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



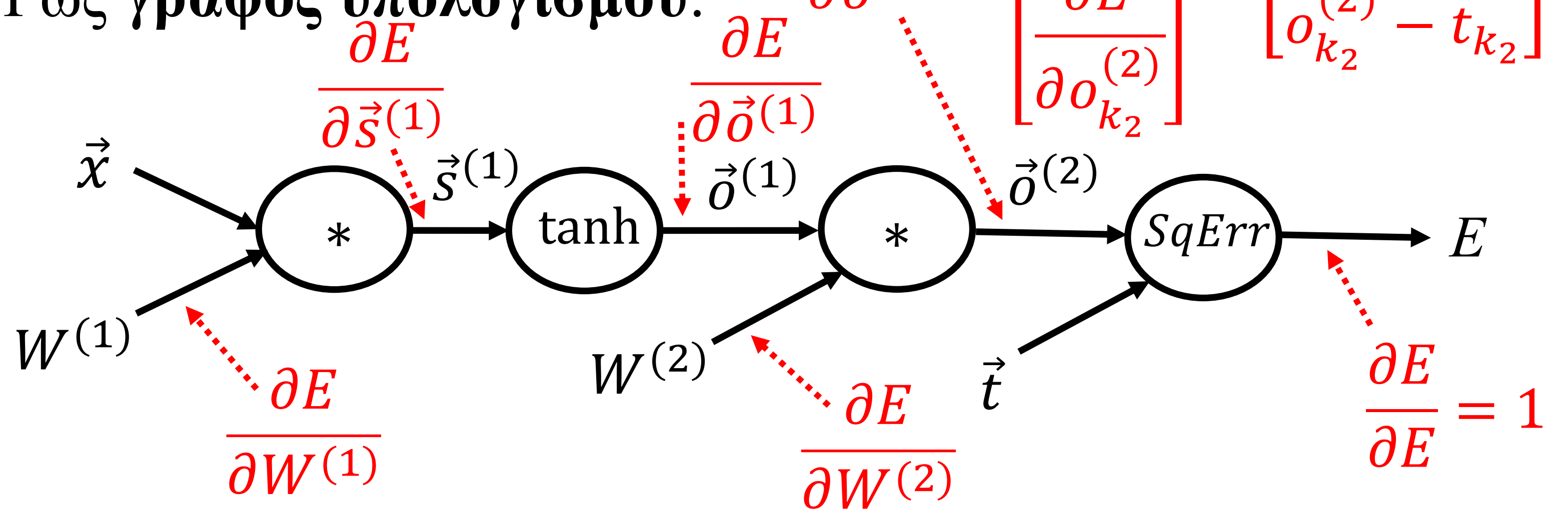
Μέσο τετραγωνικό σφάλμα για το **τρέχον διάνυσμα εισόδου**

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

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

$$\frac{\partial E}{\partial \vec{o}^{(2)}} = \begin{bmatrix} \dfrac{\partial E}{\partial o_1^{(2)}} \\ \cdots \\ \dfrac{\partial E}{\partial o_{k_2}^{(2)}} \end{bmatrix} = \begin{bmatrix} o_1^{(2)} - t_1 \\ \cdots \\ o_{k_2}^{(2)} - t_{k_2} \end{bmatrix} = \vec{o}^{(2)} - \vec{t}$$

