



RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων
Δεδομένων Παρατήρησης Γης

Εισαγωγή στη Βαθιά Μάθηση

Αθηνά Ψάλτα
Βασίλειος Τσιρώνης
Κωνσταντίνος Καράντζαλος

Εαρινό εξάμηνο 2022

Περιεχόμενα σημερινού μαθήματος

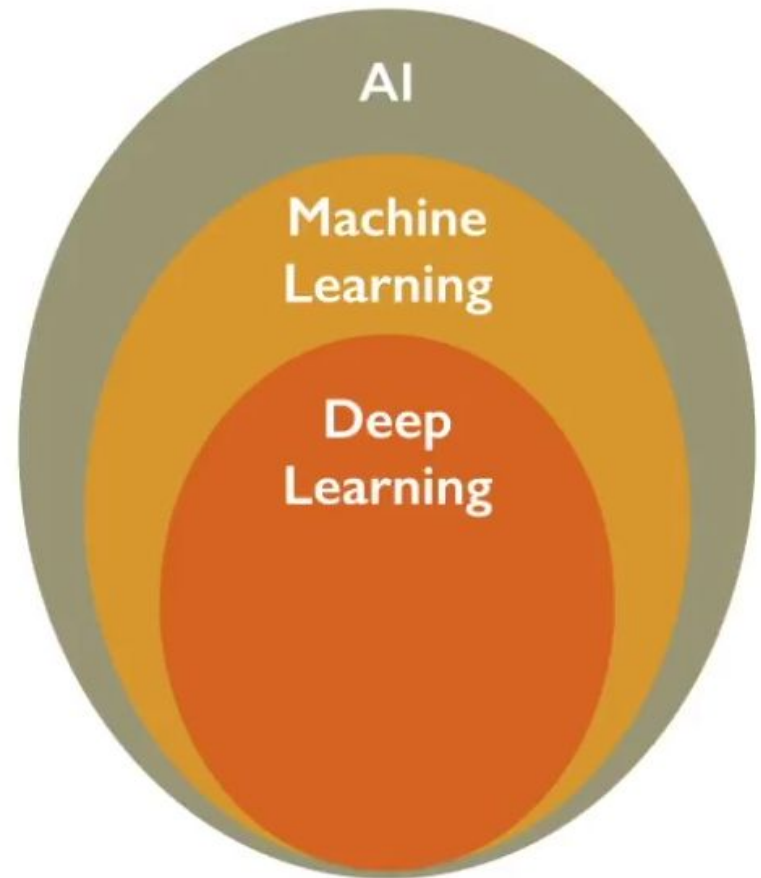
1. Μηχανική Μάθηση: μία σύντομη εισαγωγή
2. Αλγόριθμοι ταξινόμησης Μηχανικής Μάθησης
3. Εισαγωγή στη Βαθιά Μάθηση
 - a. Τι είναι ένα Perceptron;
 - b. Multi-layer Perceptrons (MLPs)
4. Συναρτήσεις κόστους

Μηχανική Μάθηση

Κάθε τεχνική που επιτρέπει σε
ένα πρόγραμμα Η/Υ να
“μιμείται” την ανθρώπινη
συμπεριφορά

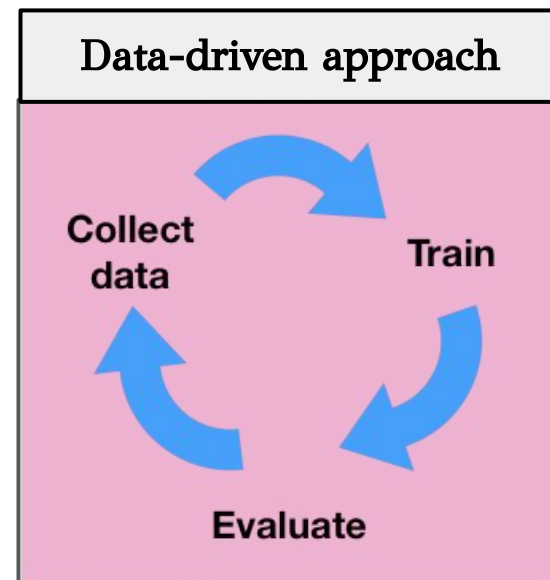
Προγράμματα Η/Υ που
“μαθαίνουν” να επιτελούν κάποια
εργασία χωρίς να έχουν ρητά
προγραμματιστεί για αυτό

Προγράμματα Η/Υ που
μαθαίνουν από τα πρωτογενή
δεδομένα αναπαραστάσεις
υψηλού επιπέδου με τη χρήση
τεχνητών νευρωνικών δικτύων

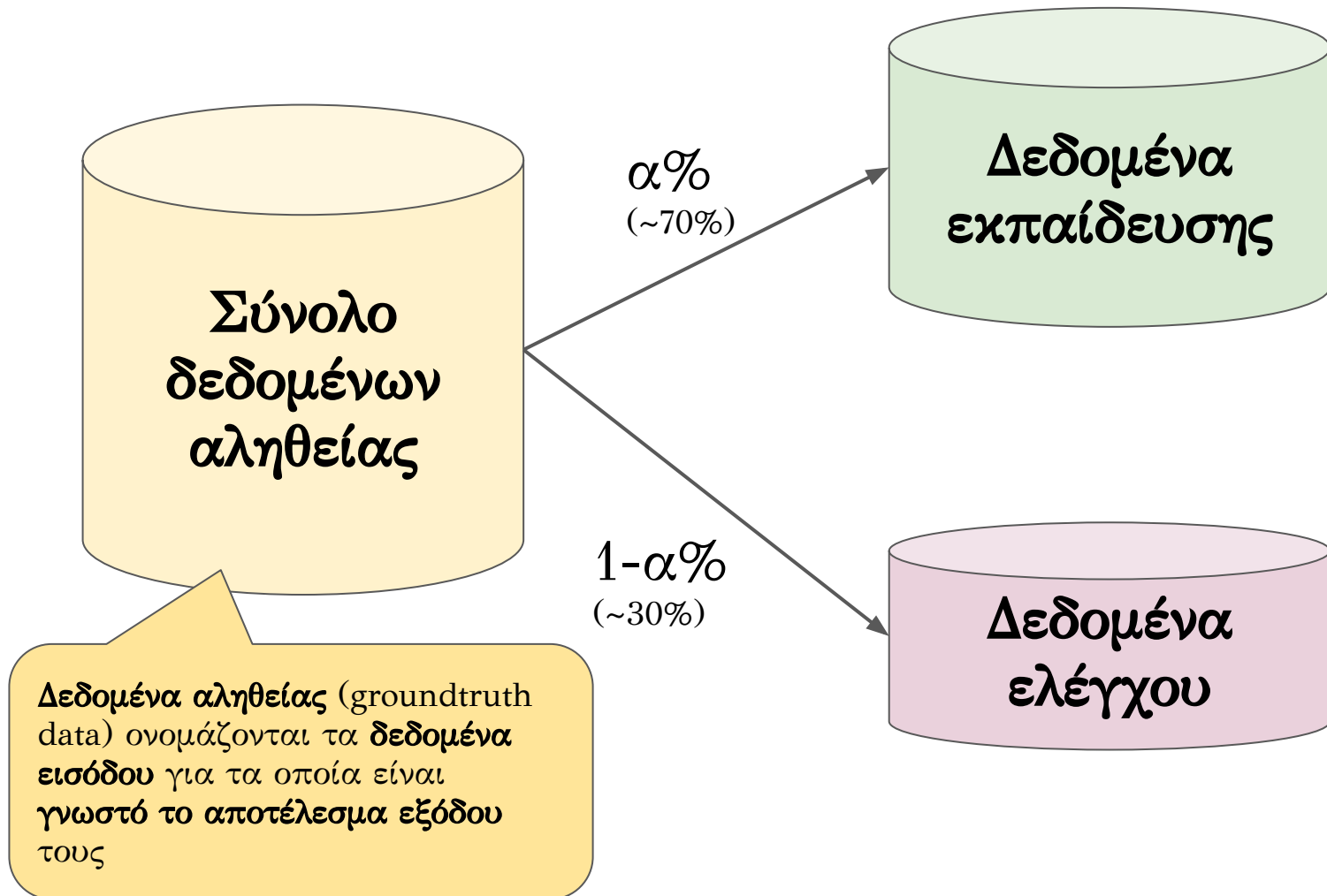


Μηχανική Μάθηση (Machine Learning)

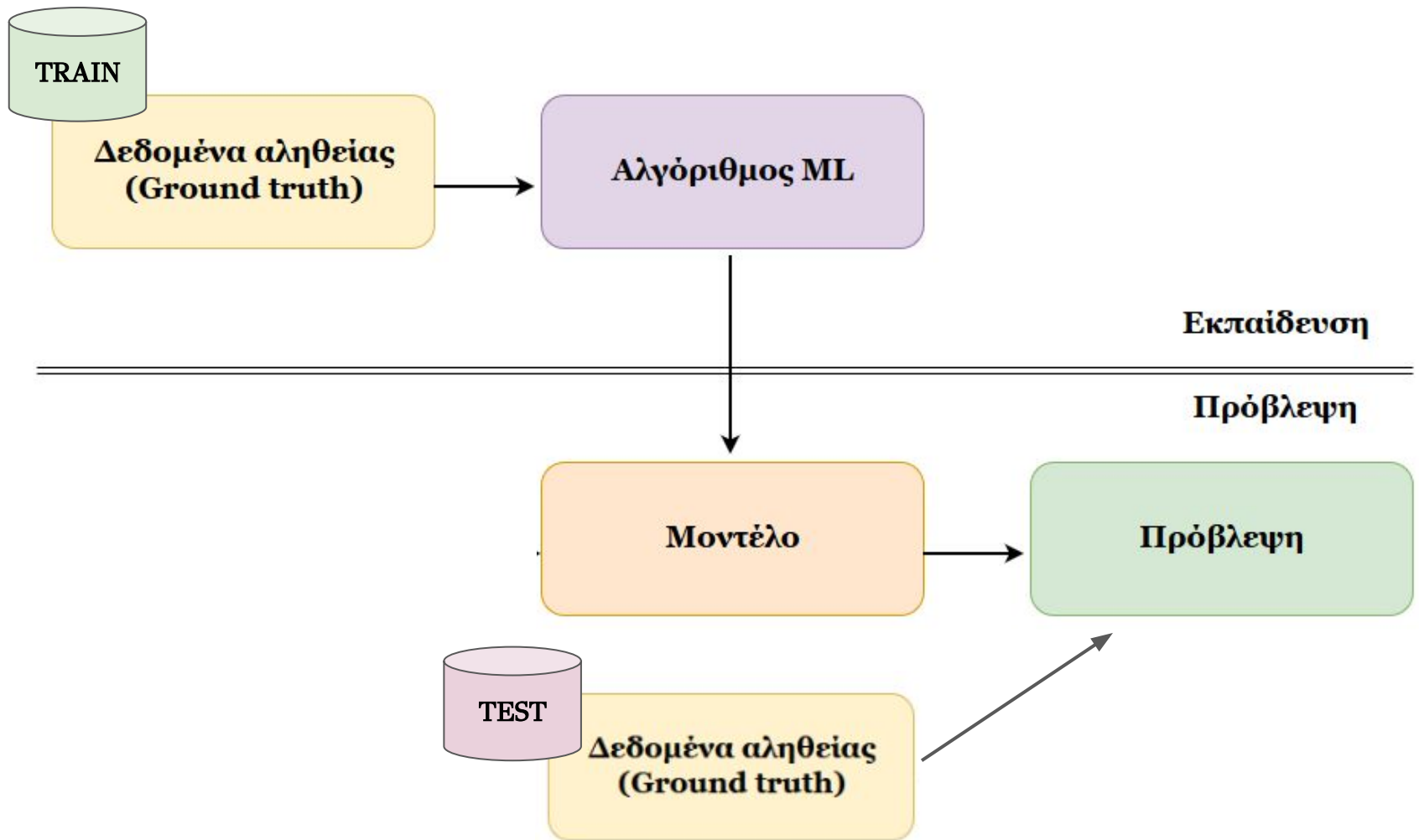
- “Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί” - Arthur Samuel, 1959
- Αλγόριθμοι που αναλύουν τα δεδομένα, μαθαίνουν από αυτά **μόνοι** τους και εφαρμόζουν αυτά που έμαθαν στη λήψη τεκμηριωμένων αποφάσεων.
 - **Επιβλεπόμενη - Supervised** → Παλινδρόμηση (R), Ταξινόμηση (Ca)
 - **Μη επιβλεπόμενη - Unsupervised** → Ομαδοποίηση (Clustering - Cu), Μείωση διάστασης (Dimensionality Reduction - DR)
 - **Ενισχυτική - Reinforcement**
- Παραδείγματα ML αλγορίθμων
 - Random Forests & Decision Trees (Ca)
 - Linear Regression (R)
 - Logistic Regression (Ca)
 - Support Vector Machines (Ca, R)
 - k-Means, DB-Scan, IsoData (Cu)
 - PCA, ICA (DR)



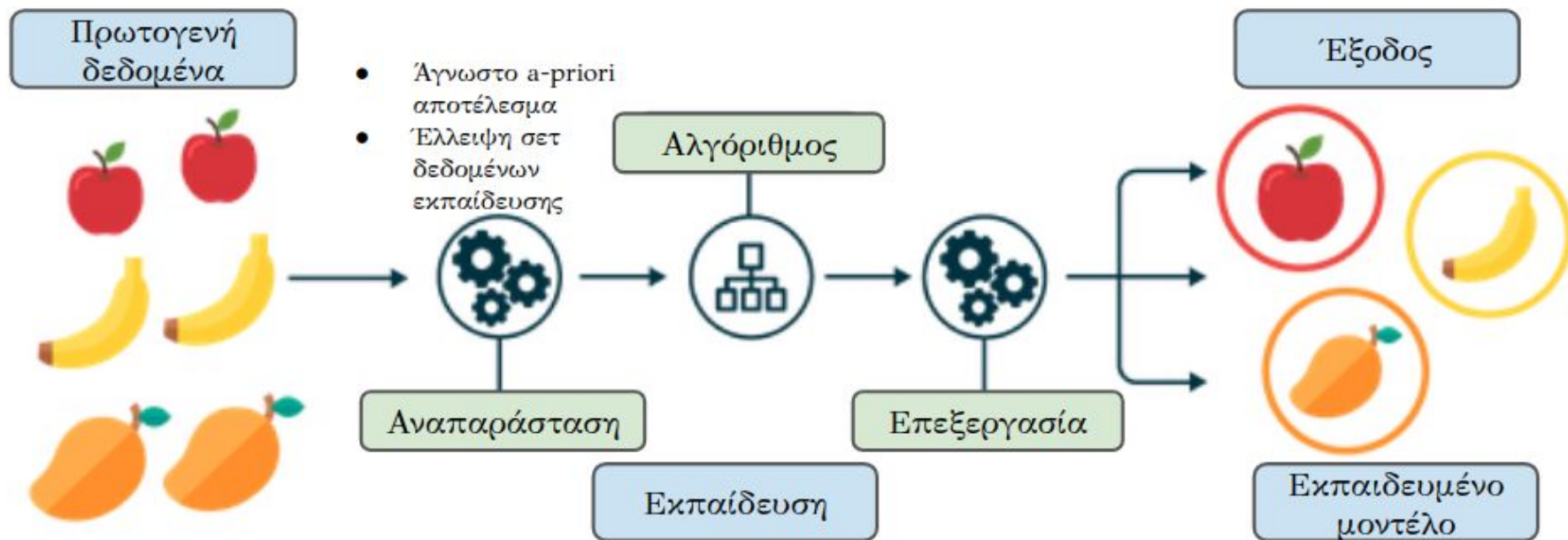
Επιβλεπόμενη Μάθηση (Supervised Learning)



Επιβλεπόμενη Μάθηση (Supervised Learning)



Μη-Επιβλεπόμενη Μάθηση (Unsupervised Learning)



Επιβλεπόμενη Ταξινόμηση

Δημοφιλείς ταξινομητές

Επιβλεπόμενη Ταξινόμηση

Ορισμός : Το πρόβλημα κατηγοριοποίησης μιας νέας παρατήρησης σε ένα σύνολο προκαθορισμένων κατηγοριών (classes) δεδομένου ενός συνόλου δεδομένων εκπαίδευσης (training set) που περιέχουν παρατηρήσεις των οποίων η κατηγορία είναι γνωστή εκ των προτέρων.

