

Νευρωνικά Δίκτυα 2016-17

1^η Εργαστηριακή Άσκηση

Μελέτη των πολυεπίπεδων Perceptrons και εφαρμογή σε προβλήματα ταξινόμησης εικόνας

Εργαστήριο Εικόνων, Βίντεο και Συστημάτων Πολυμέσων

Μέρος 1^ο

Εισαγωγή στη MATLAB

Πίνακες και MATLAB

- MATLAB = MATrix LABoratory
- Αποτελεί ισχυρό υπολογιστικό περιβάλλον και ταυτόχρονα γλώσσα προγραμματισμού που χειρίζεται με ευκολία πίνακες και πολύπλοκες πράξεις
- Βασική Δομική Μονάδα: Πίνακες (1-Δ, 2-Δ, 3-Δ,...)
- Κάθε μεταβλητή στη MATLAB αποθηκεύεται σαν ένας πίνακας 1x1
- Η MATLAB είναι case-sensitive, άρα $a \neq A$
- Το στοιχείο στη γραμμή i και στη στήλη j του πίνακα A συμβολίζεται με $A(i,j)$

Πίνακες και MATLAB

- Δημιουργία Πινάκων:
 - Ορισμός όλων των στοιχείων του πίνακα:
 - Αναλυτικά: $A=[1\ 2\ 3\ 4]$, $B=[1\ 2;3\ 4]$
 - Περιγραφικά: $C=[1:10]$, $D = [0:5:100]$
 - Φόρτωμα πίνακα από αρχείο:
 - `load myMatrix.mat`
 - Δημιουργία πίνακα χρησιμοποιώντας μια από τις έτοιμες συναρτήσεις της MATLAB:
 - $E=zeros(3,4)$, $F=ones(5,6)$, $G=eye(10)$
 - $H=rand(5,2)$
 - Δημιουργία πίνακα με χρήση δικής μας συνάρτησης:
 - $I=myFunction(E,F)$

Λήψη

- Έστω ο πίνακας $A=[1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9]$ τότε:
- $A([1\ 2], [1\ 3]) = [1\ 3; 4\ 6]$ κρατάμε τις 2 πρώτες γραμμές και τη 1^η και 3^η στήλη
- $A([1\ 2], :) = [1\ 2\ 3; 4\ 5\ 6]$ κρατάμε από τον πίνακα A μόνο τις 2 πρώτες γραμμές και όλες τις στήλες,
- $A(:, [1\ 3]) = [1\ 3; 4\ 6; 7\ 9]$ κρατάμε από τον πίνακα A όλες τις γραμμές και μόνο την 1^η και 3^η στήλη

Αλλαγή-Διαγραφή

- Αλλαγή

- $A(2,3)=40$ στην 2^η σειρά και 3^η στήλη παίρνει την τιμή 40.

$A = [1 \ 2 \ 3; 4 \ 5 \ 40; 7 \ 8 \ 9]$

- $A([1 \ 2], [1 \ 3])=[40 \ 41; 42 \ 43]$, τότε:

$A = [40 \ 2 \ 41; 42 \ 5 \ 43; 7 \ 8 \ 9]$

- $A=A(:, \text{randperm}(3))$ παίρνουμε (τυχαία) τον πίνακα $A=[2 \ 1 \ 3; 5 \ 4 \ 6; 8 \ 7 \ 9]$ (ανακατεύει τις 3 στήλες)

- Διαγραφή

- $A([2,3],:)=[]$: Διαγράφεται η 2^η και η 3^η σειρά του πίνακα A:

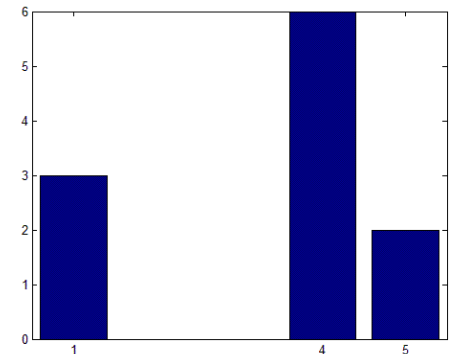
$A=[1 \ 2 \ 3]$

- $A(:,2)=[]$: Διαγράφεται η 2^η στήλη του πίνακα A:

$A=[1 \ 3; 4 \ 6; 7 \ 9]$

Χρήσιμες Συναρτήσεις

- Έστω ο πίνακας $A = \begin{bmatrix} 1 & 0 \\ 3 & 5 \end{bmatrix}$
- Πράξεις άθροισης:
 - άθροισμα στοιχείων ανά στήλη:
 $\text{sum}(A,1)=\text{sum}(A) = [4 \ 5]$
 - άθροισμα στοιχείων ανά γραμμή: $\text{sum}(A,2) = [1; 8]$
 - άθροισμα στοιχείων πίνακα:
 $\text{sum}(\text{sum}(A))=\text{sum}(A(:))=9$
- **bar**([1 4 5],[3 6 2]) Θα δημιουργήσει ένα διάγραμμα **bar** που στις τιμές 1, 4, 5 στον άξονα των x θα έχει ύψος 3, 6, 2 αντίστοιχα.



Χρήσιμες Συναρτήσεις

- Έστω ο πίνακας $A = \begin{bmatrix} 1 & 0 \\ 3 & 5 \end{bmatrix}$
- **find**: επιστρέφει δύο ορίσματα τα οποία περιέχουν τις σειρές και τις στήλες που έχουν τιμή διάφορη του 0

$[X, Y] = \text{find}(A),$
 $X = [1 \ 2 \ 2], \ Y = [1 \ 1 \ 2]$

$[X, Y] = \text{find}(A > 3)$
 $X = [2], \ Y = [2]$

Χρήσιμες πράξεις μεταξύ πινάκων

- έστω: $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ $B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$
- τότε: $A + B = \begin{pmatrix} 2 & 4 \\ 6 & 8 \end{pmatrix}$ $A - B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
 $A * B = \begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$ $A. * B = \begin{pmatrix} 1 & 4 \\ 9 & 16 \end{pmatrix}$
 $A. / B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ $A.^ B = \begin{pmatrix} 1 & 4 \\ 27 & 256 \end{pmatrix}$
 $A' = B' = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$

Χρήσιμες Πράξεις μεταξύ Πινάκων

- Πράξεις με συναρτήσεις:

$$A=[1 \ 2 \ 3] \rightarrow \sin(A) = (0.8415 \quad 0.9093 \quad 0.1411)$$

$$A=[1 \ 2 \ 3] \rightarrow \exp(A) = (2.7183 \quad 7.3891 \quad 20.0855)$$

- Λογικές πράξεις:

$$A = [1 \ 2 \ 1], B = [0 \ 3 \ 1] \rightarrow$$

$$A > B = (1 \quad 0 \quad 0), A \geq B = (1 \quad 0 \quad 1)$$

$$A < B = (0 \quad 1 \quad 0), A \leq B = (0 \quad 1 \quad 1)$$

$$A == B = (0 \quad 0 \quad 1)$$

- Πράξεις Μέγιστου/Ελάχιστου:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \rightarrow \max(A) = (3 \ 4), \max(\max(A)) = 4$$

Αλληλουχία Πινάκων

- έστω: $A = [1 \ 2 \ 3 \ 4]$
- τότε: $B = [A \ A] \rightarrow B = (1 \ 2 \ 3 \ 4 \ 1 \ 2 \ 3 \ 4)$

$$B = [A; A] \rightarrow B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

Έλεγχος Ροής

- **if** statement **else** statement **end**
- **if** statement **elseif** statement **elseif** ... **else** statement **end**
- **while** statement **end**
- **for** statement **end**
- **switch** switch_expression
 - case** case_expression(s)
statement(s)
 - case** case_expression(s)
statement(s)
 - ...
 - otherwise**
statement(s)**end**
- **break**
- Όχι σε άσκοπη χρήση **for** loops!!!

M-Files

- Δύο τρόποι για τη δημιουργία m-files:
 - File → New → Script
 - `edit file1.m` [σε command line]
- Ένα M-File μπορεί να είναι:
 - Μια συνάρτηση (function)
 - `function C = myFunction(A,B)`
`C = A + 2*B;`
 - Μια ακολουθία εντολών (script)
 - `A, B`
`C = A + 2*B`

Σχεδίαση Γραφικής Παράστασης με τη χρήση της συνάρτησης plot

- Η `plot(x,y)` δέχεται ως είσοδο δύο διανύσματα και εμφανίζει μια γραφική παράσταση του y σε συνάρτηση του x

Παράδειγμα

- Θέλουμε να σχεδιάσουμε το $\sin(x)$ για τιμές του x από 0 έως 2π :

```
t = 0:pi/100:2*pi;
```

```
y = sin(t);
```

```
plot(t,y)
```

Μέρος 2^ο

Multilayer Perceptrons

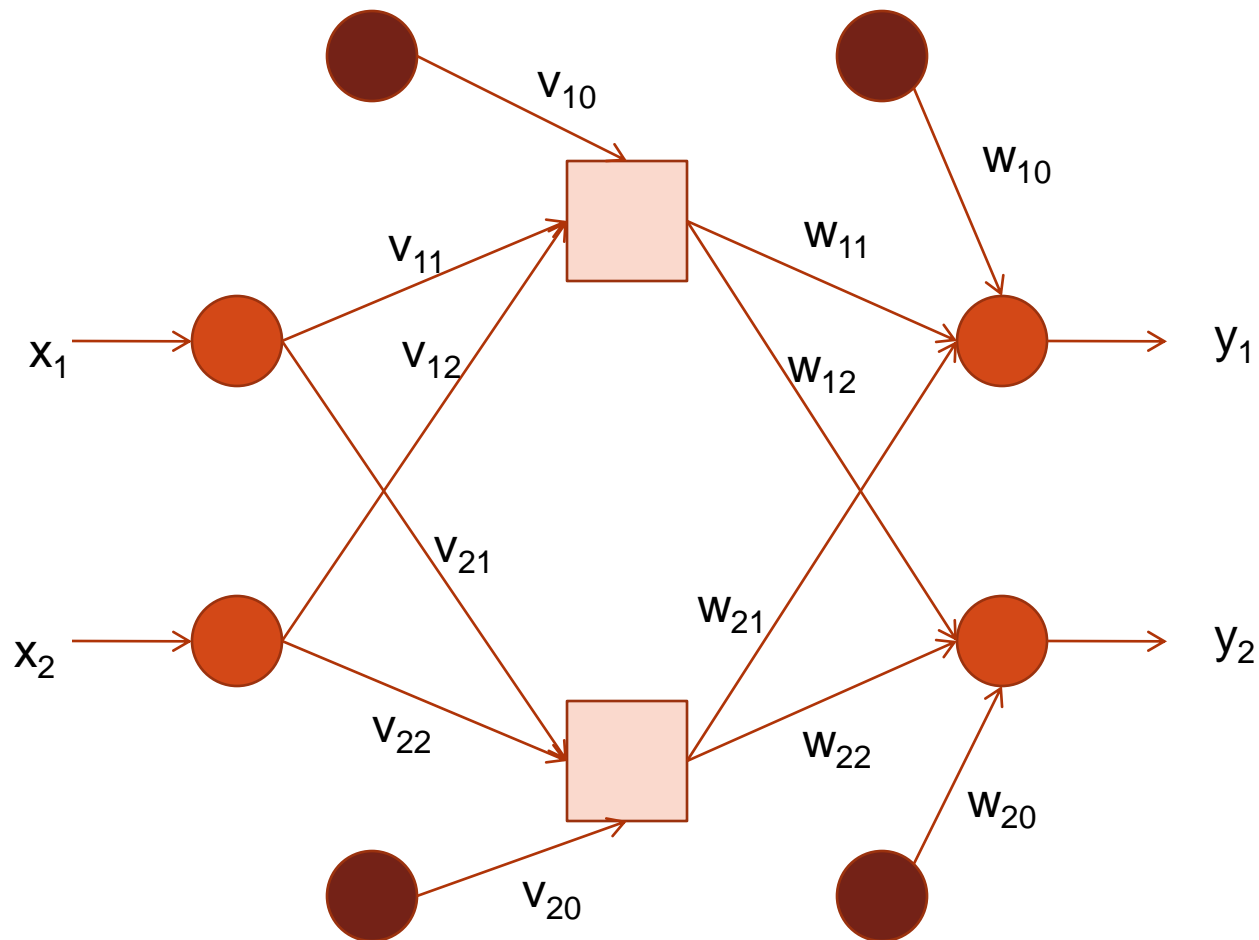
Πολυεπίπεδα Perceptrons (1)

