



Πρόγραμμα Μεταπτυχιακών Σπουδών  
"Προηγμένες Τεχνολογίες Υπολογιστών  
και Πληροφορικής"

# Υπολογιστική Νοημοσύνη

Μάθημα 1ο

Δρ. Βασίλειος Γ. Καμπουρλάζος  
Δρ. Ανέστης Γ. Χατζημιχαηλίδης

Τμήμα Μηχανικών Πληροφορικής Τ.Ε.  
ΤΕΙ Ανατολικής Μακεδονίας και Θράκης

2017-2018

# Περιεχόμενα

---

- ⊙ Αξιολόγηση μαθήματος
- ⊙ Εισαγωγή στην ΥΝ
- ⊙ Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)
- ⊙ Προγραμματισμός σε **MATLAB**
- ⊙ ΥΝ και **MATLAB**

# Αξιολόγηση Μαθήματος

## ● Τελική εξέταση (50%)

- Ερωτήσεις πολλαπλών επιλογών
- Επίλυση προβλημάτων

## ● Εργασία 1 - Ερευνητική πρακτική (25%)

- Εύρεση θέματος που άπτεται της θεματολογίας του μαθήματος από κάθε φοιτητή
- Βιβλιογραφική ανασκόπηση του θέματος
- Ανάθεση εργασιών 2η εβδομάδα μαθήματος
- Παράδοση εργασιών 4η εβδομάδα με παρουσίαση σε ανοικτή ακρόαση

## ● Εργασία 2 - Υπολογιστική νοημοσύνη στην πράξη (25%)

- Εύρεση θέματος/μεθοδολογίας από κάθε φοιτητή
- Ανάθεση εργασιών 4η εβδομάδα μαθήματος
- Υλοποίηση σε περιβάλλον **MATLAB**
- Σύνταξη έκθεσης (περιγραφή μεθόδου, πειραματικά αποτελέσματα)
- Παράδοση εργασιών την τελευταία εβδομάδα με παρουσίαση σε ανοικτή ακρόαση

# Βιβλίο Μαθήματος



Διατίθεται δωρεάν από τον σύνδεσμο  
<https://repository.kallipos.gr/handle/11419/3443>

Δρ. Βασίλειος Γ. Καμπουρλάζος

# Εφαρμογές

---

- ◉ Εργαστήριο

Αλληλεπίδρασης Ανθρώπου-με-Μηχανές  
(ΑΜΑ) του ΤΕΙ ΑΜΘ

<http://humain-lab.teiemt.gr/>

# Εισαγωγή στην ΥΝ

---

# Ορισμός 1

## Τεχνητή Νοημοσύνη

---

Ως **Τεχνητή Νοημοσύνη (TN) (*artificial intelligence*)** ορίζουμε την ικανότητα των υπολογιστών/μηχανών να συμπεριφέρονται με νοημοσύνη όπως ακριβώς οι άνθρωποι.

Κύριοι στόχοι της TN είναι :

- Αναπαράσταση της γνώσης (**knowledge representation**)
- Συλλογισμός (**reasoning**)
- Εκμάθηση (**learning**)
- Αντίληψη (**perception**)

# Υπολογιστική Νοημοσύνη

Ως **Υπολογιστική Νοημοσύνη (ΥΝ)** (**computational intelligence**) ορίζουμε μια συνεχώς εξελισσόμενη συνέργια μεθοδολογιών επεξεργασίας αριθμητικών δεδομένων, οι οποίες είναι υλοποιήσιμες σε υπολογιστή για την λήψη αποφάσεων κοινής λογικής.

Συνώνυμο του όρου ΥΝ στη βιβλιογραφία χρησιμοποιείται ο όρος **Ευέλικτος Υπολογισμός (ΕΥ)** (**soft computing**).



# Κατηγοριοποίηση (1/2)

---

## ◎ Κλασική ΥΝ

- Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)  
*(προσομοίωση της λειτουργίας τους εγκεφάλου)*
- Ασαφή Συστήματα (ΑΣ)  
*(προσομοίωση της λειτουργίας της ομιλούμενης γλώσσας)*
- Εξελικτικός Υπολογισμός (ΕΥ)  
*(προσομοίωση της Δαρβινικής φυσικής επιλογής)*

# Κατηγοριοποίηση (2/2)

---

## ◉ Διευρημένη ΥΝ

- Νευρο-Ασαφή Συστήματα (Neuro-Fuzzy)
- Δίκτυα Ακτινωτής Βάσης (Radial Basis Networks)
- Γνωσιακοί Χάρτες (Cognitive Maps)
- Μηχανές Διανυσμάτων Στήριξης (SVMs)
- Δένδρα Αποφάσεων (Decision Trees)
- Συνδυασμός Μοντέλων (Ensembles)

# Στόχος αλγορίθμων ΥΝ

Τυπικό στόχο ενός αλγόριθμου ΥΝ αποτελεί η εκτίμηση μιας συνάρτησης  $f: \mathbb{R}^N \rightarrow \mathbb{T}$ , όπου το πεδίο ορισμού  $\mathbb{R}$  είναι το σύνολο των πραγματικών αριθμών, ενώ το πεδίο τιμών  $\mathbb{T}$  μπορεί να είναι είτε  $\mathbb{T} = \mathbb{R}^M$  (σε προβλήματα παλινδρόμησης) είτε ένα σύνολο από ετικέτες (σε προβλήματα αναγνώρισης).

# Βασικές Λειτουργίες

Η διαδικασία υπολογισμού της συνάρτησης  $f: \mathbb{R}^N \rightarrow \mathbb{T}$  από το σύνολο των ζευγών  $(\mathbf{x}_1, f(\mathbf{x}_1)), \dots, (\mathbf{x}_n, f(\mathbf{x}_n))$  στα πλαίσια της ΥΝ καλείται μάθηση ή, ισοδύναμα, εκπαίδευση, ενώ η διαδικασία υπολογισμού της τιμής  $f(\mathbf{x}_0)$ , για  $\mathbf{x}_0 \neq \mathbf{x}_i, i \in \{1, \dots, n\}$ , καλείται γενίκευση. Βασικός στόχος κάθε διαδικασίας μάθησης στην ΥΝ είναι μια αποδεκτή ικανότητα γενίκευσης.

# Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

---

# Βιολογικά Νευρωνικά Δίκτυα (1/2)

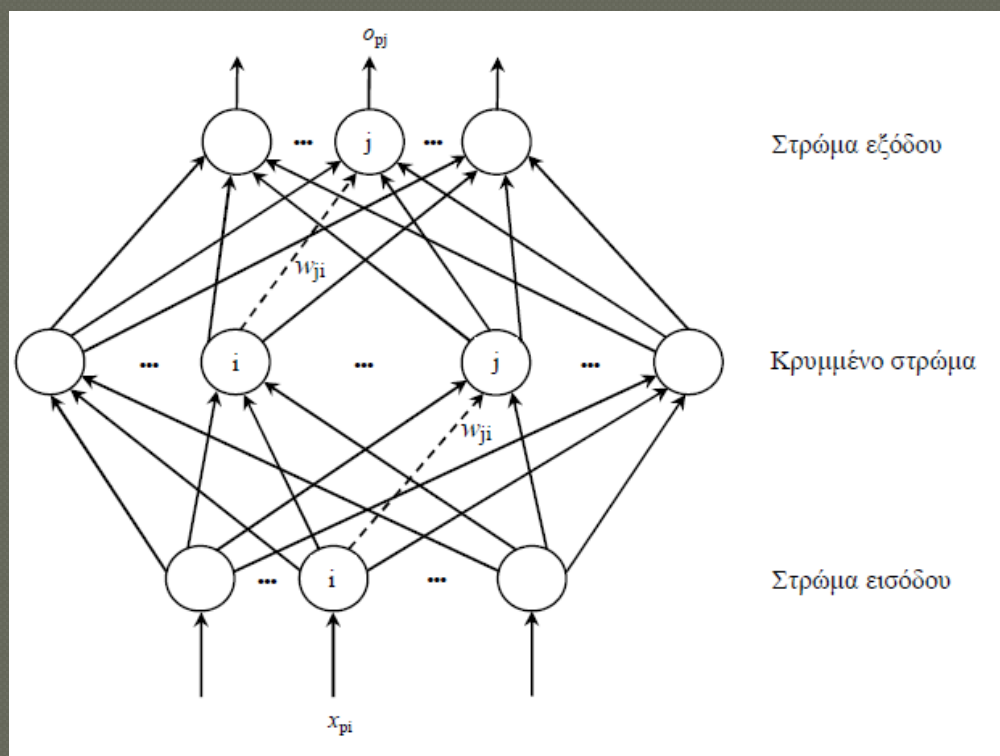
- Το βασικό στοιχείο υπολογισμού στα βιολογικά συστήματα είναι ο **νευρώνας**.
- Ένας νευρώνας είναι ένα (μικρό) **κύτταρο** που λαμβάνει ηλεκτροχημικές διεγέρσεις από πολλαπλές πηγές και αποκρίνεται δημιουργώντας ηλεκτρικά ερεθίσματα.
- Ο ανθρώπινος εγκέφαλος αποτελείται από πολλά δισεκατομμύρια ( $>10^9$ ) νευρώνες.

# Βιολογικά Νευρωνικά Δίκτυα (2/2)

- Περίπου το 10% των νευρώνων είναι είσοδοι (προσαγωγοί) και έξοδοι (επαγωγοί). Το υπόλοιπο 90% είναι αλληλοσυνδέσεις με άλλους νευρώνες, οι οποίες αποθηκεύουν πληροφορία.
- Ο ανθρώπινος εγκέφαλος λειτουργεί εκτελώντας, μαζικά, μια παράλληλη και κατανεμημένη επεξεργασία ηλεκτρικών σημάτων.

# Τεχνητά Νευρωνικά Δίκτυα

- Ένα ΤΝΔ υλοποιεί μια συνάρτηση  $f: \mathcal{R}^N \rightarrow \mathcal{S}$  χρησιμοποιώντας μια αρχιτεκτονική παρόμοια με την αρχιτεκτονική του (ανθρώπινου) εγκεφάλου.