- Ίσως το πλέον διαδεδομένο πρόβλημα της Μηχανικής Μάθησης
 - Δυαδική Ταξινόμηση (Binary Classification)
 - Ταξινόμηση σε πολλαπλές κατηγορίες (Multiclass Classification)

Ενδεικτικοί αλγόριθμοι :

- Naive Bayes
- Support Vector Machines
- k-Nearest Neighbor
- Boosting
- Random Forests
- Perceptron
- Neural Networks
-

Naive Bayes Classifiers

- Οικογένεια “πιθανοτικών” ταξινομητών
- Βασίζονται στο **Θεώρημα του Bayes**
 - Περιγράφει τη πιθανότητα ενός γεγονότος βάσει πρότερης γνώσης συνθηκών που πιθανόν να σχετίζονται με το γεγονός
- Οι τυχαίες μεταβλητές που συνθέτουν τα χαρακτηριστικά (features) θεωρούνται στατιστικά **ανεξάρτητες** μεταξύ τους

The diagram shows the formula for Bayes' Theorem: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from the labels to the corresponding parts of the formula: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

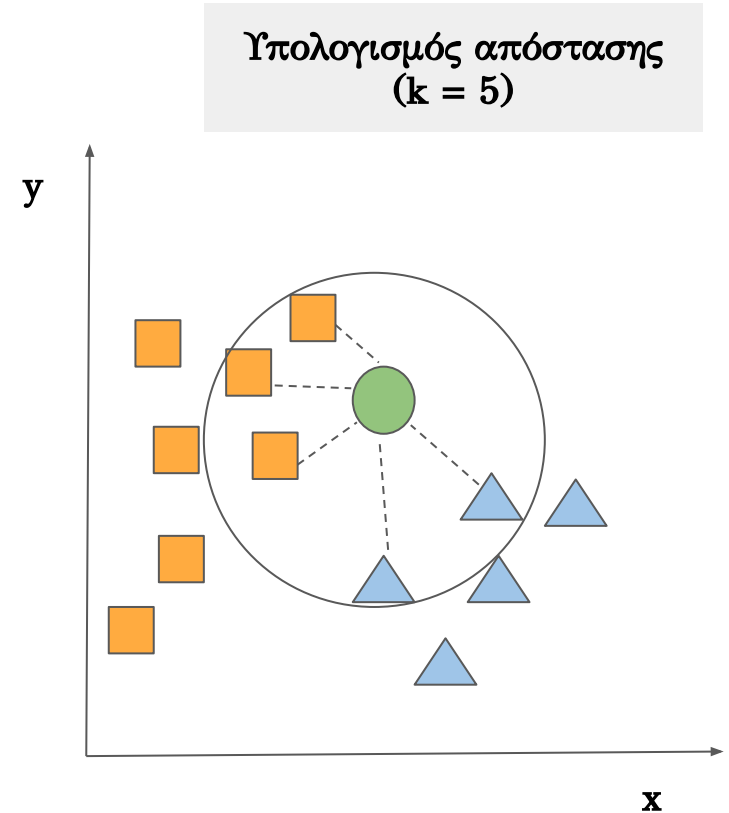
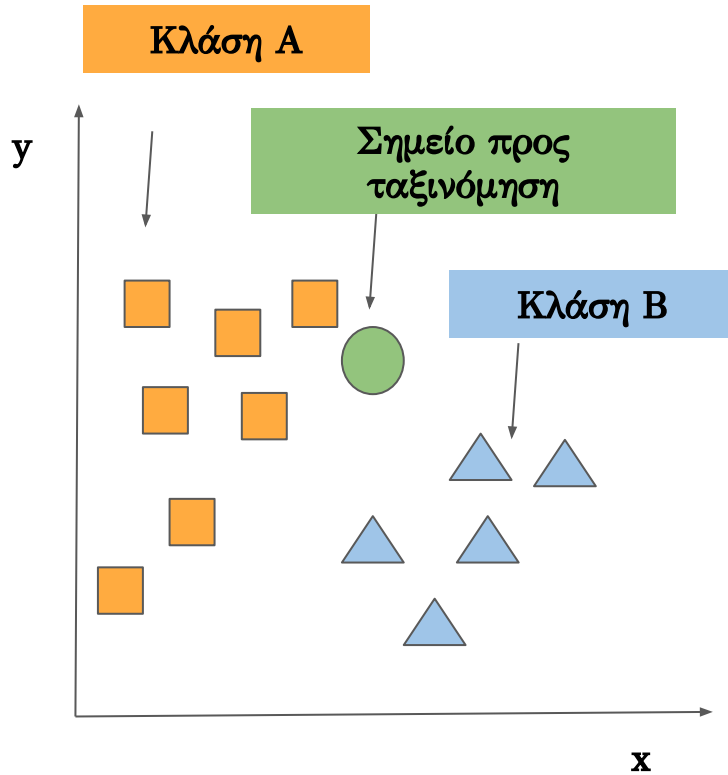
$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$

- Οι Naive Bayes ταξινομητές έχουν υψηλή κλιμακωσιμότητα απαιτώντας αριθμό παραμέτρων ευθέως ανάλογο του αριθμού των μεταβλητών.
- Η εκπαίδευσή τους με τη μέθοδο της μεγίστης πιθανοφάνειας (maximum likelihood) πραγματοποιείται μέσω ενός κλειστού τύπου σε γραμμικό χρόνο.
 - Σημαντικό πλεονέκτημα έναντι άλλων ταξινομητών που εκπαιδεύονται με επαναληπτικούς αλγορίθμους προσέγγισης.

k-Nearest Neighbor (kNN)

- Αποτελεί μια **μη παραμετρική** μέθοδο ταξινόμησης/παλινδρόμησης
 - Ταξινόμηση : κάθε “σημείο” του σετ δεδομένων κατηγοριοποιείται στην πιο συχνή κατηγορία των k-γειτόνων του.
- Ο k-NN ταξινομητής ανήκει στη κατηγορία **instance-based learning** (lazy learning), όπου η συνάρτηση απόφασης ταξινόμησης εκτιμάται σε τοπικό επίπεδο κατά την ταξινόμηση του εκάστοτε σημείου.
- Οι k-γείτονες υπολογίζονται ανά “σημείο” προς ταξινόμηση ως τα k-κοντινότερα “σημεία” του σετ εκπαίδευσης υπό κάποια μετρική απόστασης.
- Μία χρήσιμη τεχνική είναι ο ορισμός βαρών στη συμμετοχή των γειτόνων ανάλογα με την απόστασή τους από το “σημείο”.

k-Nearest Neighbor (kNN)



k-Nearest Neighbor (kNN)

Πλεονεκτήματα

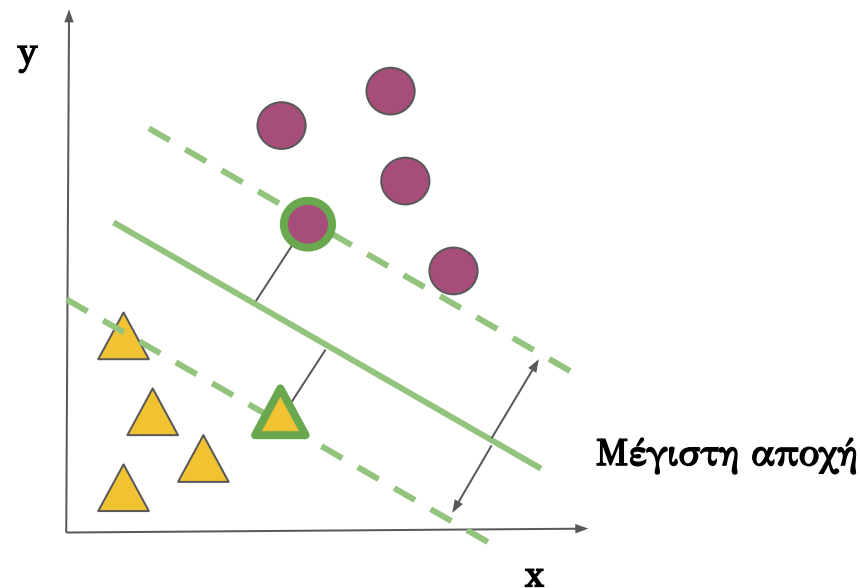
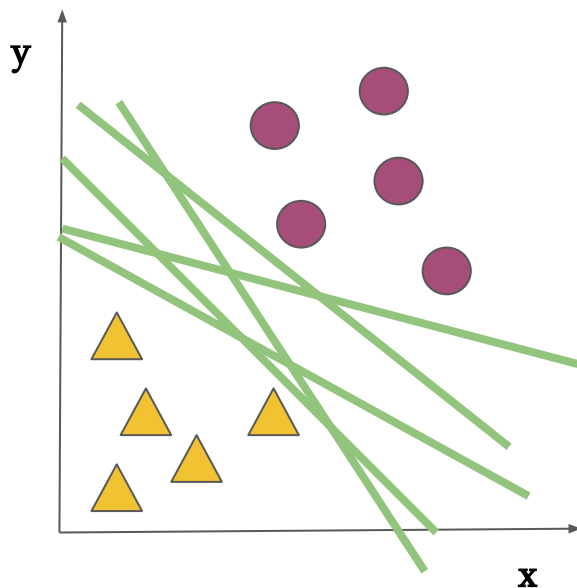
- Απλός και εύκολος στην υλοποίηση αλγόριθμος
- Δεν απαιτείται εκτίμηση παραμέτρων/βελτιστοποίηση
- Μπορεί να χρησιμοποιηθεί για ταξινόμηση και παλινδρόμηση

Μειονεκτήματα

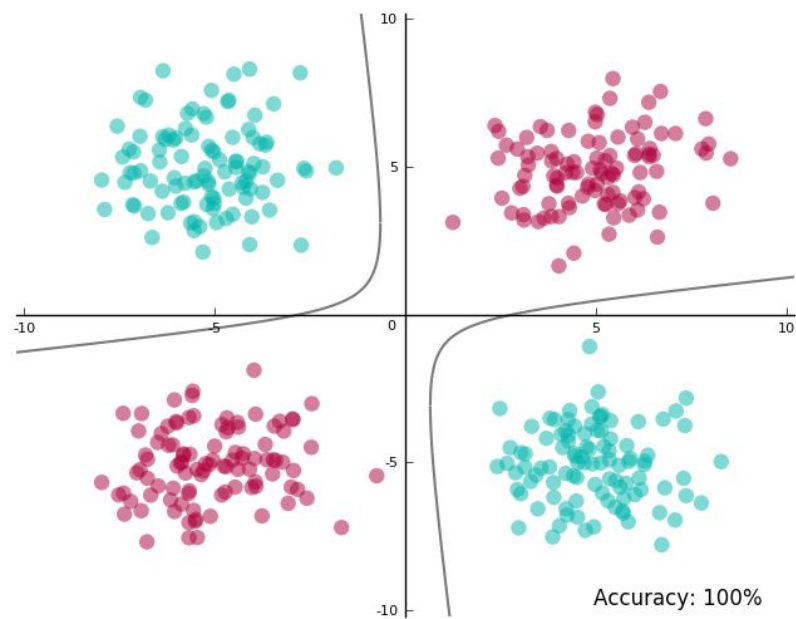
- Το υπολογιστικό κόστος αυξάνεται ανάλογα με το πλήθος των δεδομένων ή/και των ανεξάρτητων μεταβλητών.
- Υπερευαίσθητος στη κατανομή του σετ εκπαίδευσης
- Η ορθή ταξινόμηση ενός σημείου εξαρτάται από την ύπαρξη “κοντινών” σημείων στο σετ εκπαίδευσης

Γραμμικά Support Vector Machines

Στόχος είναι η εύρεση εκείνου του **υπερεπιπέδου** (hyperplane) του N -διάστατου χώρου, όπου N το πλήθος των χαρακτηριστικών (features), ως προς το οποίο η απόσταση των κοντινότερων ανά κατηγορία σημείων είναι μέγιστη.



Γραμμικά Support Vector Machines



Πηγή : <https://blog.statsbot.co/support-vector-machines-tutorial-c1618e635e93>

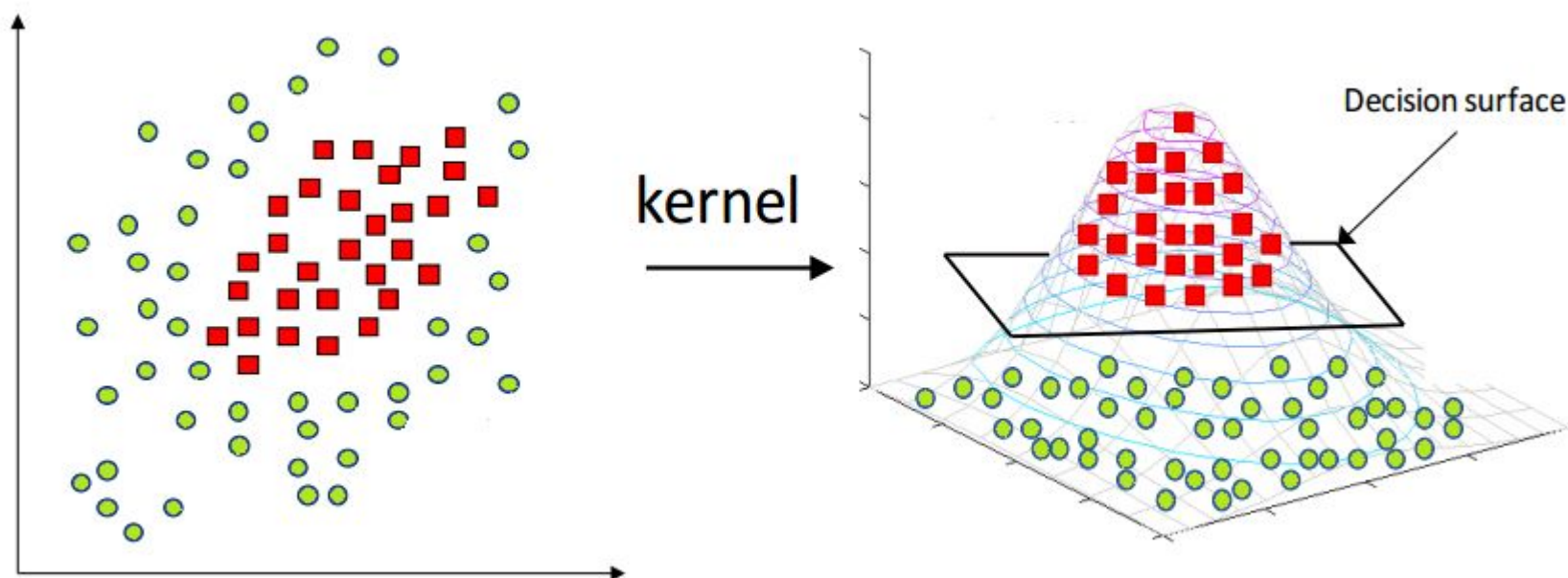
Support Vector Machines : Kernels

Οι αλγόριθμοι SVM μπορούν να χρησιμοποιηθούν και ως μη γραμμικοί ταξινομητές αξιοποιώντας τη λεγόμενη συνάρτηση “**πυρήνα**” (**kernel**) που αναφέρεται σε μία συνάρτηση μετασχηματισμού των δεδομένων εισόδου. Χαρακτηριστικά παραδείγματα τέτοιων συναρτήσεων είναι :

- Γραμμικές συναρτήσεις
- Πολυωνυμικές συναρτήσεις
- Συναρτήσεις ακτινικής βάσης (RBF)
- Σιγμοειδής συνάρτηση, κλπ.

Η συνάρτηση “πυρήνα” ουσιαστικά επιστρέφει το **εσωτερικό γινόμενο** μεταξύ δύο σημείων σε έναν κατάλληλο χώρο προτύπων. Κατά αυτό τον τρόπο ορίζονται μέτρα ομοιότητας με χαμηλό υπολογιστικό κόστος ακόμη και σε χώρους προτύπων πολλών διαστάσεων.

Support Vector Machines : Kernels



Πηγή : <https://www.hackerearth.com/blog/machine-learning/simple-tutorial-svm-parameter-tuning-python-r/>

Support Vector Machines

Πλεονεκτήματα

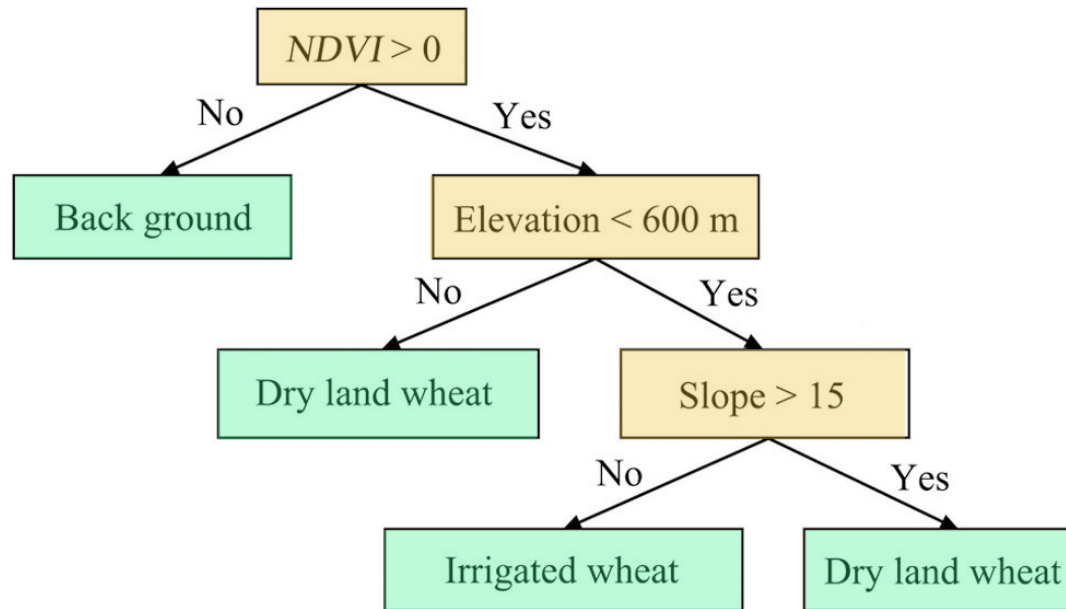
- Αποτελεσματική μέθοδος σε χώρους πολλαπλών διαστάσεων
- Αποτελεσματική μέθοδος ακόμη και σε περιπτώσεις όπου το πλήθος των διαστάσεων του χώρου προτύπων είναι μεγαλύτερο από το πλήθος των δειγμάτων
- Αποδοτική μέθοδος αφού χρησιμοποιεί ένα υποσύνολο των δεδομένων εκπαίδευσης στη συνάρτηση απόφασης (decision function)
- Μπορούμε να ορίσουμε διαφορετικές συναρτήσεις “πυρήνα” για τη συνάρτηση απόφασης.

Μειονεκτήματα

- Αν το πλήθος των χαρακτηριστικών είναι πολύ μεγαλύτερο από τον αριθμό των δειγμάτων τότε η επιλογή της συνάρτησης “πυρήνα” είναι καθοριστική για να αποφευχθούν προβλήματα υπερπροσαρμογής (overfitting)
- Δεν υπολογίζονται απευθείας οι πιθανότητες της κάθε πρόβλεψης, αλλά απαιτούνται περαιτέρω σύνθετοι υπολογισμοί που κοστίζουν υπολογιστικά.