- Γενικά:
 - Ένα από τα πιο συνηθισμένα Νευρωνικά Δίκτυα
 - Δίκτυο πρόσθιας τροφοδότησης (feed forward)
 - Γενίκευση του μονοστρωματικού perceptron
 - Επιβλεπόμενη(supervised) μάθηση
 - Αρχιτεκτονική:
 - 1 επίπεδο εισόδου
 - 1 ή περισσότερα κρυμμένα επίπεδα
 - 1 επίπεδο εξόδου
 - Εκπαίδευση με τον αλγόριθμο ανάστροφης διάδοσης σφάλματος (Back Propagation)

Πολυεπίπεδα Perceptrons (2)

- Ένα πολυεπίπεδο perceptron έχει τα εξής χαρακτηριστικά:
 - Κάθε κρυμμένος νευρώνας περιέχει μια ΜΗ-ΓΡΑΜΜΙΚΗ συνάρτηση ενεργοποίησης (activation function)
 - Η συνάρτηση αυτή είναι ΔΙΑΦΟΡΙΣΙΜΗ
 - Τα κρυμμένα επίπεδα προσδίδουν στο δίκτυο την δυνατότητα να «μάθει» πολύπλοκα πρότυπα

Παράδειγμα



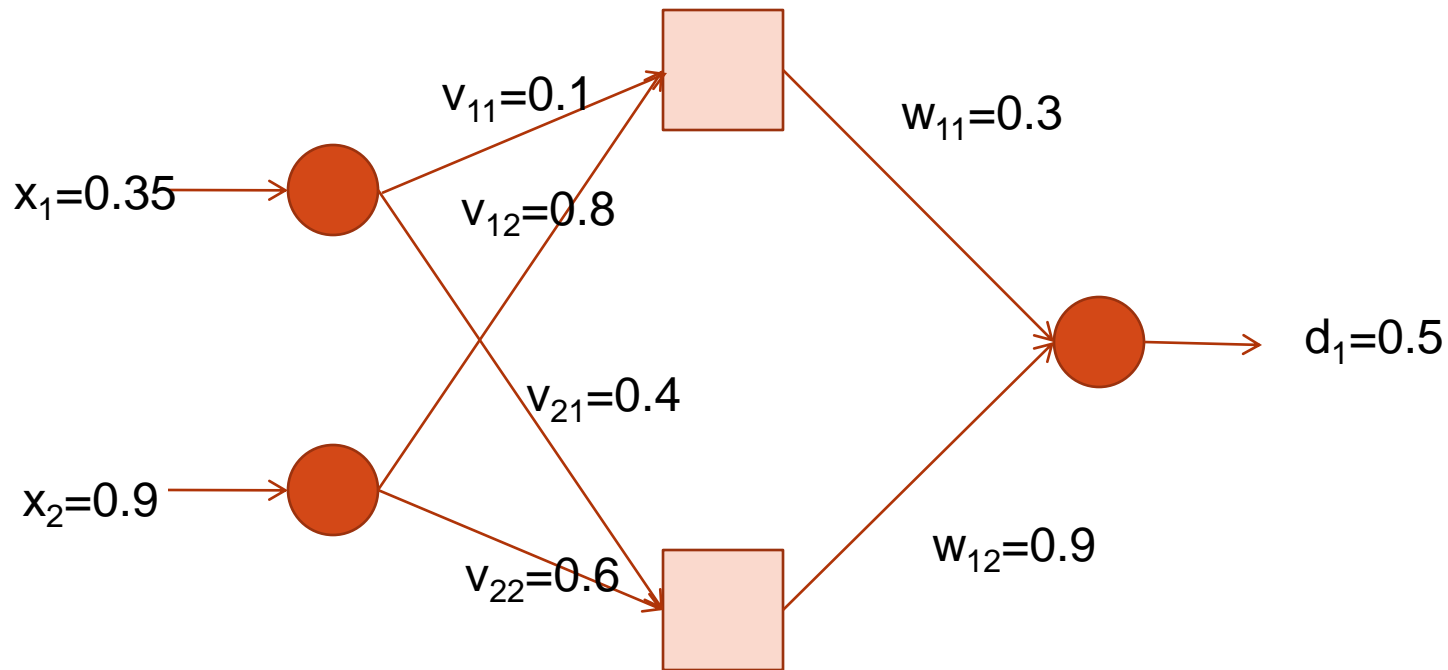
Back Propagation

- Αλγόριθμος Επιβλεπόμενης Μάθησης
- Διαδικασία Εκπαίδευσης
 - Αρχικοποιούμε τυχαία τα βάρη
 - Δίνουμε την είσοδο
 - Υπολογίζουμε την έξοδο
 - Προσαρμόζουμε τα συναπτικά βάρη ώστε να ελαχιστοποιηθεί το μέσο τετραγωνικό σφάλμα σε σχέση με την επιθυμητή έξοδο
- Τερματισμός Εκπαίδευσης
 - Μικρό σφάλμα
 - Αριθμός εποχών

Γενίκευση

- Ικανότητα ταξινόμησης προτύπων που δεν έχουν παρουσιαστεί ποτέ
- Όταν το δίκτυο «υπερεκπαιδεύει» χάνει την ικανότητα για γενίκευση
- Early Stopping
 - Δεδομένα εκπαίδευσης
 - Δεδομένα επαλήθευσης
 - Δεδομένα ελέγχου
 - Εκπαίδευση μόνο με τα δεδομένα εκπαίδευσης
 - Υπολογισμός σφάλματος για δεδομένα επαλήθευσης
 - Όταν το σφάλμα αρχίσει να αυξάνεται σταματάμε

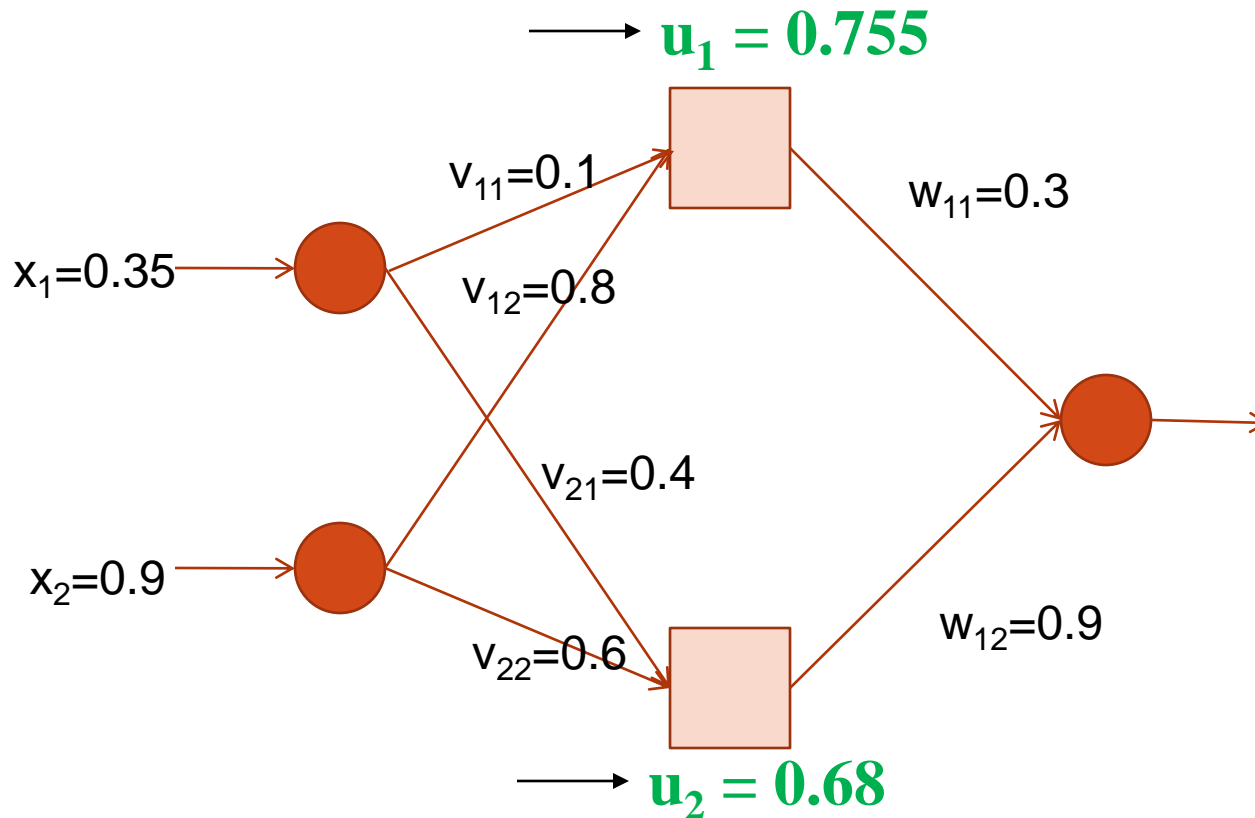
Παράδειγμα



Είσοδος: $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.35 \\ 0.9 \end{pmatrix}$ Επιθυμητή Έξοδος: $(d_1) = (0.5)$