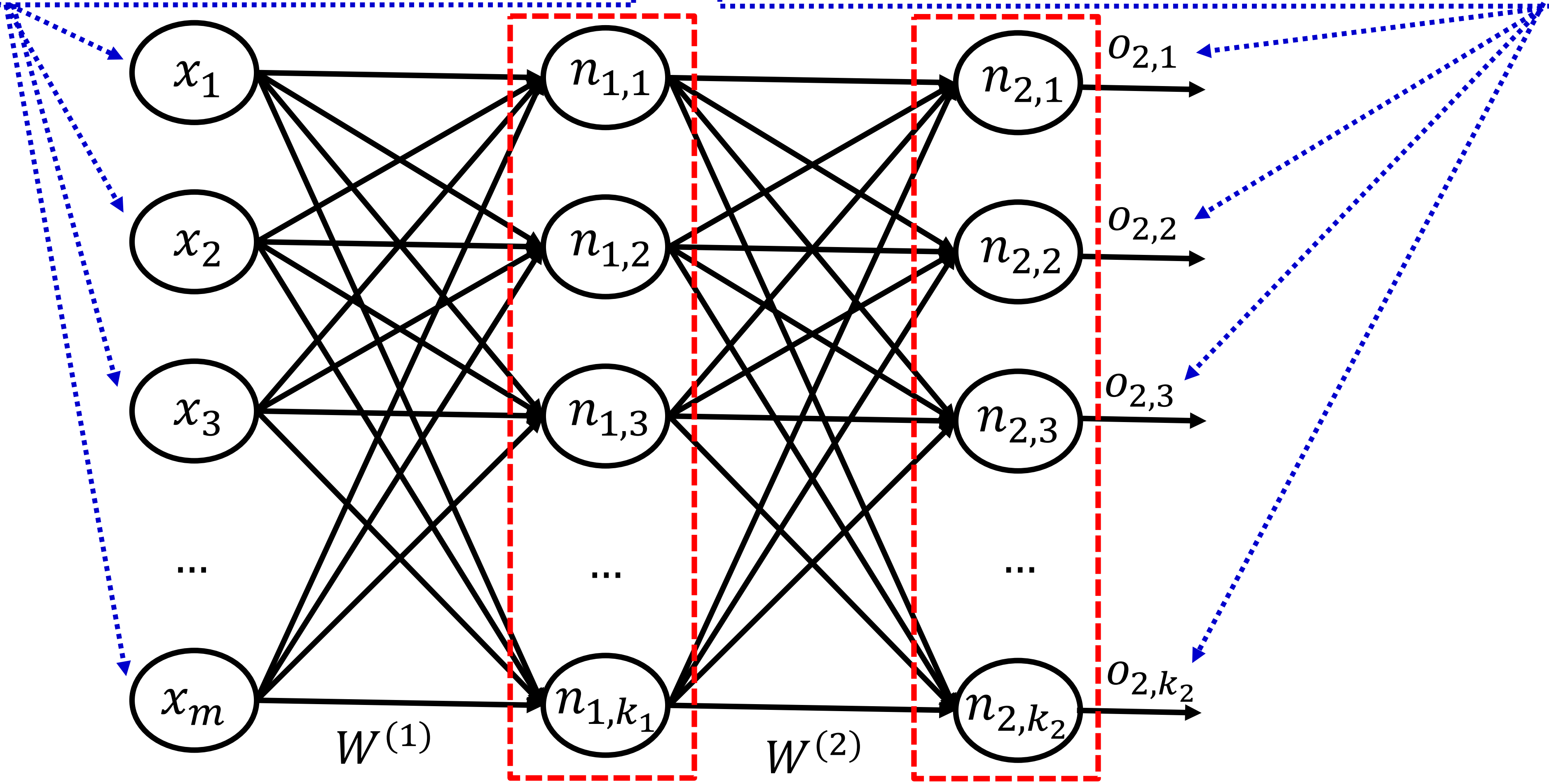
Ή ως **γράφος υπολογισμού**:



Παράδειγμα κατηγοριοποίησης

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

Πόσο πιθανό θεωρεί το σύστημα να ανήκει η είσοδος σε κάθε μία από τις $\boldsymbol{k_2}$ **κατηγορίες**.



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

$$\vec{o}^{(2)} = \mathrm{softmax}\left(W^{(2)}\vec{o}^{(1)}\right)$$

Θεωρούμε 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**.