Decision Tree Classifier

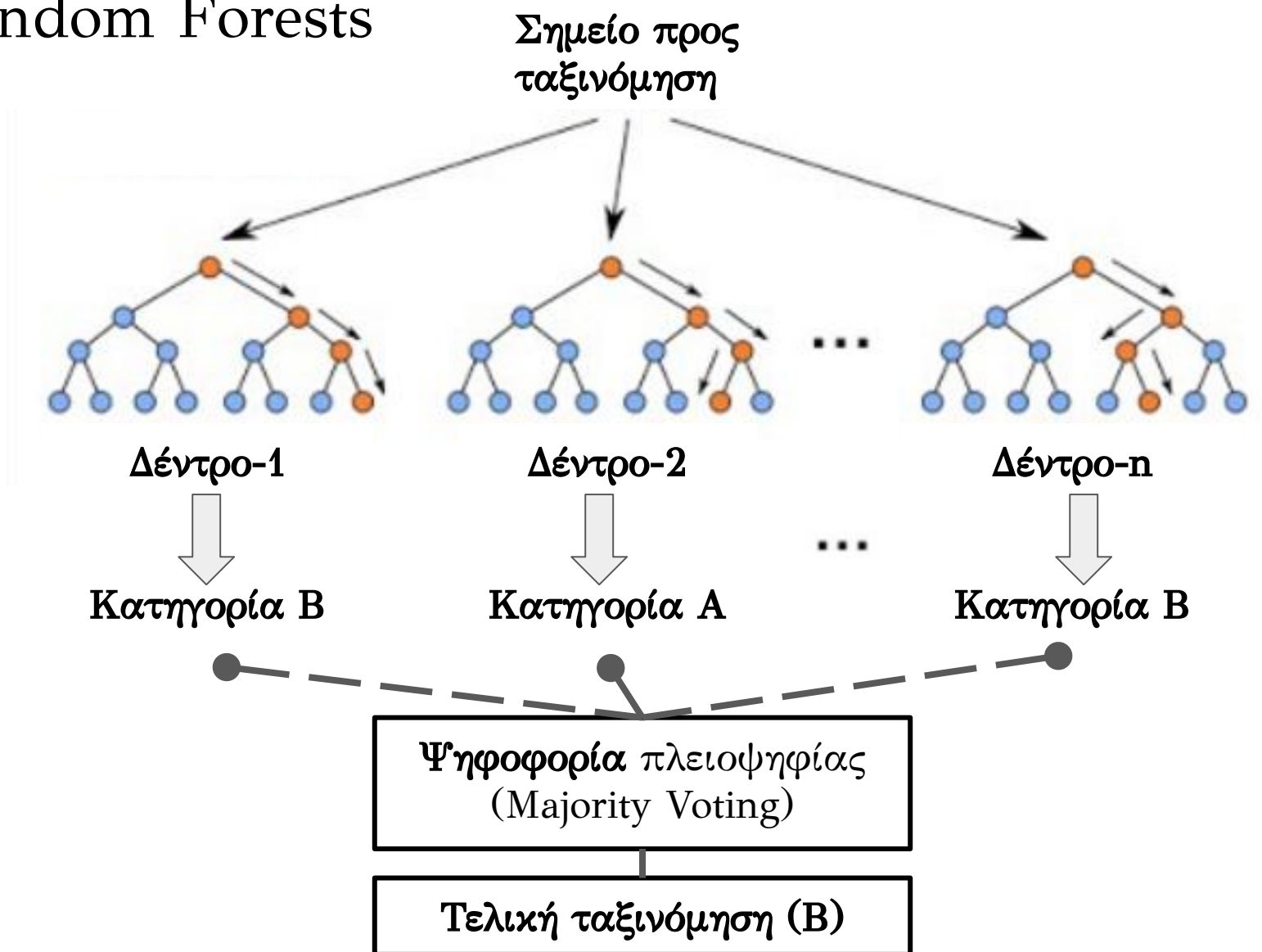
Τα “δέντρα αποφάσεων” αποτελούν μια δημοφιλή μέθοδο ταξινόμησης που βασίζεται στη δυνατότητα μετάβασης από μια σειρά παρατηρήσεων σε συμπεράσματα για την κατηγορία στην οποία ανήκει ένα σημείο.



Random Forests

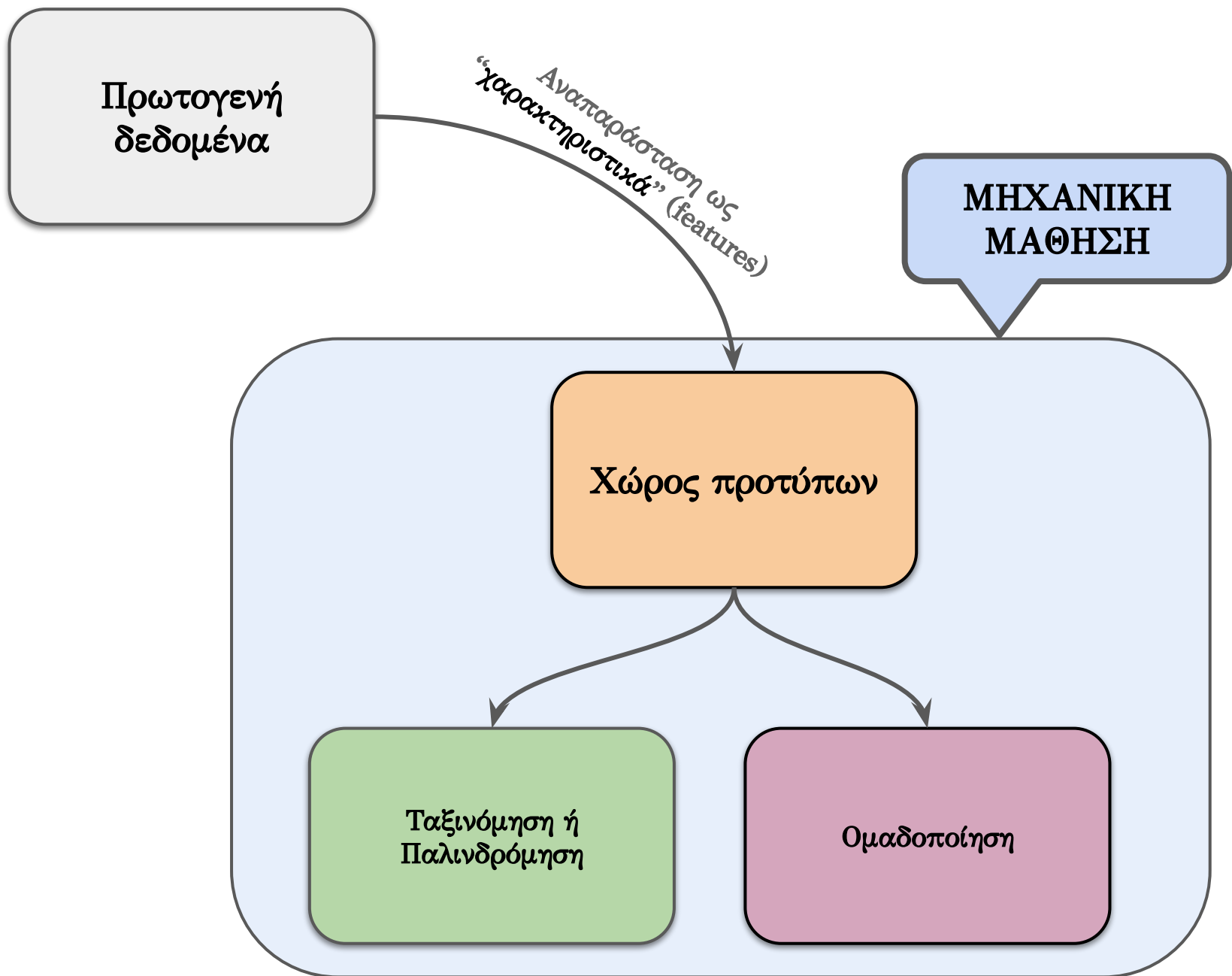
- Δημοφιλείς ταξινομητές που ανήκουν στην κατηγορία τεχνικών ensemble-learning
- Βασίζονται στην κατασκευή πολλαπλών “δέντρων αποφάσεων” κατά το χρόνο εκπαίδευσης βάσει των οποίων προκύπτει η τελική ταξινόμηση του κάθε σημείου ως η πλέον συχνή κατηγορία
- Συγκριτικά με τα απλά “δέντρα αποφάσεων”, οι αλγόριθμοι RF δεν εμφανίζουν το ίδιο συχνά προβλήματα υπερπροσαρμογής (overfitting) στο σετ εκπαίδευσης

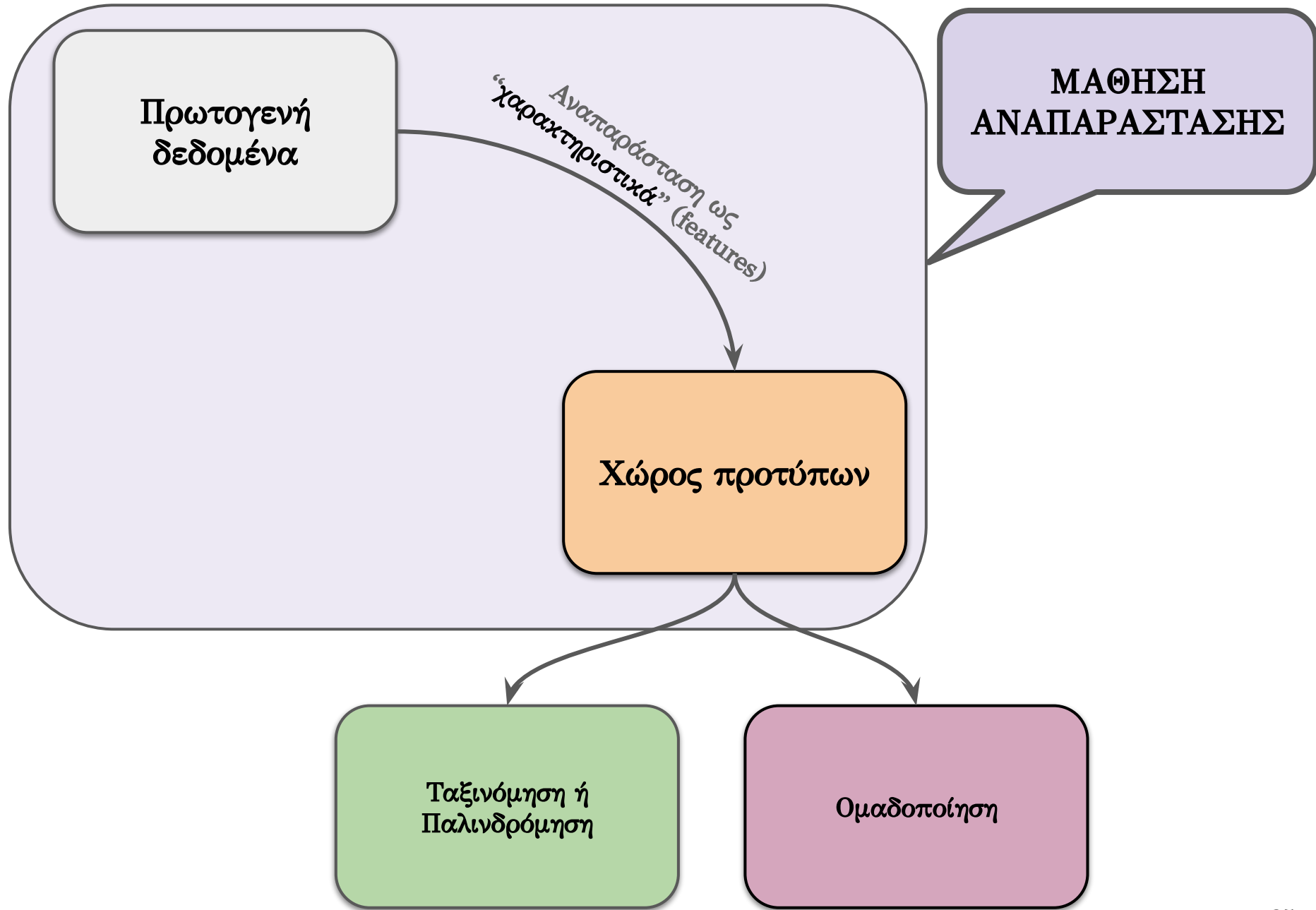
Random Forests

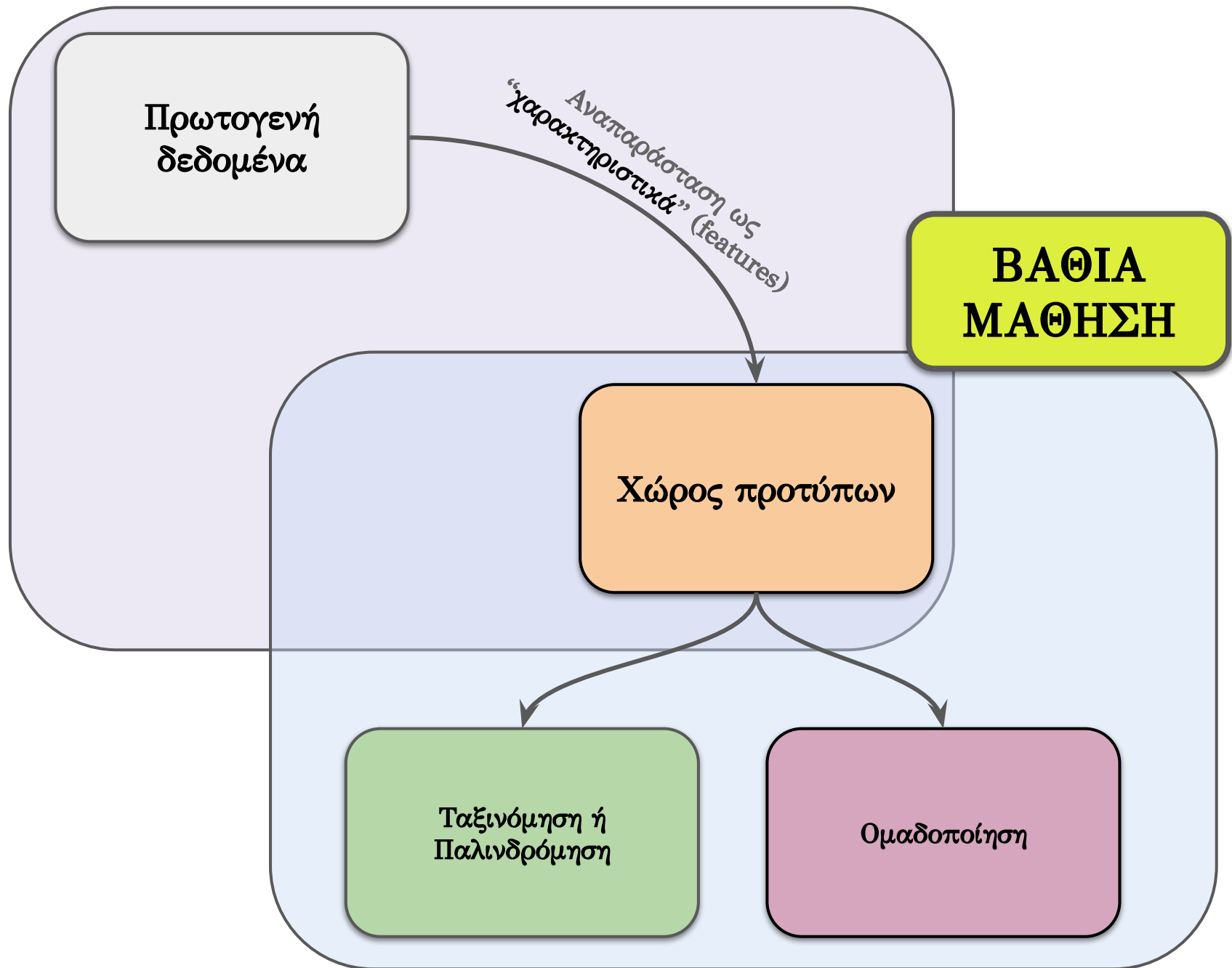


Βαθιά Μάθηση

Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα







Βαθιά Μάθηση (Deep Learning)

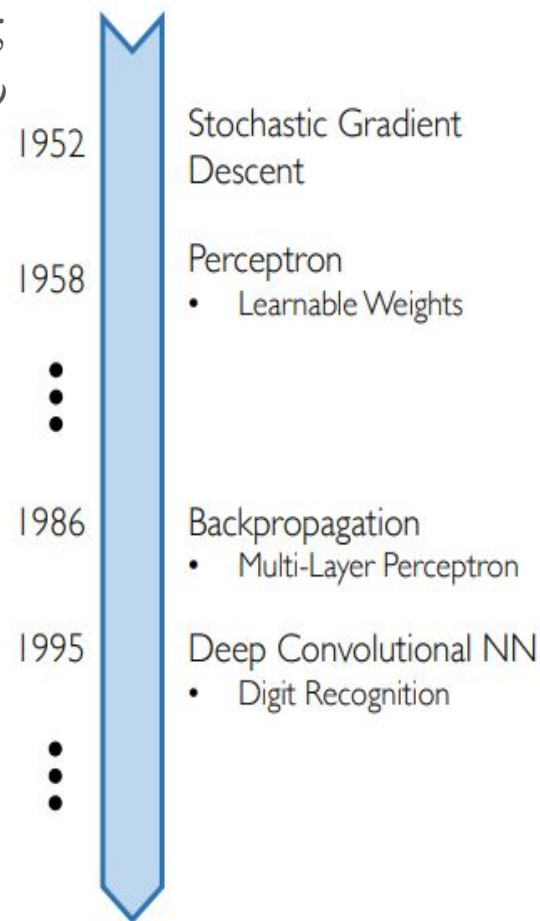
Μέλος της ευρύτερης οικογένειας αλγορίθμων Μηχανικής Μάθησης βασιζόμενο στην εκμάθηση αναπαραστάσεων υψηλού επιπέδου για:

- Ταξινόμηση δεδομένων
- Παλινδρόμηση
- Ομαδοποίηση δεδομένων

Η Βαθιά Μάθηση μπορεί να εφαρμοστεί με τρόπο:

- Επιβλεπόμενο
- Μη-επιβλεπόμενο
 - Ημι-επιβλεπόμενο (;)