Παράδειγμα

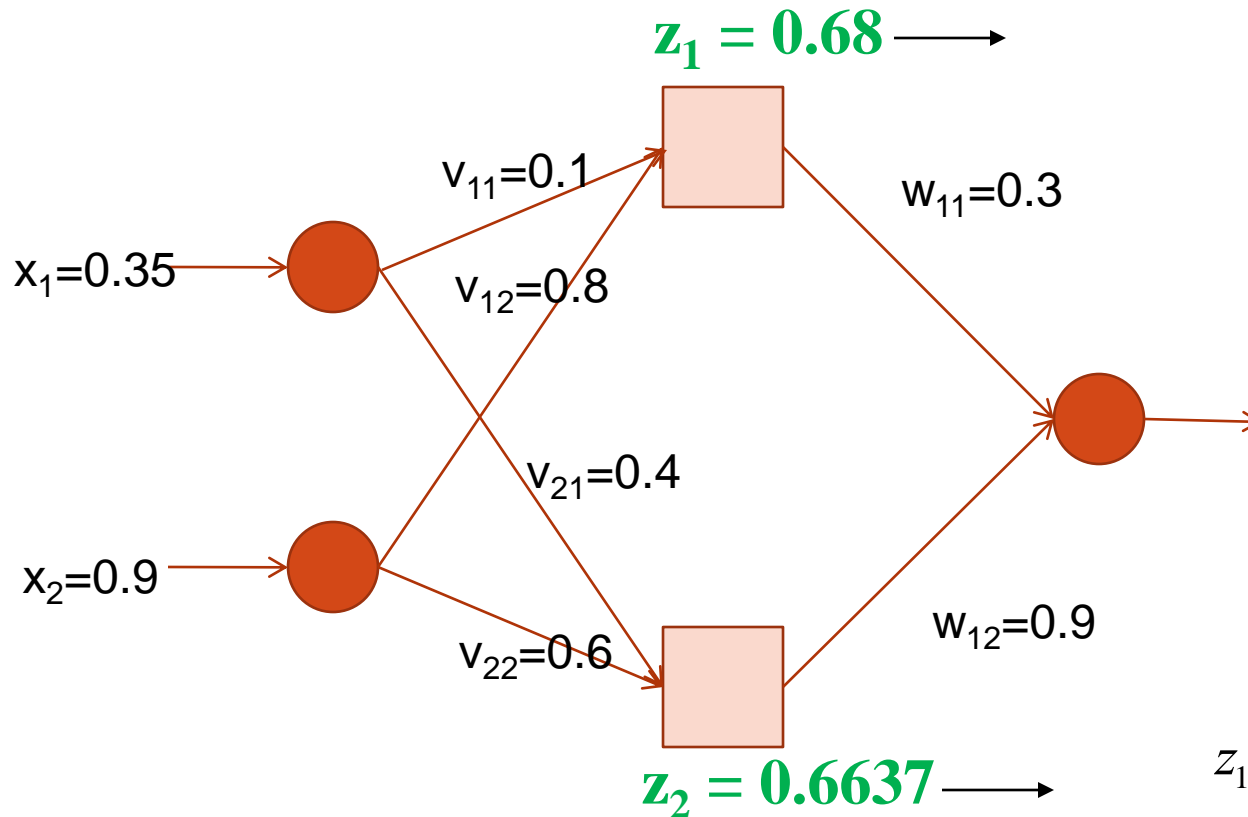
$$u_k = \sum_i w_{ki} \cdot x_i$$



- Πέρασμα κατά την **ΕΥΘΕΙΑ** φορά $u_1 = v_{11} \cdot x_1 + v_{12} \cdot x_2 = 0.1 \cdot 0.35 + 0.8 \cdot 0.9 = 0.755$
 - Υπολογίζουμε τις ενεργοποιήσεις $u_2 = v_{21} \cdot x_1 + v_{22} \cdot x_2 = 0.4 \cdot 0.35 + 0.6 \cdot 0.9 = 0.68$
- 1^{ου} επιπέδου

Παράδειγμα

$$y_k = f(u_k) = \frac{1}{1 + e^{-u_k}}$$



- Υπολογίζουμε τις εξόδους του πρώτου επιπέδου, περνώντας τις ενεργοποιήσεις που υπολογίσαμε προηγουμένως στη συνάρτηση ενεργοποίησης:

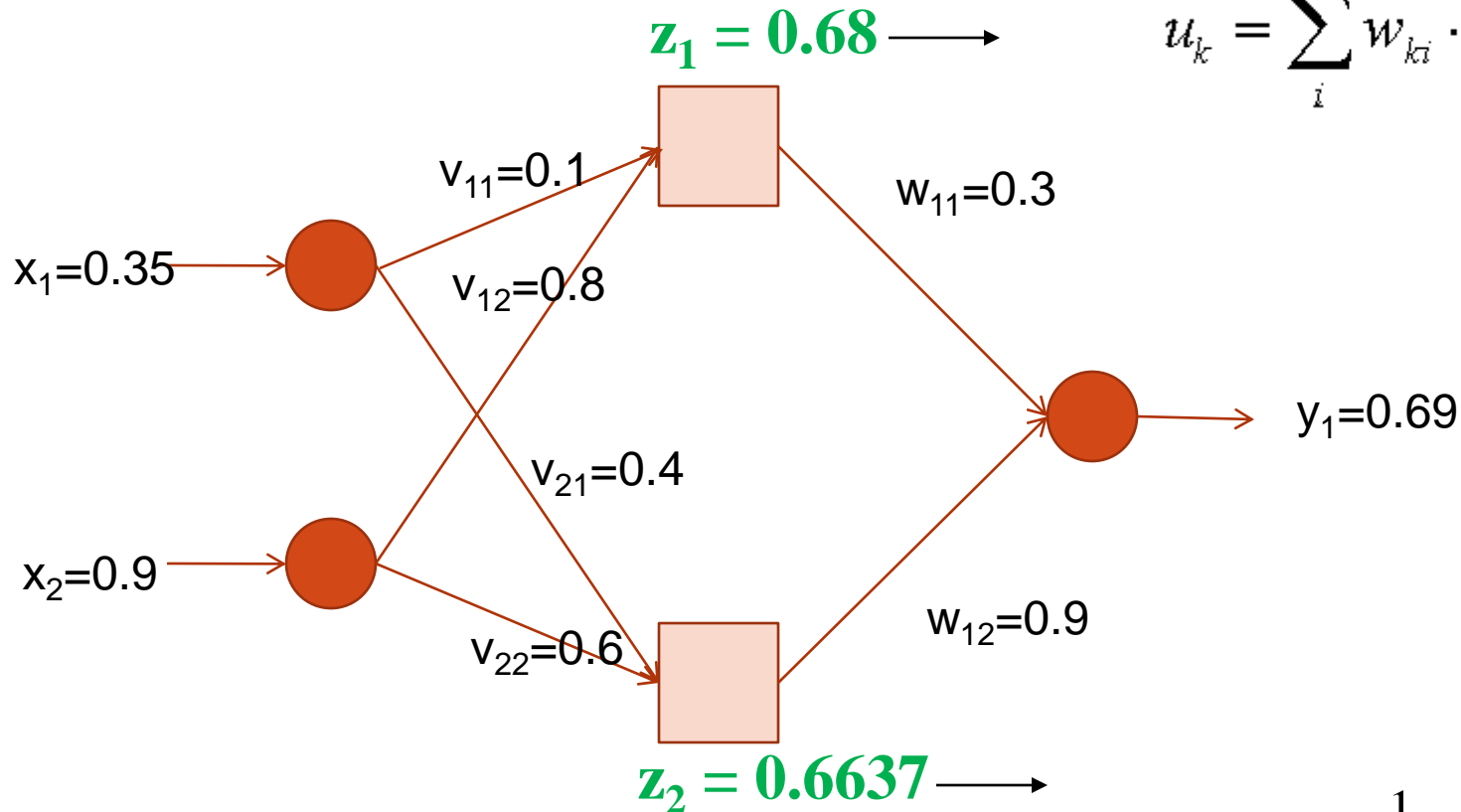
$$z_1 = \frac{1}{1 + e^{-0.755}} = 0.68$$

$$z_2 = \frac{1}{1 + e^{-0.68}} = 0.6637$$

Παράδειγμα

$$y_k = f(u_k) = \frac{1}{1 + e^{-u_k}}$$

$$u_k = \sum_i w_{ki} \cdot x_i$$

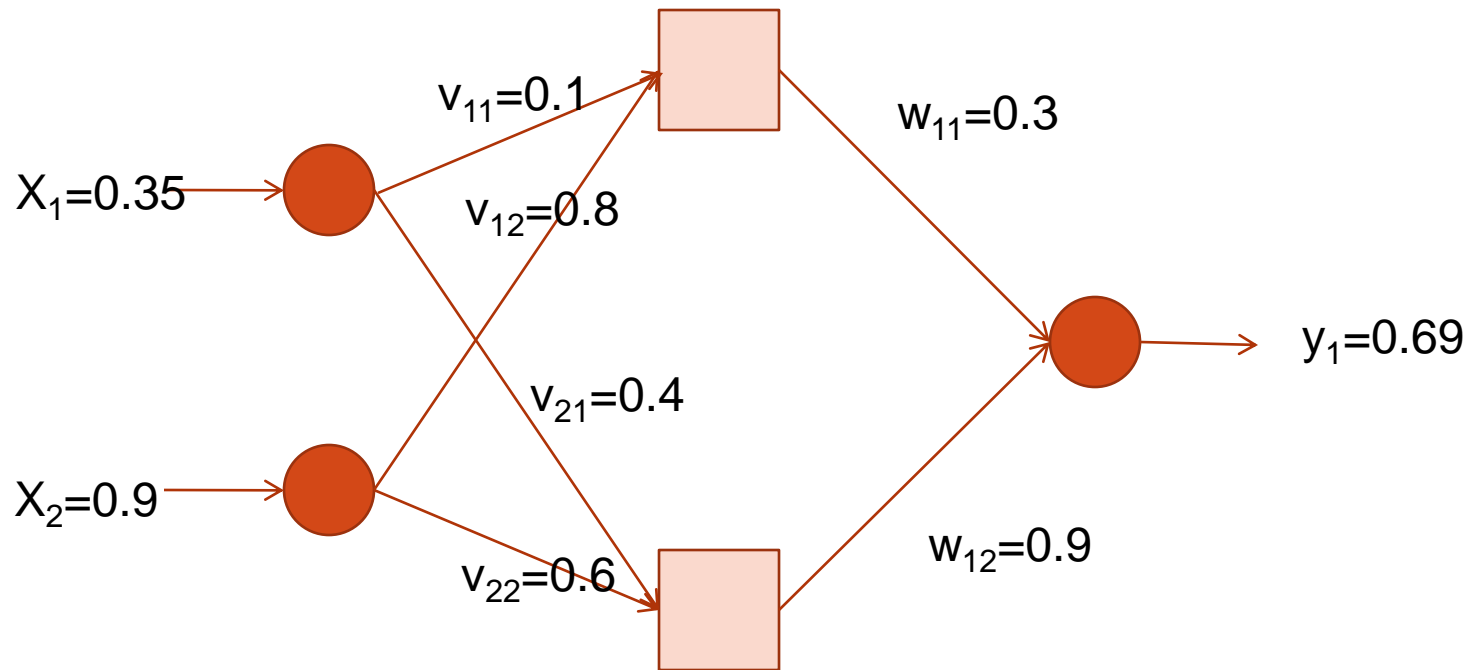


- Υπολογίζουμε την έξοδο του δευτέρου επιπέδου:

