



Machine Learning & Knowledge Extraction

DR KONSTANTINOS KARAMPIDIS

Πληροφορίες Μαθήματος

- ▶ Ωράριο:
 - ▶ Θεωρία: **Κάθε Τρίτη 09:00-13:00 – Αίθουσα 207**
 - ▶ Εργαστήριο: **13:00-14:00 ΕΡΓ6 (σύμφωνα με το πρόγραμμα που είναι αναρτημένο στο eclass)**
- ▶ Εργασίες
 - ▶ **1 project – Ομάδες έως 2 ατόμων – 80%**
 - ▶ **Εργαστηριακές ασκήσεις – 20%**
- ▶ Προαπαιτούμενα: Κανένα

Περιεχόμενο Μαθήματος

- ▶ Εισαγωγή στη Μηχανική Μάθηση - τι είναι, γιατί μας ενδιαφέρει, παραδείγματα προβλημάτων, η μηχανική μάθηση ως αναζήτηση, υπόθεση επαγωγικής μάθησης
- ▶ Επεξεργασία εισόδου – Μείωση διαστατικότητας- Αξιόλογηση
- ▶ Μέθοδοι επιβλεπόμενης μάθησης
- ▶ **Νευρωνικά Δίκτυα**
- ▶ Εξελικτική Μάθηση – Γενετικοί Αλγόριθμοι
- ▶ Μηχανική Μάθηση Βασιζόμενη σε Κανόνες
- ▶ Ενισχυτική Μάθηση
- ▶ Μάθηση Αναπαράστασης
- ▶ Εξόρυξη Δεδομένων

Εισαγωγή στα Νευρωνικά Δίκτυα

- ▶ Ιδιαίτερη προσέγγιση στη δημιουργία συστημάτων με νοημοσύνη
 - ▶ Δεν αναπαριστούν ρητά τη γνώση
 - ▶ Δεν υιοθετούν ειδικά σχεδιασμένους αλγορίθμους αναζήτησης
- ▶ Βασίζονται σε βιολογικά πρότυπα
- ▶ Μάθηση και μνήμη βασίζονται στη μεταβολή στην αγωγιμότητα των συνάψεων

Ιστορική Αναδρομή

- Η ανάπτυξη των νευρωνικών δικτύων χρονολογείται από τις αρχές της δεκαετίας του 1940. Η δημοτικότητά τους αυξήθηκε στα τέλη της δεκαετίας του 1980. Αυτό ήταν αποτέλεσμα της ανακάλυψης νέων τεχνικών και εξελίξεων και της γενικής προόδου στην τεχνολογία του υλικού των υπολογιστών.
- Ορισμένα NN είναι μοντέλα βιολογικών νευρωνικών δικτύων και ορισμένα όχι, αλλά ιστορικά, μεγάλο μέρος της έμπνευσης για τον τομέα των NN προήλθε από την επιθυμία να παραχθούν τεχνητά συστήματα ικανά για εξελιγμένους, ίσως και ευφυείς, υπολογισμούς παρόμοιους με αυτούς που εκτελεί συνήθως ο ανθρώπινος εγκέφαλος, και έτσι ενδεχομένως να βελτιωθεί η κατανόησή μας για τον ανθρώπινο εγκέφαλο.
- Τα περισσότερα NN έχουν κάποιου είδους κανόνα εκπαίδευσης. Με άλλα λόγια, τα NN μαθαίνουν από παραδείγματα (όπως τα παιδιά μαθαίνουν να αναγνωρίζουν σκύλους από παραδείγματα σκύλων) και παρουσιάζουν κάποια ικανότητα γενίκευσης πέρα από τα δεδομένα εκπαίδευσης.
- Οι νευρωνικοί υπολογιστές δεν πρέπει να θεωρούνται ανταγωνιστές των συμβατικών υπολογιστών. Αντίθετα, θα πρέπει να θεωρείται συμπληρωματικός, καθώς οι πιο επιτυχημένες νευρωνικές λύσεις ήταν εκείνες που λειτουργούσαν σε συνδυασμό με τις υπάρχουσες, παραδοσιακές τεχνικές.

Τι είναι τα Νευρωνικά Δίκτυα ?

- ❑ Μοντέλα του εγκεφάλου και του νευρικού συστήματος
- ❑ Εξαιρετικά παράλληλα
 - ❑ Επεξεργάζονται πληροφορίες πολύ περισσότερο σαν τον εγκέφαλο, από ό,τι ένας σειριακός υπολογιστής
- ❑ Μάθηση

- ❑ Πολύ απλές αρχές
- ❑ Πολύ σύνθετες συμπεριφορές

- ❑ Εφαρμογές
 - ❑ Ως ισχυροί λύτες προβλημάτων
 - ❑ Ως βιολογικά μοντέλα

Τεχνικές Νευρωνικών Δικτύων

- ❑ Οι υπολογιστές πρέπει να προγραμματίζονται ρητά
 - Αναλύστε το πρόβλημα που πρέπει να επιλυθεί.
 - Γράψτε τον κώδικα σε μια γλώσσα προγραμματισμού.
- ❑ Τα νευρωνικά δίκτυα μαθαίνουν από παραδείγματα
 - Δεν απαιτείται ρητή περιγραφή του προβλήματος.
 - Δεν απαιτείται προγραμματιστής.
 - Ο νευρωνικός υπολογιστής προσαρμόζεται κατά τη διάρκεια μιας περιόδου εκπαίδευσης, με βάση παραδείγματα παρόμοιων προβλημάτων, ακόμη και χωρίς την επιθυμητή λύση σε κάθε πρόβλημα. Μετά από επαρκή εκπαίδευση ο νευρωνικός υπολογιστής είναι σε θέση να συσχετίσει τα δεδομένα του προβλήματος με τις λύσεις, τις εισόδους με τις εξόδους, και στη συνέχεια είναι σε θέση να προσφέρει μία βιώσιμη λύση σε ένα ολοκαίνουργιο πρόβλημα.
 - Είναι σε θέση να γενικεύει ή να χειρίζεται ελλιπή δεδομένα.

NNs vs Computers (1/2)

- Ψηφιακοί υπολογιστές
 - Συμπερασματικός συλλογισμός. Εφαρμόζουμε γνωστούς κανόνες σε δεδομένα εισόδου για να παράγουμε αποτελέσματα.
 - Ο υπολογισμός είναι συγκεντρωτικός, σύγχρονος και σειριακός.
 - Η μνήμη είναι πακεταρισμένη, κυριολεκτικά αποθηκευμένη και με δυνατότητα διευθυνσιοδότησης θέσης.
 - Δεν έχει ανοχή σε σφάλματα. Ένα τρανζίστορ χαλάει και δεν λειτουργεί πλέον.
 - Ακρίβεια.
 - Στατική συνδεσιμότητα.
 - Εφαρμόζεται αν υπάρχουν καλά καθορισμένοι κανόνες με ακριβή δεδομένα εισόδου.

NNs vs Computers (2/2)

□ Νευρωνικά δίκτυα

- Επαγωγική συλλογιστική. Δεδομένων των δεδομένων εισόδου και εξόδου (παραδείγματα εκπαίδευσης), κατασκευάζουμε τους κανόνες.
- Ο υπολογισμός είναι συλλογικός, ασύγχρονος και παράλληλος.
- Η μνήμη είναι κατανομημένη, εσωτερικευμένη, βραχυπρόθεσμη και με δυνατότητα διευθυνσιοδότησης περιεχομένου.
- Ανοχή σε σφάλματα, πλεονασμός και καταμερισμός ευθυνών.
- Ανακριβής.
- Δυναμική συνδεσιμότητα.

- Εφαρμόζεται εάν οι κανόνες είναι άγνωστοι ή περίπλοκοι ή εάν τα δεδομένα είναι θορυβώδη ή μερικά.

Εφαρμογές Νευρωνικών Δικτύων (1/2)

- Ταξινόμηση
 - Μάρκετινγκ: ταξινόμηση των καταναλωτικών δαπανών
 - Άμυνα: ταξινόμηση εικόνων ραντάρ και σόναρ
 - Γεωργία και αλιεία: ταξινόμηση φρούτων και αλιευμάτων
 - Ιατρική: εικόνα υπερήχων/ηλεκτροκαρδιογραφήματος ταξινόμηση, ηλεκτροεγκεφαλογραφήματα
- Αναγνώριση και ταυτοποίηση
 - Γενική πληροφορική και τηλεπικοινωνίες: ομιλία, υπολογιστική όραση, και αναγνώρισης γραφής
 - Οικονομικά: επαλήθευση υπογραφής και επαλήθευση τραπεζικών χαρτονομισμάτων

Εφαρμογές Νευρωνικών Δικτύων (2/2)

- Αξιολόγηση
 - Μηχανική: παρακολούθηση και έλεγχος επιθεώρησης προϊόντων
 - Άμυνα: εντοπισμός στόχων
 - Ασφάλεια: ανίχνευση κίνησης, επιτήρηση με κάμερες, ταυτοποίηση δακτυλικών αποτυπωμάτων
- Πρόγνωση και πρόβλεψη
 - Στα οικονομικά: πρόβλεψη συναλλαγματικών ισοτιμιών και μετοχών
 - Στη γεωργία: πρόβλεψη της απόδοσης των καλλιεργειών
 - Στο μάρκετινγκ: πρόβλεψη πωλήσεων
 - Στη μετεωρολογία: πρόβλεψη καιρού

Βιολογική έμπνευση

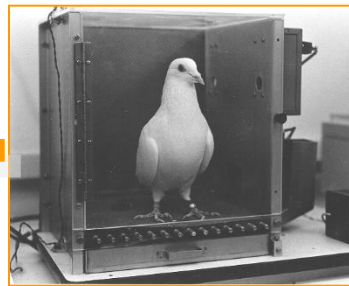
- Τα ζώα είναι σε θέση να αντιδρούν προσαρμοστικά στις αλλαγές του εξωτερικού και του εσωτερικού τους περιβάλλοντος και χρησιμοποιούν το νευρικό τους σύστημα για να εκτελέσουν αυτές τις συμπεριφορές.
- Ένα κατάλληλο μοντέλο/προσομοίωση του νευρικού συστήματος θα πρέπει να είναι σε θέση να παράγει παρόμοιες αντιδράσεις και συμπεριφορές σε τεχνητά συστήματα.
- Το νευρικό σύστημα δομείται από σχετικά απλές μονάδες, τους νευρώνες, οπότε η αντιγραφή της συμπεριφοράς και της λειτουργικότητάς τους θα πρέπει να είναι η λύση.