Ως θεωρία υφίσταται σε μεγάλο βαθμό από τα τέλη του 20ου αιώνα



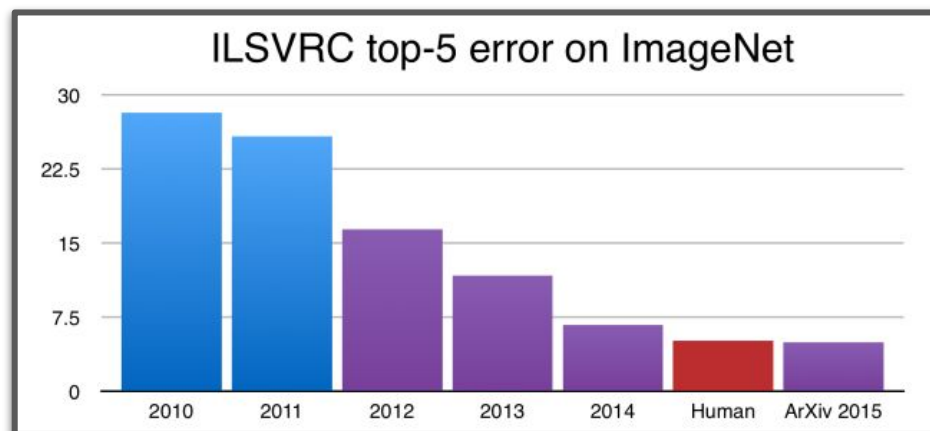
Βαθιά Μάθηση (Deep Learning)

Πλεονεκτήματα

- + Καλύτερες επιδόσεις
- + Κλιμακωσιμότητα
- + Μη-γραμμικότητα
- + Έλλειψη ανάγκης για “χειροποίητες” (και σύνθετες) αναπαραστάσεις
- + Προσαρμοστικότητα και δυνατότητα μεταφοράς “γνώσης” μεταξύ εφαρμογών

Μειονεκτήματα

- Απαίτηση για υπέρογκα δεδομένα εκπαίδευσης
- Υπολογιστικό κόστος
- Χρονοβόρα εκπαίδευση
- Έλλειπής θεωρητική τεκμηρίωση
- Δύσκολη ερμηνεία



Βαθιά Μάθηση - Η “έκρηξη”

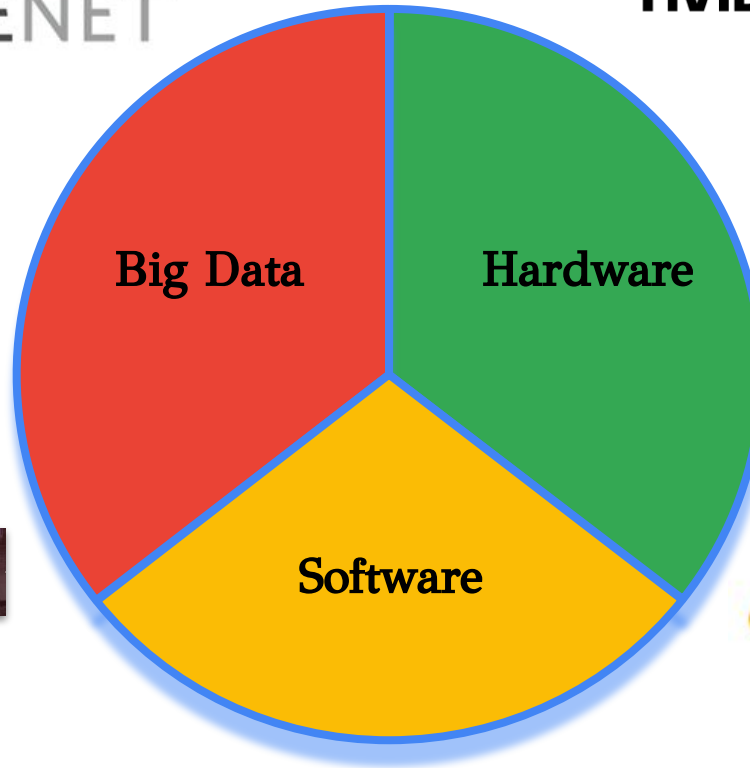
IMAGENET



DEEP
LEARNING
INSTITUTE



kaggle



aws

Open Images Dataset V4

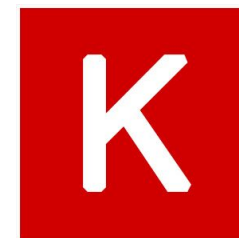
Google Cloud

Microsoft Azure

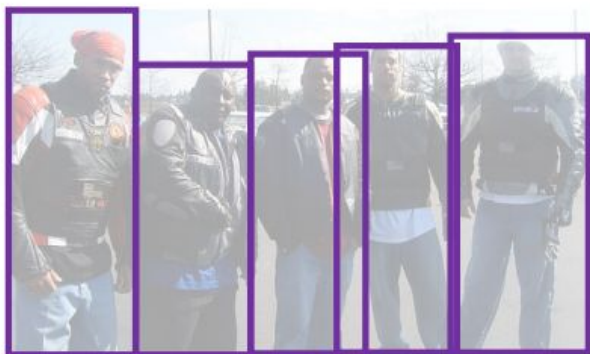
PYTORCH



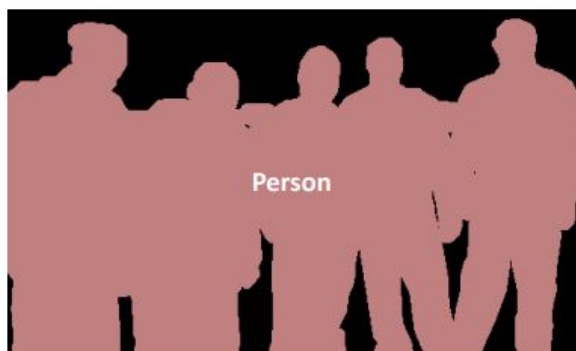
TensorFlow



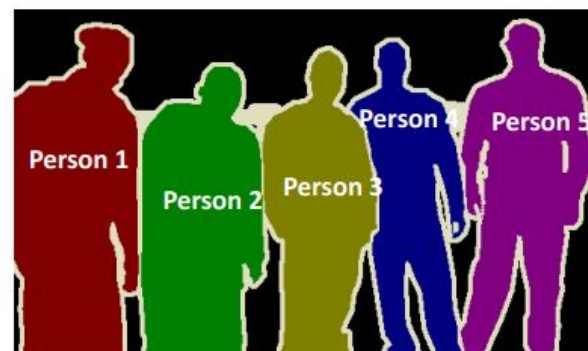
Βαθιά Μάθηση - Εφαρμογές στην Όραση Υπολογιστών



Object Detection



Semantic Segmentation

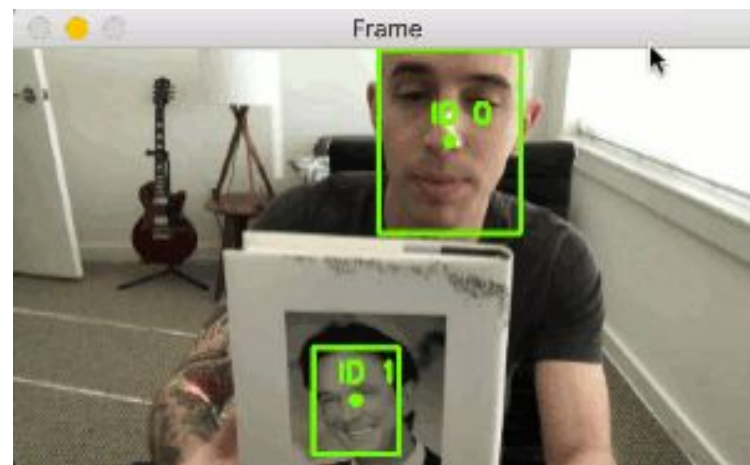


Instance Segmentation



Style transfer

Βαθιά Μάθηση - Εφαρμογές στην Όραση Υπολογιστών

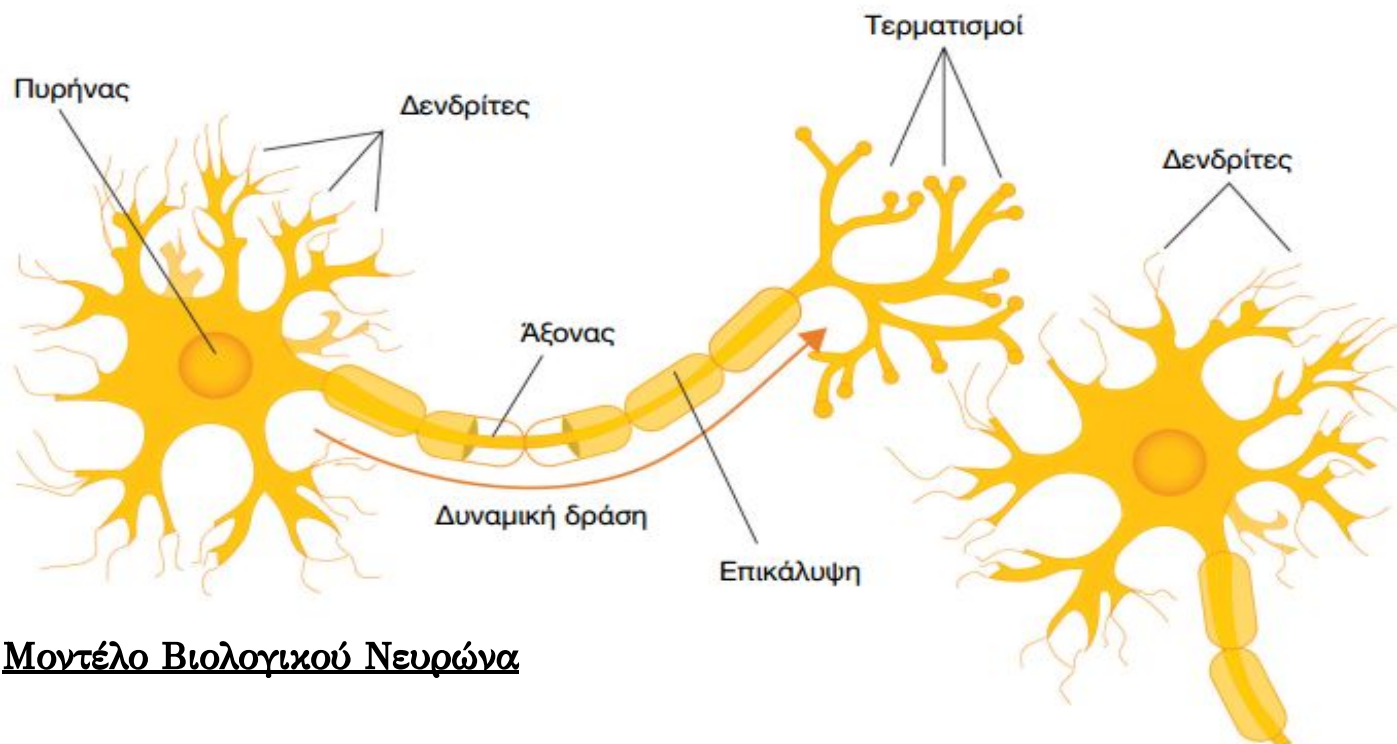


Perceptron

Οι βασικές αρχές μιας “στρώσης” (layer) ενός
νευρωνικού δικτύου

Εισαγωγή στο Perceptron

- Δυαδική ταξινόμηση - Binary Classification
 - $\{A, B\} \rightarrow \{-1, 1\}$ ή (ισοδύναμα) $\{1, 0\}$
- Το Perceptron αποτελεί το “απλούστερο” νευρωνικό δίκτυο
 - Απαιτεί τα δεδομένα να είναι γραμμικά διαχωρίσιμα
 - Ιστορικά έχει εμπνευστεί από τον βιολογικό νευρώνα



Μοντέλο Βιολογικού Νευρώνα

Εισαγωγή στο Perceptron



$\{x_n\}$: δεδομένα εισόδου \rightarrow τιμές χαρακτηριστικών

$\{w_n\}$: τιμές βαρών \rightarrow αρχικά τυχαίες τιμές!

Γραμμικός συνδυασμός : $\hat{y} = \sum_{i=1}^n w_i x_i$

Εισαγωγή στο Perceptron

b : δείκτης μεροληψίας (bias) \rightarrow σταθερή τιμή

$$\hat{y} = \sum_{i=1}^n w_i x_i + b$$

Εισαγωγή στο Perceptron

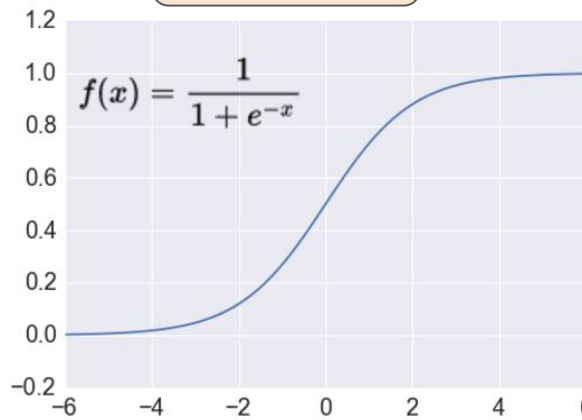


$$\hat{y} = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

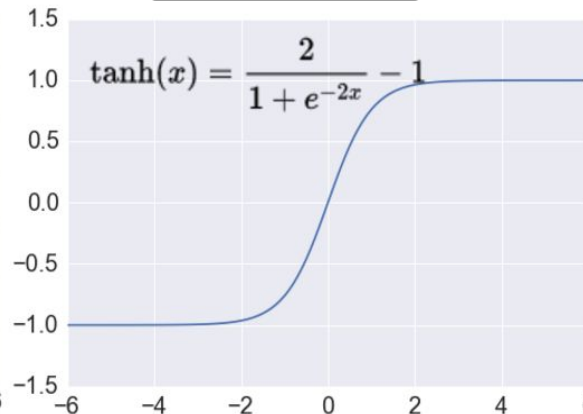
Συναρτήσεις ενεργοποίησης (Activation functions)

- Απαραίτητες διότι προσδίδουν την απαιτούμενη **μη γραμμικότητα** στο μοντέλο!
- Ιστορικά, αρχικά χρησιμοποιούνταν μόνο η βηματική συνάρτηση (step function / heaviside) που λαμβάνει την τιμή 1 όταν $y > 0$ και την τιμή 0 αλλιώς.

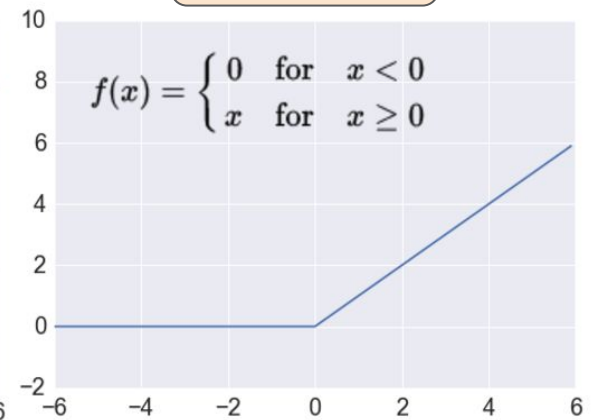
Σιγμοειδής



Υπερβολική
εφαπτομένη

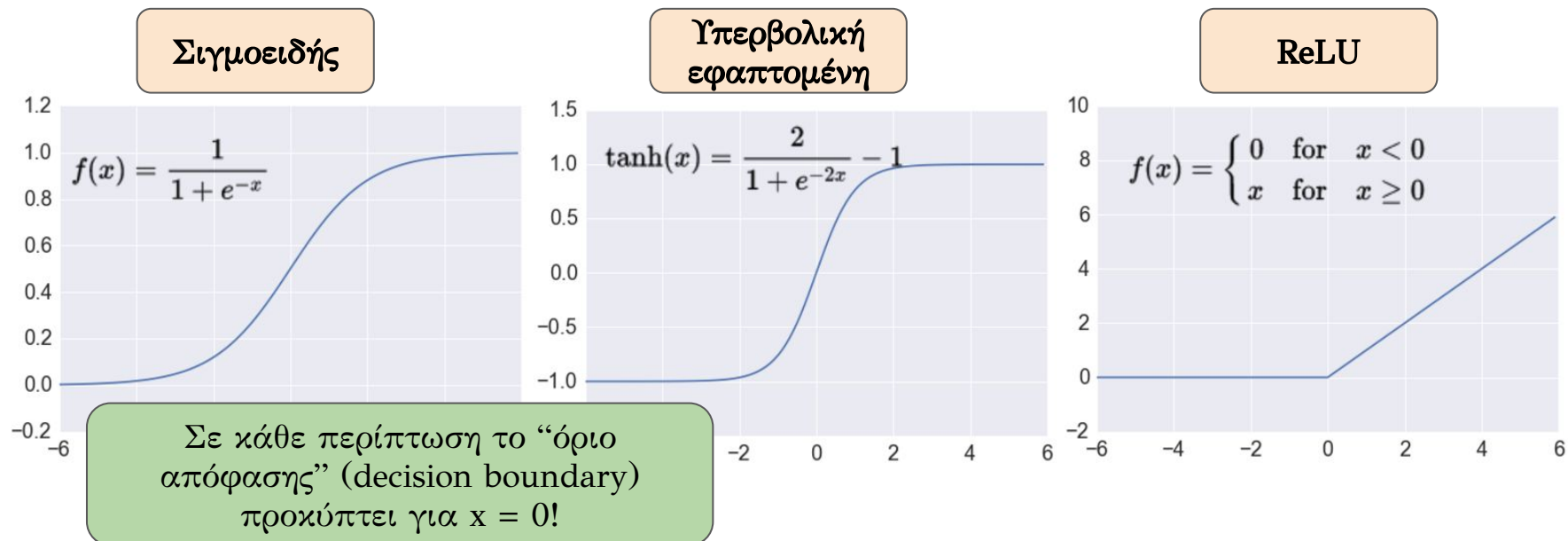


ReLU

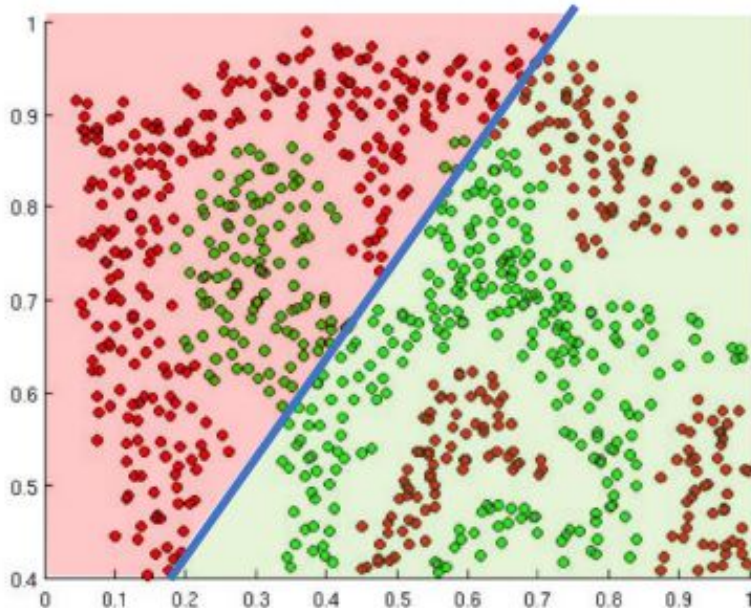


Συναρτήσεις ενεργοποίησης (Activation functions)