$$y_1 = \frac{1}{1 + e^{-0.8013}} = 0.69$$

Παράδειγμα

$$e_k = d_k - y_k$$



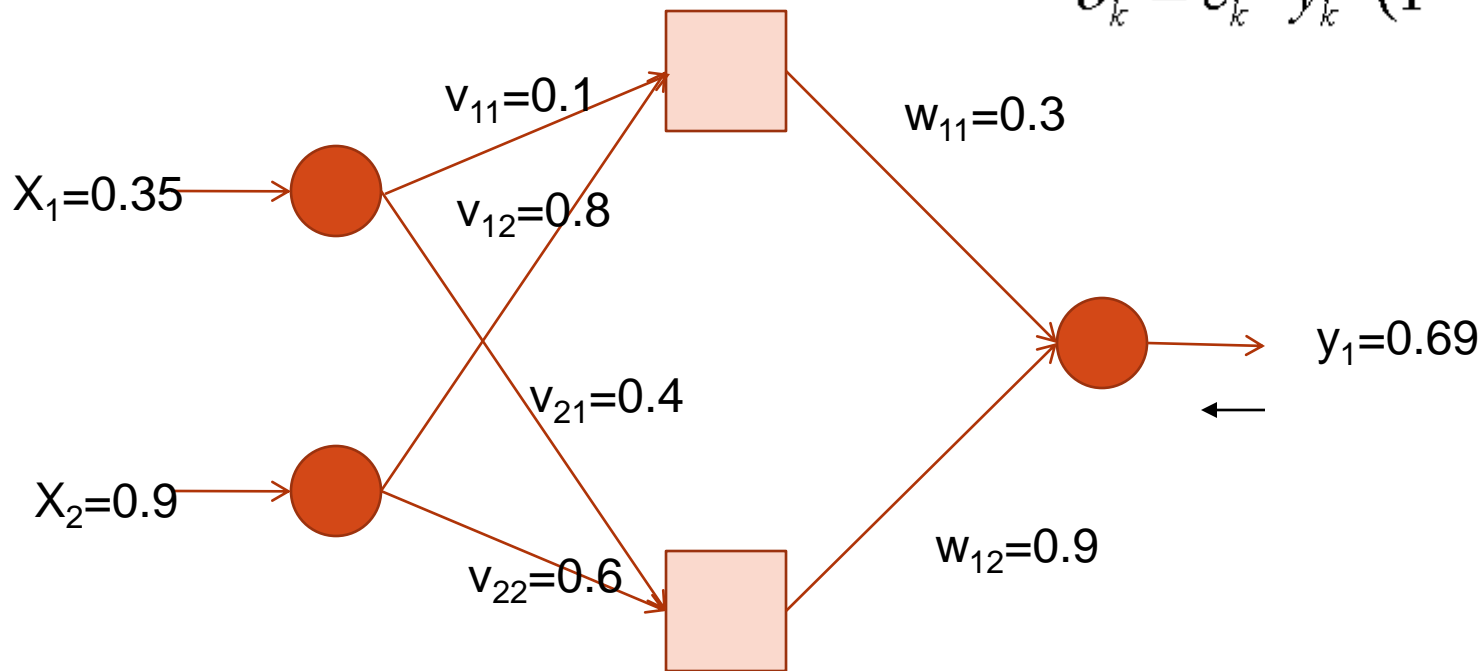
- Υπολογίζουμε το σφάλμα στην έξοδο: $e_1 = d_1 - y_1 = 0.5 - 0.69 = -0.19$

Παράδειγμα

$$e_k = d_k - y_k$$

$$\delta_k = e_k \cdot f'(u_k)$$

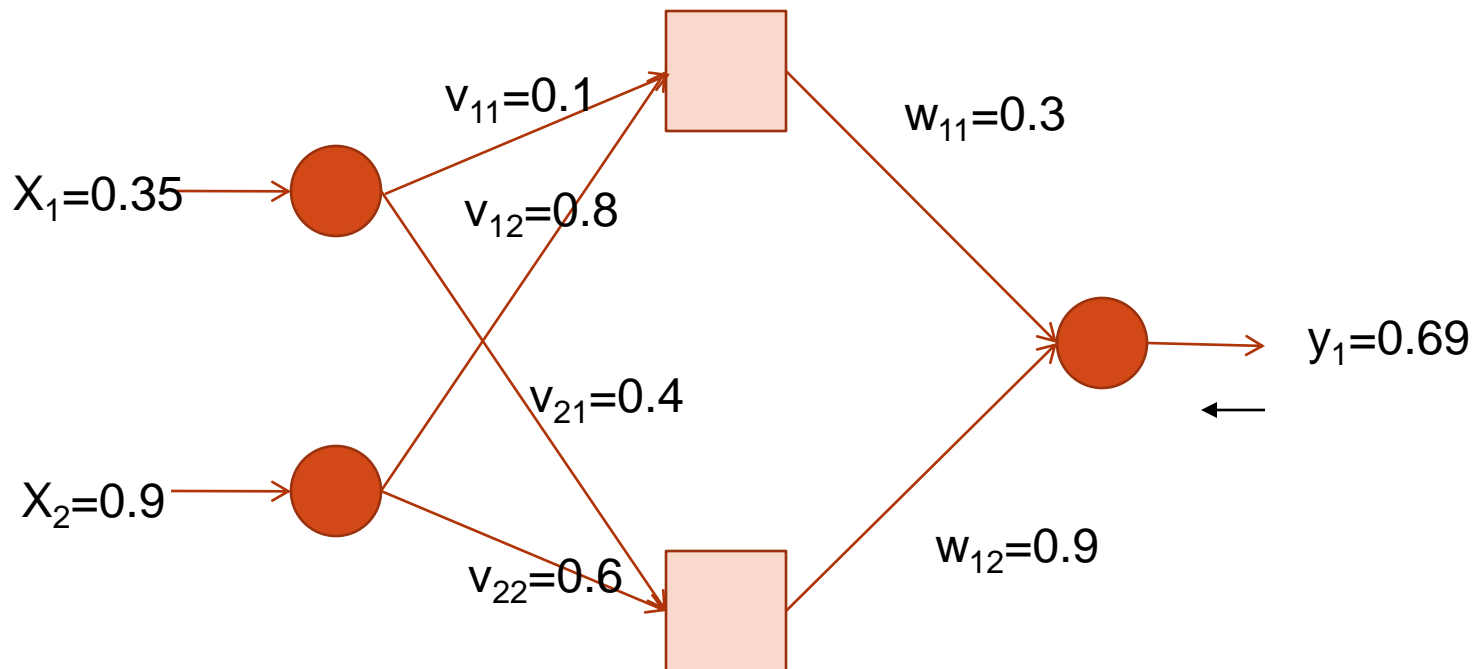
$$\delta_k = e_k \cdot y_k \cdot (1 - y_k)$$



- Υπολογίζουμε το σφάλμα στην έξοδο $e_1 = d_1 - y_1 = 0.5 - 0.69 = -0.19$ και την τοπική κλίση:
$$\delta_1 = -0.19 \cdot 0.69 \cdot (1 - 0.69) = -0.0406$$