Βιολογικά NNs (1/3)

□ Τα περιστέρια ως ειδικοί στην τέχνη (Watanabe et al. 1995)

➤ Πείραμα:

- Περιστέρι σε κουτί Skinner
- Παρουσίαση πινάκων ζωγραφικής δύο διαφορετικών καλλιτεχνών
- Επιβράβευση όταν παρουσιάζεται ένας συγκεκριμένος καλλιτέχνης

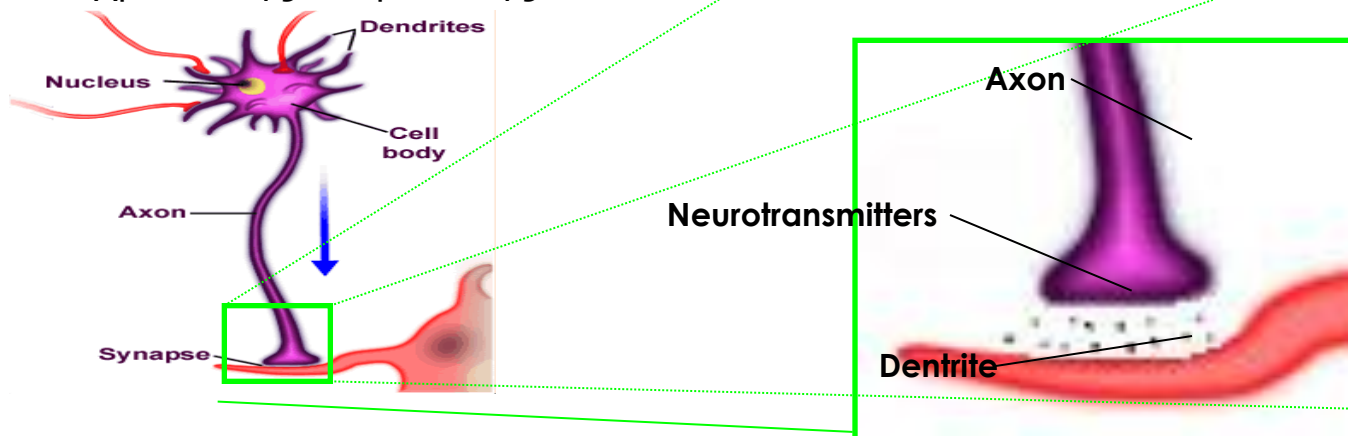


Βιολογικά NNs (2/3)

- ❑ Τα περιστέρια μπόρεσαν να διακρίνουν μεταξύ Βαν Γκογκ και Σαγκάλ με ακρίβεια 95% (όταν τους παρουσιάστηκαν εικόνες στις οποίες είχαν εκπαιδευτεί).
- ❑ Η διάκριση εξακολουθούσε να είναι επιτυχής κατά 85% για πίνακες των καλλιτεχνών που δεν είχαν δει προηγουμένως
- ❑ Τα περιστέρια δεν απομνημονεύουν απλώς τους πίνακες
- ❑ Μπορούν να εξάγουν και να αναγνωρίζουν μοτίβα (το «στυλ»)
- ❑ Γενικεύουν για να κάνουν προβλέψεις

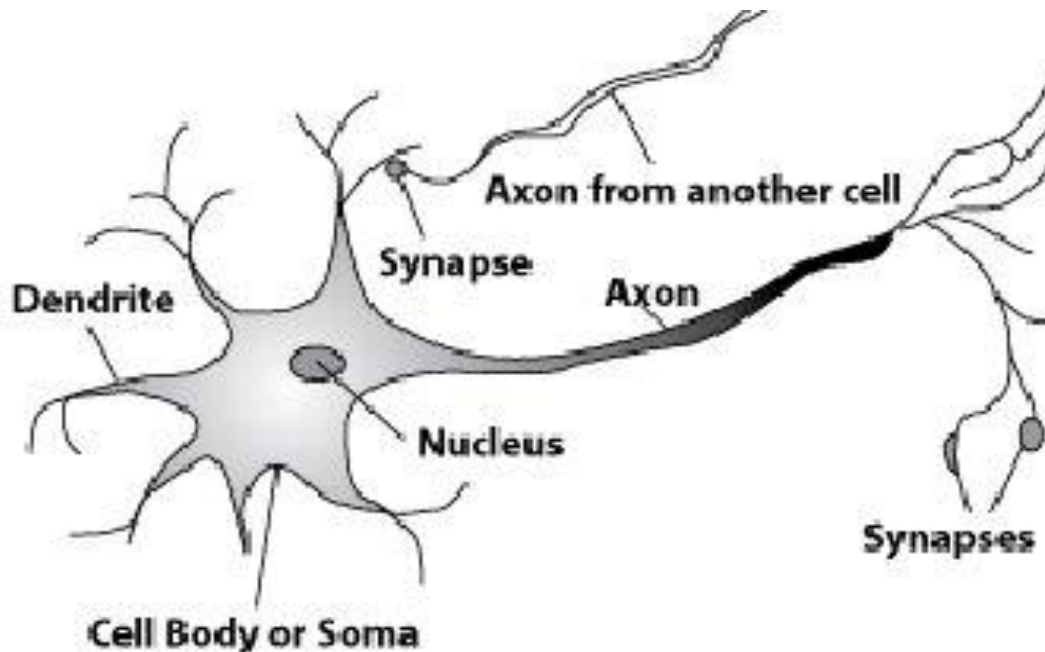
Βιολογικά NNs (3/3)

- Εγκέφαλος: περίπου 100 δισεκατομμύρια νευρώνες. Κάθε νευρωνικό κύτταρο χρησιμοποιεί βιοχημικές αντιδράσεις για να λαμβάνει, να επεξεργάζεται και να μεταδίδει πληροφορίες.
- Κάθε τερματικό κουμπί συνδέεται με άλλους νευρώνες μέσω ενός μικρού χάσματος που ονομάζεται σύναψη.
- Το δενδριτικό δέντρο του νευρώνα συνδέεται με χιλιάδες γειτονικούς νευρώνες. Όταν ένας από αυτούς τους νευρώνες πυροδοτείται, ένα θετικό ή αρνητικό φορτίο λαμβάνεται από έναν από τους δενδρίτες. Οι εντάσεις όλων των φορτίων που λαμβάνονται αθροίζονται μέσω των διαδικασιών της χωρικής και χρονικής άθροισης



Βιολογικός Νευρώνας

- ▶ Τα σήματα εισέρχονται στο σώμα του νευρώνα και συνδυάζονται
- ▶ Αν το αποτέλεσμα ξεπερνά ένα συγκεκριμένο κατώφλι, τότε διαδίδεται μέσω του άξονα σε άλλους νευρώνες

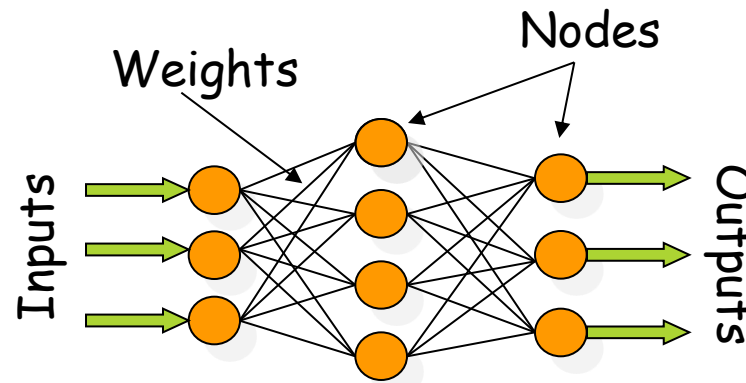


Βιολογικοί Νευρώνες

- ▶ Σε ένα ανθρώπινο εγκέφαλο υπάρχουν περίπου 100 δισεκατομμύρια νευρώνες.
- ▶ Κάθε νευρώνας συνδέεται περίπου με άλλους 1000 νευρώνες
- ▶ Υπάρχουν περίπου 100 τρισεκατομμύρια συνάψεις που επηρεάζουν τη λειτουργία του εγκεφάλου
- ▶ Η αντιγραφή της δομής και λειτουργίας του εγκεφάλου σε τέτοια κλίμακα είναι αδύνατη
 - ▶ Τα μοντέλα που κατασκευάζονται έχουν μερικές χιλιάδες τεχνητούς νευρώνες, περίπου 1 εκ. συνάψεις και πολύ περιορισμένη λειτουργικότητα
- ▶ Ο εγκέφαλος είναι σε θέση να πάρει πολύπλοκες αποφάσεις πολύ γρήγορα αν και ο χρόνος απόκρισης των νευρώνων είναι της τάξης των msec
- ▶ Πρόκειται για παράλληλο και κατανεμημένο σύστημα καθώς η υπολογιστική ικανότητα και η πληροφορία είναι διαμοιρασμένη σε όλο του τον όγκο

Τεχνητά NNs: Βασικά στοιχεία

- Τα ANN ενσωματώνουν τα δύο θεμελιώδη συστατικά των βιολογικών νευρωνικών δικτύων:
 - Νευρώνες -> Κόμβοι
 - συνάψεις -> βάρη



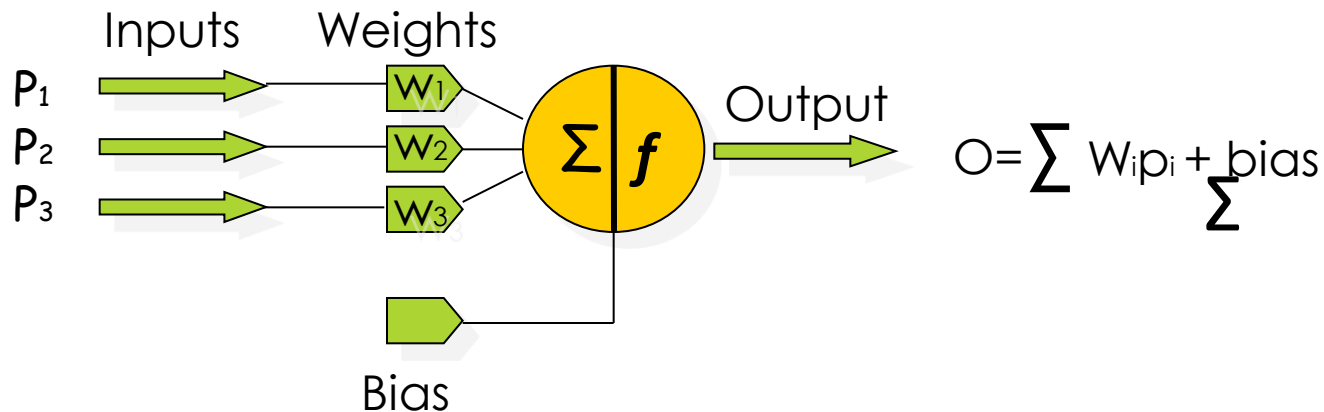
- Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από πολλούς τεχνητούς νευρώνες που συνδέονται μεταξύ τους σύμφωνα με μια συγκεκριμένη αρχιτεκτονική δικτύου. Ο στόχος του νευρωνικού δικτύου είναι να μετατρέψει τις εισόδους σε εξόδους με νόημα.

Τεχνητός Νευρώνας

- ▶ Υπολογιστικό μοντέλο με μέρη που αντιστοιχούν σε αυτά του βιολογικού νευρώνα
- ▶ Δέχεται κάποια σήματα εισόδου που αντιστοιχούν σε συνεχείς μεταβλητές (σε αντίθεση με τους ηλεκτρικούς παλμούς του εγκεφάλου)
- ▶ Το σήμα εισόδου μεταβάλλεται από μια τιμή βάρους που έχει τον ρόλο αντίστοιχο μιας σύναψης. Η τιμή μπορεί να είναι θετική ή αρνητική αντιστοιχιζόμενη με την επιταχυντική ή επιβραδυντική λειτουργία της σύναψης
- ▶ Το σώμα του νευρώνα χωρίζεται σε δύο μέρη
 - ▶ *Αθροιστής*: προσθέτει τα σήματα εισόδου με βάρη και παράγει την S ποσότητα
 - ▶ *Συνάρτηση ενεργοποίησης ή κατωφλιού*, μη γραμμικό φίλτρο που διαμορφώνει την τελική τιμή της εξόδου y σε συνάρτηση με την ποσότητα S

Τεχνητά NNs: Βασικά στοιχεία

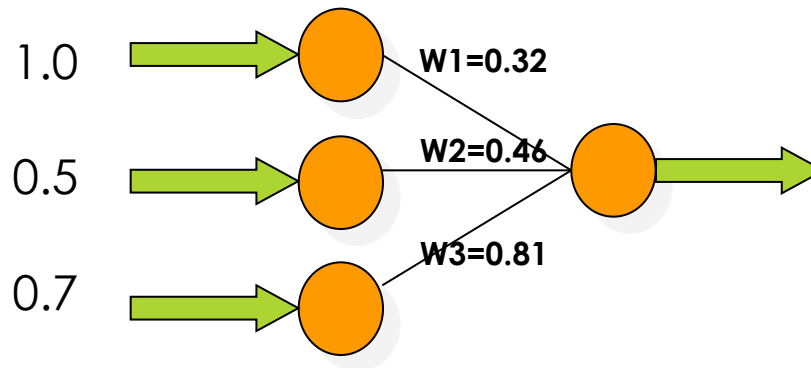
- ▶ Απαιτείται ένας αριθμός νευρώνων, οι οποίοι πρέπει να συνδεθούν μεταξύ τους σε ένα νευρωνικό δίκτυο. Οι νευρώνες διατάσσονται σε στρώματα.