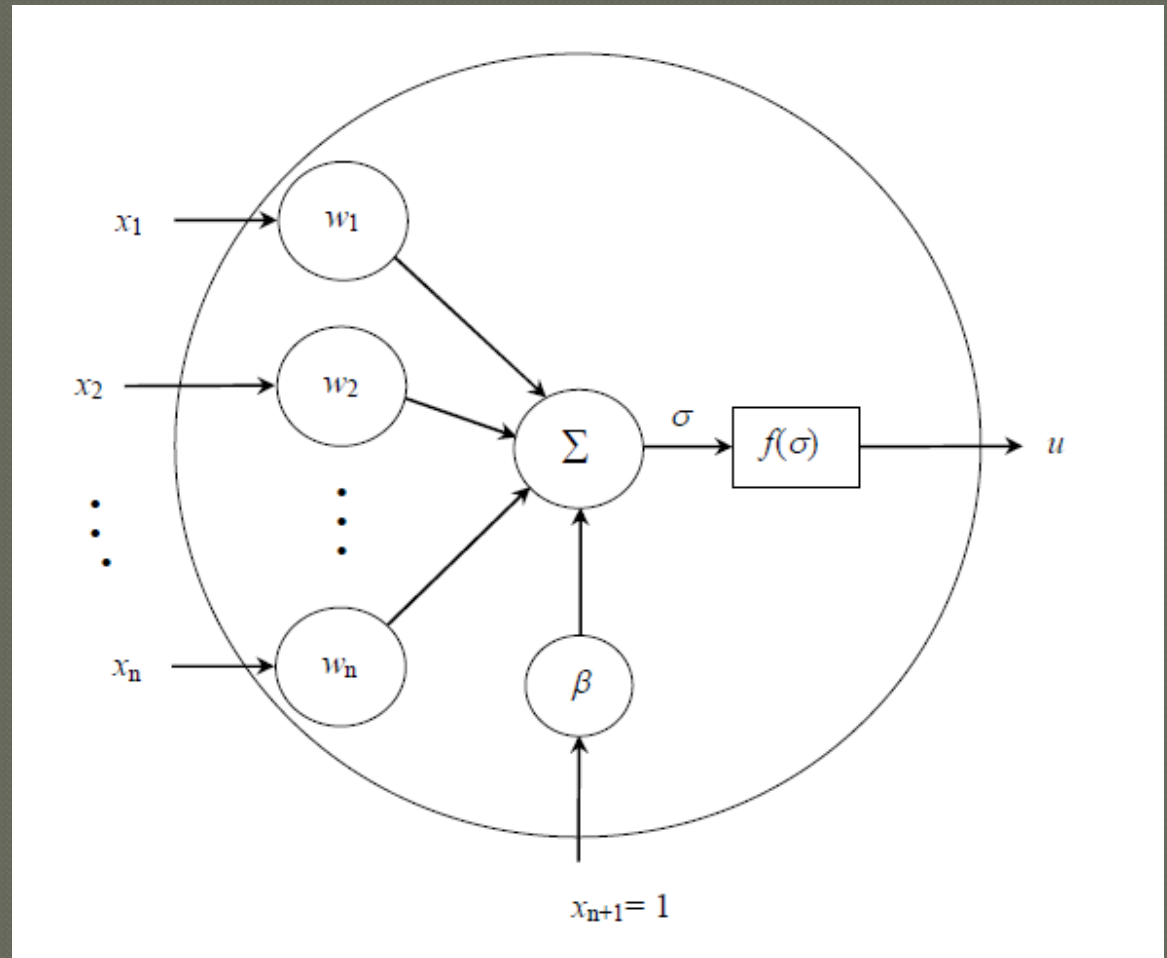


Δρ. Βασίλειος Γ. Καμπουρλάζος



# Τυπικός Νευρώνας

$$\sigma = \sum_{i=1}^n w_i x_i + \beta = \sum_{i=1}^{n+1} w_i x_i = \mathbf{w} \cdot \mathbf{x}$$



# Συναρτήσεις Ενεργοποίησης/Μεταφοράς

1) Γραμμική συνάρτηση μεταφοράς:  $f(\sigma) = \sigma$ .

2) Συνάρτηση μεταφοράς τύπου Perceptron:

$$f(\sigma) = \begin{cases} \sigma, & \sigma \geq 0 \\ 0 & \sigma < 0 \end{cases}$$

3) Δυαδικά (δίτιμη) συνάρτηση μεταφοράς με κατώφλι  $T$ :

$$f(\sigma) = \begin{cases} 1, & \sigma \geq T \\ 0 & \sigma < T \end{cases}$$

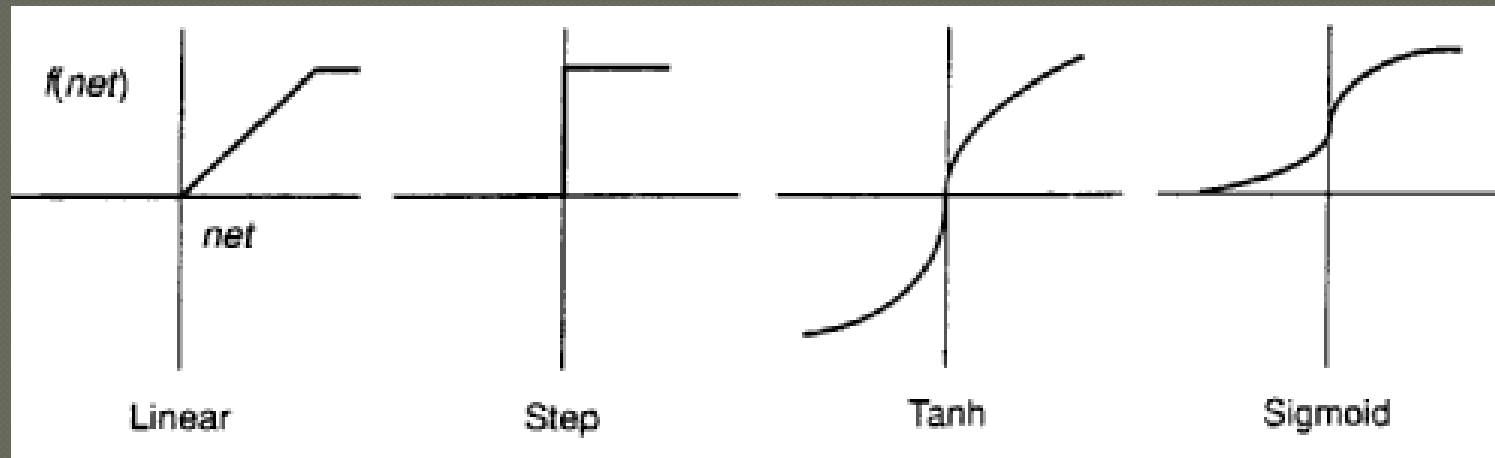
4) Συνάρτηση μεταφοράς τύπου υπερβολικής εφαπτομένης:

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

5) Σιγμοειδής (ή, Λογιστική) συνάρτηση μεταφοράς:

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

# Μορφή Συναρτήσεων Ενεργοποίησης



# Δομή ενός ΤΝΔ

---

- Αν και ένα ΤΝΔ μπορεί να αποτελείται από περισσότερα των τριών στρωμάτων νευρώνων, έχει αποδειχθεί ότι μια οποιαδήποτε συναρτησιακή σχέση μπορεί να υλοποιηθεί με 1 ΤΝΔ τριών στρωμάτων (1 κρυφό στρώμα) (Cybenko, 1989).

# Πλεονεκτήματα των ΤΝΔ

---

1. Μαθαίνουν από εμπειρία (δεδομένα – data) συναρτήσεις εισόδου-εξόδου.
2. Μπορούν να προσεγγίζουν μη-γραμμικές συναρτήσεις εισόδου-εξόδου.
3. Έχουν ανεκτικότητα σε βλάβες λόγω της παράλληλης δομής και λειτουργίας τους.
4. Έχουν την ικανότητα γενίκευσης.
5. Έχουν κατανομημένα και παράλληλη τοπολογία.
6. Μπορούν να επεξεργάζονται τόσο αριθμητικά όσο και γλωσσικά δεδομένα.