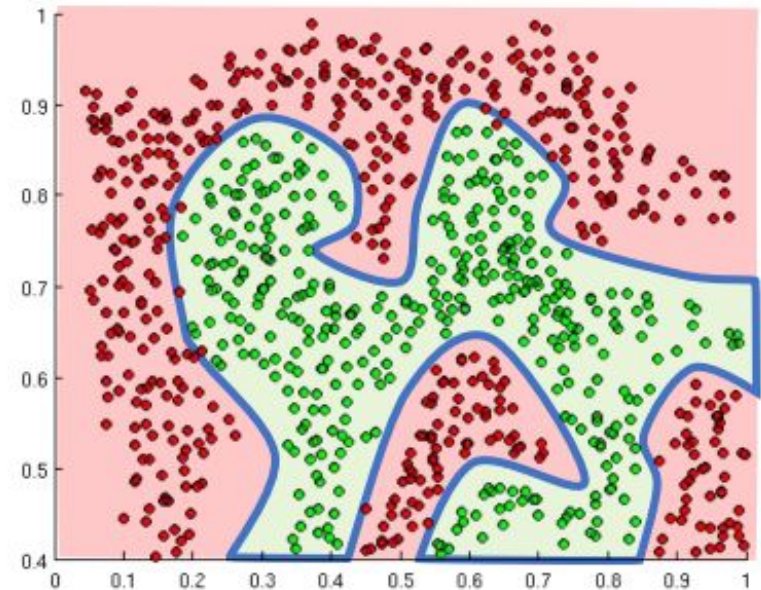
- Απαραίτητες διότι προσδίδουν την απαιτούμενη **μη γραμμικότητα** στο μοντέλο!
- Ιστορικά, αρχικά χρησιμοποιούνταν μόνο η βηματική συνάρτηση (step function / heaviside) που λαμβάνει την τιμή 1 όταν $y > 0$ και την τιμή 0 αλλιώς.



Συναρτήσεις ενεργοποίησης (Activation functions)



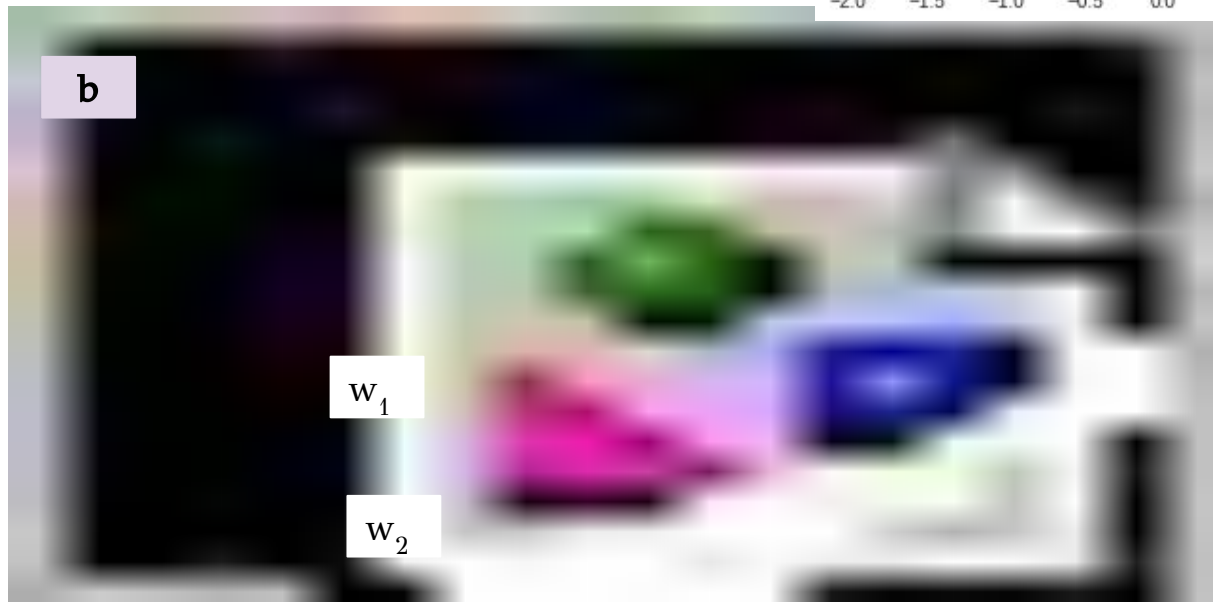
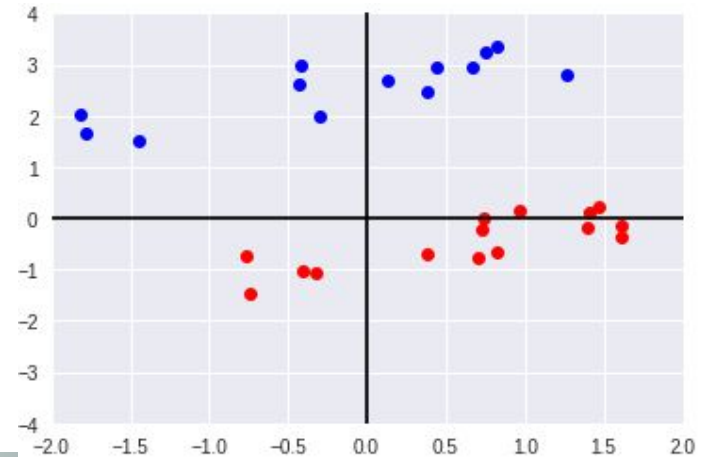
Οι γραμμικές συναρτήσεις ενεργοποίησης εξάγουν γραμμικές “αποφάσεις” χωρίς να λαμβάνουν υπόψη το μέγεθος του δικτύου



Οι μη γραμμικότητες μας επιτρέπουν να εξετάσουμε πιο περίπλοκα πρότυπα στα δεδομένα μας

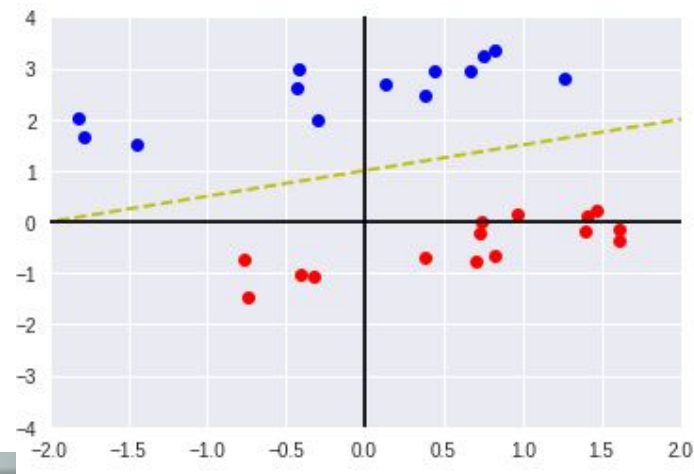
Απλή εφαρμογή : Επιβλεπόμενη Ταξινόμηση

Θέλουμε να βρούμε το μοντέλο εκείνο που διαχωρίζει τα δεδομένα εισόδου σε 2 γνωστές κατηγορίες. Για το λόγο αυτό, υλοποιούμε το παρακάτω perceptron :



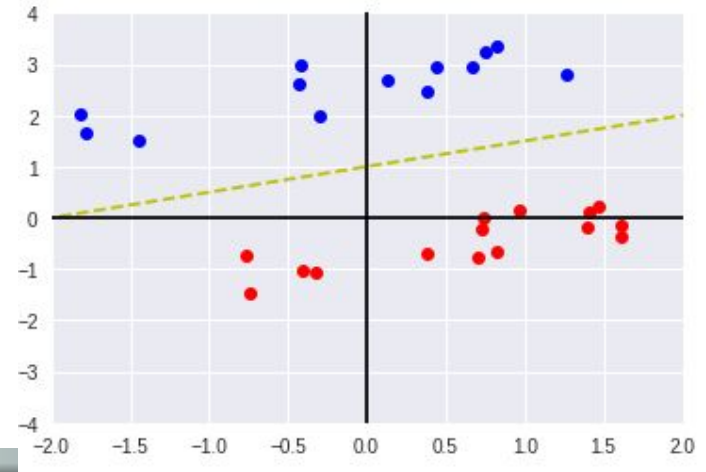
Απλή εφαρμογή : Επιβλεπόμενη Ταξινόμηση

Παρατηρούμε ότι για $b=1$, $w_1=0.5$ και $w_2=-1$ το μοντέλο διαχωρίζει ορθά τις 2 κατηγορίες για τα δεδομένα εκπαίδευσης



Απλή εφαρμογή : Επιβλεπόμενη Ταξινόμηση

Παρατηρούμε ότι για $b=1$, $w_1=0.5$ και $w_2=-1$ το μοντέλο διαχωρίζει ορθά τις 2 κατηγορίες για τα δεδομένα εκπαίδευσης

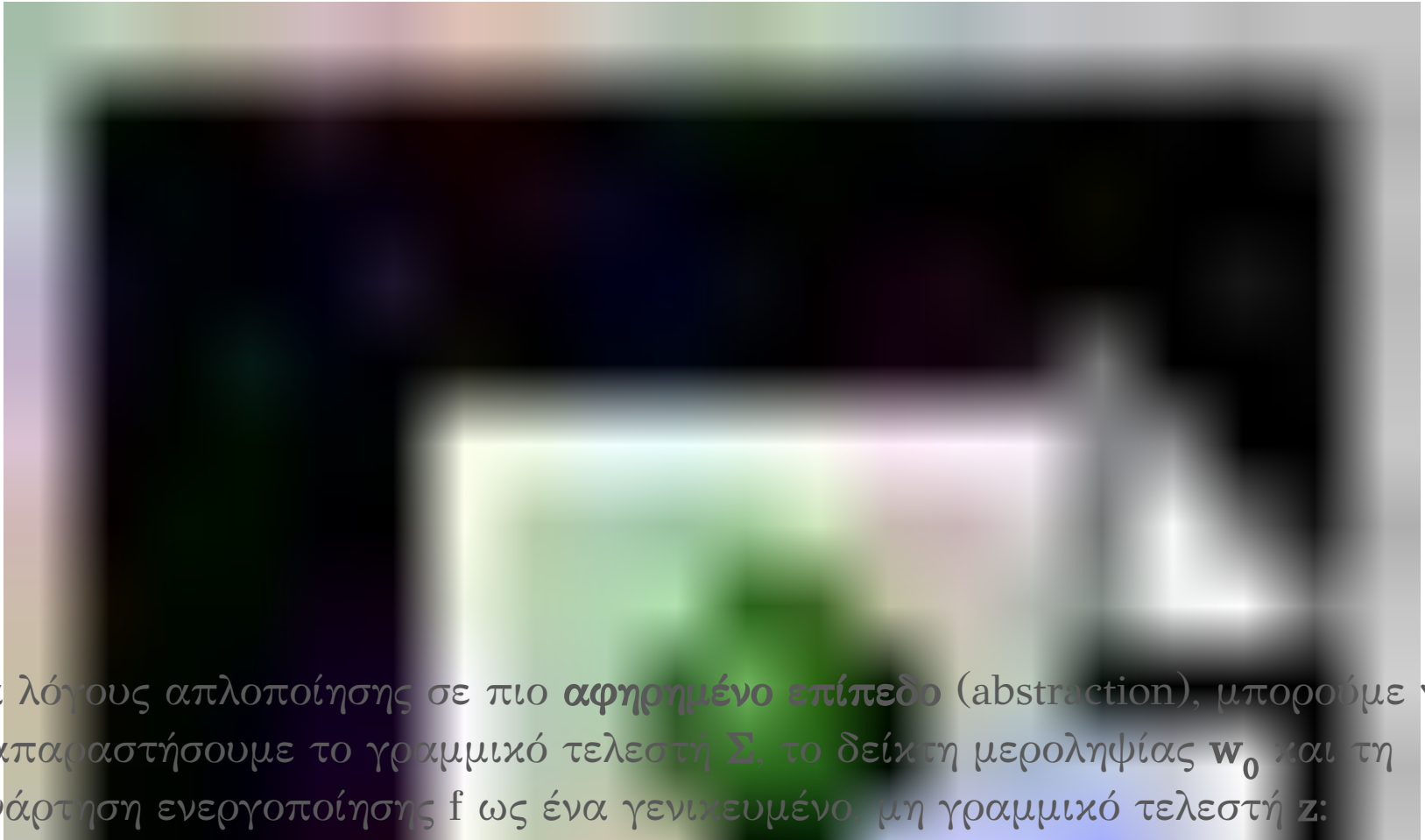


Θέτοντας $y = 0 \Rightarrow$
 $x_2 = 0.5 * x_1 + 1$
Εξίσωση ευθείας στο $x_1 x_2$ επίπεδο

Multi-Layer Perceptron

Ένα απλοϊκό νευρωνικό δίκτυο

Απλοποιημένο Perceptron



Για λόγους απλοποίησης σε πιο αφηρημένο επίπεδο (abstraction), μπορούμε να αναπαραστήσουμε το γραμμικό τελεστή Σ , το δείκτη μεροληψίας w_0 και τη συνάρτηση ενεργοποίησης f ως ένα γενικευμένο, μη γραμμικό τελεστή z :

$$z: \mathbb{R}^n \rightarrow \mathbb{R} \mid z(x_1, \dots, x_n) = f(\sum_{i=1}^n w_i x_i + w_0)$$

Multi-Layer Perceptron (MLP)



Το σύνολο νευρώνων $\{z_1, \dots, z_m\}$ καλείται “**κρυφή**” στρώση (hidden layer)

- Η στρώση των νευρώνων (στη συνήθη περίπτωση) είναι **πλήρως συνδεδεμένη**, δηλαδή σε κάθε νευρώνα της στρώσης εισάγονται ως “είσοδος” όλες οι αποκρίσεις της προηγούμενης
- Στη βιβλιογραφία/APIs μία τέτοια στρώση συναντάται ως **fully-connected** ή **dense**

Multi-Layer Perceptron (MLP)



Ένα ενδεικτικό MLP δίκτυο με δύο “κρυφές” στρώσεις.

- Το πλήθος των νευρώνων σε κάθε στρώση μπορεί να είναι διαφορετικό.
- Οι “κρυφές” στρώσεις αυξάνουν (κατά την πολυωνυμική έννοια) την τάξη μη-γραμμικότητας του δικτύου

Multi-Layer Perceptron (MLP)



Ισοδύναμος συμβολισμός πλήρους συνδεδεμένης στρώσης

Multi-Layer Perceptron (MLP)



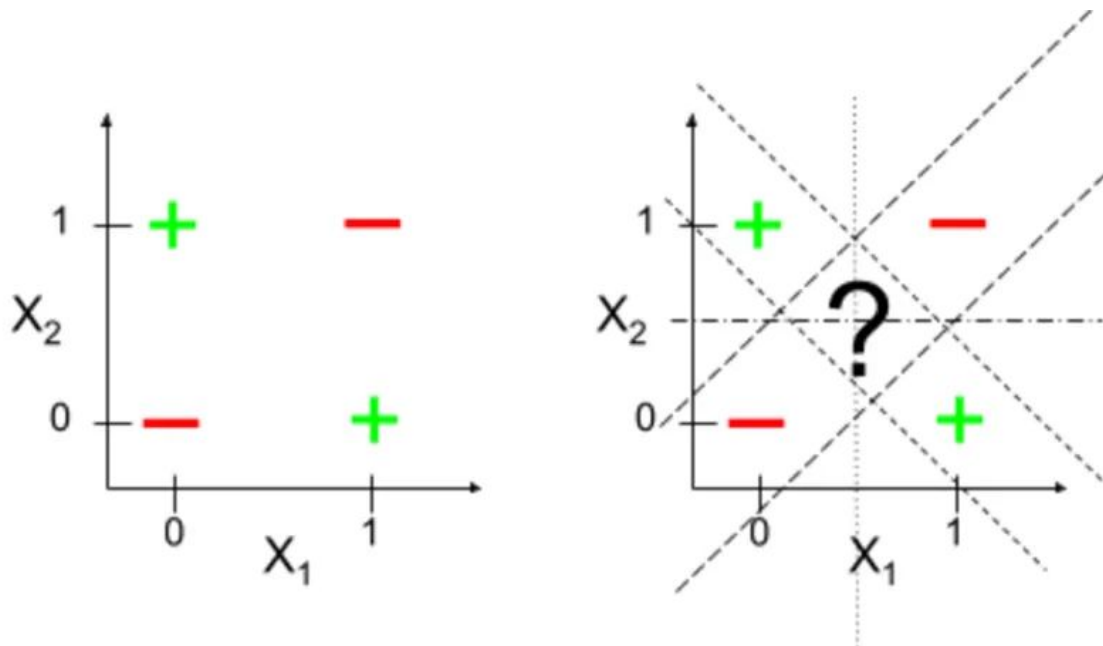
Στη γενική περίπτωση ένα MLP μπορεί να έχει l “πυκνές” κρυφές στρώσεις

Εφαρμογή : Το πρόβλημα XOR

- Απλό μη γραμμικό πρόβλημα ταξινόμησης που ορίζεται από τον παρακάτω πίνακα αληθείας :

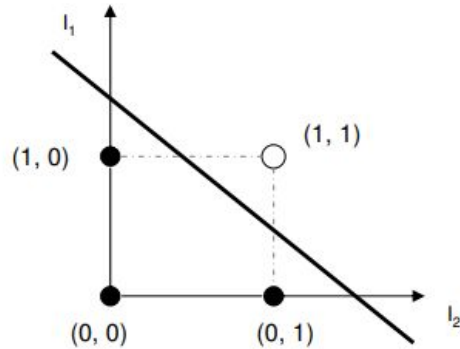
Είσοδος 1	Είσοδος 2	Έξοδος
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE

- Τα δεδομένα εισόδου είναι μη γραμμικά διαχωρίσιμα!!!