Παράδειγμα

$$\Delta w_{ki} = \gamma \cdot \delta_k \cdot y_i$$

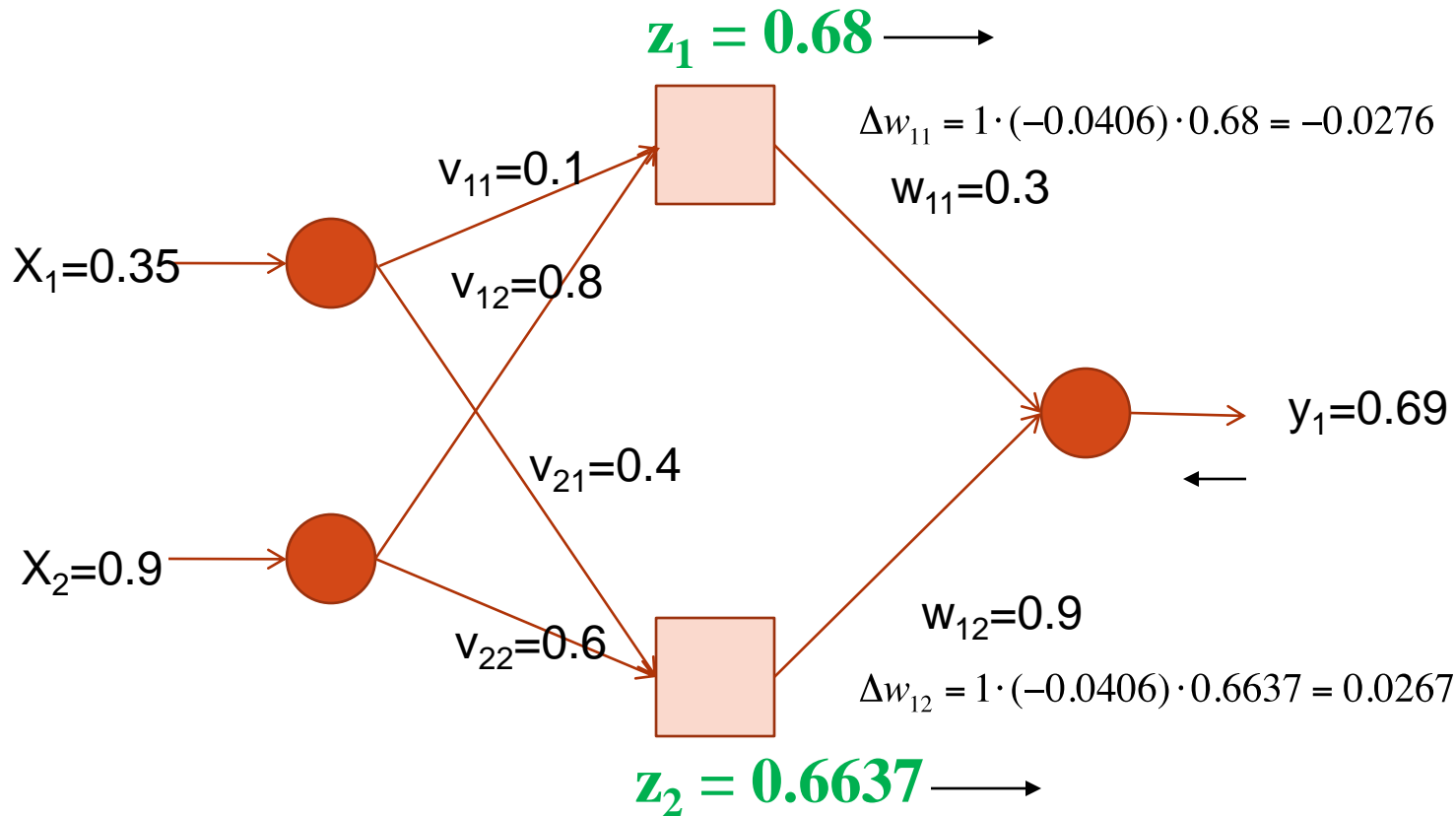


- Υπολογίζουμε το : Δw_{ki}

$$\delta_1 = -0.0406$$

Παράδειγμα

$$\Delta w_{ki} = \gamma \cdot \delta_k \cdot y_i$$

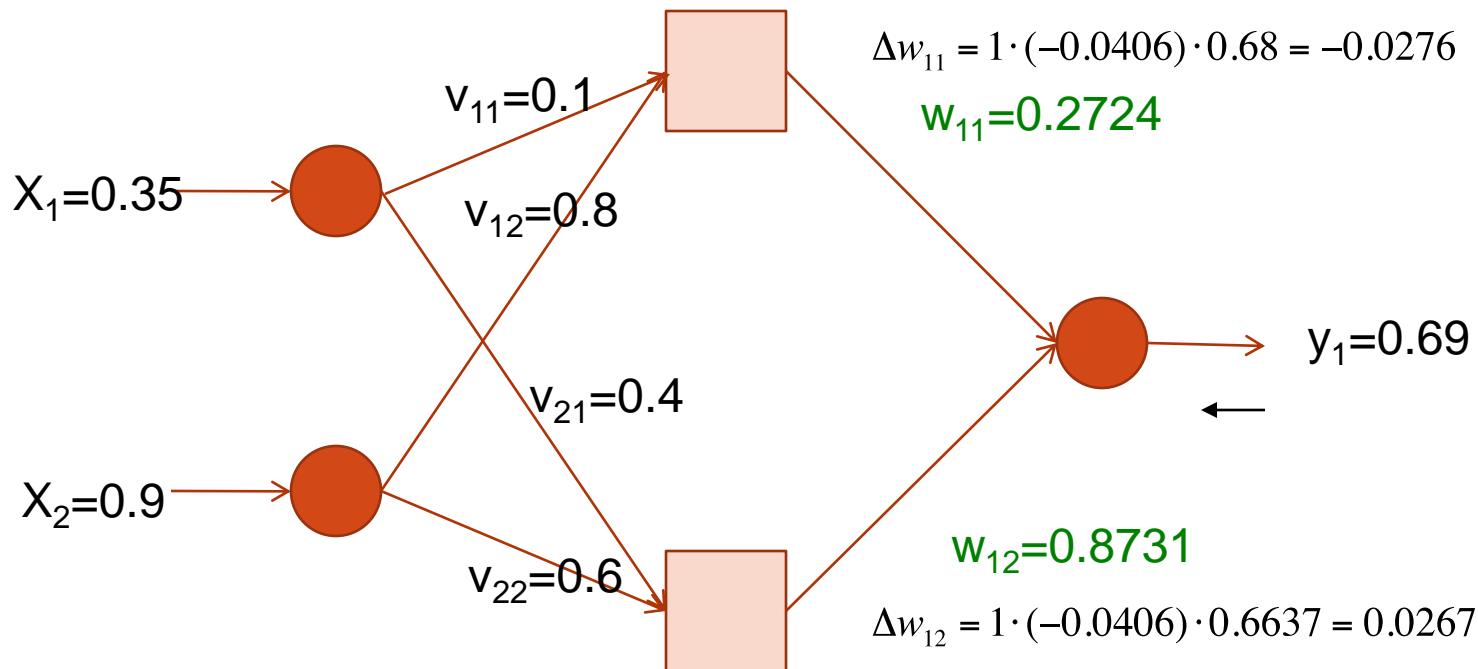


- Υπολογίζουμε το : Δw_{ki}

$$\delta_1 = -0.0406$$

Παράδειγμα

$$\Delta w_{ki} = \gamma \cdot \delta_k \cdot y_i$$



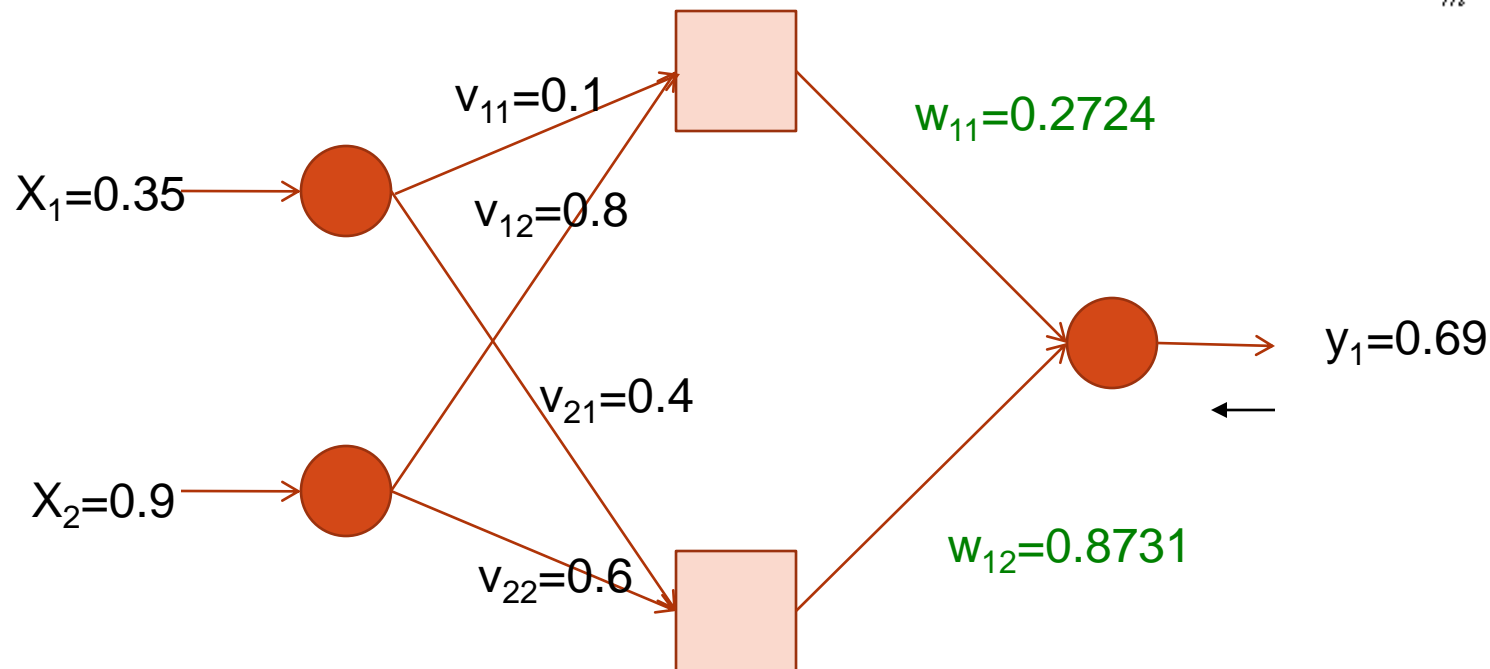
- Ανανεώνουμε τα βάρη

$$\delta_1 = -0.0406$$

Παράδειγμα

$$\Delta w_{ji} = \gamma \cdot \delta_j \cdot y_i$$

$$\delta_j = y_j \cdot (1 - y_j) \cdot \sum_m \delta_m \cdot w_{mj}$$



- Υπολογίζουμε το σφάλμα προς τα πίσω

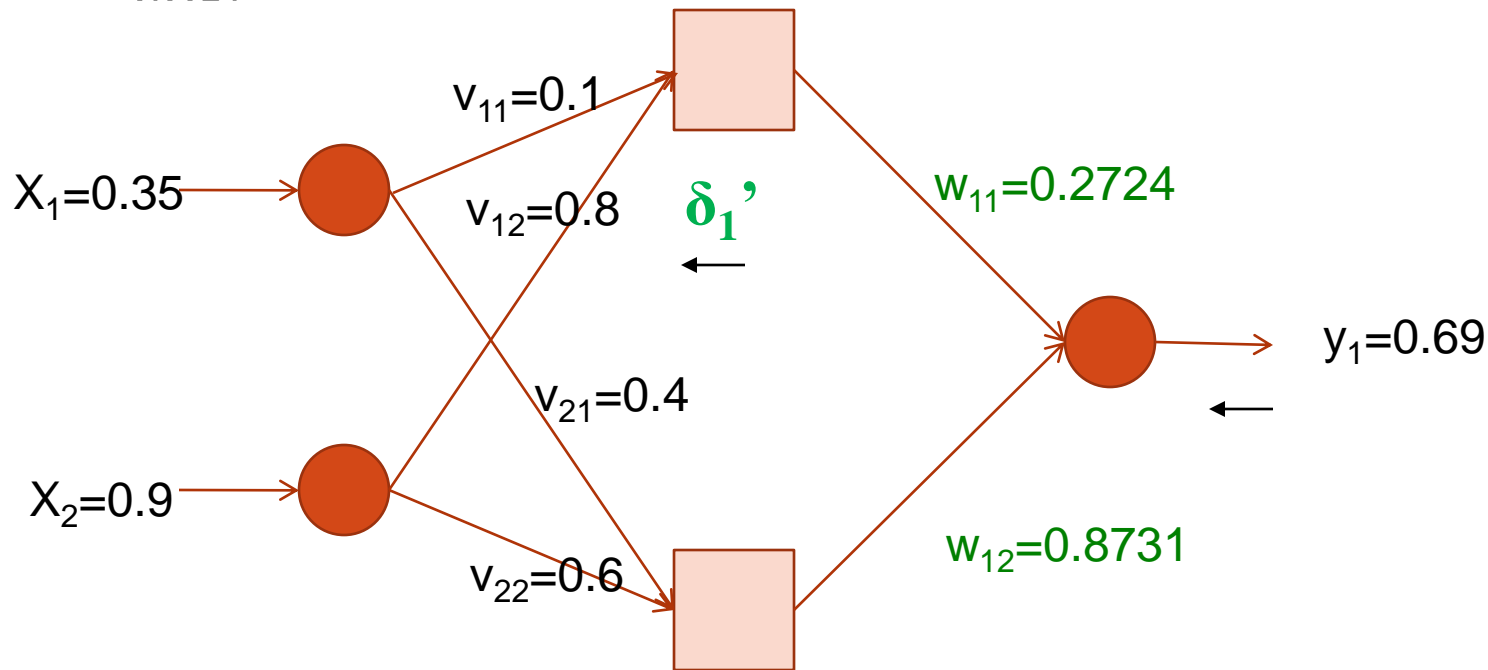
Παράδειγμα

$$\begin{aligned}\delta_1' &= y_1 \cdot (1 - y_1) \cdot \delta_1 \cdot w_{11} \\ &= 0.68 \cdot (1 - 0.68) \cdot (-0.0406) \cdot 0.2724 \\ &= -0.0024\end{aligned}$$

$$\Delta w_{ji} = \gamma \cdot \delta_j \cdot y_i$$

$$\delta_j = y_j \cdot (1 - y_j) \cdot \sum_m \delta_m \cdot w_{mj}$$

$$z_1 = 0.68 \rightarrow$$



$$\begin{aligned}\delta_2' &= y_2 \cdot (1 - y_2) \cdot \delta_1 \cdot w_{12} \\ &= 0.6637 \cdot (1 - 0.6637) \cdot (-0.0406) \cdot 0.8731 \\ &= -0.0079\end{aligned}$$