# Γνωστά Νευρωνικά Δίκτυα

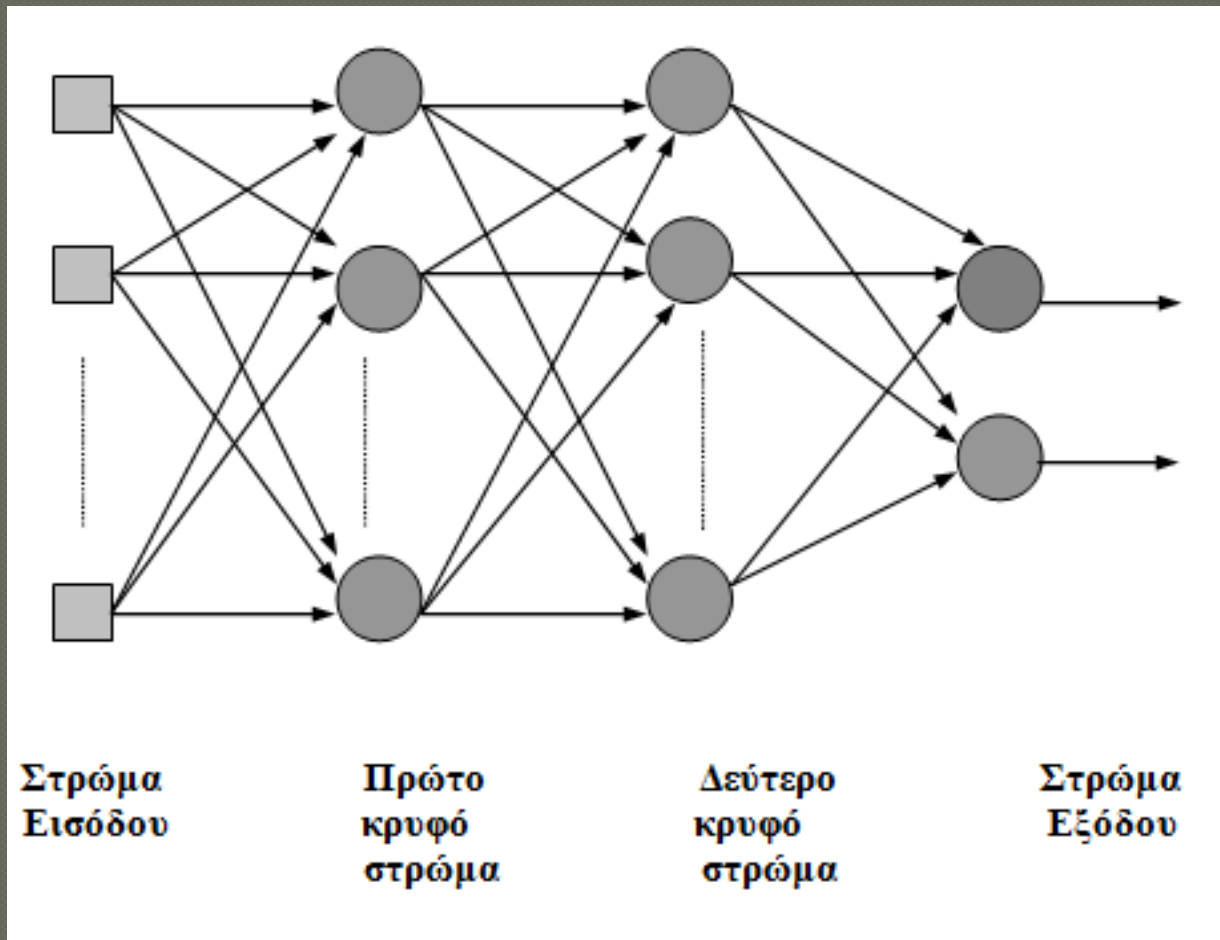
---

- ◉ Multi-layer Perceptron
- ◉ Self-organizing Maps
- ◉ Regression NNs
- ◉ Radial Basis NNs
- ◉ Recurrent NNs
- ◉ Time Delay NNs
- ◉ Polynomial NNs
- ◉ Spiking NNs

# Μάθηση στα ΤΝΔ

- Η μάθηση των ΤΝΔ πραγματοποιείται με τη μεταβολή των βαρών των συνδέσεων (συνάψεων) έτσι ώστε να ελαχιστοποιείται ένα καλώς ορισμένο σφάλμα.Κ
- Κύριοι τύποι μάθησης:
  - Με εποπτεία (**supervised**) – (with completely labeled training data)
  - Χωρίς εποπτεία (**unsupervised**) - (without any labeled training data)
  - Με λίγο εποπτεία (**semi-supervised learning**) - (with some labeled training data)
- Γνωστοί αλγόριθμοι μάθησης:
  - **Gradient-descent / steepest descent**
  - **Delta-rule**
  - **Backpropagation**
  - **Competitive learning**
  - **Heuristic learning**