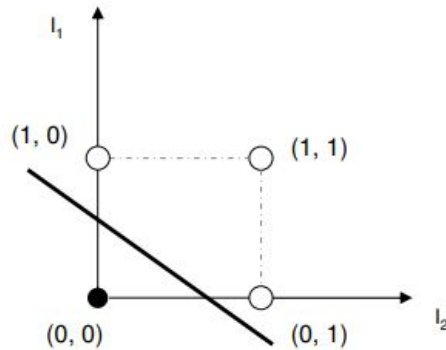


Εφαρμογή : Το πρόβλημα XOR

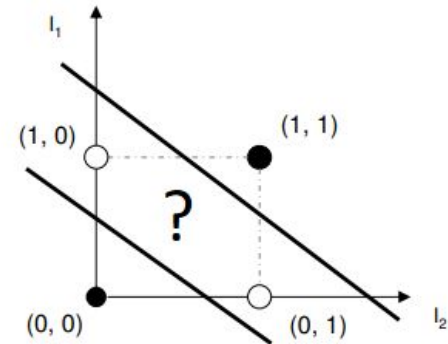
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1



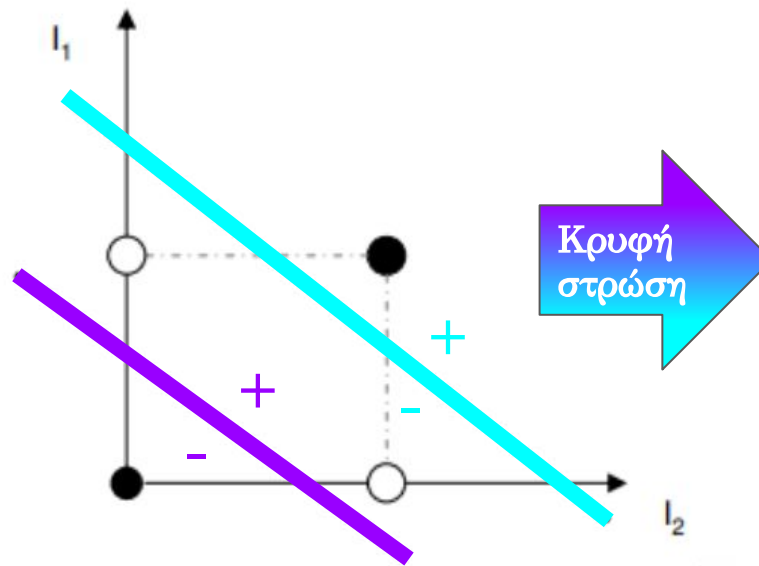
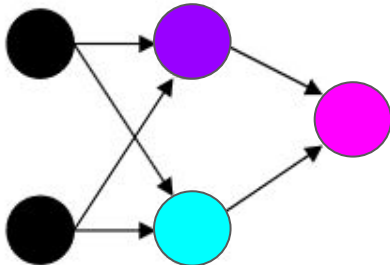
XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



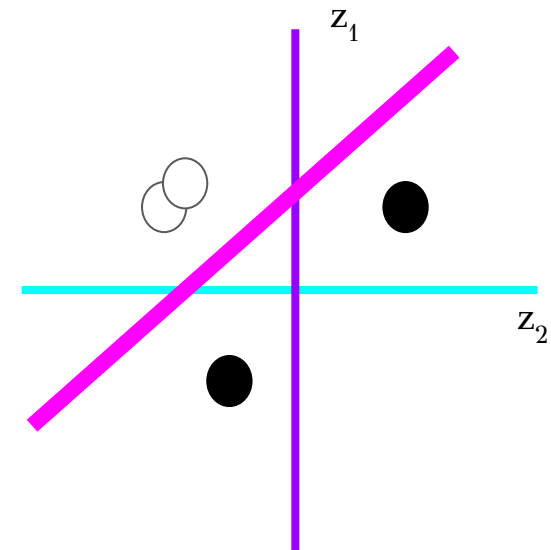
Χρήση ενός ή
δύο perceptron?

Εφαρμογή : Το πρόβλημα XOR

XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0



Χρήση MLP

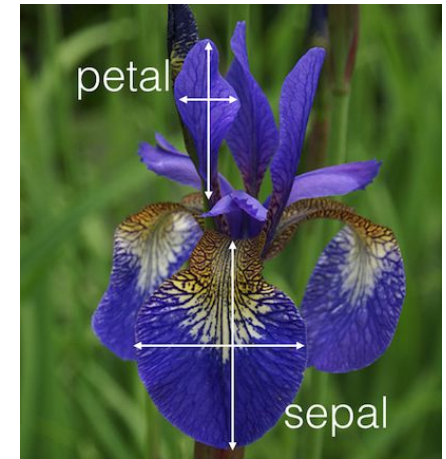


Συναρτήσεις κόστους

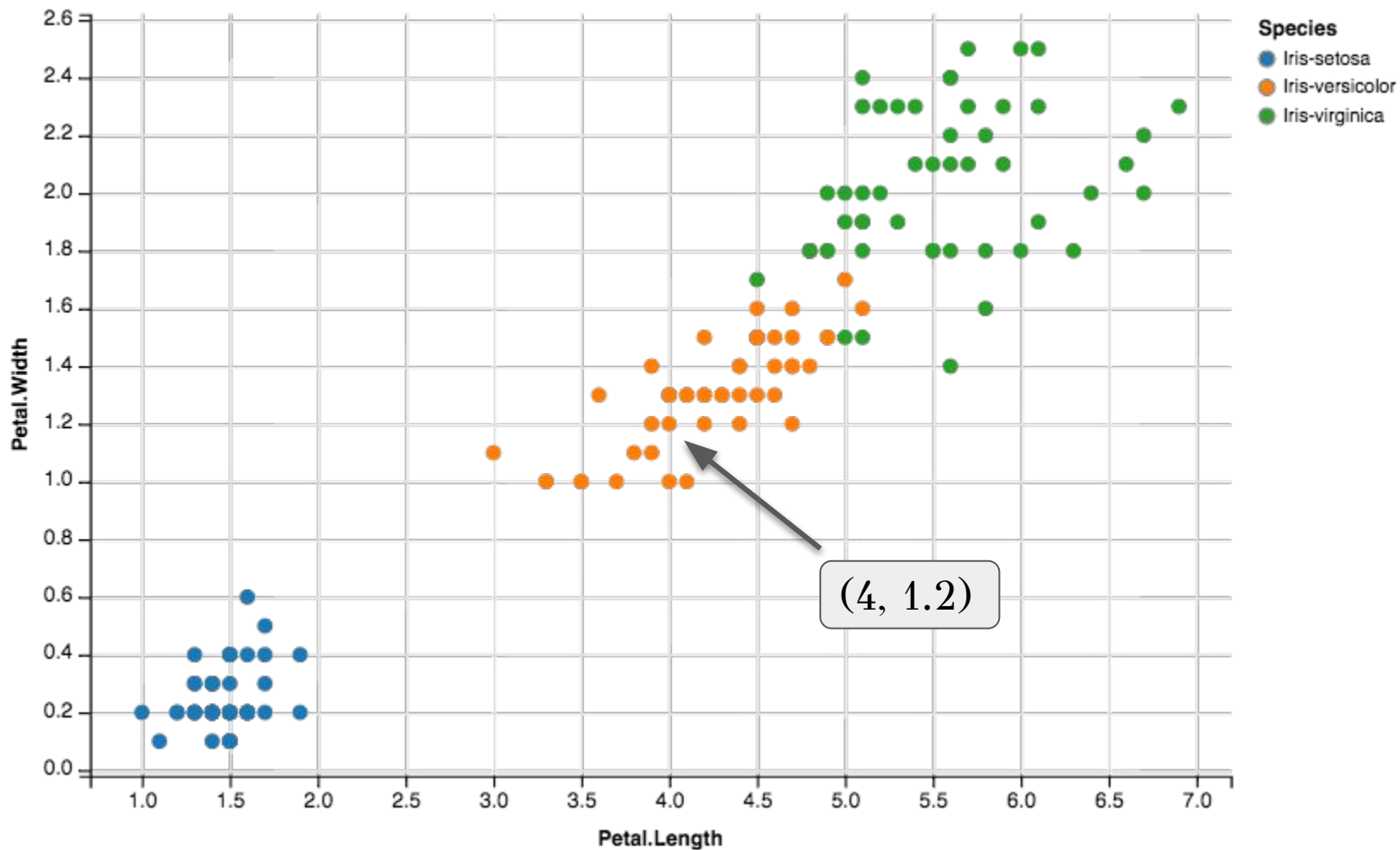
Ένα νευρωνικό δίκτυο στην πράξη

Παράδειγμα : Iris dataset

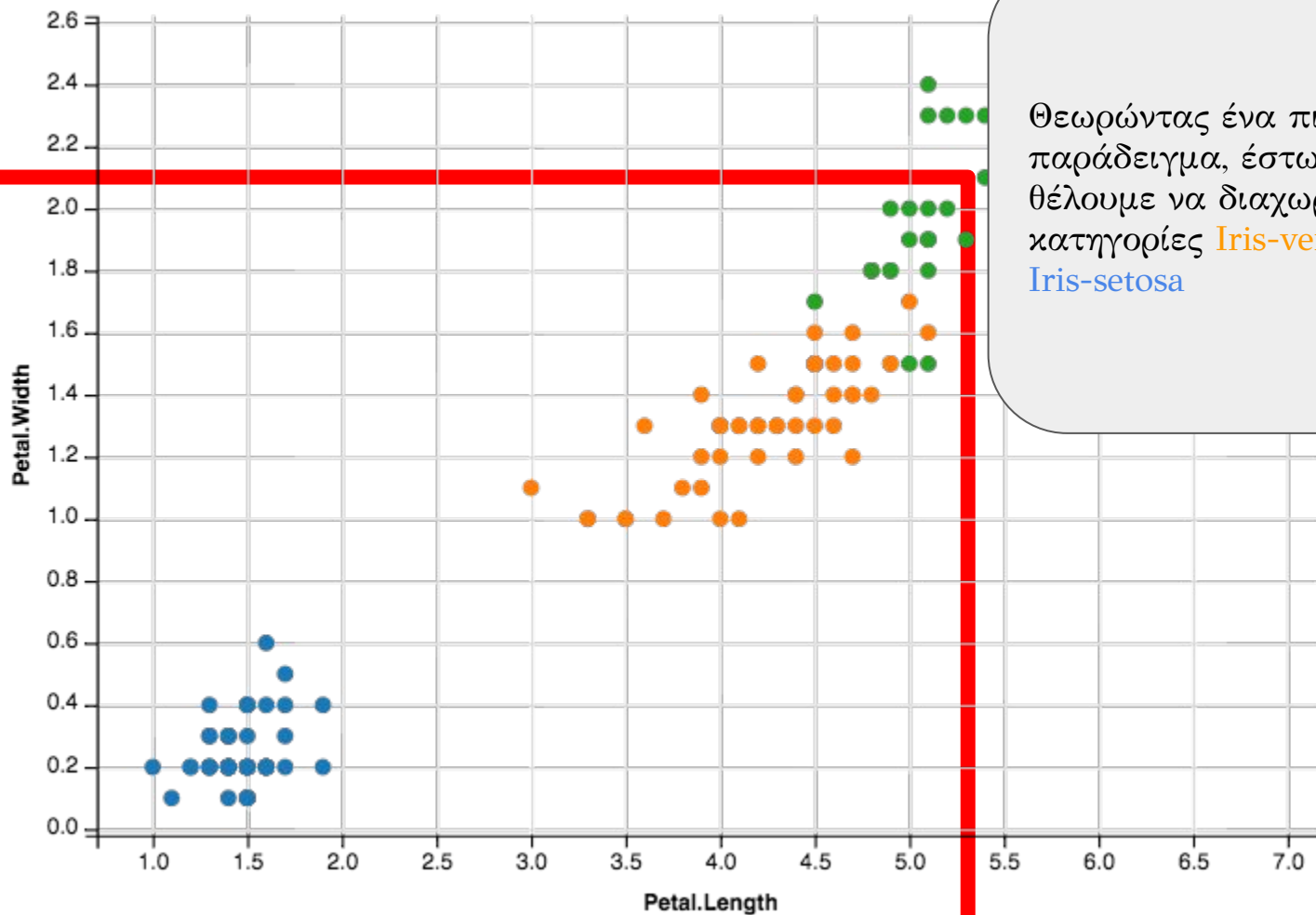
- Δείγματα για καθένα από τα εξής είδη λουλουδιών :
 - Iris setosa
 - Iris virginica
 - Iris versicolor
- Τα χαρακτηριστικά (features) που εμπεριέχει είναι τα εξής :
 - Petal_length
 - Petal_width
 - Sepal_length
 - Sepal_width
- Θα κατασκευάσουμε ένα νευρωνικό δίκτυο που θα **ταξινομεί** τα τρία διαφορετικά είδη λουλουδιών λαμβάνοντας υπόψη μόνο δύο χαρακτηριστικά (για λόγους απλότητας).



Παράδειγμα : Iris dataset

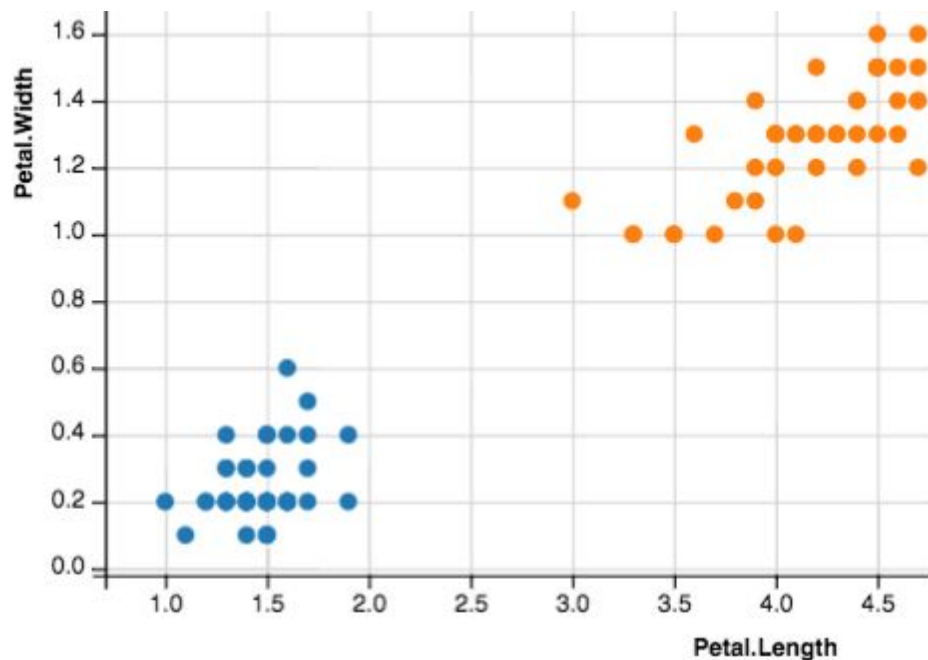


Παράδειγμα : Iris dataset

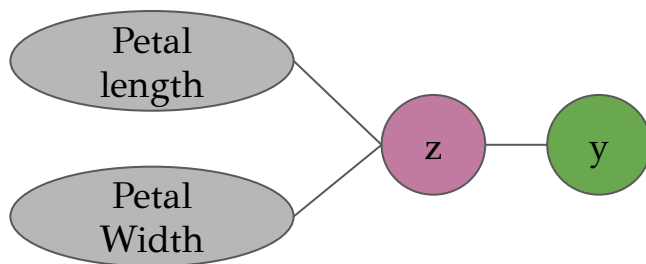


Θεωρώντας ένα πιο απλό παράδειγμα, έστω ότι θέλουμε να διαχωρίσουμε τις κατηγορίες *Iris-versicolor* και *Iris-setosa*

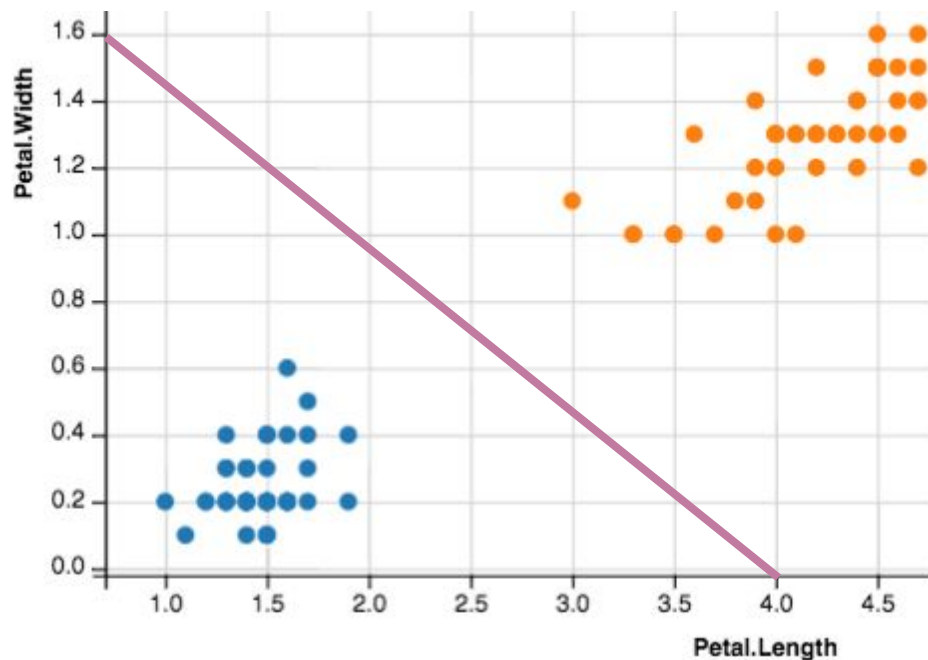
Παράδειγμα : Iris dataset



Οι κατηγορίες είναι γραμμικά διαχωρίσιμες, άρα μπορούμε να χρησιμοποιήσουμε ένα απλό **perceptron** για την επίλυση του προβλήματος!