- ▶ Κάθε νευρώνας στο δίκτυο είναι συνήθως μια απλή μονάδα επεξεργασίας που δέχεται μία ή περισσότερες εισόδους και παράγει μια έξοδο. Σε κάθε νευρώνα, κάθε είσοδος έχει ένα σχετικό βάρος το οποίο τροποποιεί την ισχύ κάθε εισόδου. Ο νευρώνας απλά προσθέτει όλες τις εισόδους και υπολογίζει μια έξοδο που θα μεταβιβαστεί.

Τεχνητά NNs: Βασικά στοιχεία

- ▶ Τροφοδότηση δεδομένων μέσω του δικτύου:



$$\text{Έξοδος} = (1.0 \times 0.32) + (0.5 \times 0.46) + (0.7 \times 0.81) = 1.117$$

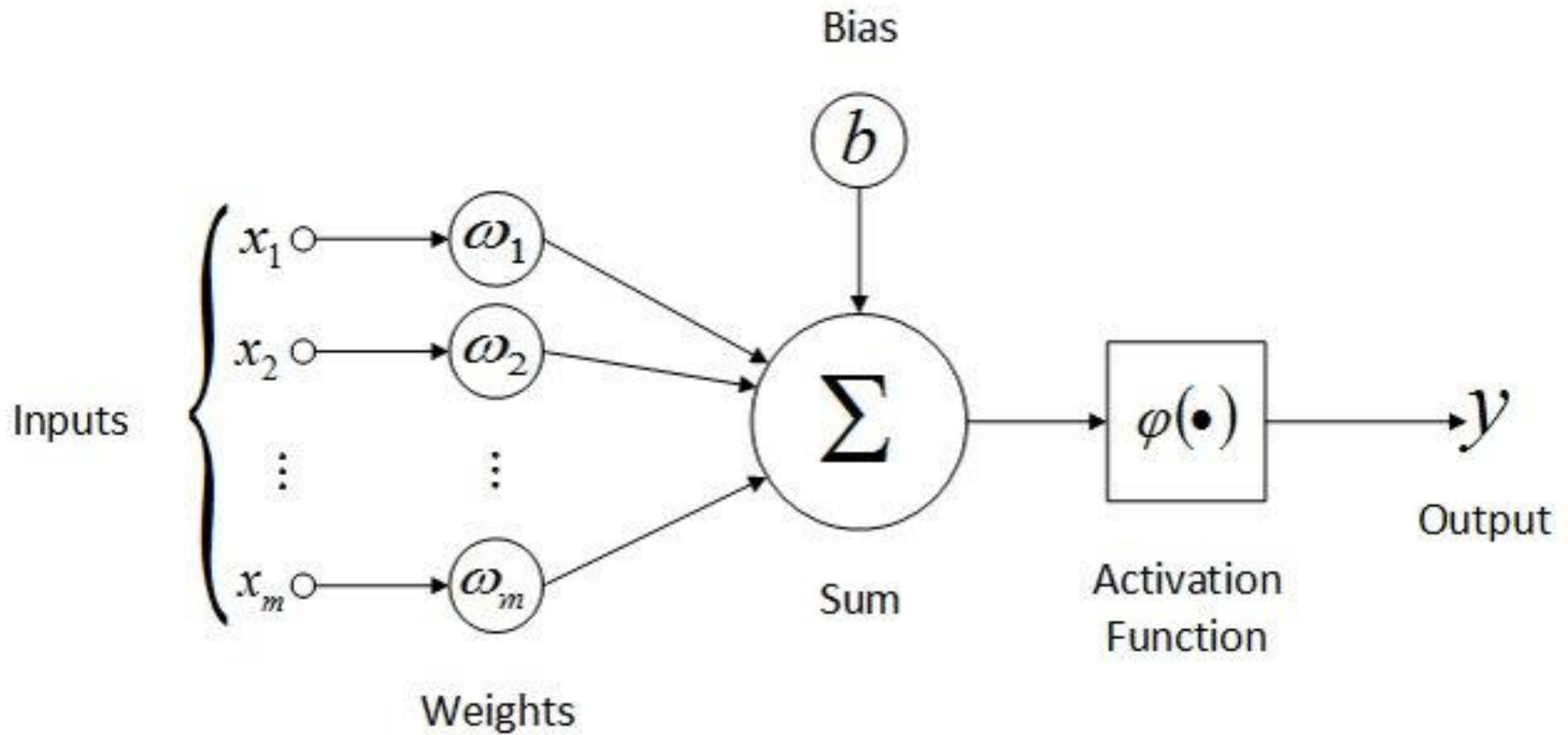
- ▶ Περιορισμός της εξόδου μεταξύ του εύρους 0 -1

$$\frac{1}{1 + e^{1.117}} = 0.466$$

Τροφοδότηση NNs

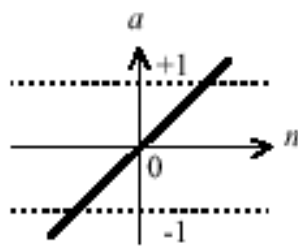
- Τα δεδομένα παρουσιάζονται στο δίκτυο με τη μορφή ενεργοποιήσεων στο στρώμα εισόδου.
- Παραδείγματα
 - Ένταση εικονοστοιχείου (για εικόνες)
 - Τιμές μετοχών (για πρόβλεψη χρηματιστηρίου)
- Τα δεδομένα απαιτούν συνήθως προεπεξεργασία
- Πώς να αναπαραστήσουμε πιο αφηρημένα δεδομένα, π.χ. ένα όνομα;
 - Επιλέξτε ένα μοτίβο, π.χ.
 - 0-0-1 για το «Chris»
 - 0-1-0 για το «Becky».

Τεχνητός Νευρώνας



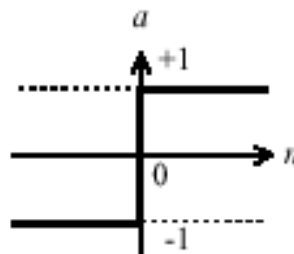
Συναρτήσεις ενεργοποίησης

- Η συνάρτηση ενεργοποίησης περιγράφει τη συμπεριφορά εξόδου των νευρώνων. Γενικά είναι μη γραμμική. Οι γραμμικές συναρτήσεις είναι περιορισμένες επειδή η έξοδος είναι απλά ανάλογη της εισόδου.



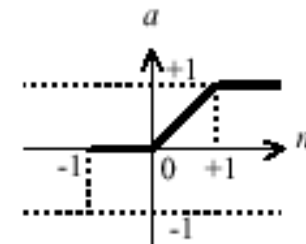
$$a = \text{purelin}(n)$$

Linear Transfer Function



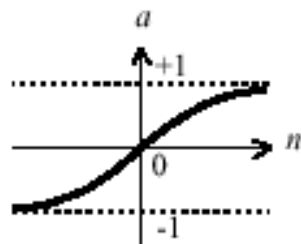
$$a = \text{hardlims}(n)$$

Symmetric Hard Limit Trans. Funct.



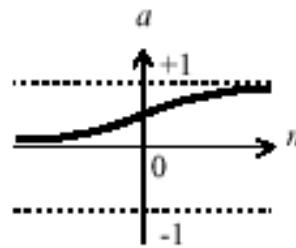
$$a = \text{satlin}(n)$$

Satlin Transfer Function



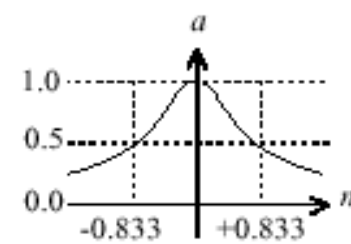
$$a = \text{tansig}(n)$$

Tan-Sigmoid Transfer Function



$$a = \text{logsig}(n)$$

Log-Sigmoid Transfer Function



$$a = \text{radbas}(n)$$

Radial Basis Function

Τυπικές Συναρτήσεις Ενεργοποίησης

- ▶ *Βηματική (step)*: δίνει στην έξοδο αποτέλεσμα (συνήθως 1) μόνο όταν η τιμή εισόδου είναι πάνω από ένα κατώφλι T
- ▶ *Προσήμου (sign)*: δίνει στην έξοδο αρνητική (θετική) τιμή (+/-1) αν η τιμή εισόδου που παρέχει ο αθροιστής είναι μικρότερη (μεγαλύτερη) από ένα κατώφλι T
- ▶ *Σιγμοειδής (sigmoid)*: εκφράζεται από τη φόρμουλα:

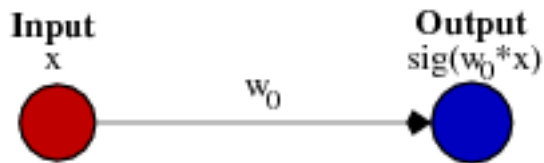
$$\Phi(S) = \frac{1}{1 + e^{-aS}}$$

- ▶ όπου a είναι ο συντελεστής ρύθμισης της ταχύτητας μετάβασης μεταξύ δύο ασυμπτωτικών γραμμών
- ▶ παρέχει μη γραμμικότητα στο νευρώνα

Bias

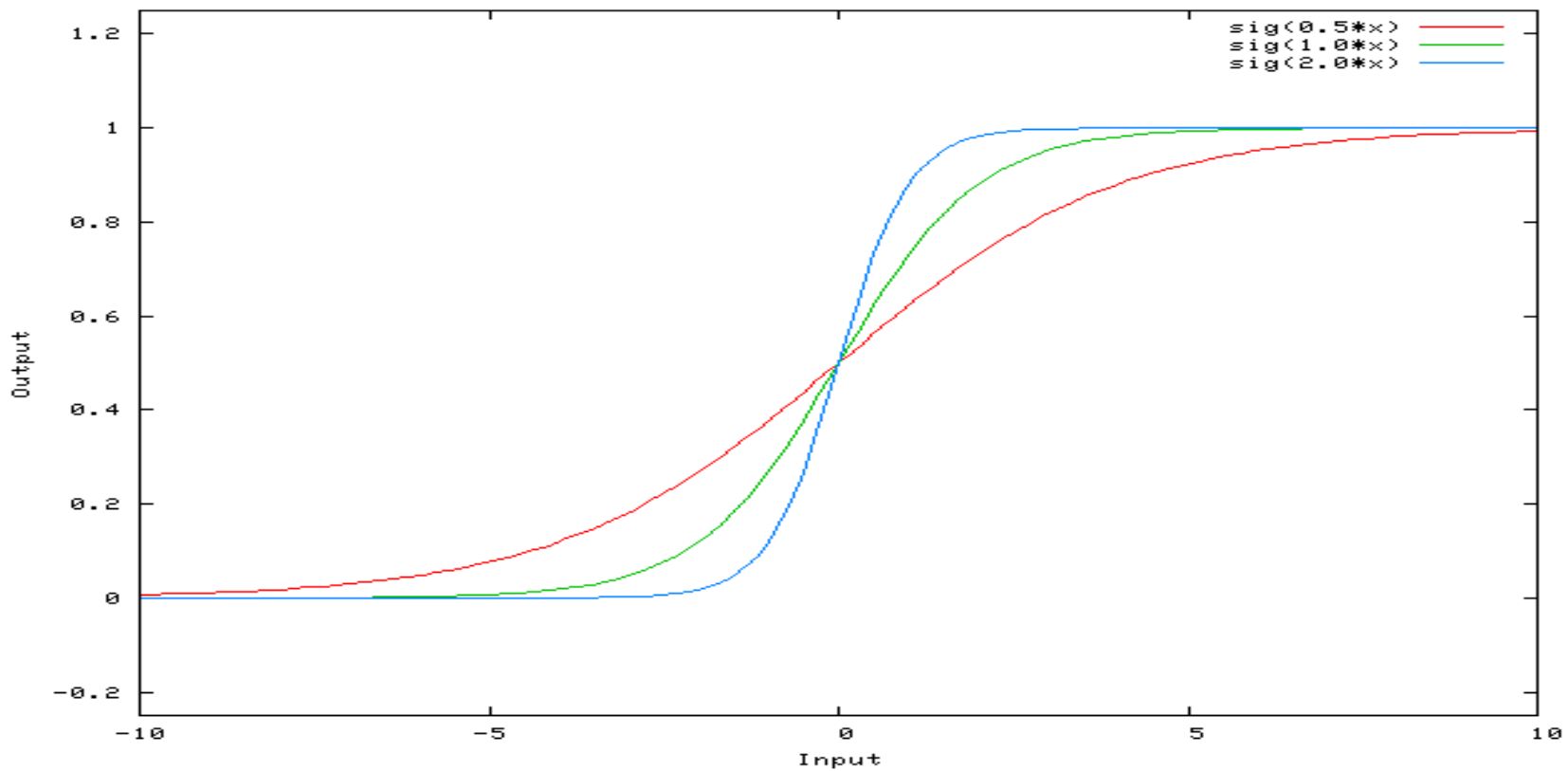
A bias value allows you to shift the activation function to the left or right, which may be critical for successful learning

Consider this 1-input, 1-output network that has no bias:



The output of the network is computed by multiplying the input (x) by the weight (w_0) and passing the result through an activation function (e.g. a sigmoid function.)