# Perceptron Πολλαπλών Στρωμάτων

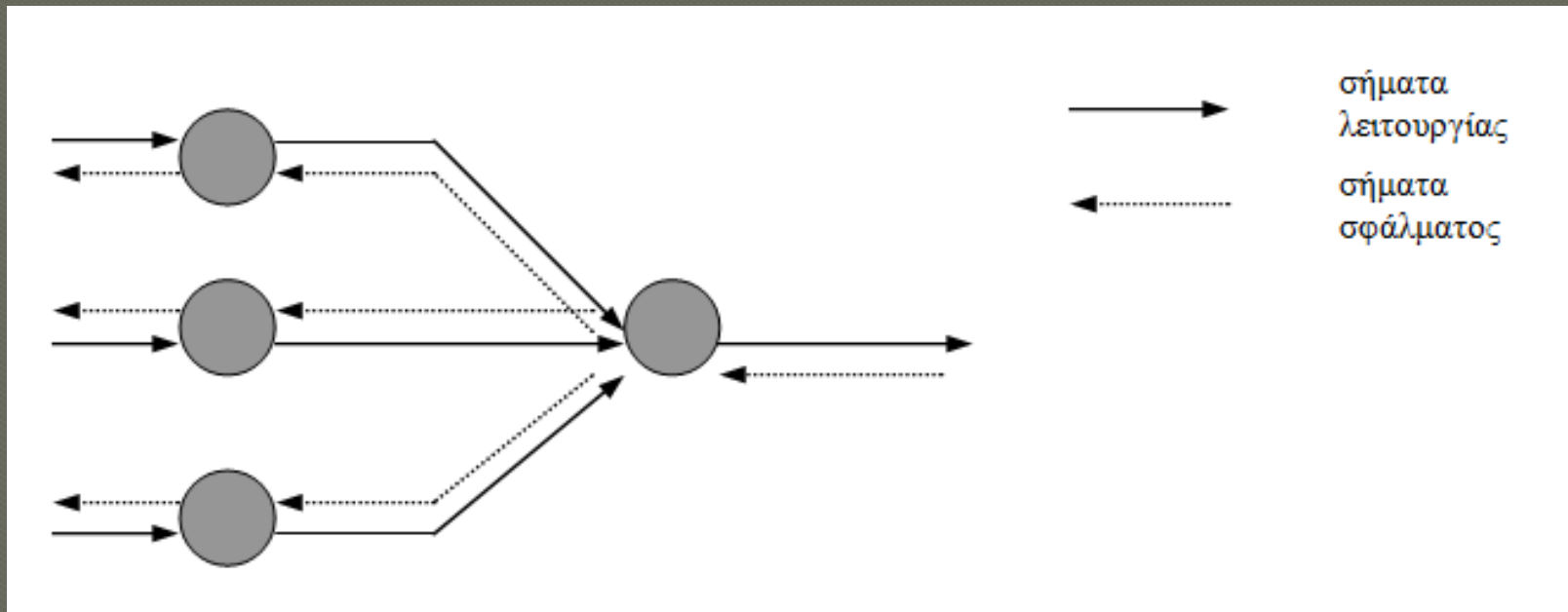




# Κανόνας Δέλτα (Delta Rule)

$$\begin{bmatrix} \text{Διόρθωση} \\ \text{Βάρους} \\ \Delta w_{ji}(n) \end{bmatrix} = \begin{bmatrix} \text{Παράμετρος} \\ \text{ρυθμού μάθησης} \\ \eta \end{bmatrix} \times \begin{bmatrix} \text{Τοπική} \\ \text{βαθμίδα} \\ \delta_j(n) \end{bmatrix} \times \begin{bmatrix} \text{Σήμα εισόδου} \\ \text{του νευρώνα } j \\ y_i(n) \end{bmatrix}$$

# Οπισθόδρομης Μάθησης (1/3) (Backpropagation)



# Οπισθόδρομης Μάθησης (2/3) (Backpropagation)

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2$$

$$\Delta_p w_{ji} = -\alpha \frac{\partial E_p}{\partial w_{ji}}$$

$$-\frac{\partial E_p}{\partial w_{ji}} = -\frac{\partial E_p}{\partial \sigma_{pj}} \frac{\partial \sigma_{pj}}{\partial w_{ji}}$$

$$\sigma_{pj} = \sum_k w_{jk} o_{pk}$$

# Οπισθόδρομης Μάθησης (3/3) (*Backpropagation*)

- ◎ Δύο τρόποι λειτουργίας:
  - **Batch mode \ offline**
    - Ανανέωση βαρών μετά την εμφάνιση όλων των δεδομένων εκπαίδευσης (*εποχή*).
    - Σταθερή λειτουργία.
    - Αργή λειτουργία.
  - **Incremental \ online**
    - Ανανέωση βαρών μετά την εμφάνιση κάθε δεδομένου εκπαίδευσης.
    - Ασταθής λειτουργία.
    - Λειτουργία σε πραγματικό χρόνο.

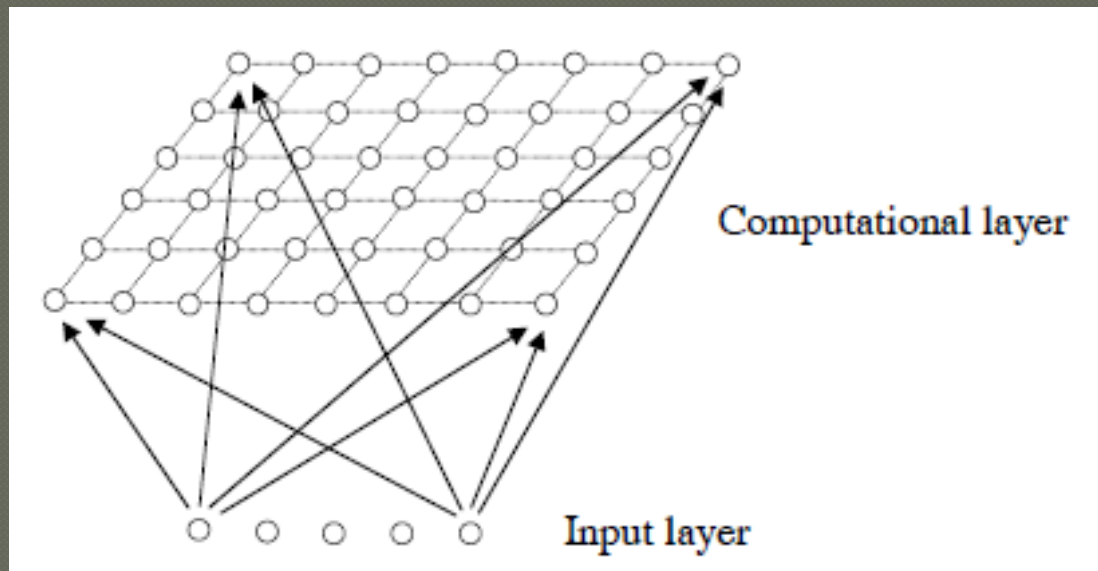
# Ανοιχτά θέματα

---

- ◉ Αρχικοποίηση των βαρών.
- ◉ Επιλογή του αριθμού των νευρώνων των στρωμάτων.
- ◉ Ερμηνεία της προκύπτουσας δομής.
- ◉ Ευστάθεια εκμάθησης

# Αυτό-οργανούμενοι Χάρτες (Self-organizing Maps)

- Προτάθηκαν από τον Kohonen (1980)
- Η λειτουργία τους βασίζεται στον τρόπο που διάφορα τμήματα του ανθρώπινου εγκέφαλου διαχειρίζονται διαφορετικές πληροφορίες των αισθητήρων (οπτικά, ακουστικά, κλπ).
- Μή εποπτευόμενη εκμάθηση (unsupervised)
- Ανταγωνιστική εκμάθηση (competitive learning)