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

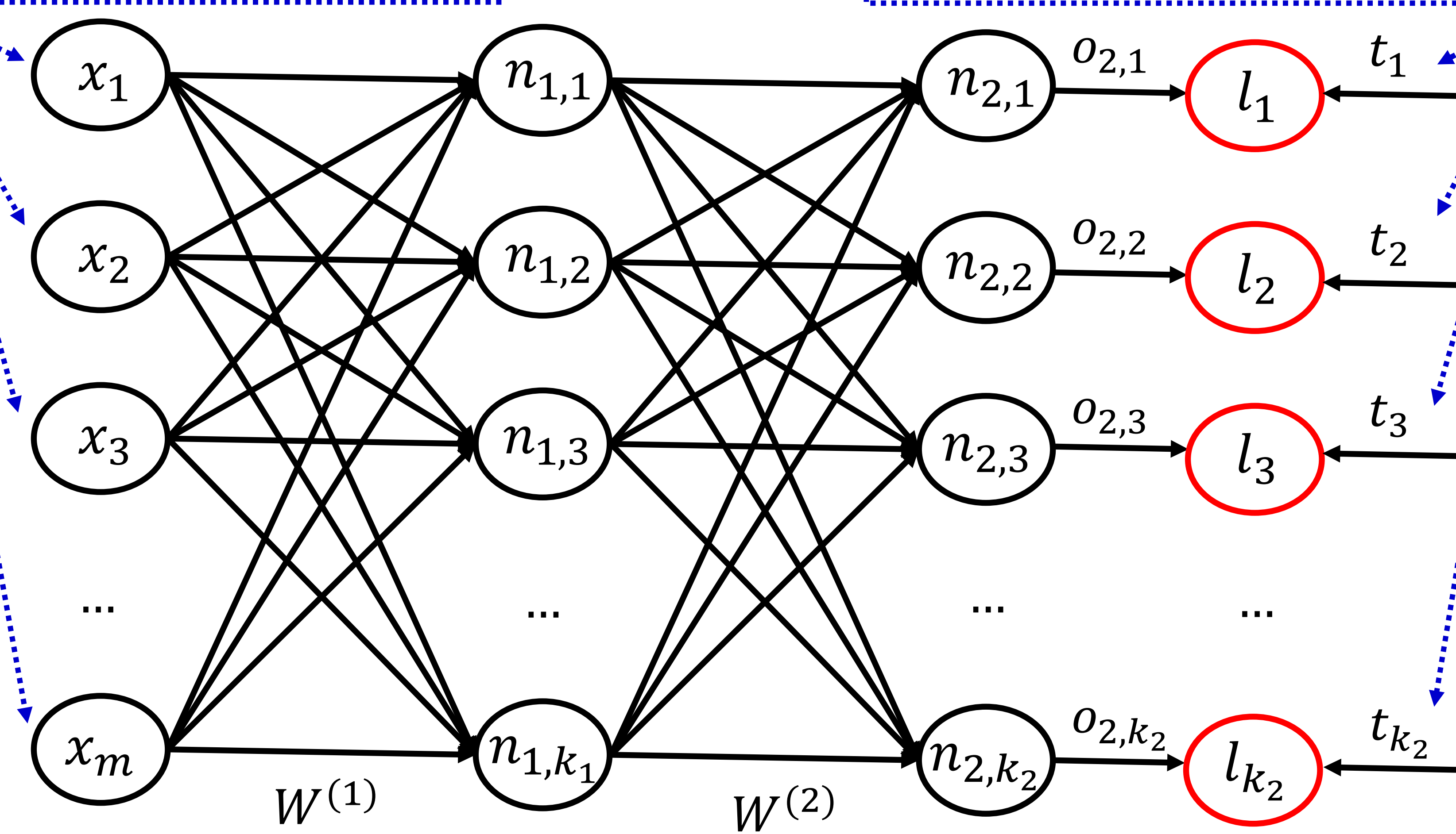
$$= \begin{bmatrix} \dfrac{\exp\left(s_{2,1}\right)}{\sum_{j=1}^{k_2} \exp\left(s_{2,j}\right)} \\ \dfrac{\exp\left(s_{2,2}\right)}{\sum_{j=1}^{k_2} \exp\left(s_{2,j}\right)} \\ \dots \\ \dfrac{\exp\left(s_{2,k_2}\right)}{\sum_{j=1}^{k_2} \exp\left(s_{2,j}\right)} \end{bmatrix}$$

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)



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

$$\vec{o}^{(2)} = \mathrm{softmax}\left(W^{(2)}\vec{o}^{(1)}\right)$$

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

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

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.