Bias

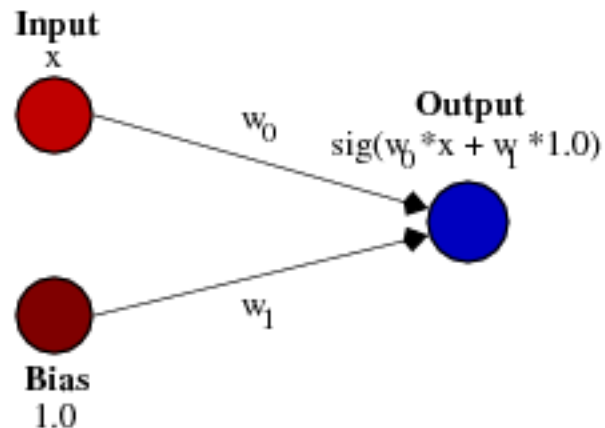


Here is the function that this network computes, for various values of w_0

Bias

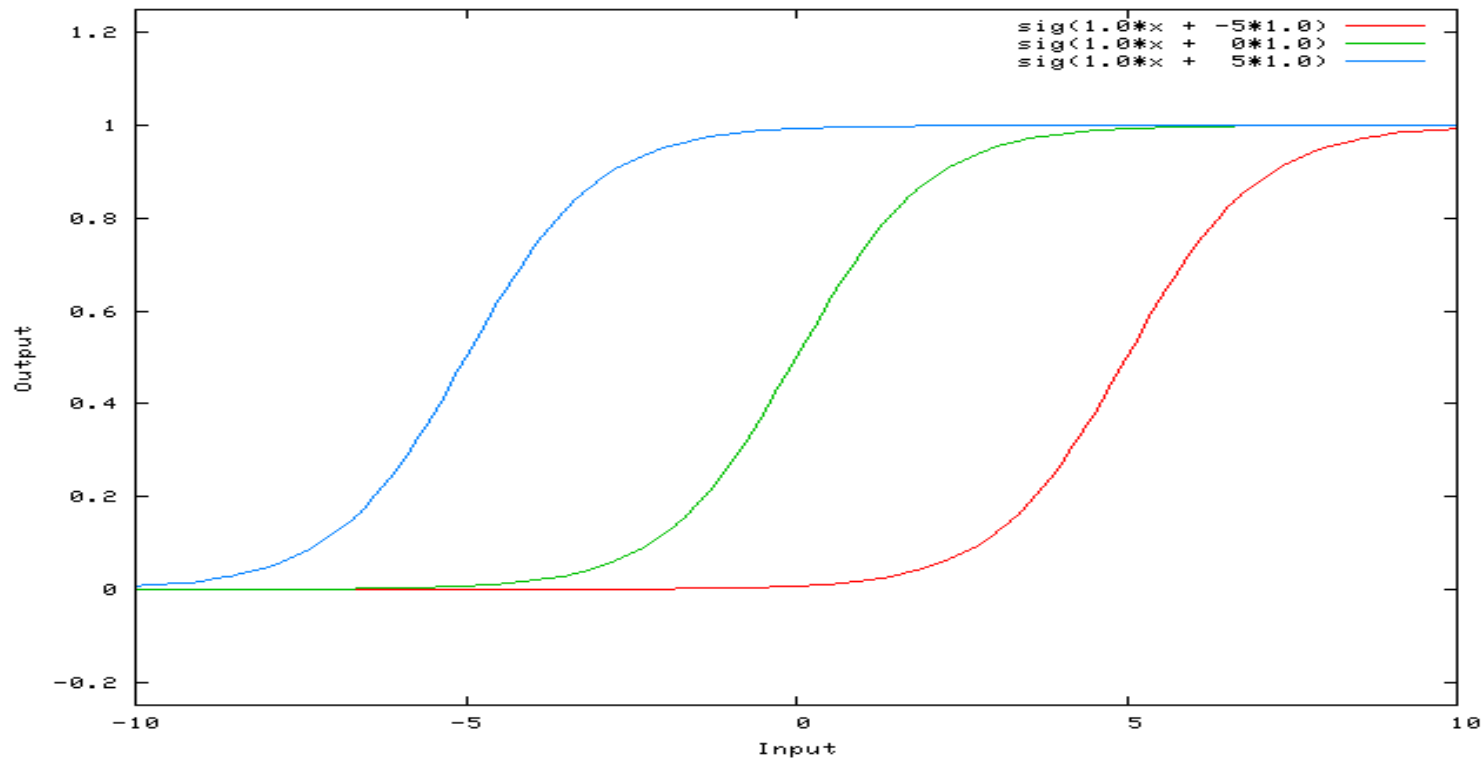
Changing the weight w_0 essentially changes the "steepness" of the sigmoid. That's useful, but what if you wanted the network to output 0 when x is 2? Just changing the steepness of the sigmoid won't really work -- you want to be able to shift the entire curve to the right.

That's exactly what the bias allows you to do. If we add a bias to that network



Bias

The output of the network becomes $\text{sig}(w_0 * x + w_1 * 1.0)$. Here is what the output of the network looks like for various values of w_1

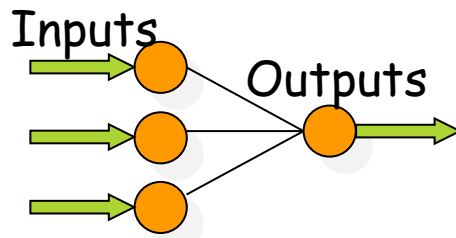


Τεχνητά Νευρωνικά Δίκτυα

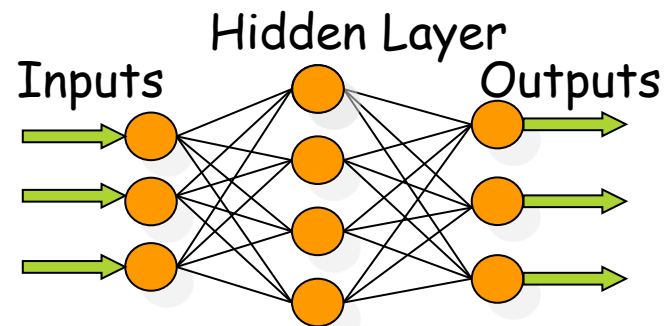
- ▶ Συστήματα επεξεργασίας δεδομένων αποτελούμενα από πλήθος τεχνητών νευρώνων σε δομή παρόμοια με αυτή του ανθρώπινου εγκεφάλου
- ▶ Οργάνωση σε σειρά από στρώματα / επίπεδα (layers)
- ▶ Το πρώτο επίπεδο, το επίπεδο εισόδου, χρησιμοποιείται για εισαγωγή δεδομένων
 - ▶ Δεν έχει νευρώνες και δεν εκτελεί υπολογισμούς
- ▶ Έπειτα υπάρχει μια σειρά από ενδιάμεσα ή κρυφά επίπεδα
- ▶ Τέλος υπάρχει το επίπεδο εξόδου

Τεχνητά Νευρωνικά Δίκτυα

- ▶ Ένα κρυφό επίπεδο μαθαίνει να επανακωδικοποιεί (ή να παρέχει μια αναπαράσταση για) τις εισόδους. Μπορούν να χρησιμοποιηθούν περισσότερα από ένα κρυφά στρώματα.
- ▶ Αυτή η αρχιτεκτονική είναι πιο ισχυρή από τα δίκτυα ενός επιπέδου: μπορεί να αποδειχθεί ότι οποιαδήποτε απεικόνιση μπορεί να μαθευτεί, δεδομένων δύο κρυφών επιπέδων.