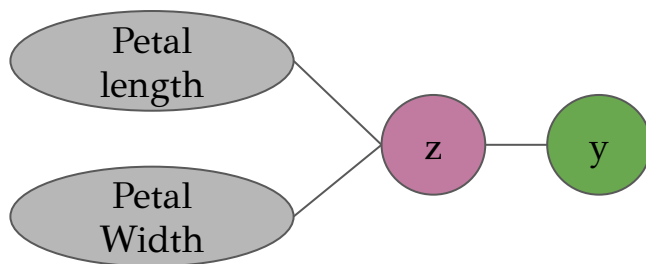


Παράδειγμα : Iris dataset

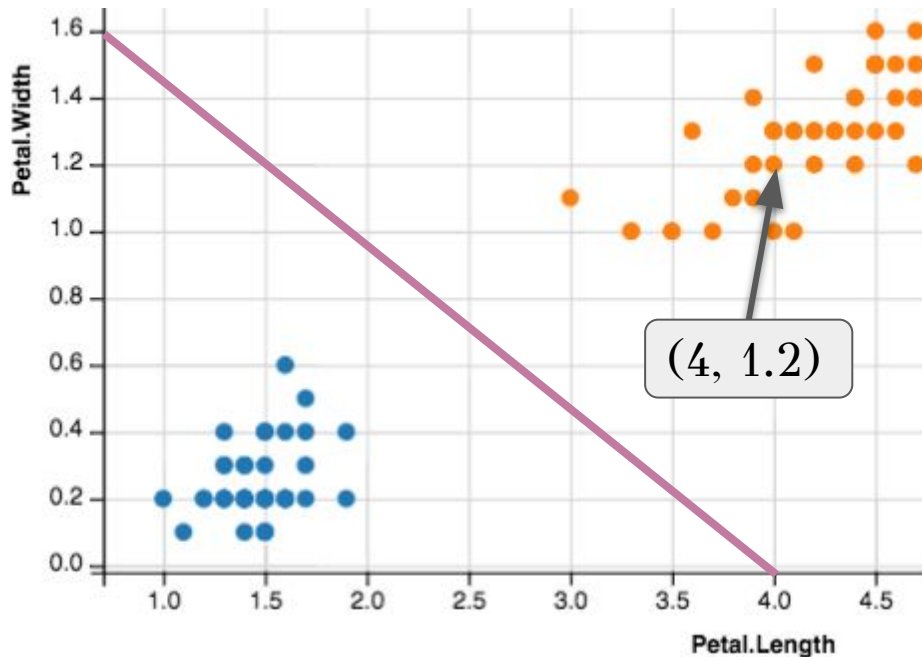


Οι κατηγορίες είναι γραμμικά διαχωρίσιμες, άρα μπορούμε να χρησιμοποιήσουμε ένα απλό **perceptron** για την επίλυση του προβλήματος!

w_0	=	-1.6
w_1	=	+0.4
w_2	=	+1
f	=	sigmoid

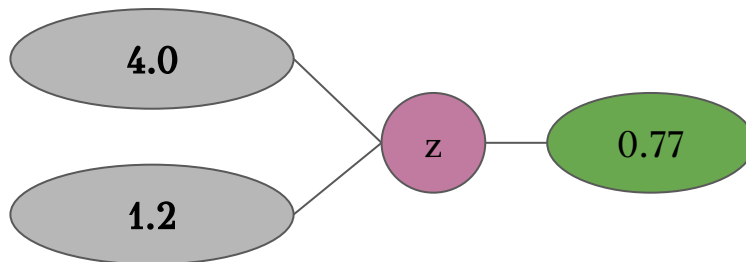


Παράδειγμα : Iris dataset



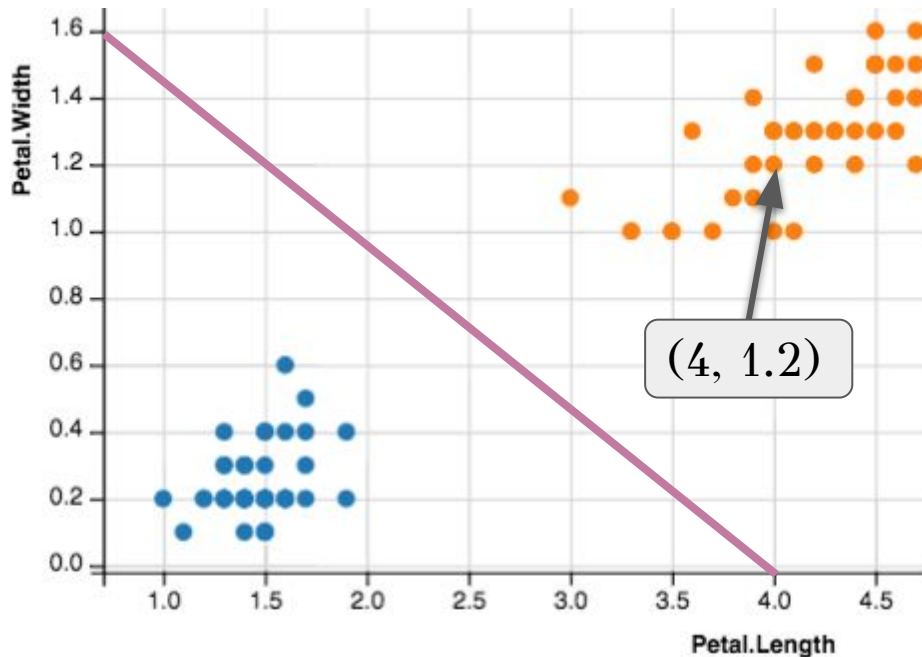
Οι κατηγορίες είναι γραμμικά διαχωρίσιμες, άρα μπορούμε να χρησιμοποιήσουμε ένα απλό **perceptron** για την επίλυση του προβλήματος!

w_0	=	-1.6
w_1	=	+0.4
w_2	=	+1
f	=	sigmoid



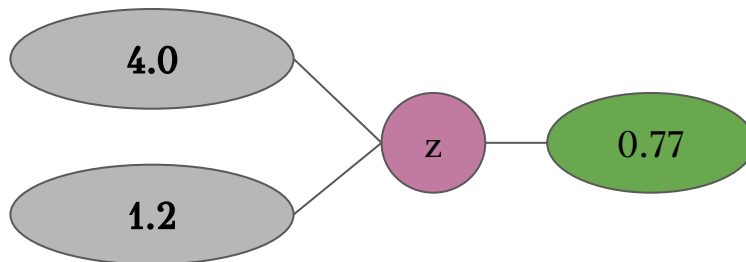
$$y_{\text{pred}} = \sigma[0.4*(4.0) + 1*(1.2) - 1.6] = 0.77$$

Παράδειγμα : Iris dataset



Για να **αξιολογήσουμε** το αποτέλεσμα θα πρέπει η τιμή y_{pred} που υπολογίσαμε να μπορεί να συγκριθεί με την πραγματική κατηγορία του σημείου y .

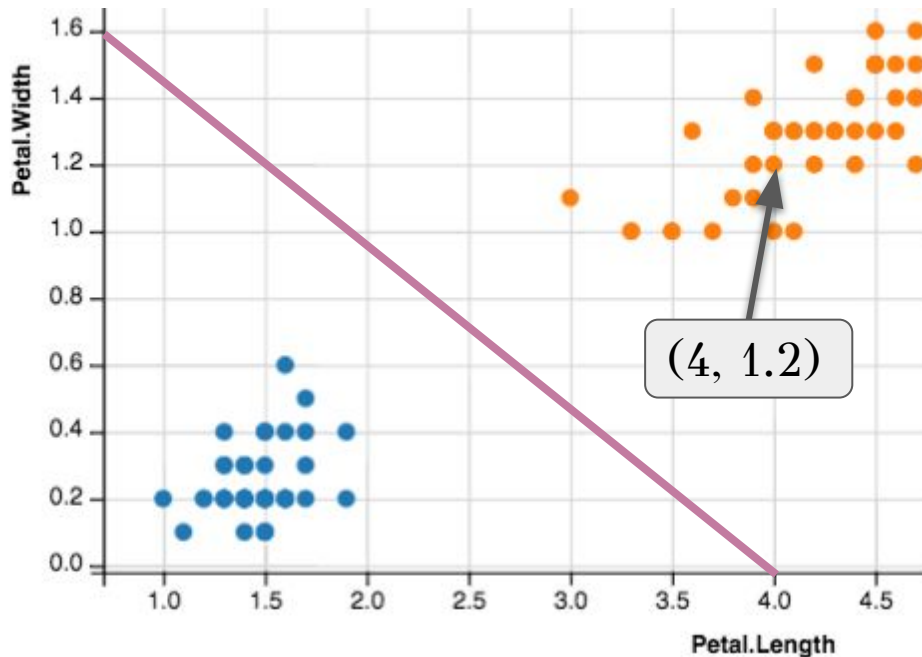
$$\begin{aligned} w_0 &= -1.6 \\ w_1 &= +0.4 \\ w_2 &= +1 \\ f &= \text{sigmoid} \end{aligned}$$



$$y_{\text{pred}} = 0.77$$

$$y = 0 \text{ ή } y = 1$$

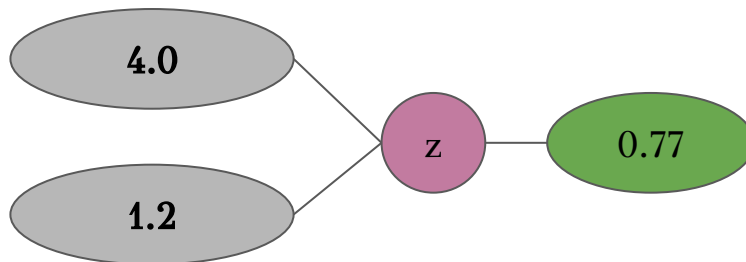
Παράδειγμα : Iris dataset



Για να συγκρίνουμε τις δύο τιμές y_{pred} και y ορίζουμε τη συνάρτηση κόστους \mathcal{L} (loss-function)

$$\mathcal{L}(y_{\text{pred}}, y)$$

- *Mean Squared error*
- *(Binary) Cross-entropy*



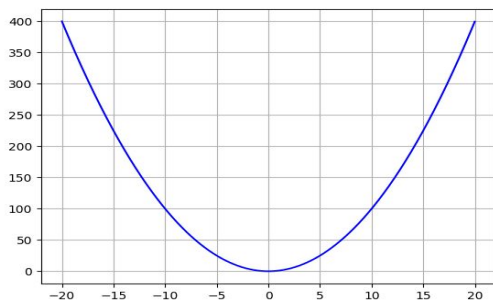
Για να ορίσουμε μία συνάρτηση κόστους για όλα τα δεδομένα ελέγχου, χρησιμοποιούμε την **Empirical-Loss \mathcal{J}**

$$J(W) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x^{(i)}; W), y^{(i)})$$

Συναρτήσεις Κόστους (Loss-functions)

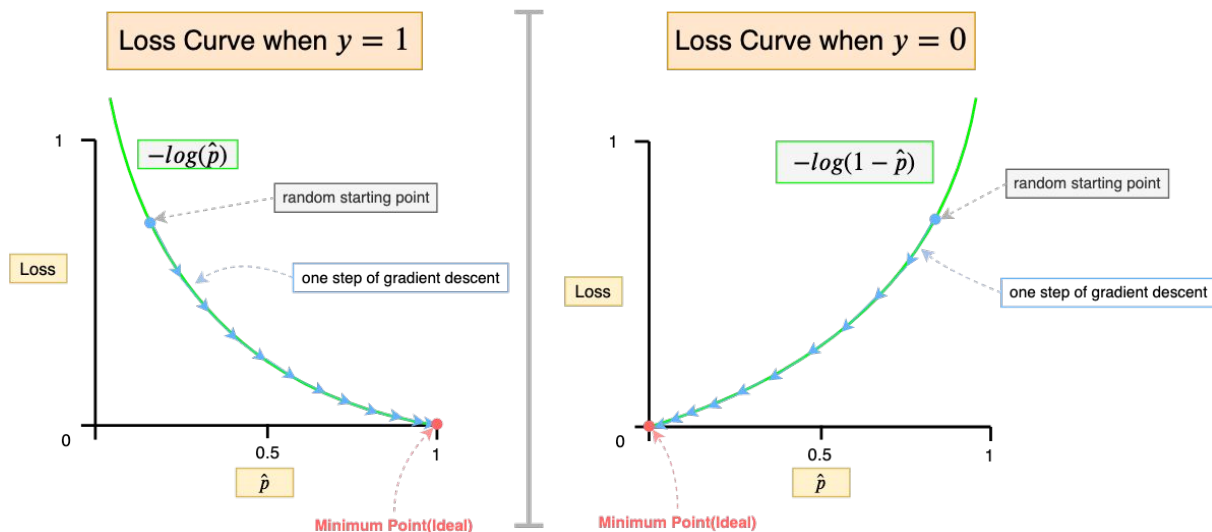
Μέσο τετραγωνικό σφάλμα (MSE):

$$MSE = \frac{1}{N} \sum_{i=0}^N (\hat{y}_i - y_i)^2$$



(Binary) Cross Entropy (BCE):

$$BCE = -\frac{1}{N} \sum_{i=0}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$



Συναρτήσεις Κόστους (Loss-functions)

Μέσο τετραγωνικό σφάλμα (MSE):

$$MSE = \frac{1}{N} \sum_{i=0}^N (\hat{y}_i - y_i)^2$$

Παλινδρόμηση σε
συνεχείς μεταβλητές

(Binary) Cross Entropy (BCE):

$$BCE = -\frac{1}{N} \sum_{i=0}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

Δυαδική ταξινόμηση

Συναρτήσεις Κόστους (Loss-functions)

(Binary) Cross Entropy (BCE):

Δυαδική ταξινόμηση

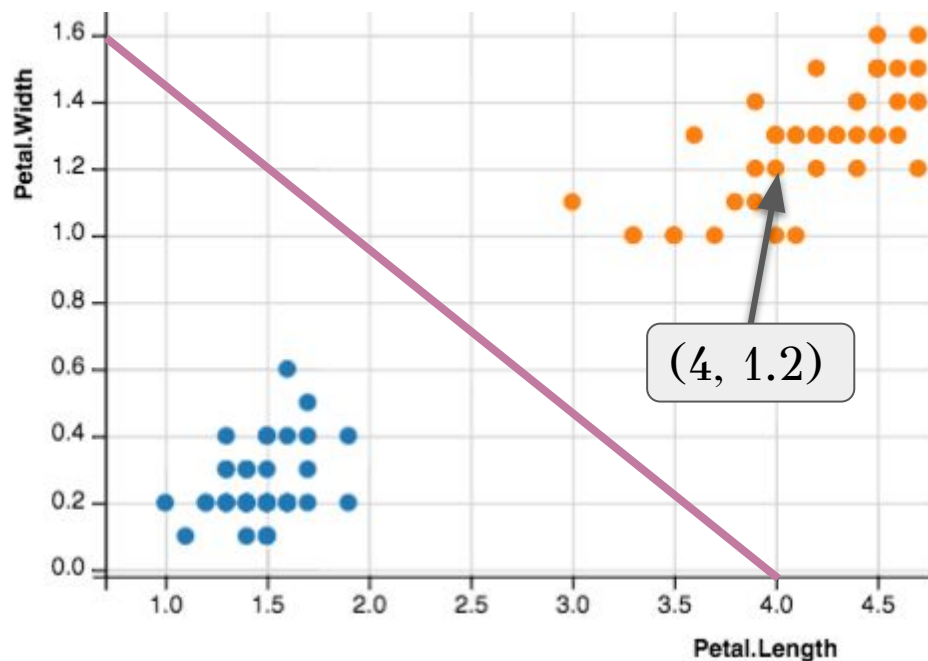
$$BCE = -\frac{1}{N} \sum_{i=0}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

Η BCE συνδυάζεται σχεδόν **αποκλειστικά με sigmoid** συνάρτηση ενεργοποίησης γιατί:

- Η sigmoid έχει **πεδίο τιμών στο (0, 1)**
- Το exp της σιγμοειδούς με τους νεπέριους λογάριθμους της BCE αλληλοαναιρούνται μερικώς **“απλοποιώντας” την έξοδο** σχετικά με τη τιμή της συνάρτησης κόστους

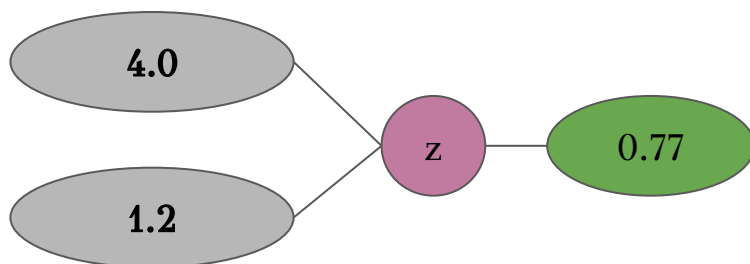
$$\log\left(\frac{1}{1 + \exp(-x)}\right) \xrightarrow{x \text{ is real}} -\log(e^{-x} + 1)$$