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

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

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** $P_m(c_i)$:

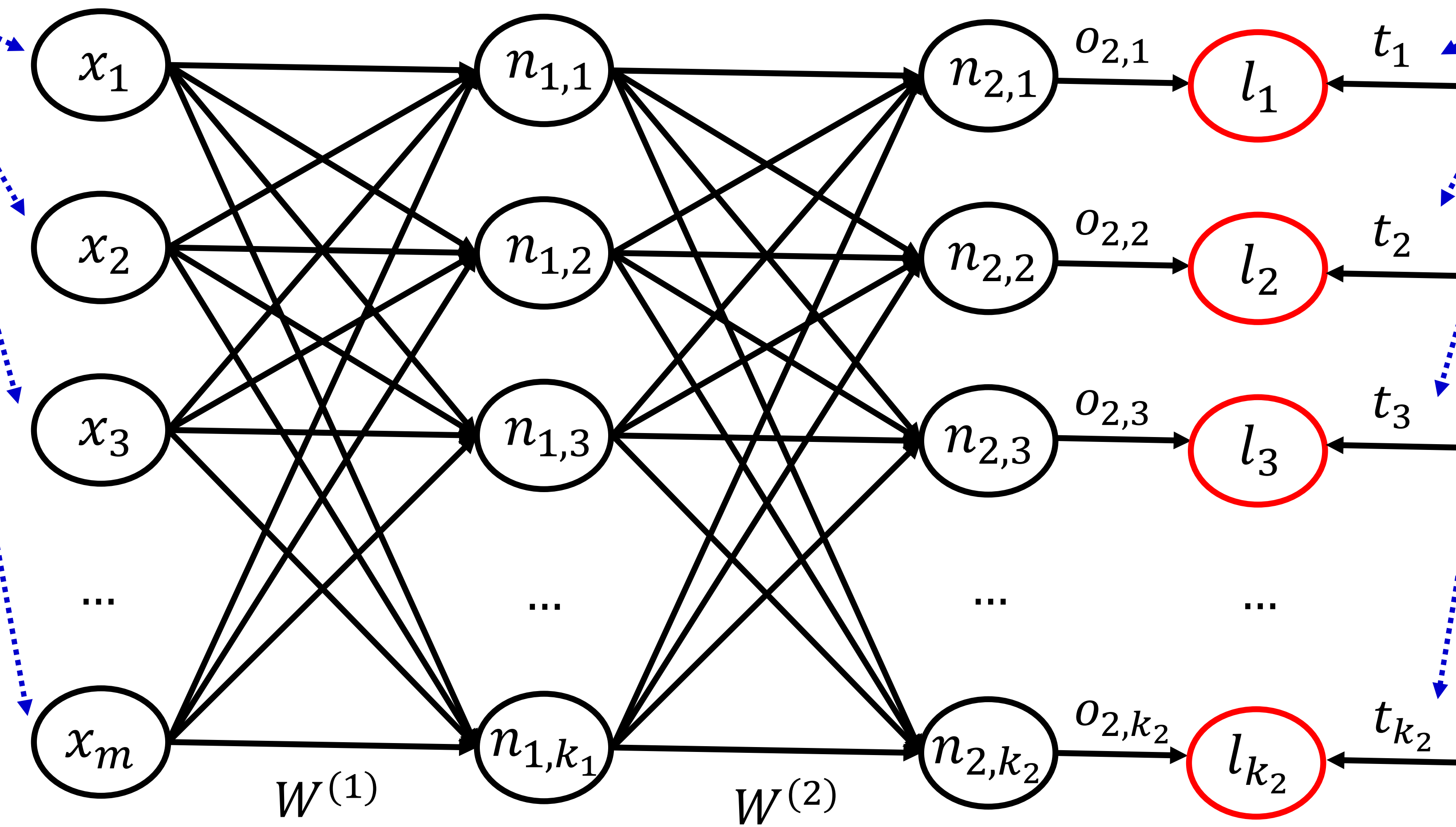
$$H_{P_m}(C) = -\sum_{c_i} P(C = c_i) \cdot \log_2 P_m(C = c_i) \geq H(C)$$