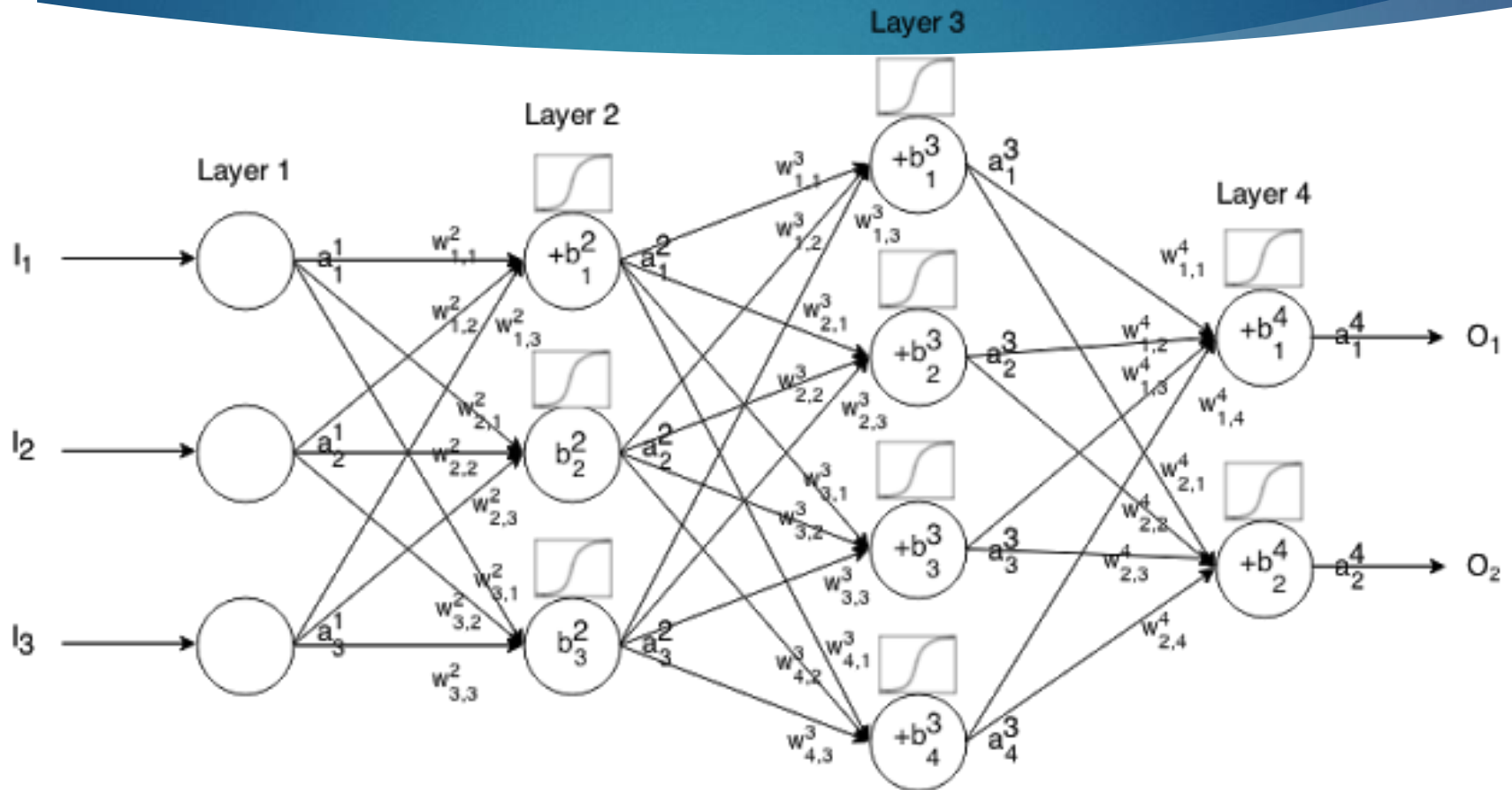


Single Layer ANN



ANN with hidden layer

Τεχνητά Νευρωνικά Δίκτυα – ΤΝΔ



Χαρακτηριστικά ΤΝΔ

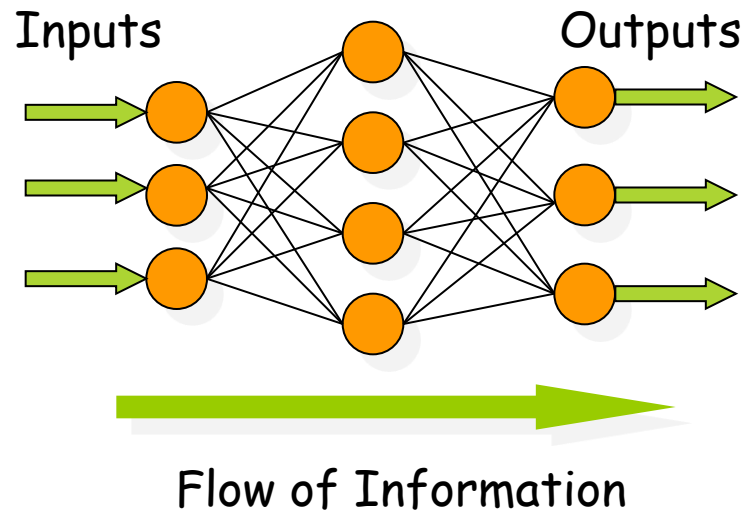
- ▶ Οι νευρώνες σε κάθε στρώμα μπορεί να είναι είτε:
 - ▶ *Πλήρως συνδεδεμένοι*: συνδέονται με όλους τους νευρώνες του επόμενου επιπέδου
 - ▶ *Μερικώς συνδεδεμένοι*: συνδέονται με μερικούς νευρώνες του επόμενου επιπέδου
- ▶ Δύο είδη ΤΝΔ:
 - ▶ *Απλής τροφοδότησης (feedforward)*: οι συνδέσεις ακολουθούν μόνο την πρόσθια κατεύθυνση
 - ▶ *Ανατροφοδότησης (feedback)*: οι συνδέσεις μπορεί να έχουν και οπίσθια κατεύθυνση. Επίσης, μπορεί να υπάρχουν συνδέσεις μεταξύ νευρώνων του ίδιου επιπέδου

Χαρακτηριστικά ΤΝΔ

- ▶ ΤΝΔ ανατροφοδότησης:
 - ▶ Υπολογισμός διαφόρων μεγεθών γίνεται σε 2 στάδια:
 - ▶ Πρώτο στάδιο: υπολογισμός μεγεθών με βάση συνδέσεις απλής τροφοδότησης
 - ▶ Δεύτερο στάδιο: υπολογισμός μεγεθών με βάση συνδέσεις ανατροφοδότησης
- ▶ Οι περισσότερες εφαρμογές ΤΝΔ αντιστοιχούν σε ΤΝΔ απλής τροφοδότησης

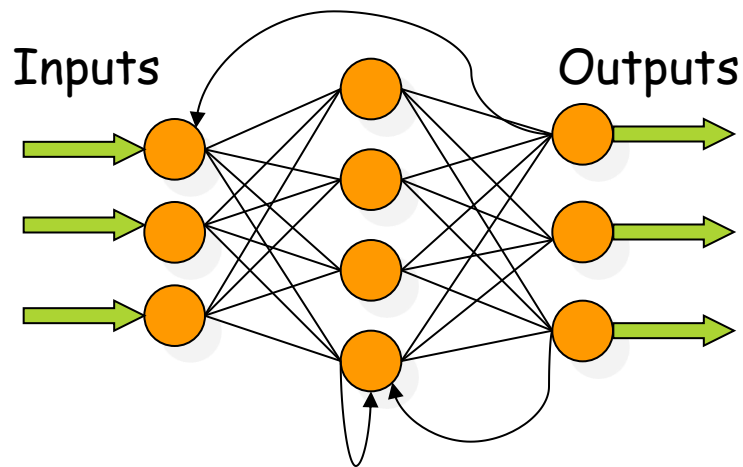
Αρχιτεκτονικές

- Τα feed-forward ANN επιτρέπουν στα σήματα να ταξιδεύουν προς μία μόνο κατεύθυνση: από την είσοδο στην έξοδο. Δεν υπάρχει ανατροφοδότηση (βρόχοι), δηλαδή η έξοδος οποιουδήποτε επιπέδου δεν επηρεάζει το ίδιο επίπεδο. Τα feed-forward ANN τείνουν να είναι απλά δίκτυα που συνδέουν τις εισόδους με τις εξόδους. Χρησιμοποιούνται ευρέως στην αναγνώριση προτύπων.



Αρχιτεκτονικές

- Τα δίκτυα ανατροφοδότησης μπορούν να έχουν σήματα που ταξιδεύουν και προς τις δύο κατευθύνσεις εισάγοντας βρόχους στο δίκτυο. Τα δίκτυα ανατροφοδότησης είναι πολύ ισχυρά και μπορούν να γίνουν εξαιρετικά περίπλοκα. Τα δίκτυα ανατροφοδότησης είναι δυναμικά- η «κατάστασή» τους αλλάζει συνεχώς μέχρι να φτάσουν σε ένα σημείο ισορροπίας. Παραμένουν στο σημείο ισορροπίας έως ότου αλλάξει η είσοδος και πρέπει να βρεθεί μια νέα ισορροπία



Μάθηση και Ανάκληση σε ΤΝΔ

- ▶ *Μάθηση / Εκπαίδευση*: διαδικασία τροποποίησης τιμής βαρών ΤΝΔ ώστε με βάση ένα συγκεκριμένο διάνυσμα εισόδου να παραχθεί ένα συγκεκριμένο διάνυσμα εξόδου
- ▶ *Ανάκληση*: διαδικασία υπολογισμού διανύσματος εξόδου με βάση ένα διάνυσμα εισόδου και τις τιμές βαρών (που έχουν μαθευτεί)

Μάθηση σε ΤΝΔ

▶ 3 Είδη Μάθησης ΤΝΔ:

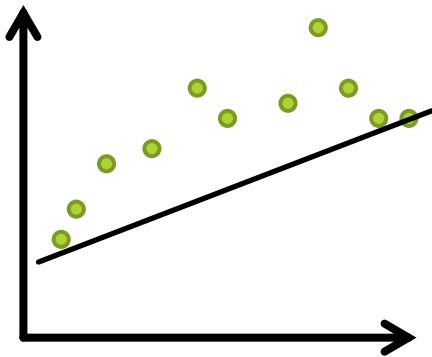
- ▶ *Επαγωγική μάθηση*: δίδονται ζευγάρια διανυσμάτων εισόδου-εξόδου για εκπαίδευση. Αρχικά, οι τιμές βαρών δεν παράγουν την επιθυμητή έξοδο. Η διαφορά πραγματικής με παραγόμενης εξόδου ονομάζεται σφάλμα. Με βάση αυτή γίνεται η αναπροσαρμογή των βαρών.
- ▶ *Βαθμολογημένη μάθηση*: η έξοδος χαρακτηρίζεται με βάση μια κλίμακα ως καλή ή κακή. Ο χαρακτηρισμός αυτός οδηγεί την αναπροσαρμογή βαρών.
- ▶ *Μη επαγωγική μάθηση*: αυτό-οργάνωση ΤΝΔ με βάση τα διανύσματα εισόδου. Η εσωτερική οργάνωση πραγματοποιείται έτσι ώστε ένας μόνο νευρώνας να αντιδρά σθεναρά σε ένα σύνολο εισόδου. Τα σύνολα εισόδου είναι οι πραγματικές έννοιες που το ΤΝΔ θα πρέπει να μάθει

Μάθηση σε ΤΝΔ

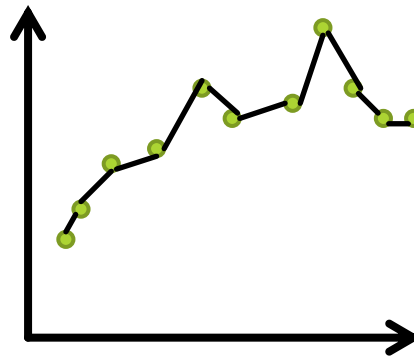
- ▶ Κύριο είδος μάθησης: επαγωγική. Οι περισσότερες εφαρμογές την υιοθετούν.
- ▶ 4 αλγόριθμοι επαγωγικής μάθησης ΤΝΔ
 - ▶ *Μάθηση Κανόνα Δέλτα*: το λάθος ελαχιστοποιείται μέσω διαδικασίας ελάχιστων τετραγώνων
 - ▶ *Ανάστροφης Μετάδοσης Λάθους*: ο μετασχηματισμός των βαρών εξαρτάται από το μέγεθος της συνεισφοράς τους στο συνολικό λάθος
 - ▶ *Ανταγωνιστικής Μάθησης*: οι τεχνητοί νευρώνες συναγωνίζονται μεταξύ τους. Ο νευρώνας με τη μεγαλύτερη απόκριση (στην τρέχουσα είσοδο) τροποποιεί τα βάρη του
 - ▶ *Τυχαίας Μάθησης*: ο μετασχηματισμός των βαρών γίνεται με τυχαίο τρόπο. Αν η έξοδος βελτιώνεται (ανίχνευση μέσω κριτηρίων διδόμενων από το χρήστη), τότε ο μετασχηματισμός υιοθετείται

Ζητούμενο Μάθησης

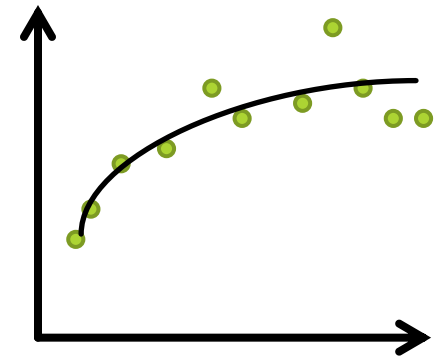
- ▶ Καλή προσαρμογή ως προς το σύνολο δεδομένων εκπαίδευσης



Υπο-προσαρμογή
(under-fitting)



Υπερ-προσαρμογή
(over-fitting)



Καλή προσαρμογή

Επιπλέον Χαρακτηριστικά Μάθησης

- ▶ Η εκπαίδευση μπορεί να γίνει σε κύκλους με ονομασία εποχές (epochs)
 - ▶ Μπορεί να πραγματοποιηθεί μάθηση δέσμης, αυξητική μάθηση ή συνδυασμός αυτών
- ▶ Τερματισμός: κριτήριο ελέγχου ποιότητας ΤΝΔ μεγαλύτερο από ένα κατώφλι
- ▶ Κριτήρια ελέγχου ποιότητας που μπορούν να χρησιμοποιηθούν:
 - ▶ Μέσο σφάλμα
 - ▶ Μεταβολή του μέσου σφάλματος

Πλεονεκτήματα των ΤΝΔ

- ▶ Ικανότητα οργάνωσης της πληροφορίας εισόδου σε διάφορες χρήσιμες μορφές / μοντέλα
- ▶ Ικανότητα μάθησης μέσω παραδειγμάτων
- ▶ Ικανότητα κατανεμημένης μνήμης και μνήμης συσχέτισης
 - ▶ Κατανεμημένη μνήμη: κωδικοποιημένη πληροφορία σε όλα τα βάρη της συνδεσμολογίας ενός ΤΝΔ
 - ▶ Μνήμη συσχέτισης: αποθηκεύει πληροφορία συσχετίζοντας αποθηκευμένα δεδομένα μεταξύ τους ενώ η ανάκληση πληροφορίας γίνεται με βάση το περιεχόμενο.

Προβληματισμοί σχεδιασμού

- Ποια συνάρτηση ενεργοποίησης πρέπει να χρησιμοποιηθεί;
- Πόσες εισόδους χρειάζεται το δίκτυο;
- Πόσα κρυφά στρώματα χρειάζεται το δίκτυο;
- Πόσοι κρυφοί νευρώνες ανά κρυφό επίπεδο;
- Πόσες εξόδους πρέπει να έχει το δίκτυο;

Δεν υπάρχει τυποποιημένη μεθοδολογία για τον προσδιορισμό αυτών των τιμών. Ακόμη και αν υπάρχουν κάποια ευρετικά σημεία, οι τελικές τιμές καθορίζονται με τη διαδικασία δοκιμής και σφάλματος.