Παράδειγμα

$$\Delta v_{11} = 1 \cdot (-0.0024) \cdot 0.35 = -0.00084$$

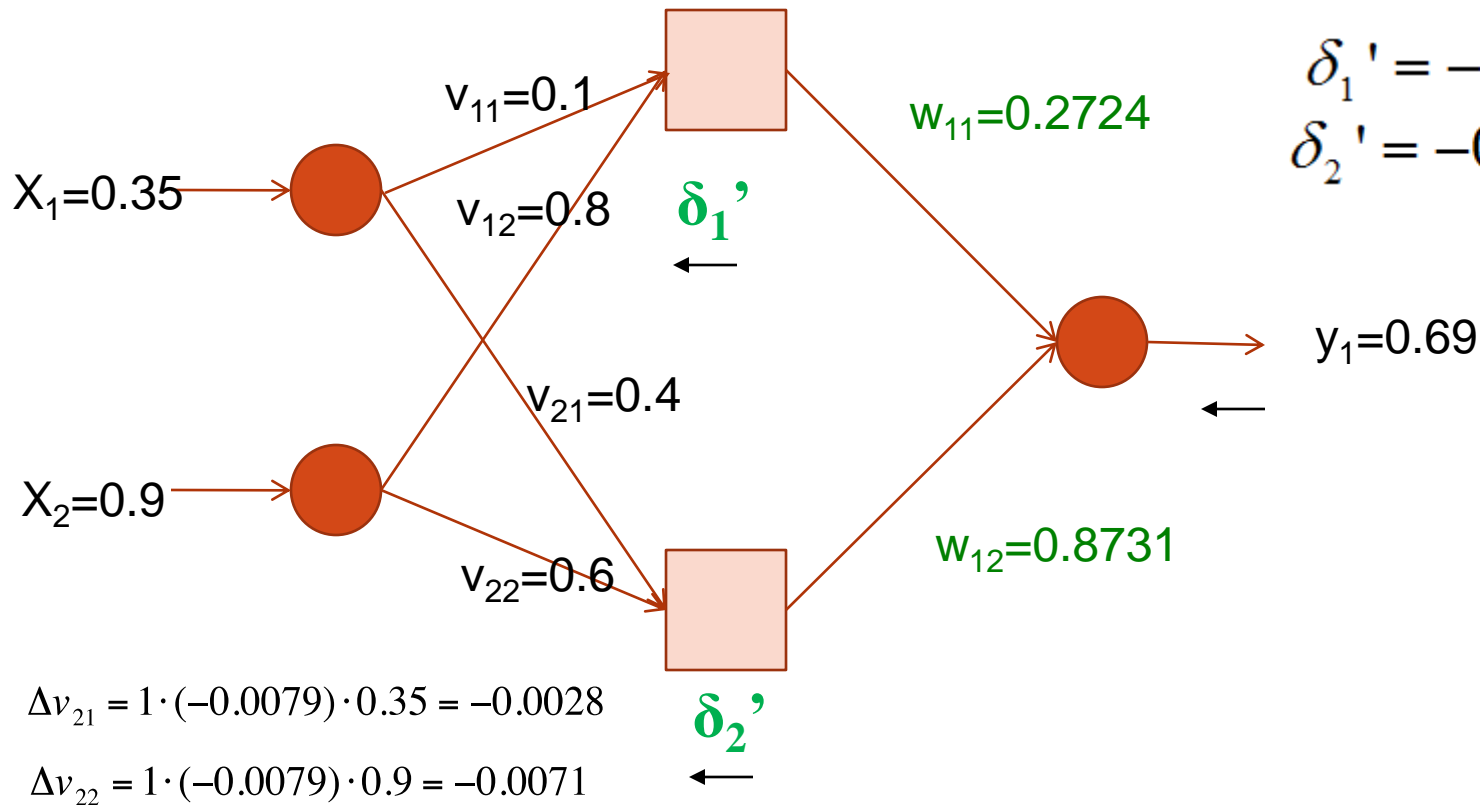
$$\Delta v_{12} = 1 \cdot (-0.0024) \cdot 0.9 = -0.0022$$

$$\Delta w_{ji} = \gamma \cdot \delta_j \cdot y_i$$

$$\delta_j = y_j \cdot (1 - y_j) \cdot \sum_m \delta_m \cdot w_{mj}$$

$$\delta_1' = -0.0024$$

$$\delta_2' = -0.0079$$



Παράδειγμα

$$\Delta v_{11} = 1 \cdot (-0.0024) \cdot 0.35 = -0.00084$$

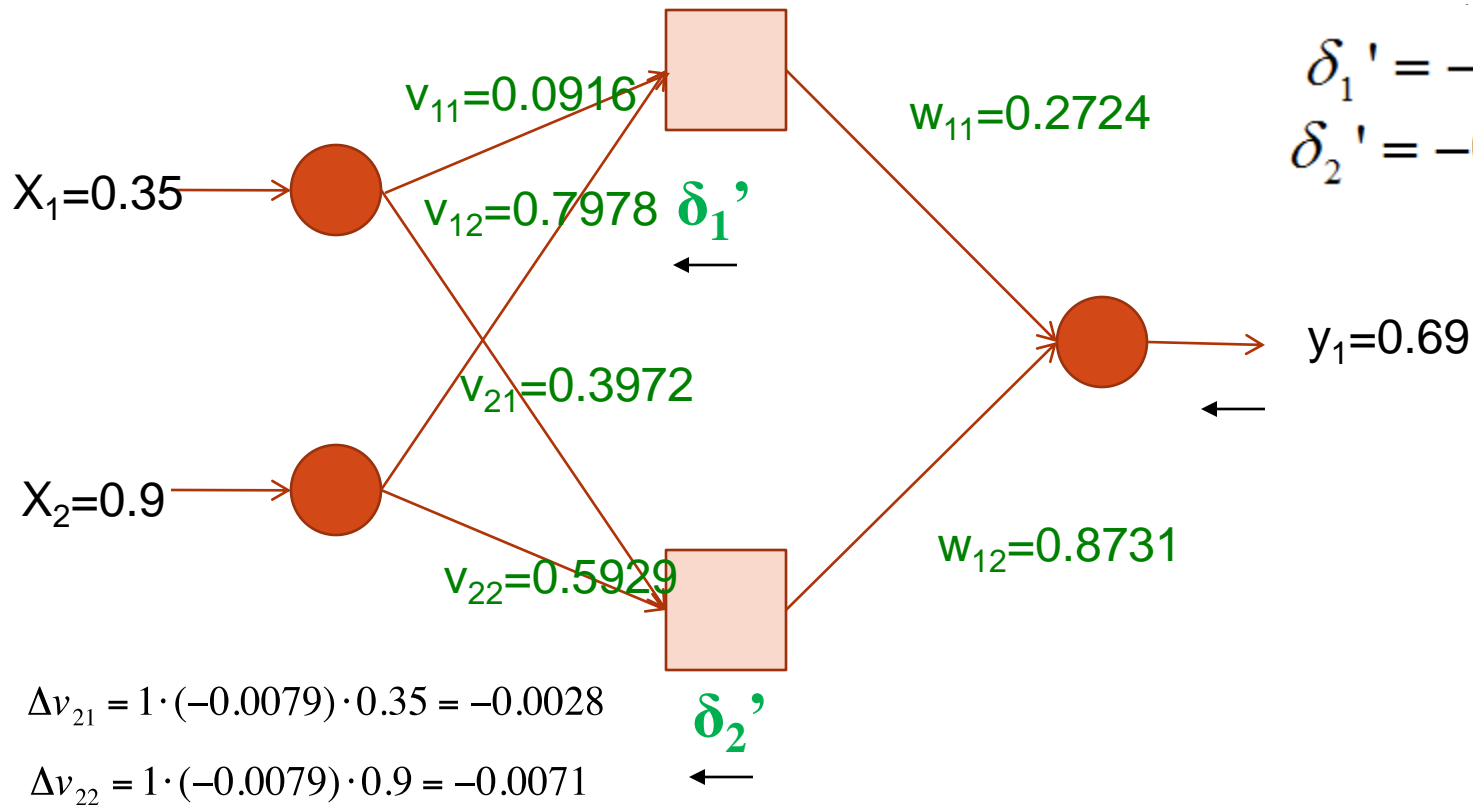
$$\Delta v_{12} = 1 \cdot (-0.0024) \cdot 0.9 = -0.0022$$