Παράδειγμα : Iris dataset



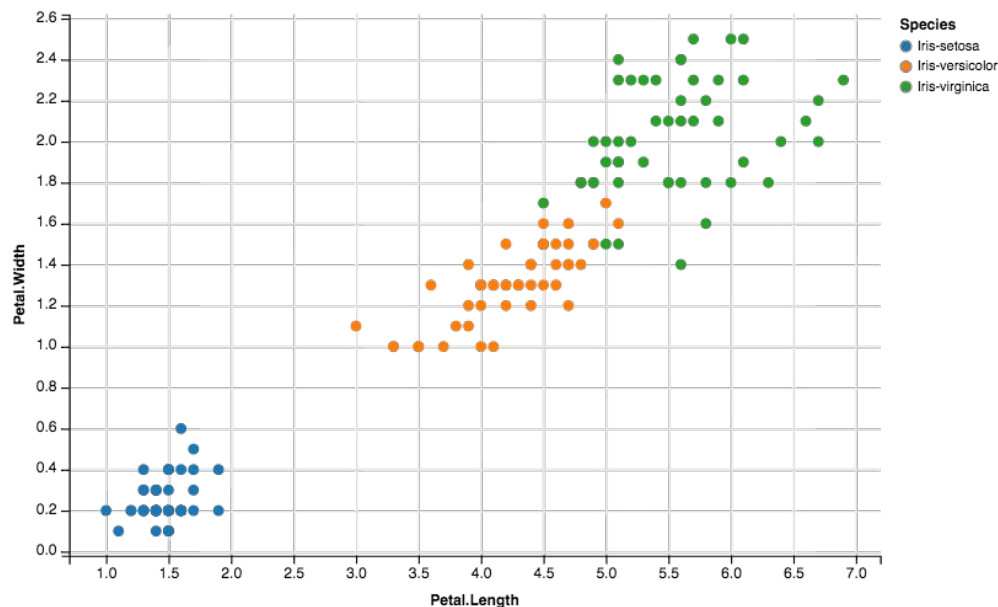
$$\text{MSE} = 0.0529$$
$$\text{BCE} = 0.26$$

$$\begin{aligned} w_0 &= -1.6 \\ w_1 &= +0.4 \\ w_2 &= +1 \\ f &= \text{sigmoid} \end{aligned}$$

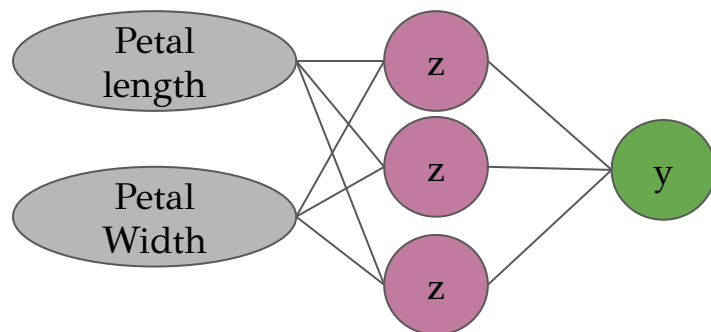


$$y_{\text{pred}} = 0.77, y = 1$$

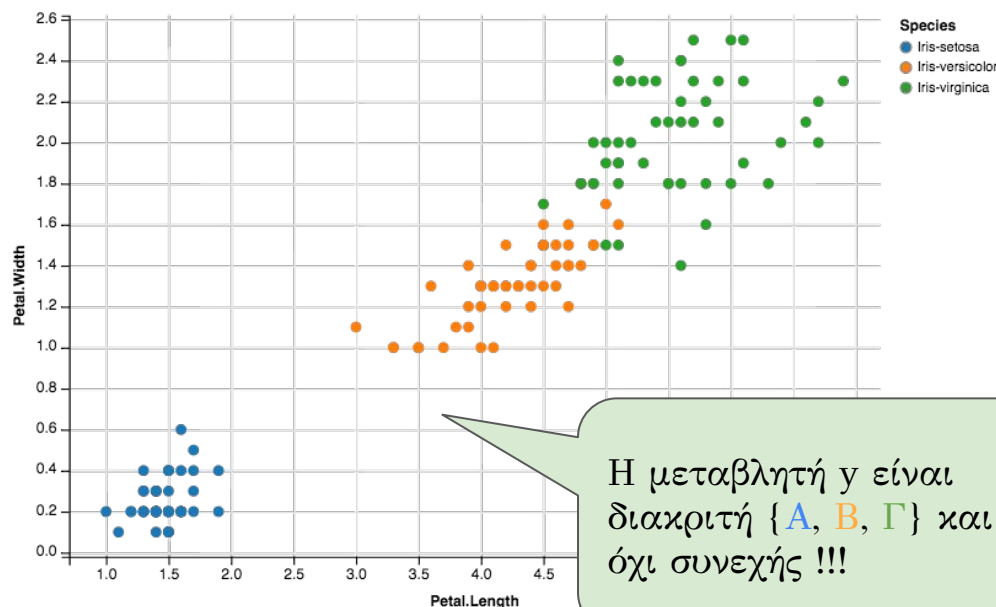
Ταξινόμηση σε πολλαπλές κατηγορίες



Αν το πρόβλημα είναι η ταξινόμηση σε τρεις κατηγορίες τότε το παρακάτω δίκτυο θα ήταν το ιδανικό για την επίλυση του προβλήματος????

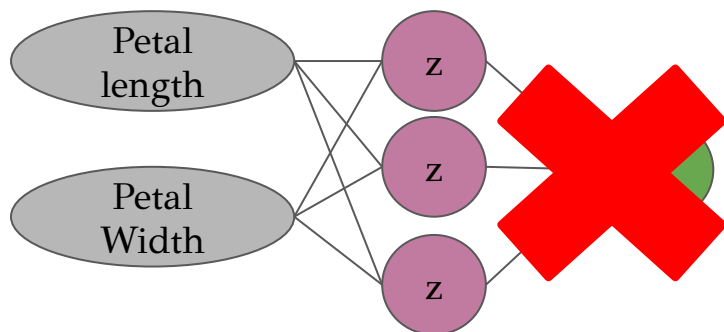


Ταξινόμηση σε πολλαπλές κατηγορίες



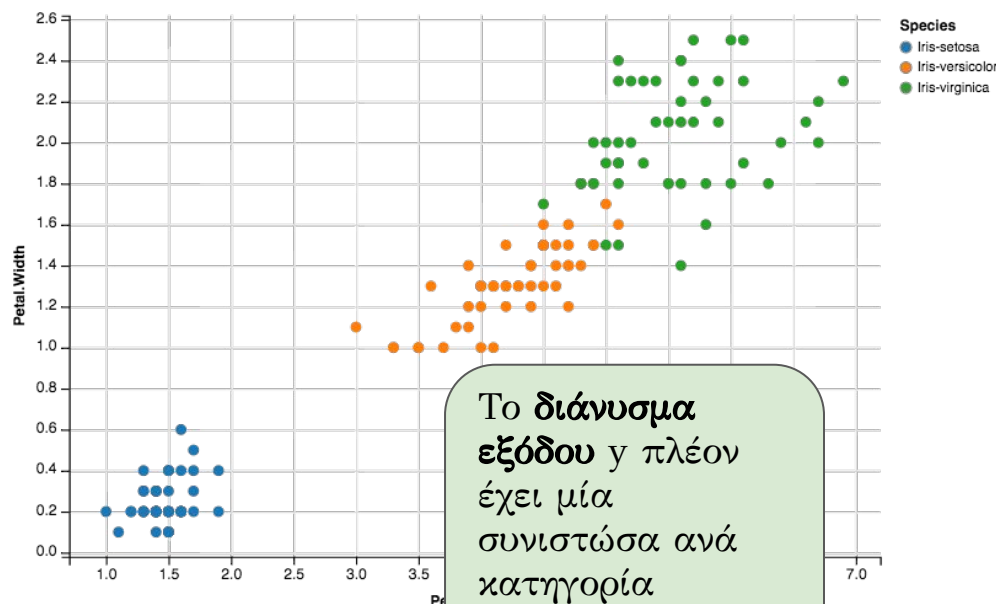
Η μεταβλητή y είναι διακριτή {Α, Β, Γ} και όχι συνεχής !!!

Αν το πρόβλημα είναι η ταξινόμηση σε τρεις κατηγορίες τότε το παρακάτω δίκτυο θα ήταν το ιδανικό για την επίλυση του προβλήματος????



Μια τέτοια διατύπωση δεν θα μπορούσε να υποστηρίξει πολλαπλές κλάσεις (2 στάθμες/νευρώνα) και θα προέκυπτε ένα πρόβλημα παλινδρόμησης!!

Ταξινόμηση σε πολλαπλές κατηγορίες



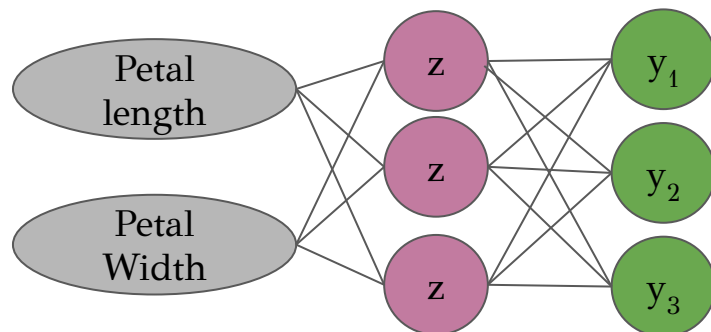
Το διάνυσμα
εξόδου y πλέον
έχει μία
συνιστώσα ανά
κατηγορία
 $y = \{y_1, y_2, y_3\}$

Μπορούμε να
γενικεύσουμε το **one-hot
encoding** σε 3 διακριτές
μεταβλητές ως εξής:

A : $y = \{1, 0, 0\}$

B : $y = \{0, 1, 0\}$

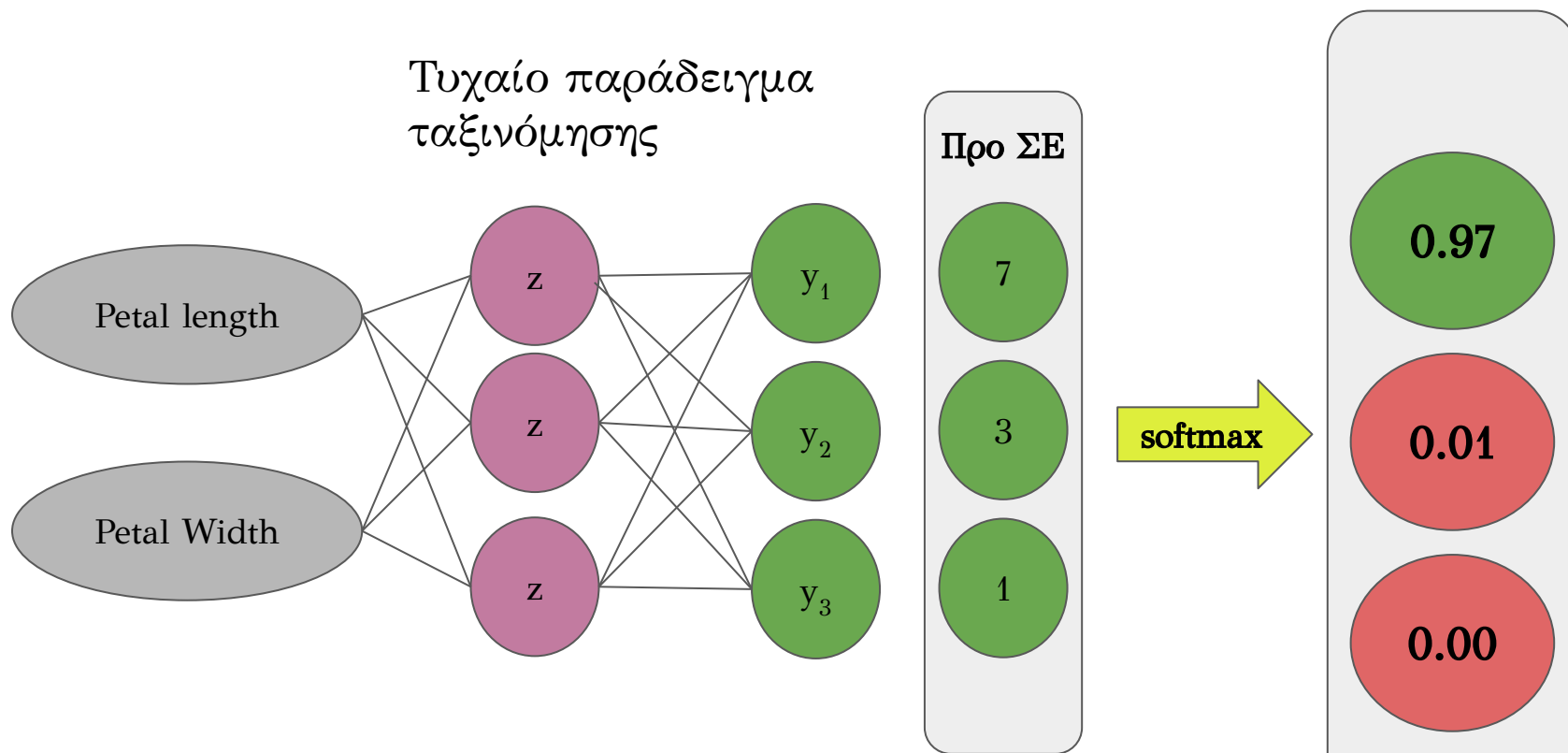
C : $y = \{0, 0, 1\}$



**Categorical Cross Entropy
Empirical Loss:**

$$CCE = -\frac{1}{N} \sum_{i=0}^N \sum_{j=0}^J y_j \cdot \log(\hat{y}_j) + (1 - y_j) \cdot \log(1 - \hat{y}_j)$$

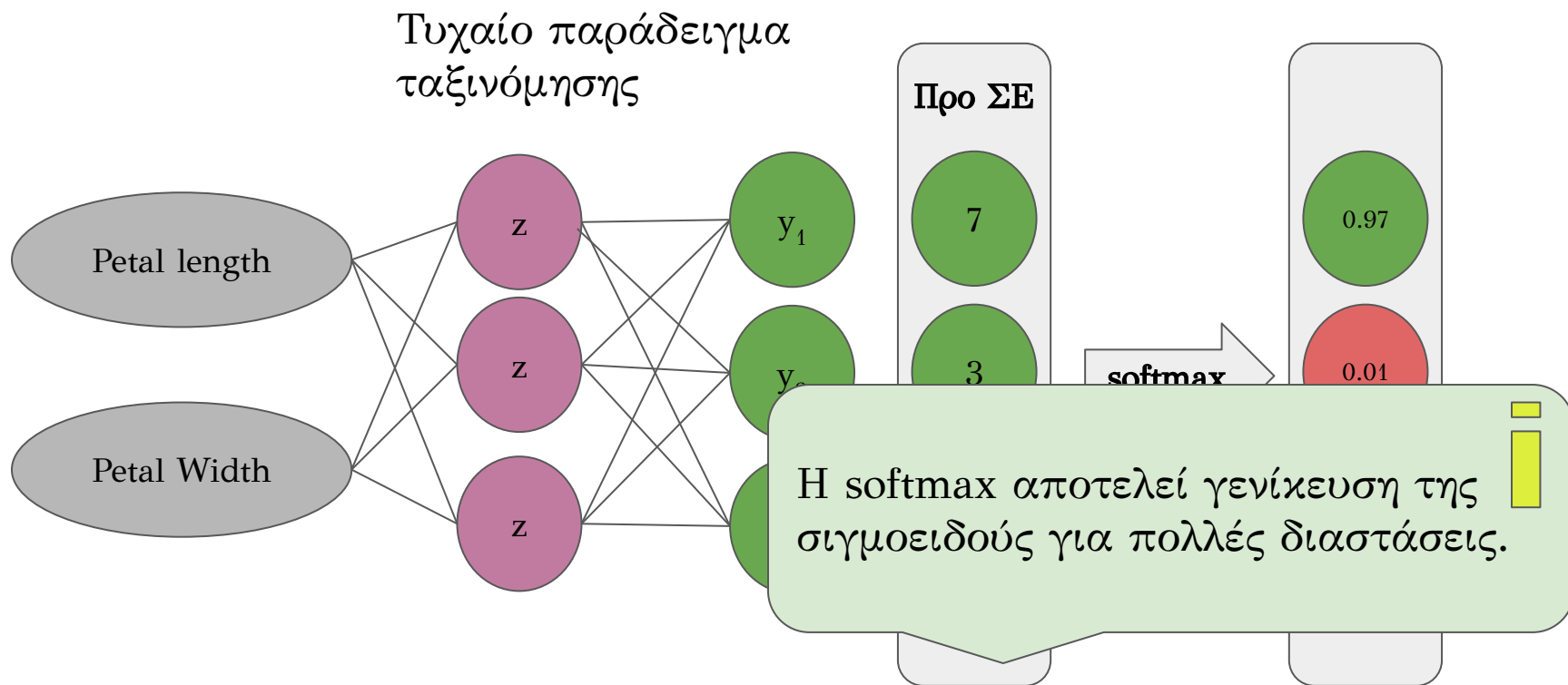
Ταξινόμηση σε πολλαπλές κατηγορίες



Softmax
συνάρτηση:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

Ταξινόμηση σε πολλαπλές κατηγορίες



Softmax
συνάρτηση:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$



RSLab

Remote Sensing Laboratory
National Technical University of Athens



Διαχείριση και Επεξεργασία Μεγάλων
Δεδομένων Παρατήρησης Γης



GitHub

<https://github.com/rslab-ntua>