Πλεονεκτήματα των ΤΝΔ

- ▶ Απλή μορφή ΤΝΔ
- ▶ Αποτελούνται από 1 επίπεδο εισόδου, 1 εξόδου και προαιρετικά 1 ή περισσότερα κρυφά επίπεδα
- ▶ Μάθηση: κυρίως επαγωγική
- ▶ Τοπολογία:
 - ▶ Πόσα κρυφά επίπεδα και πόσοι νευρώνες ανά επίπεδο; Ποια είναι η συνδεσμολογία;
 - ▶ Τα δεδομένα εισόδου-εξόδου βοηθούν στον υπολογισμό του αριθμού των νευρώνων ανά επίπεδο

Πλεονεκτήματα των ΤΝΔ

- ▶ Μεγάλη ανοχή σε σφάλματα
 - ▶ Τα ΤΝΔ είναι ανεκτικά σε μικροαλλαγές εισόδου
 - ▶ Αλλά και σε δομικά σφάλματα (πχ. καταστροφή νευρώνων ή συνδέσεων) λόγω της διάχυσης της πληροφορίας.
 - ▶ Μέγεθος σφάλματος λόγω δομικών σφαλμάτων είναι ανάλογο του ποσοστού των κατεστραμμένων στοιχείων
 - ▶ ΤΝΔ υλοποιημένα σε κύκλωμα ιδανικά για χρήση σε αυτοματισμούς ανθεκτικούς σε αντίξοες συνθήκες (πχ. χώροι με ραδιενέργεια)
- ▶ Ικανότητα αναγνώρισης προτύπων

Perceptron

- ▶ Πρώτη προσέγγιση ΤΝΔ
- ▶ Αποτελείται από ένα νευρώνα και χρησιμοποιεί την βηματική συνάρτηση.
- ▶ Υιοθετεί επαγωγική μάθηση για εκπαίδευση.
- ▶ Αλγόριθμος μεταβολής βαρών:
Επανάλαβε μέχρι να ικανοποιηθεί η συνθήκη τερματισμού
Για κάθε ζευγάρι διανύσματος εισόδου x με επιθυμητή έξοδο t
 1. Υπολόγισε το y
 2. Αν $y=t$, τότε μην κάνεις αλλαγή
 3. Αν $y \neq t$, τότε τροποποίησε τα βάρη των ενεργών γραμμών (αυτών με σήμα $\neq 0$) κατά $\Delta w = d^*(t - y) * x$ ώστε το y να πλησιάζει το t
- ▶ Η παράμετρος d είναι ο ρυθμός μάθησης

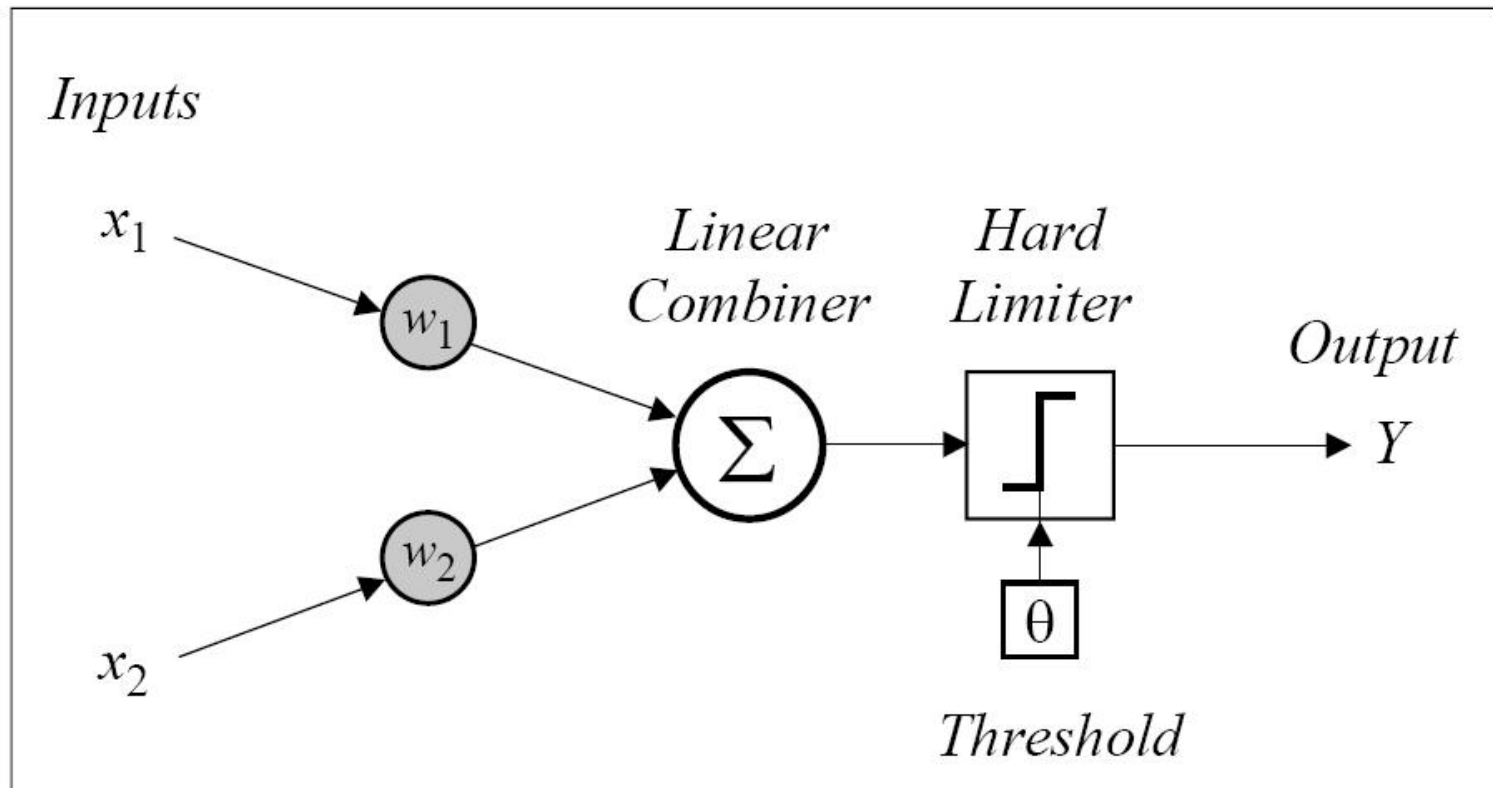
Perceptron

Can a single neuron learn a task?

- In 1958, **Frank Rosenblatt** introduced a training algorithm that provided the first procedure for training a simple ANN: a **perceptron**.
- The perceptron is the simplest form of a neural network. It consists of a single neuron with *adjustable* synaptic weights and a *hard limiter*.

Perceptron

Single-layer two-input perceptron



Perceptron

The Perceptron

- The operation of Rosenblatt's perceptron is based on the **McCulloch and Pitts neuron model**. The model consists of a linear combiner followed by a hard limiter.
- The weighted sum of the inputs is applied to the hard limiter, which produces an output equal to $+1$ if its input is positive and -1 if it is negative.

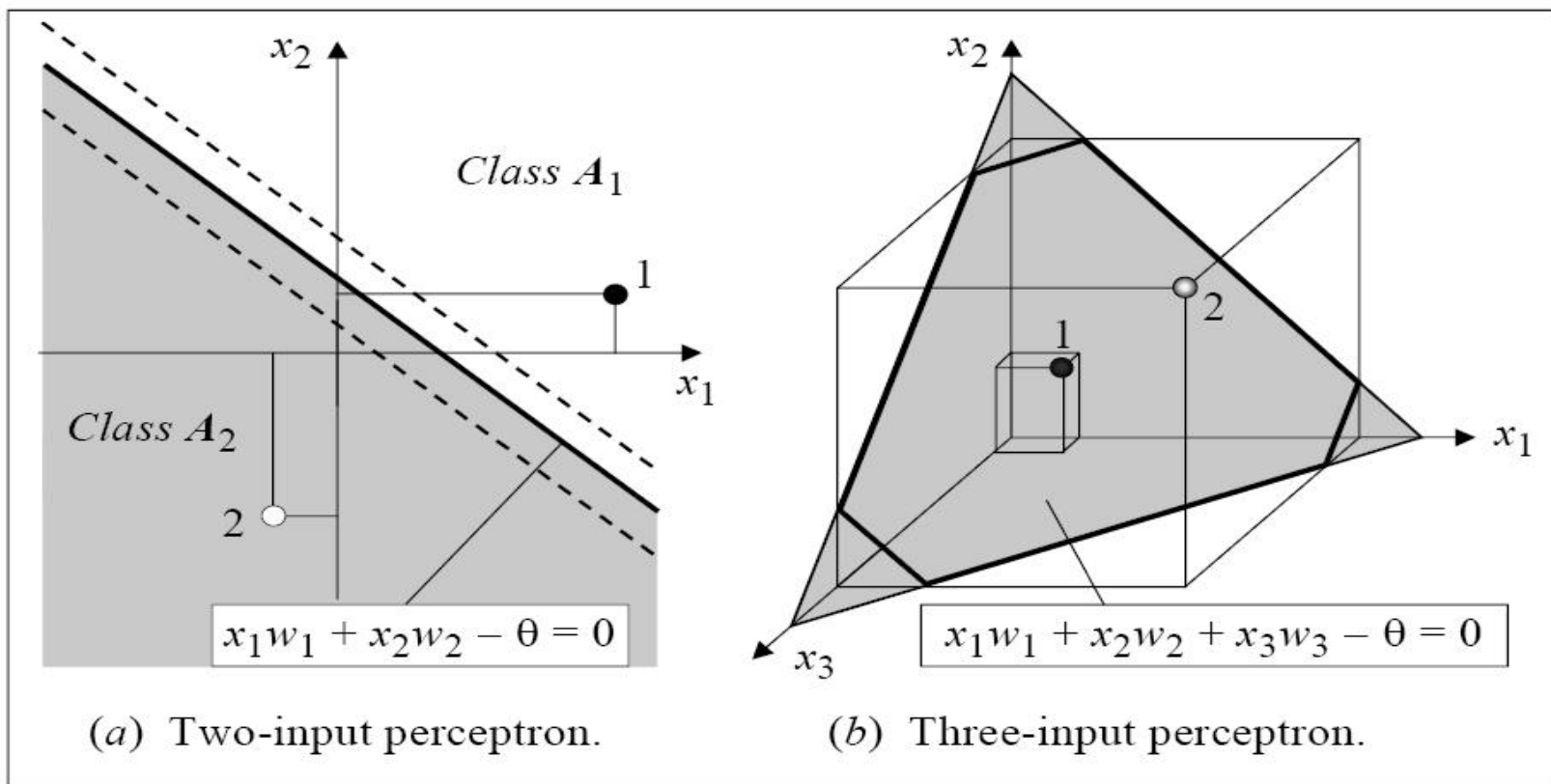
Perceptron

- The aim of the perceptron is to classify inputs, x_1, x_2, \dots, x_n , into one of two classes, say A_1 and A_2 .
- In the case of an elementary perceptron, the n -dimensional space is divided by a *hyperplane* into two decision regions. The hyperplane is defined by the *linearly separable function*:

$$\sum_{i=1}^n x_i w_i - \theta = 0$$

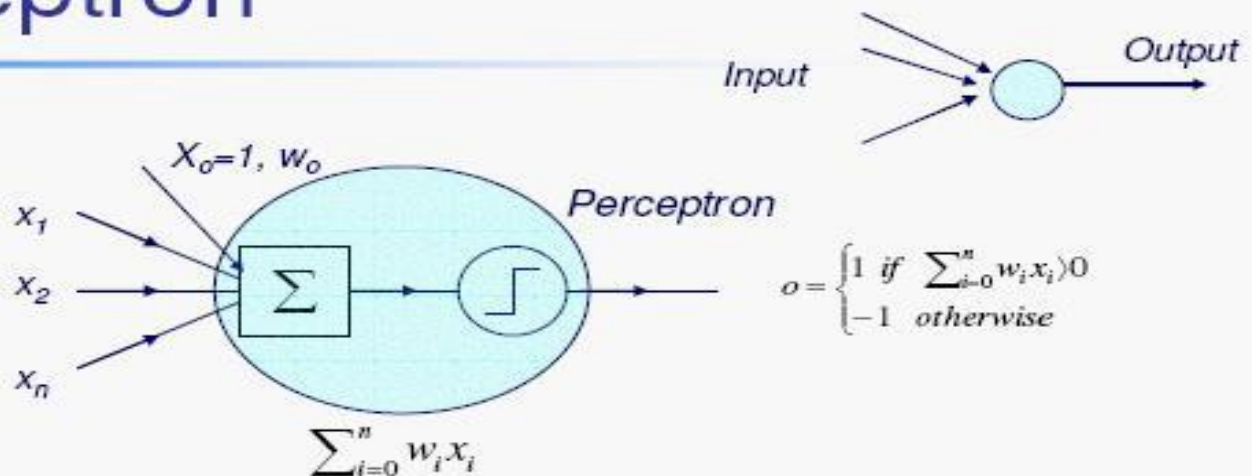
Perceptron

Linear separability in the perceptrons



Perceptron

Perceptron



The McCulloch-Pitts model

The perceptron calculates a weighted sum of inputs and compares it to a threshold. If the sum is higher than the threshold, the output is set to 1, otherwise to -1.