$$\Delta w_{ji} = \gamma \cdot \delta_j \cdot y_i$$

$$\delta_j = y_j \cdot (1 - y_j) \cdot \sum_m \delta_m \cdot w_{mj}$$

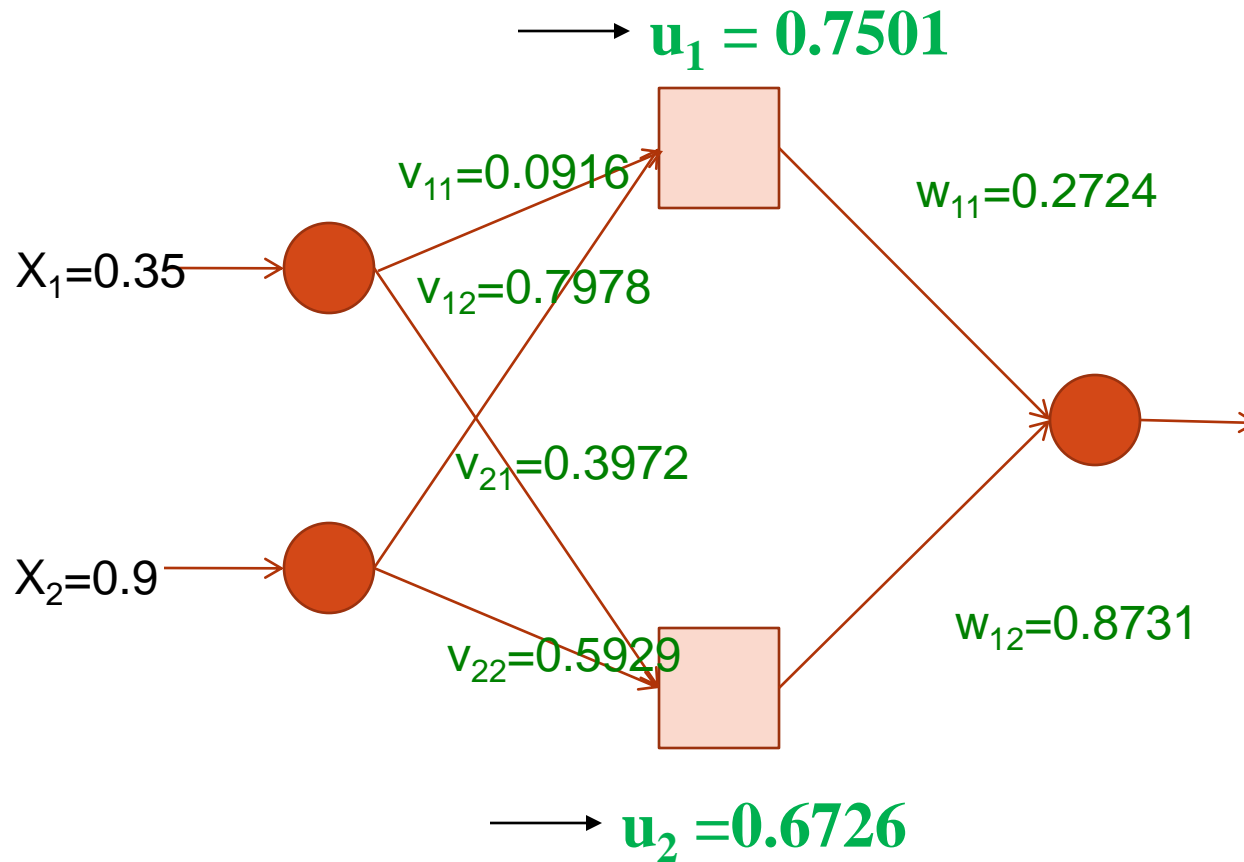
$$\delta_1' = -0.0024$$

$$\delta_2' = -0.0079$$



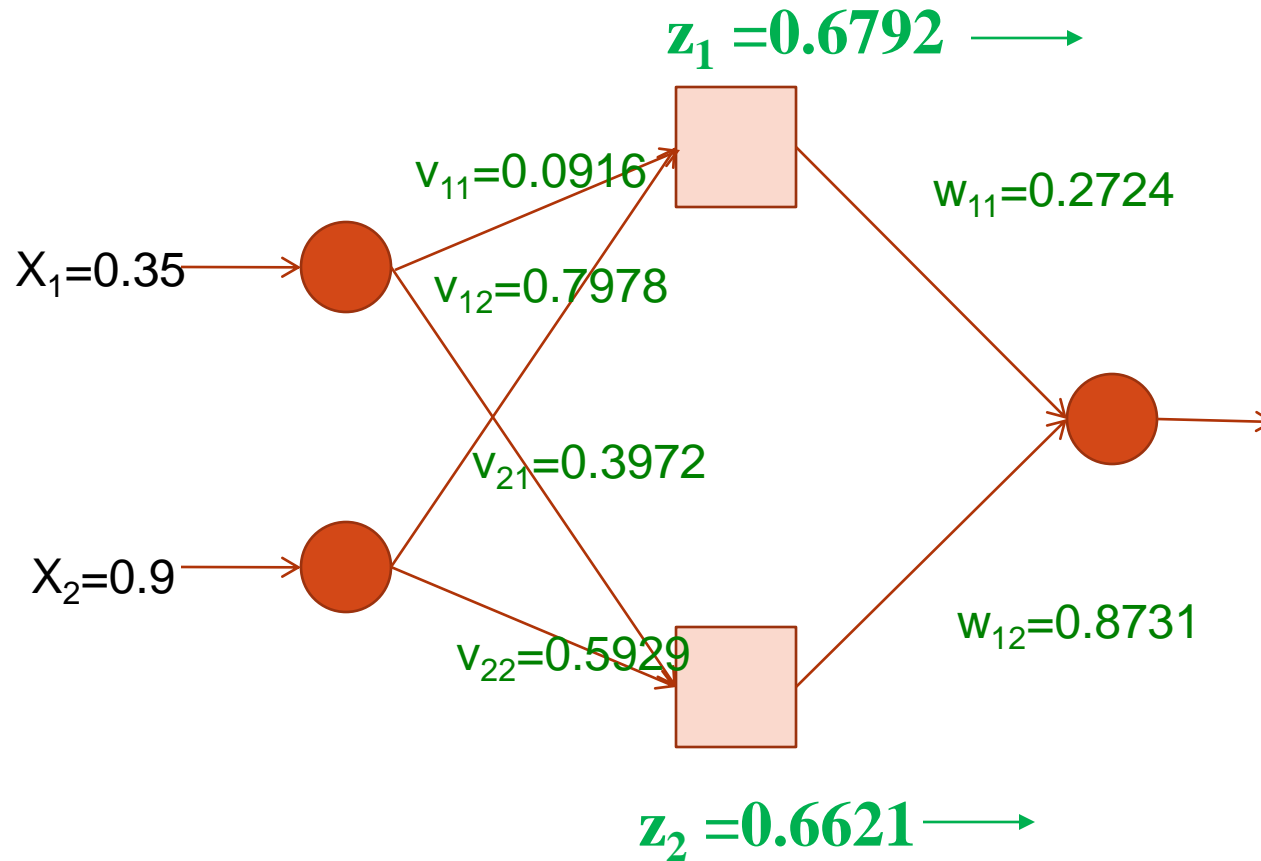
Παράδειγμα

$$u_k = \sum_i w_{ki} \cdot x_i$$



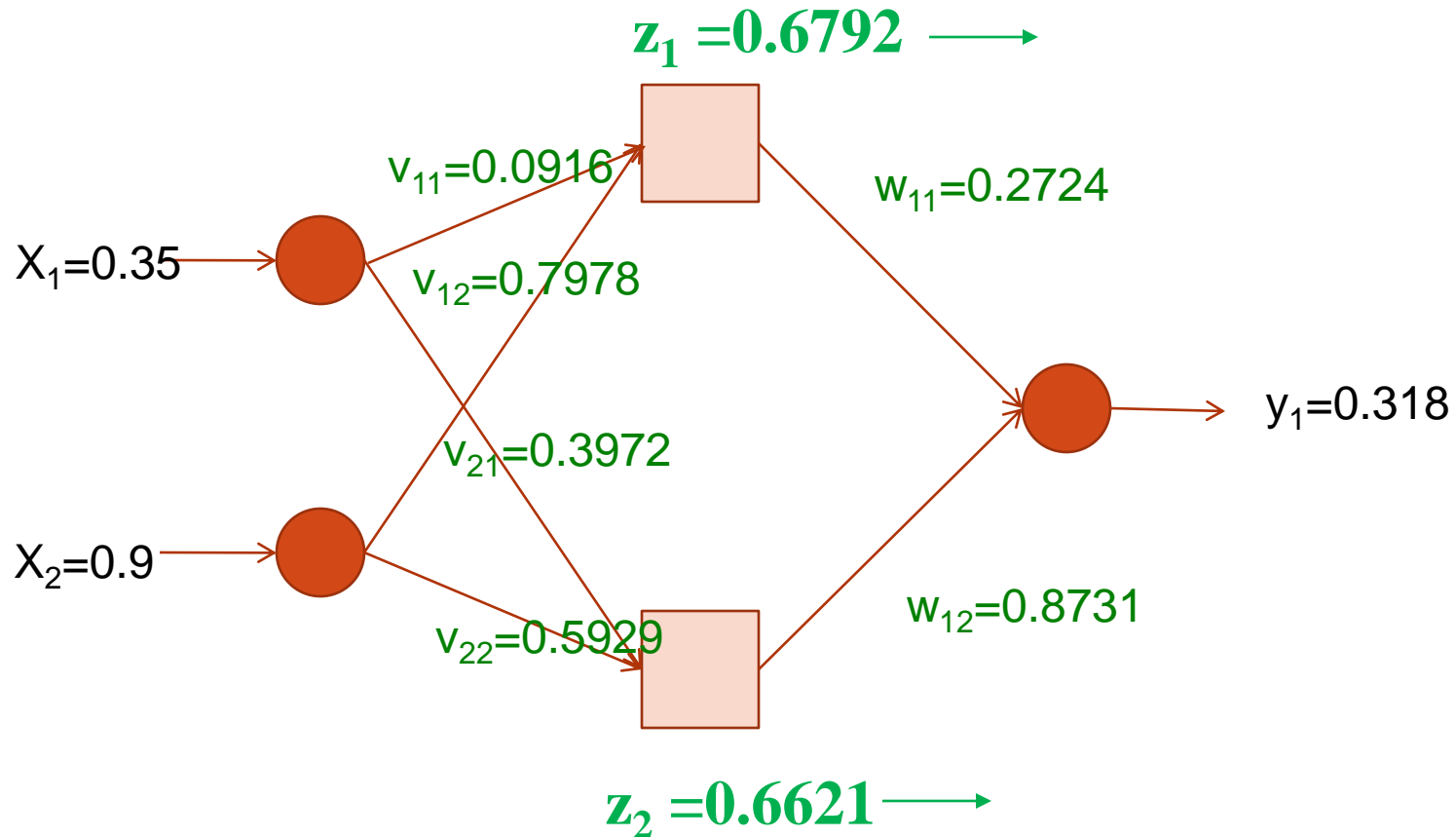
Παράδειγμα

$$y_k = f(u_k) = \frac{1}{1 + e^{-u_k}}$$



Παράδειγμα

$$y_k = f(u_k) = \frac{1}{1 + e^{-u_k}}$$



Το σφάλμα από 0.19 έχει γίνει 0.182

Μέρος 3^ο

Εισαγωγή στην άσκηση

Άσκηση 1



Άσκηση 1

Τμηματοποίηση



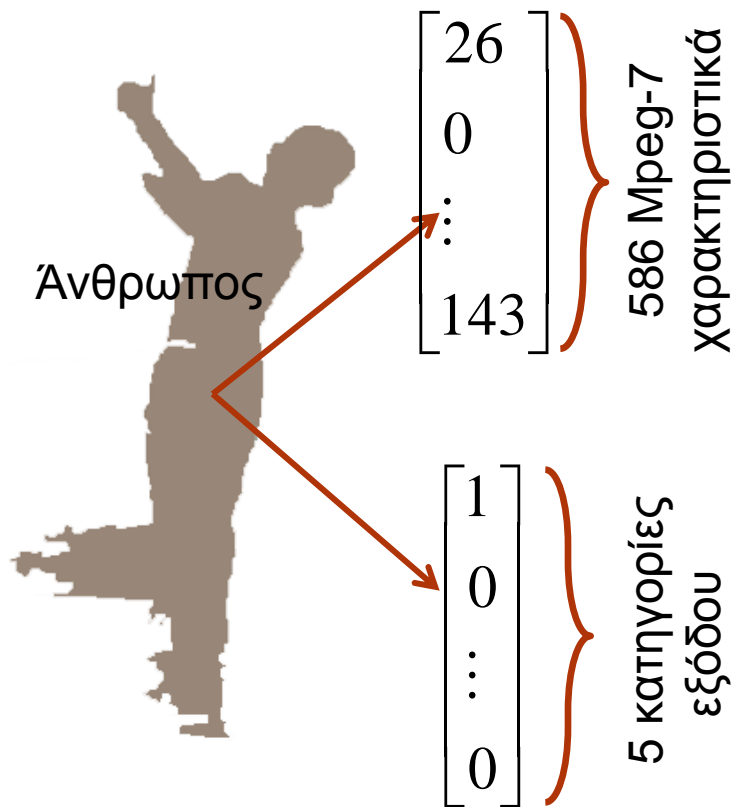
Άσκηση 1

Χαρακτηρισμός κάθε τμήματος της εικόνας



Κατηγορίες: Άνθρωπος, Άμμος, Κτήριο, Πεζοδρόμιο, Θάλασσα, Ουρανός, Κύμα, Φυτό, Γρασίδι, Δέντρο, Κορμός, Χώμα

Άσκηση 1



- Για κάθε τμήμα εικόνας δημιουργείται ένα διάνυσμα εισόδου και ένα διάνυσμα εξόδου
 - **Διάνυσμα εισόδου:** Προκύπτει από την εξαγωγή των MPEG-7 χαρακτηριστικών του τμήματος της εικόνας (Τα χαρακτηριστικά προκύπτουν από την επεξεργασία της εικόνας και μπορεί να σχετίζονται με στοιχεία όπως το χρώμα, η υφή κ.ο.κ.)
 - **Διάνυσμα εξόδου:** ν-διάστατο διάνυσμα που έχει την τιμή 1 μόνο στην γραμμή που αντιστοιχεί στην έννοια άνθρωπος και 0 στις άλλες γραμμές.