# Ανταγωνιστική Εκμάθηση (Competitive Learning)

- Τα βάρη αρχικοποιούνται σε τυχαίες μικρές τιμές.
- Για κάθε δεδομένο εκπαιδευσης υπολογίζεται η Ευκλείδεια απόσταση του με όλα τα βάρη.
- Ο νευρώνας με βάρη που έχουν την μικρότερη απόσταση κερδίζει.
- Ο νικητής νευρώνας και οι γειτονικοί του νευρώνες τροποποιούν τα βάρη τους με βάση τον τύπο:

$$\mathbf{W}_v(s + 1) = \mathbf{W}_v(s) + \Theta(u, v, s) \alpha(s)(\mathbf{D}(t) - \mathbf{W}_v(s))$$

**W**: weight vector

**v**: neuron number

**s**: step/iteration

**t**: training index

$\Theta$ : neighborhood function

**D**: input data

$\alpha$ : learning coefficient/monotonically decr.

**u**: the winner neuron

# Σύγχρονες Τάσεις

## Βαθιά Μάθηση (Deep Learning)

---

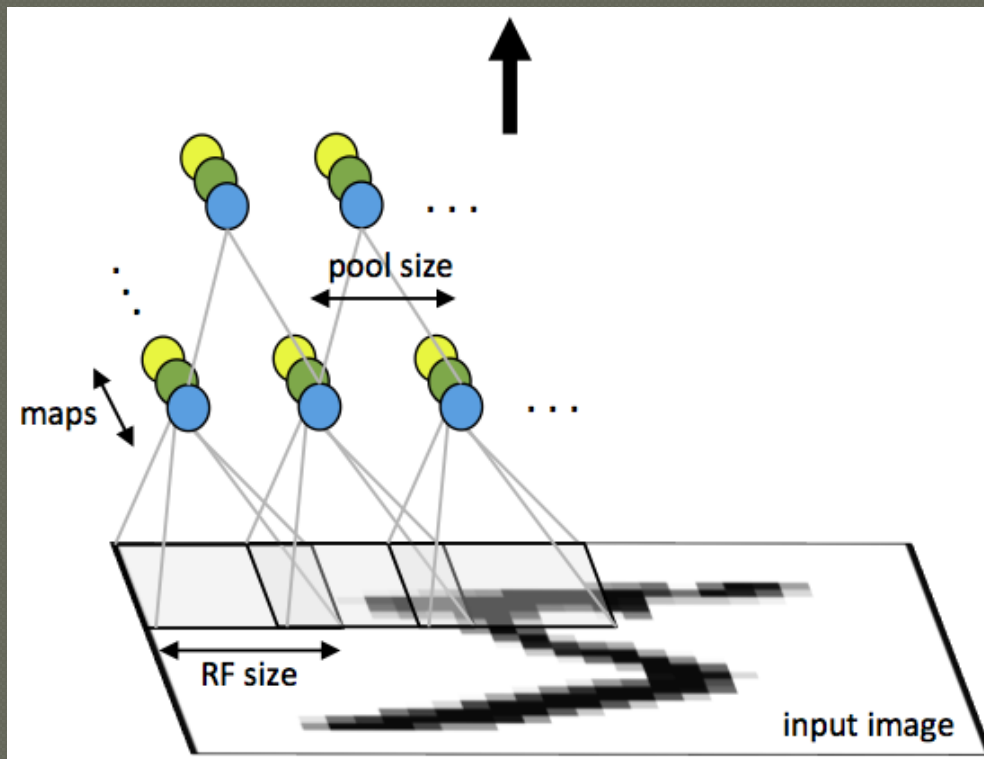
- Το χαρακτηριστικό της βαθιάς μάθησης είναι η χρήση υπολογιστικών αρχιτεκτονικών ΤΝΔ με *πολλαπλά στρώματα νευρώνων*.
- Έτσι, καθίσταται δυνατή η μάθηση αναπαραστάσεων με *πολλαπλά επίπεδα αφαίρεσης*.
- Ιδανικά για δεδομένα μεγάλης κλίμακας.
- Συνήθεις αρχιτεκτονικές:
  - **deep neural networks,**
  - **convolutional deep neural networks,**
  - **deep belief networks**
  - **recurrent neural networks**



# ΤΝΔ βαθιάς συνέλιξης (Convolutional Deep Neural Networks)

- Εφαρμόζονται κυρίως στην Τεχνητή Όραση.
- Εκπαιδεύονται εύκολα (**backpropagation**).
- Έχουν λιγότερες παραμέτρους που θα πρέπει να βελτιστοποιηθούν σε σύγκριση με τα **full connected NNs**.
- Όλοι οι κρυφοί νευρώνες έχουν τα ίδια βάρη και πολώσεις (**bias**).
- Κάνουν χρήση των εννοιών: **receptive field**, **feature map**, **pooling**.
- Γνωστό **CNN**: **LeNet-5**, used 6 feature maps, each associated to a **5×5 local receptive field**, to recognize **MNIST digits**