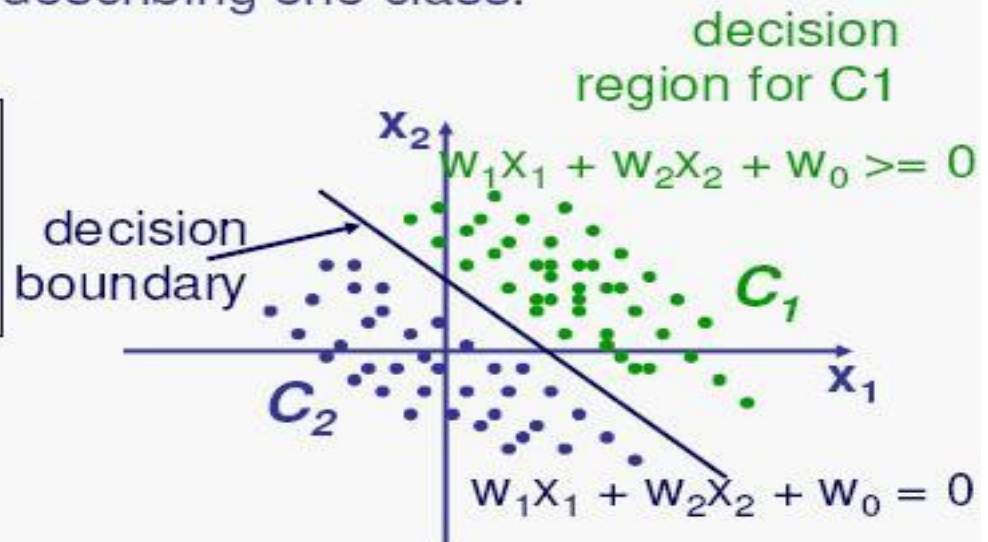
Learning is finding weights w_i

Perceptron

Perceptron

The equation below describes a (hyper-)plane in the input space consisting of real valued m -dimensional vectors. The plane splits the input space into two regions, each of them describing one class.

$$\sum_{i=1}^m w_i X_i + w_0 = 0$$



Perceptron

Perceptron Learning

*We have either (-1) or (+) as the output and inputs are either 0 or 1
There are 4 cases*

- The output is suppose to be +1 and perceptron returns +1*
- The output is suppose to be -1 and perceptron returns -1*
- The output is suppose to be +1 and perceptron returns -1*
- The output is suppose to be -1 and perceptron returns +1*

If Case 1 or 2, do nothing since the perceptron returns right result

*If Case 3 $w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0$ we need to increase the weights so that
the left side of the equation will become greater than 0*

If Case 4, the weights must be decreased

So we can use following update rule that satisfies this

$$w_i \leftarrow w_i + \Delta w_i$$
$$\Delta w_i = \eta(t - o)x_i$$

t is the target output, o is the output generated by the perceptron and η is a positive constant known as the learning rate.

Perceptron

How does the perceptron learn its classification tasks?

This is done by making small adjustments in the weights to reduce the difference between the actual and desired outputs of the perceptron. The initial weights are randomly assigned, usually in the range $[-0.5, 0.5]$, and then updated to obtain the output consistent with the training examples.

Perceptron's training algorithm

Step 1: Initialisation

Set initial weights w_1, w_2, \dots, w_n and threshold θ to random numbers in the range $[-0.5, 0.5]$.

If the error, $e(p)$, is positive, we need to increase perceptron output $Y(p)$, but if it is negative, we need to decrease $Y(p)$.

Perceptron's training algorithm

Step 2: Activation

Activate the perceptron by applying inputs $x_1(p)$, $x_2(p), \dots, x_n(p)$ and desired output $Y_d(p)$.

Calculate the actual output at iteration $p = 1$

$$Y(p) = \text{step} \left[\sum_{i=1}^n x_i(p) w_i(p) - \theta \right]$$

where n is the number of the perceptron inputs, and step is a step activation function.

Perceptron's training algorithm

Step 3: Weight training

Update the weights of the perceptron

$$w_i(p+1) = w_i(p) + \Delta w_i(p)$$

where Δw_i is the weight correction at iteration p .

The weight correction is computed by the **delta rule**:

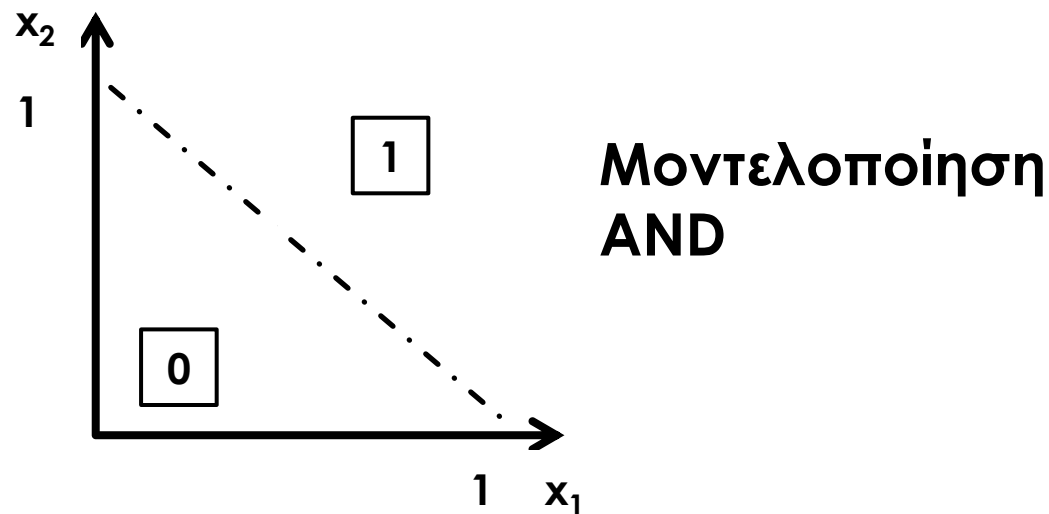
$$\Delta w_i(p) = \alpha \cdot x_i(p) \cdot e(p)$$

Step 4: Iteration

Increase iteration p by one, go back to *Step 2* and repeat the process until convergence.

Χαρακτηριστικά Perceptron

- ▶ Αντιμετωπίζει προβλήματα γραμμικώς διαχωρίσιμα (πχ. Στοιχειώδης λογικές πύλες)



- ▶ Αλλά όχι διαφορετικά προβλήματα διότι τότε απαιτείται ΤΝΔ με κρυφά επίπεδα (πχ. περίπτωση λογικής πύλης XOR)

Χαρακτηριστικά Perceptron

Example of perceptron learning: the logical operation *AND*

Epoch	Inputs		Desired output Y_d	Initial weights		Actual output Y	Error e	Final weights	
	x_1	x_2		w_1	w_2			w_1	w_2
1	0	0	0	0.3	-0.1	0	0	0.3	-0.1
	0	1	0	0.3	-0.1	0	0	0.3	-0.1
	1	0	0	0.3	-0.1	1	-1	0.2	-0.1
	1	1	1	0.2	-0.1	0	1	0.3	0.0
2	0	0	0	0.3	0.0	0	0	0.3	0.0
	0	1	0	0.3	0.0	0	0	0.3	0.0
	1	0	0	0.3	0.0	1	-1	0.2	0.0
	1	1	1	0.2	0.0	1	0	0.2	0.0
3	0	0	0	0.2	0.0	0	0	0.2	0.0
	0	1	0	0.2	0.0	0	0	0.2	0.0
	1	0	0	0.2	0.0	1	-1	0.1	0.0
	1	1	1	0.1	0.0	0	1	0.2	0.1
4	0	0	0	0.2	0.1	0	0	0.2	0.1
	0	1	0	0.2	0.1	0	0	0.2	0.1
	1	0	0	0.2	0.1	1	-1	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1
5	0	0	0	0.1	0.1	0	0	0.1	0.1
	0	1	0	0.1	0.1	0	0	0.1	0.1
	1	0	0	0.1	0.1	0	0	0.1	0.1
	1	1	1	0.1	0.1	1	0	0.1	0.1

Threshold: $\theta = 0.2$; learning rate: $\alpha = 0.1$

Χαρακτηριστικά Perceptron

Finding Weights Analytically for the AND Network

We have two weights w_1 and w_2 and the threshold θ , and for each training pattern we need to satisfy

$$out = \text{sgn}(w_1 in_1 + w_2 in_2 - \theta)$$

So the training data lead to four inequalities:

in_1	in_2	out
0	0	0
0	1	0
1	0	0
1	1	1

 \Rightarrow

$w_1 0 + w_2 0 - \theta < 0$
$w_1 0 + w_2 1 - \theta < 0$
$w_1 1 + w_2 0 - \theta < 0$
$w_1 1 + w_2 1 - \theta \geq 0$

 \Rightarrow

$\theta > 0$
$w_2 < \theta$
$w_1 < \theta$
$w_1 + w_2 \geq \theta$

It is easy to see that there are an infinite number of solutions. Similarly, there are an infinite number of solutions for the NOT and OR networks.

Χαρακτηριστικά Perceptron

Convergence of Perceptron Learning

The weight changes Δw_{ij} need to be applied repeatedly – for each weight w_{ij} in the network, and for each training pattern in the training set. One pass through all the weights for the whole training set is called one *epoch* of training.

Eventually, usually after many epochs, when all the network outputs match the targets for all the training patterns, all the Δw_{ij} will be zero and the process of training will cease. We then say that the training process has *converged* to a solution.

It can be shown that if there does exist a possible set of weights for a Perceptron which solves the given problem correctly, then the Perceptron Learning Rule will find them in a finite number of iterations

Moreover, it can be shown that if a problem is linearly separable, then the Perceptron Learning Rule will find a set of weights in a finite number of iterations that solves the problem correctly.

Χαρακτηριστικά Perceptron

Presentation of Training Examples

- Presenting all training examples once to the ANN is called an *epoch*.
- In incremental stochastic gradient descent training examples can be presented in
 - Fixed order (1,2,3...,M)
 - Randomly permuted order (5,2,7,....,3)
 - Completely random (4,1,7,1,5,4,.....)