  - 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)



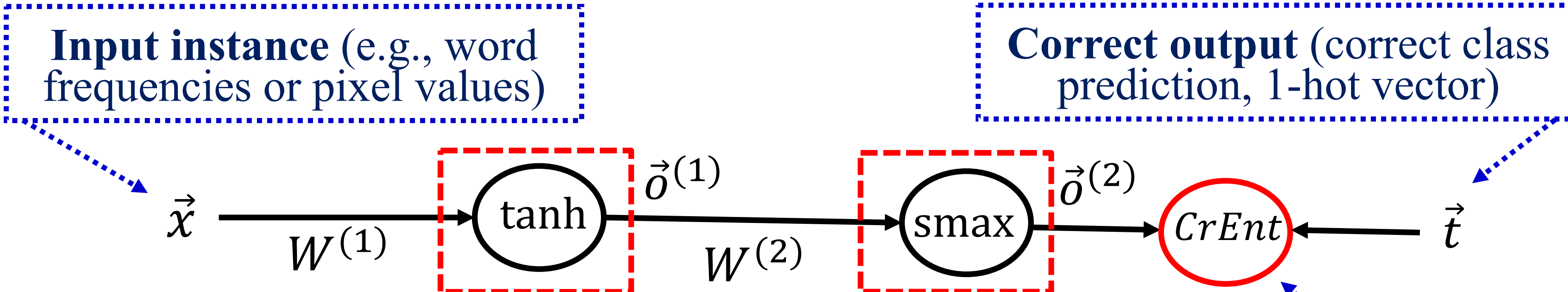
$$\vec{o}^{(1)} = \tanh\left(W^{(1)}\vec{x}\right)$$

$$\vec{o}^{(2)} = \mathrm{softmax}\left(W^{(2)}\vec{o}^{(1)}\right)$$

Cross entropy loss at the **current** training instance.

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

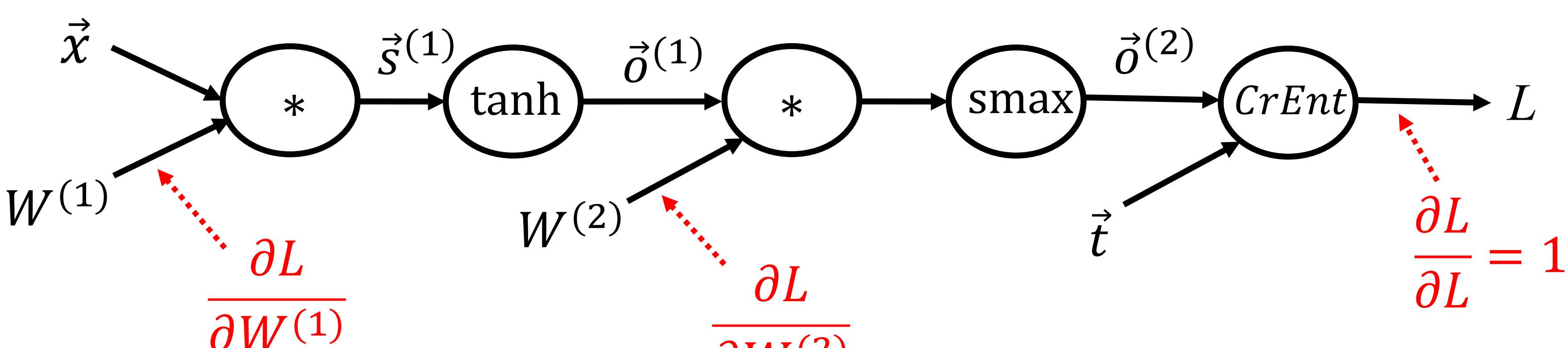
Single-label classification – more compact



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

$$\vec{o}^{(2)} = \mathrm{softmax}\left(W^{(2)}\vec{o}^{(1)}\right)$$

Or as a **computation graph**:



Βιβλιογραφία

- Russel & Norvig (4η 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.