# Δομή ενός CNN



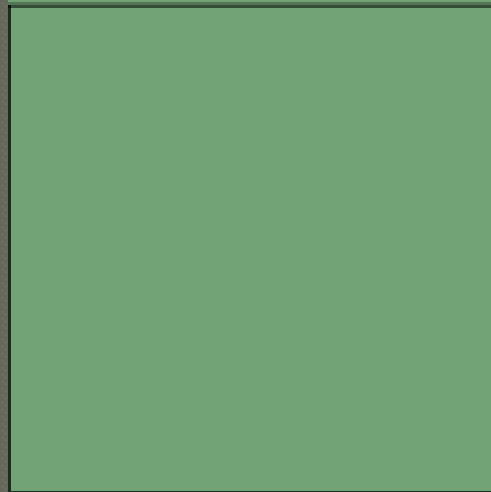
Έξοδος κρυφού  
νευρώνα

$$\sigma \left( b + \sum_{l=0}^4 \sum_{m=0}^4 w_{l,m} a_{j+l,k+m} \right)$$

# CNN για το MNIST πρόβλημα

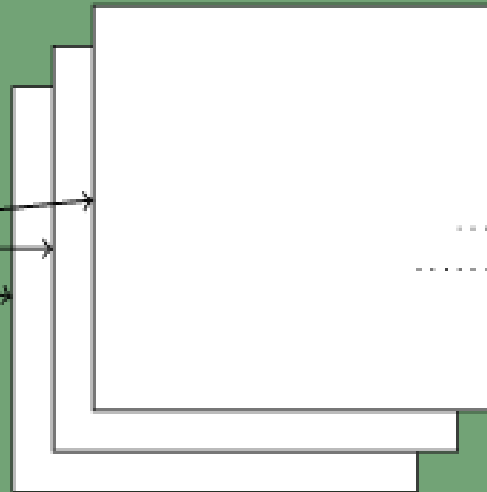
Αρχική  
εικόνα

$28 \times 28$



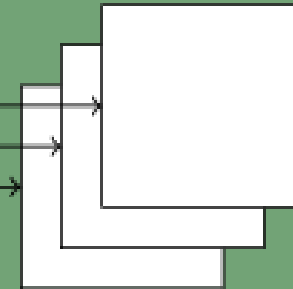
Επίπεδο  
συνέλιξης

$3 \times 24 \times 24$



Επίπεδο  
pooling

$3 \times 12 \times 12$



Επίπεδο  
εξόδου



# Προγραμματισμός σε **MATLAB**

---

# Χαρακτηριστικά του MATLAB

- Πρόκειται για ένα IDE (Integrated Development Environment).
  - Editor
  - Compiler (Interpreter)
  - Debugger
- Επιστημονικά αποδεκτό.
- Κάθε μεταβλητή αναπαρίσταται ως πίνακας.
- Προγραμματισμός C-like.
- Συλλογή βιβλιοθηκών συναρτήσεων (toolboxes) ανά επιστημονικό πεδίο.
- Δυνατότητα δημιουργίας standalone εφαρμογής.
- Δυνατότητα κλήσεων συναρτήσεων (wrapper) από άλλες εφαρμογές.

# Χρήσιμες Βιβλιοθήκες

---

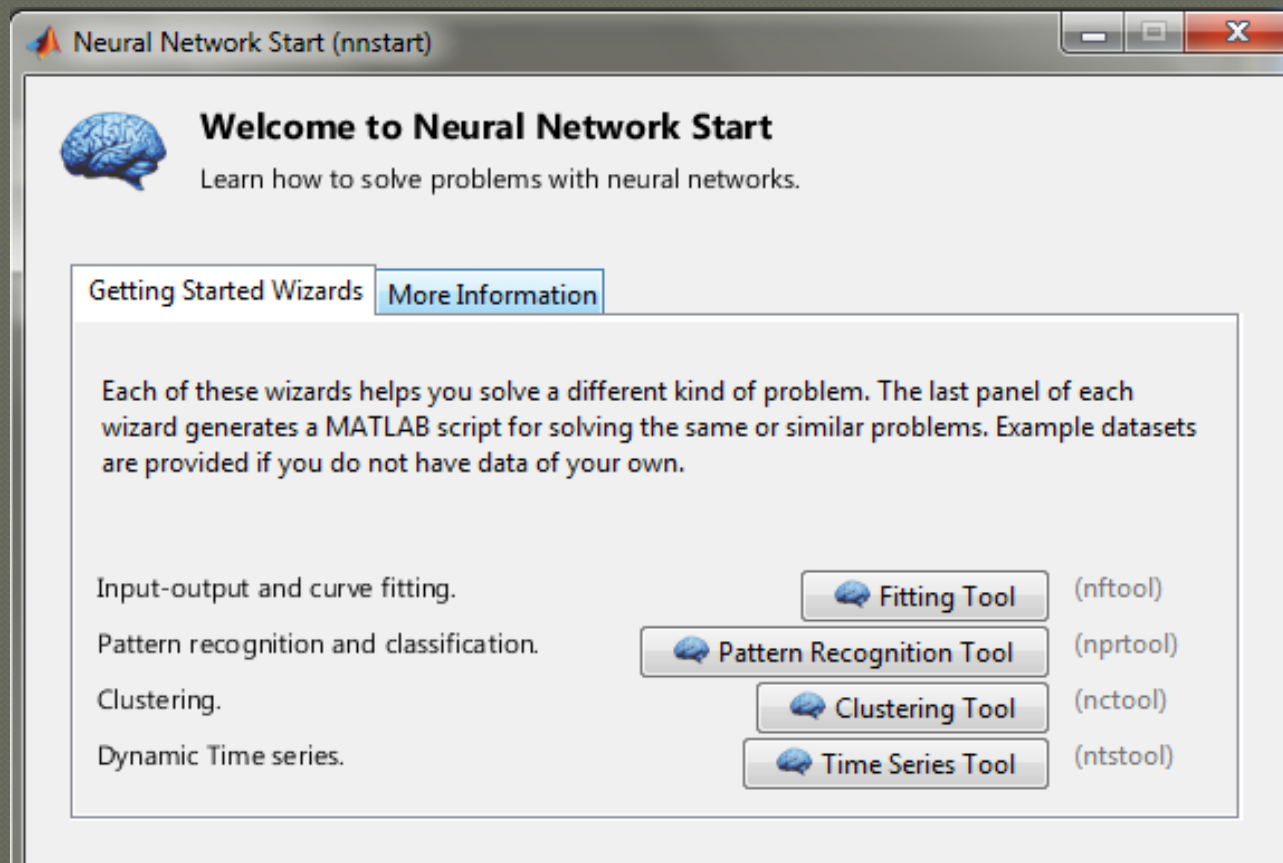
- ◉ Neural Network Toolbox
- ◉ Fuzzy Systems Toolbox
- ◉ Global Optimization Toolbox
  - Genetic Algorithm
  - Particle Swarm Optimization

# ΥΝ και MATLAB

---

# Εργαλείο του MATLAB για NN

- Πληκτρολογούμε **'nnstart'**.