Άσκηση 1

Στόχος της άσκησης

- Δημιουργία Νευρωνικού Δικτύου
 - Είσοδος: MPEG-7 χαρακτηριστικά που περιγράφουν το τμήμα μιας εικόνας
 - Έξοδος: Ένα διάνυσμα n -διαστάσεων όπου η μεγαλύτερη τιμή σε κάθε διάσταση θα προσδιορίζει σε ποια κατηγορία ανήκει το τμήμα
- Δεδομένα
 - Δεδομένα Εκπαίδευσης Νευρωνικού Δικτύου
 - TrainData: Κάθε στήλη περιέχει τα MPEG-7 χαρακτηριστικά του τμήματος μιας εικόνας
 - TrainDataTargets: Το διάνυσμα εξόδου που προσδιορίζει σε ποια κατηγορία ανήκει το συγκεκριμένο τμήμα
 - Δεδομένα Αξιολόγησης Νευρωνικού Δικτύου
 - TestData, TestDataTargets: αντίστοιχη δομή με τα προηγούμενα δεδομένα

Προεπεξεργασία(1)

Επιλογή παρεμφερή αριθμού δεδομένων από κάθε κατηγορία

- Αρχικά έχουμε τα MPEG-7 χαρακτηριστικά και τις αντίστοιχες κατηγορίες για 645 τμήματα εικόνων εκπαίδευσης και 195 τμήματα εικόνων ελέγχου.
- Μετά την επεξεργασία θα επιλέξουμε στήλες έτσι ώστε να έχουμε παρεμφερή αριθμό δεδομένων εκπαίδευσης για κάθε κατηγορία.
- Δηλαδή k τμήματα που περιγράφουν πρόσωπο, k τμήματα που περιγράφουν αεροπλάνο κ.τ.λ.

TrainData

| | | | |
|-----|-----|-----|-----|
| 26 | 2 | ... | 0 |
| 0 | 0 | ... | 0 |
| ... | ... | ... | ... |
| 143 | 97 | ... | 79 |

TestData

| | | | |
|-----|-----|-----|-----|
| 0 | 0 | ... | 2 |
| 0 | 0 | ... | 0 |
| ... | ... | ... | ... |
| 73 | 14 | ... | 135 |

TrainDataTargets

| | | | |
|---|---|-----|---|
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 1 | 1 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 1 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |

TestDataTargets

| | | | |
|---|---|-----|---|
| 1 | 1 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 1 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |

Προεπεξεργασία(2)

- Αν θέλω για παράδειγμα να επιλέξω k εισόδους/εξόδους από την κατηγορία i (σύνολο 5 κατηγορίες):
index_i=find(TrainDataTargets(i,:),k);
%Αποθηκεύει στον πίνακα index_i τα indexes των k πρώτων στηλών που περιέχουν μονάδα στην i σειρά
- indexes= [index_1 index_2 index_3 index_4 index_5]
permutation=randperm(size(indexes,2));
indexes=indexes (permutation);
TrainData=TrainData(:,indexes);
TrainDataTargets=TrainDataTargets(:,indexes) ;

Προεπεξεργασία(3)

Remove constant rows

- Διαγράφουμε από τον πίνακα TrainData τις γραμμές που είναι σταθερές για όλες τις στήλες

TrainData

$$\begin{pmatrix} 26 & 2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 143 & 97 & \dots & 79 \end{pmatrix}$$

TestData

$$\begin{pmatrix} 0 & 0 & \dots & 2 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 73 & 14 & \dots & 135 \end{pmatrix}$$

TrainDataTargets

$$\begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

TestDataTargets

$$\begin{pmatrix} 1 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Προεπεξεργασία(4)

Removeconstantrows

- Διαγράφουμε από τον πίνακα TrainData τις γραμμές που είναι σταθερές για όλες τις στήλες
- Τις ίδιες γραμμές πρέπει να τις διαγράψουμε και από τον πίνακα TestData χρησιμοποιώντας την συνάρτηση removeconstantrows με παράμετρο 'apply'

TrainData

| | | | |
|-----|-----|-----|-----|
| 26 | 2 | ... | 0 |
| 0 | 0 | ... | 0 |
| ... | ... | ... | ... |
| 143 | 97 | ... | 79 |

TestData

| | | | |
|-----|-----|-----|-----|
| 0 | 0 | ... | 2 |
| 0 | 0 | ... | 0 |
| ... | ... | ... | ... |
| 73 | 14 | ... | 135 |

TrainDataTargets

| | | | |
|---|---|-----|---|
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 1 | 1 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 1 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |

TestDataTargets

| | | | |
|---|---|-----|---|
| 1 | 1 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 1 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |
| 0 | 0 | ... | 0 |

Ανάλυση σε Κύριες Συνιστώσες (processpca)

- Μας επιτρέπει σε ένα σύνολο δεδομένων να μειώσουμε τις διαστάσεις των δεδομένων χωρίς να χάσουμε σημαντική πληροφορία.
- Χρησιμοποιεί έναν ορθογώνιο μετασχηματισμό για να μετατρέψει ένα σύνολο συσχετιζόμενων τιμών σε γραμμικά ασυσχέτιστες

Προεπεξεργασία(5)

Χρήση `mapstd` πριν την εφαρμογή της `processpca`

Processpca

- Εφαρμόζουμε τη συνάρτηση `processpca` στον πίνακα `TrainData` με παράμετρο τέτοια ώστε να προκύψει ένας πίνακας με περίπου 20 γραμμές

```
[TrainData, ps]=  
    processpca(TrainData,0.002)
```

| TrainData | | | | | TestData | | | | | | |
|------------------|---|-----|-----|-----|-----------------|-----|---|-----|-----|-----|-----|
| 586 | { | 26 | 2 | ... | 0 | 586 | { | 0 | 0 | ... | 2 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | ... | ... | ... | ... | | | ... | ... | ... | ... |
| | | 143 | 97 | ... | 79 | | | 73 | 14 | ... | 135 |
| | | | | | | | | | | | |
| TrainDataTargets | | | | | TestDataTargets | | | | | | |
| | { | 0 | 0 | ... | 0 | | { | 1 | 1 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 1 | 1 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 1 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 1 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |
| | | 0 | 0 | ... | 0 | | | 0 | 0 | ... | 0 |

Προεπεξεργασία(6)

Χρήση `mapstd` πριν την εφαρμογή της `processpca`

$$\begin{matrix} \text{TrainData} & & \text{TestData} \\ \rightsquigarrow & \begin{pmatrix} -3.66 & -1.27 \dots -4.26 \\ -3.64 & -6.25 \dots 8.13 \\ \dots & \dots \ddots \dots \\ -1.32 & 1.41 \dots -0.56 \end{pmatrix} & \rightsquigarrow & \begin{pmatrix} -2.60 & 15.9 \dots -2.08 \\ -1.15 & -4.68 \dots 4.61 \\ \dots & \dots \ddots \dots \\ -0.93 & -0.30 \dots -0.07 \end{pmatrix} \end{matrix}$$

Processpca

- Εφαρμόζουμε τη συνάρτηση `processpca` στον πίνακα `TrainData` με παράμετρο τέτοια ώστε να προκύψει ένας πίνακας με περίπου 20 γραμμές

`[TrainData, ps]=
processpca(TrainData,0.002)`

- Εφαρμόζουμε τον ίδιο μετασχηματισμό και στα `TestData` με παράμετρο `'apply'` στη συνάρτηση `processpca`

TrainDataTargets

$$\begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

TestDataTargets

$$\begin{pmatrix} 1 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Μελέτη Αρχιτεκτονικής Νευρωνικού Δικτύου(1)

Δημιουργία νευρωνικού δικτύου

- Π.χ. Δημιουργούμε ένα νευρωνικό με 2 κρυμμένα επίπεδα που το πρώτο έχει 10 και το δεύτερο 15 νευρώνες.

```
net=newff(TrainData,TrainDataTargets,[10 15])
```

- Χωρίζουμε τα δεδομένα μας σε ένα σύνολο εκπαίδευσης και ένα σύνολο επαλήθευσης έτσι ώστε να χρησιμοποιηθεί η μέθοδος Early Stopping για τον τερματισμό της εκπαίδευσης του νευρωνικού

```
net.divideParam.trainRatio=0.8  
net.divideParam.valRatio=0.2  
net.divideParam.testRatio=0
```

- Επιλέγω μέγιστο αριθμό εποχών για την εκπαίδευση π.χ. 1000

```
net.trainParam.epochs=1000
```

Μελέτη Αρχιτεκτονικής Νευρωνικού Δικτύου(2)

Εκπαίδευση νευρωνικού δικτύου

- Εκπαιδεύω το νευρωνικό δίκτυο χρησιμοποιώντας την συνάρτηση `train` με εισόδους τα `TrainData` και `TrainDataTargets`.

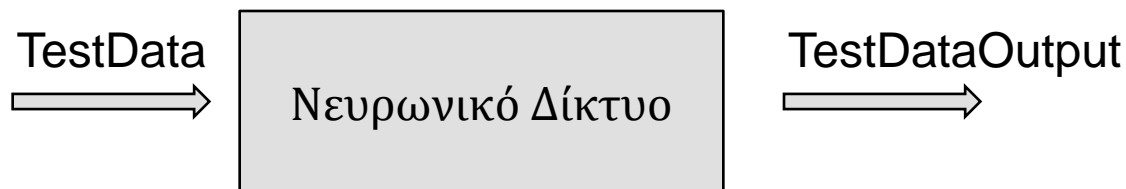
```
net=train(net,TrainData,TrainDataTargets)
```

- Σκοπός της εκπαίδευσης είναι να μάθει το νευρωνικό δίκτυο τα δεδομένα που βρίσκονται στον πίνακα `TrainData` και σε κάθε είσοδο από το `TrainData` να βγάζει την αντίστοιχη έξοδο από το `TrainDataTargets`.

Μελέτη Αρχιτεκτονικής Νευρωνικού Δικτύου(3)

Αξιολόγηση νευρωνικού δικτύου

- Αξιολογώ το νευρωνικό δίκτυο χρησιμοποιώντας τα δεδομένα των πινάκων `TestData` και `TestDataTargets`.
- Εφαρμόζω τα δεδομένα του `TestData` στο νευρωνικό δίκτυο και στην έξοδο παίρνω τον πίνακα `TestDataOutput`



`TestDataOutput=sim(net,TestData)`

Μελέτη Αρχιτεκτονικής Νευρωνικού Δικτύου(4)

Αξιολόγηση νευρωνικού δικτύου

- Χρησιμοποιώντας την συνάρτηση **eval_Accuracy_Precision_Recall** που μου δίνεται από τη σελίδα του μαθήματος υπολογίζω το accuracy, το precision, και το recall για τα TestData:

```
[accuracy,precision,recall]=  
    eval_Accuracy_Precision_Recall(TestDataOutput,TestDataTargets)
```

Μελέτη Αρχιτεκτονικής Νευρωνικού Δικτύου(5)

Μελέτη της επίδρασης διαφόρων παραμέτρων του νευρωνικού δικτύου

- Υπολογίζω τα accuracy, τα precision και recall μεταβάλλοντας το ρυθμό μάθησης και τον αριθμό των εποχών του νευρωνικού δικτύου
- Υπολογίζω τα accuracy, τα precision και recall για διάφορους συνδυασμούς αριθμού νευρώνων με ένα ή δύο κρυμμένα επίπεδα.
- Υπολογίζω τα accuracy, τα precision και recall για διαφορετικές συναρτήσεις εκπαίδευσης