# Δύο Εφαρμογές

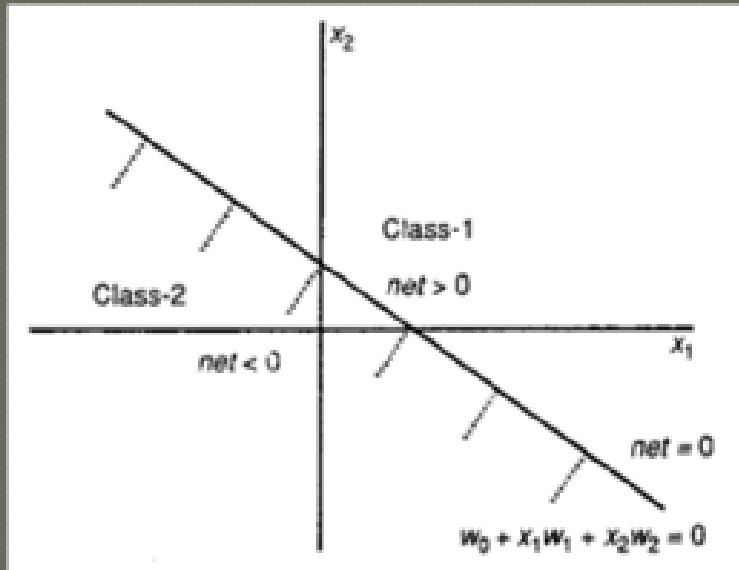
## FUNCTION APPROXIMATION

- Μαθηματική συνάρτηση μίας μεταβλητής ( $\sin(x)$ )
- ΤΝΔ : 1 - X - 1

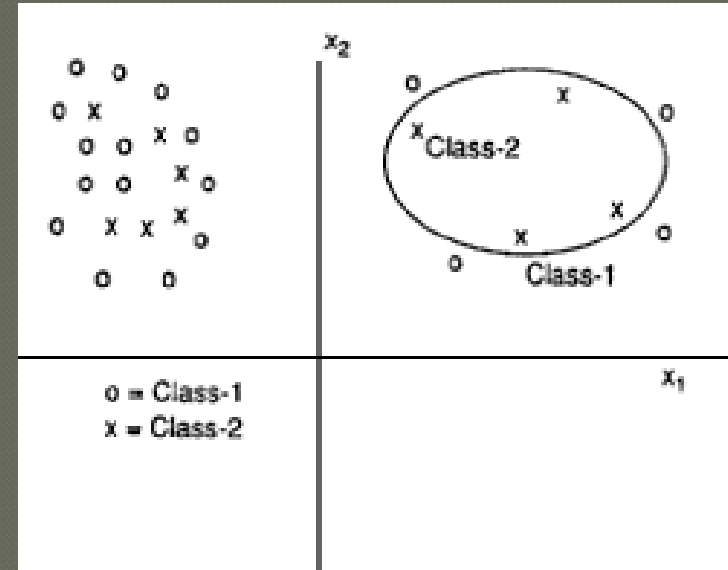
## PATTERN CLASSIFICATION

- Ταξινόμηση 3 λουλουδιών
- Χρήση 4 χαρακτηριστικών
- 150 δείγματα (50/κλάση)
- ΤΝΔ: 4 - X - X - 3

# Αναγνώριση Προτύπων



Γραμμικά  
διαχωρίσιμες  
κατηγορίες



Μη-γραμμικά  
διαχωρίσιμες  
κατηγορίες

# Source Code 1 – Function Approximation

```
%% Αρχικοποίηση του περιβάλλοντος MATLAB
clear all; close all; clc;

%% Κατασκευή των δεδομένων εκπαίδευσης (training data)
x_train = 0:pi/16:2*pi;
y_train = sin(x_train);
plot(x_train, y_train); % Προβολή των δεδομένων εκπαίδευσης

%% Κατασκευή των δεδομένων δοκιμής (testing data)
x_test = 0:pi/7:2*pi;
y_test = sin(x_test);

%% Κατασκευή ενός feedforward νευρωνικού δικτύου (1 κρυφό στρώμα 50 νευρώνων)
net = newff(x_train, y_train, 50);

%% Εκπαίδευση του νευρωνικού δικτύου
net = train(net, x_train, y_train);

%% Υπολογισμός των εξόδων του νευρωνικού δικτύου, με εισόδους τα δεδομένα
% δοκιμής
y_net = net(x_test);

%% Προβολή της εξόδου του νευρωνικού δικτύου σε σύγκριση με την επιθυμητή
% έξοδο
figure;
plot(x_test, y_test, x_test, y_net, '--r');
legend('Training Sine', 'Simulated Sine');
```

# Source Code 2 – Pattern Classification

```
%% Αρχικοποίηση του περιβάλλοντος MATLAB
clear all; close all; clc;

%% Εισαγωγή δεδομένων iris data
load fisheriris

%% Κατασκευή των προτύπων
y = zeros(3,length(species));
y(1,1:50) = 1; y(2,51:100) = 1; y(3,101:150) = 1;

%% Κατασκευή δεδομένων εκπαίδευσης (70%) και δοκιμής (30%)
inds = randperm(size(meas,1));
x_train = meas(inds(1:105),1:4)'; y_train = y(:,inds(1:105));
x_test = meas(inds(106:end),1:4)'; y_test = y(:,inds(106:end));

%% Κατασκευή ενός feedforward νευρωνικού δικτύου (2 κρυφά στρώματα των 20 νευρώνων)
net = newff(x_train,y_train,[20 20]);

%% Εκπαίδευση του νευρωνικού δικτύου
net = train(net,x_train,y_train);

%% Υπολογισμός των εξόδων του νευρωνικού δικτύου, με εισόδους τα δεδομένα
%δοκιμής
y_net = net(x_test);

%% Υπολογισμός της απόδοσης ταξινόμησης
[vals1,y_labels] = max(y_test);
[vals2,y_net_labels] = max(y_net);
cp = classperf(y_labels, y_net_labels);

%% Εκτύπωση της ποσοστιαίας ακρίβειας ταξινόμησης
disp(['Classification Rate(%) = ' num2str(cp.CorrectRate*100)]);
```

# Στοιχεία Επικοινωνίας

---

Δρ. Βασίλειος Γ. Καμπουράζος

[vgkabs@teiemt.gr](mailto:vgkabs@teiemt.gr)

Τηλ. 2520 462 320

Γραφείο Β 1 22 (Κτήριο βιβλιοθήκης)

---

ΕΥΧΑΡΙΣΤΩ