Χαρακτηριστικά Perceptron

Limitations of the Perceptron

- Only binary input-output values
- Only two layers
- Separates the space linearly

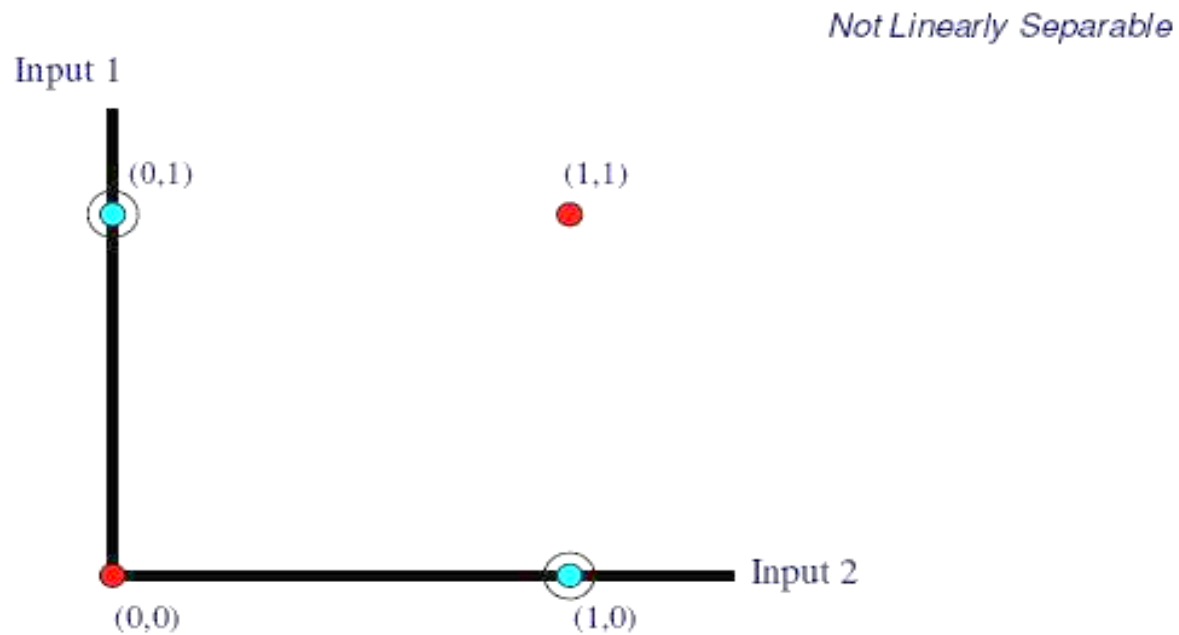
Χαρακτηριστικά Perceptron

Only two layers

- Minsky and Papert (1969) showed that a two-layer Perceptron cannot represent certain logical functions
- Some of these are very fundamental, in particular the exclusive or (XOR)
- Do you want coffee XOR tea?

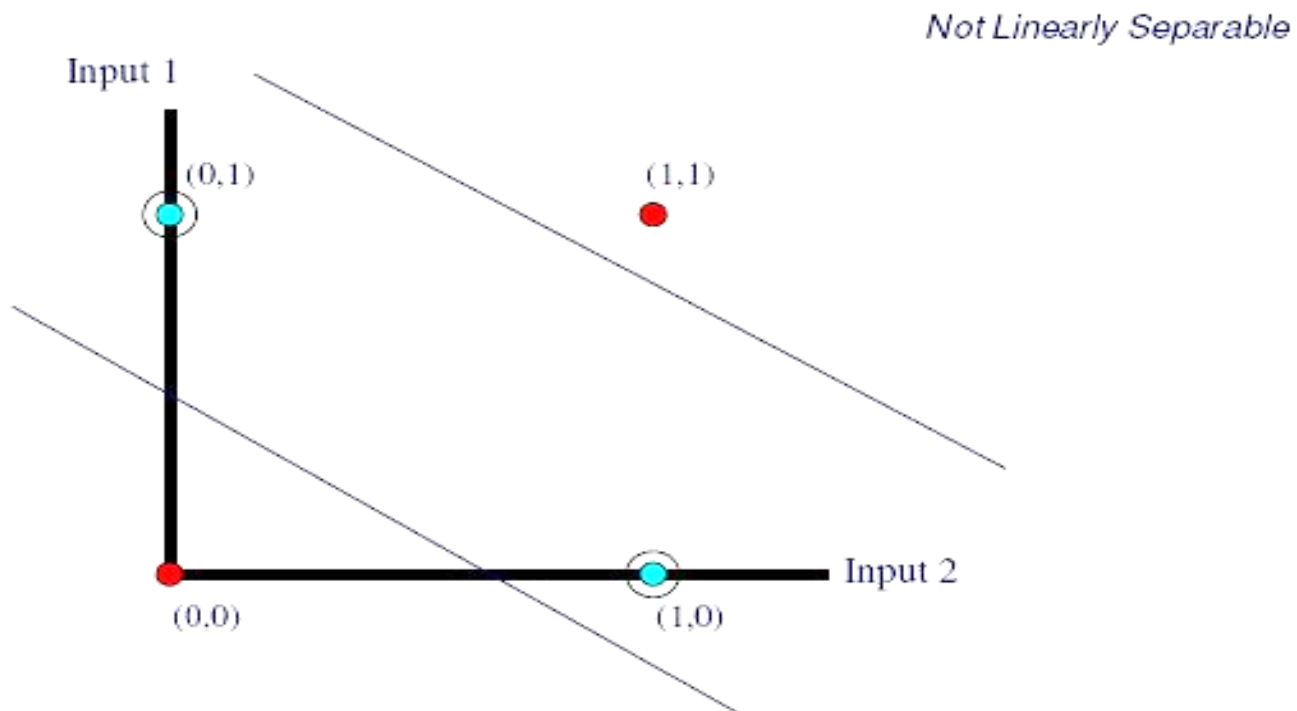
Χαρακτηριστικά Perceptron

Learning XOR



Χαρακτηριστικά Perceptron

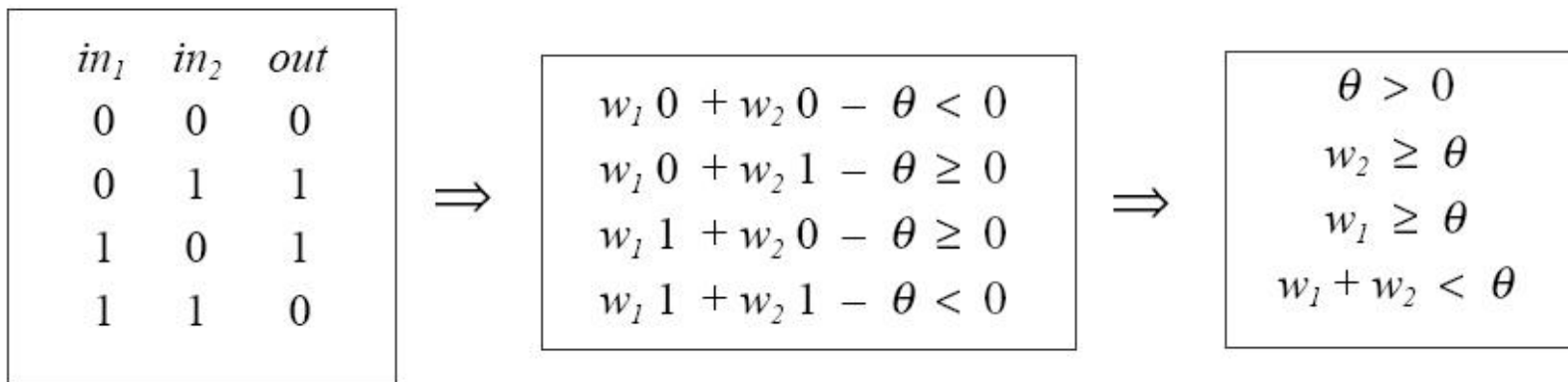
Learning XOR



Perceptron

Limitations of Simple Perceptrons

We can follow the same procedure for the XOR network:



Clearly the second and third inequalities are incompatible with the fourth, so there is in fact no solution. We need more complex networks, e.g. that combine together many simple networks, or use different activation/thresholding/transfer functions.

Perceptron

Solution to Linear Inseparability

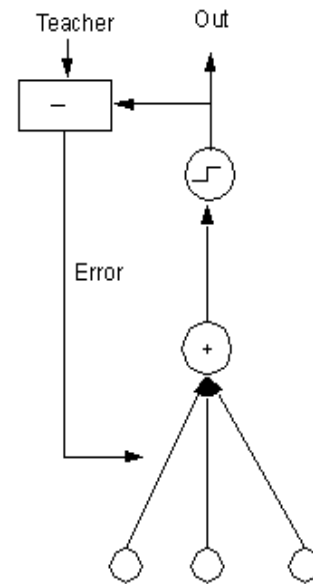
- *Use another training rule (delta rule)*
- *Backpropagation*

Perceptron – Delta Rule

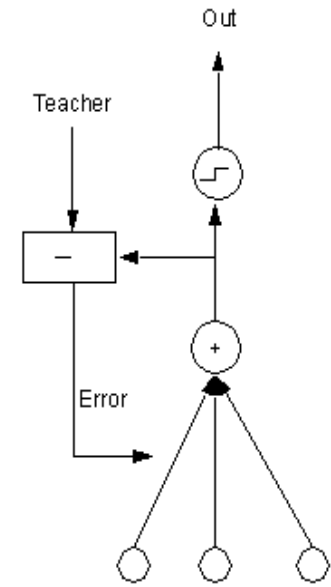
Delta Rule

- Adaline,
- Least Means Square

Change from Perceptron



Perceptron Learning



Delta Rule

Αλγόριθμος Κανόνα Δέλτα

- ▶ Αποτελεί γενίκευση του αλγόριθμου εκπαίδευσης του perceptron
- ▶ Δεν μπορεί να εφαρμοστεί σε ΤΝΔ με κρυφά επίπεδα
- ▶ Ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα:
όπου p είναι τα διανύσματα εισόδου και t_k η επιθυμητή έξοδος για το διάνυσμα εισόδου k

$$E = \frac{1}{p} \sum_{k=1}^p \left(t_k - \Phi \left(\sum_{i=1}^n w_i g x_i \right) \right)^2$$

Αλγόριθμος Κανόνα Δέλτα

- ▶ Η μεταβολή στην τιμή βάρους με βάση το τρέχων διάνυσμα εκπαίδευσης είναι η εξής:

$$\Delta_{w_i} = w_{i(new)} - w_{i(old)} = d \left(t - \Phi \left(\sum_{l=1}^n w_l x_l \right) \right) x_i$$

- ▶ Λογική: ακολουθούμε την αρνητική κλίση της επιφάνειας σφάλματος με κατεύθυνση προς το ελάχιστό της

$$\Delta_{w_i} \propto - \frac{\partial E}{\partial w_i}$$

Gradient Descent

Gradient Descent

- Training error of a hypothesis:

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

*D is the set of training examples,
T_d is the target output for training example d,
and o_d is the output of the linear unit for training example d.*

Gradient Descent

How to find Δw ?

Derivation of the Gradient Descent Rule

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right] \quad \text{Direction of the steepest descent along the error surface}$$

$$\vec{w} \leftarrow \vec{w} + \Delta \vec{w} \quad \text{where} \quad \Delta \vec{w} = -\eta \nabla E(\vec{w})$$

The negative sign is present as we want to go in the direction that decreases E .

For the i^{th} component:

$$w_i \leftarrow w_i + \Delta w_i \quad \text{where} \quad \Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

Gradient Descent

How to find Δw ?

$$\frac{\partial E}{\partial w_i} = \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 = \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2$$

$$\frac{\partial E}{\partial w_i} = \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial (t_d - o_d)}{\partial w_i} = \sum_{d \in D} (t_d - o_d) \frac{\partial (t_d - \bar{w} \cdot \bar{x}_d)}{\partial w_i}$$

$$\frac{\partial E}{\partial w_i} = \sum_{d \in D} (t_d - o_d) (-x_{id})$$

Where x_{id} is the single input component x_i for training example d

Hence
$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id}$$

Gradient Descent

Gradient-Descent Algorithm

- Each training example is a pair of the form $\langle \vec{x}, t \rangle$ where
 - \vec{x} is the vector of input values, and t is the target output value and η is the learning rate (e.g. 0.5)
- Initialize each w_i to some small random value
- Until the termination condition is met, Do
 - Initialize each Δw_i to zero.
 - For each $\langle \vec{x}, t \rangle$ in training examples, Do
 - Input the instance \vec{x} to the unit and compute the output o
 - For each linear unit weight w_i , Do
$$\Delta w_i \leftarrow \Delta w_i + \eta(t - o)x_i$$
 - For each linear unit weight w_i , Do
$$w_i \leftarrow w_i + \Delta w_i$$

Training Strategies

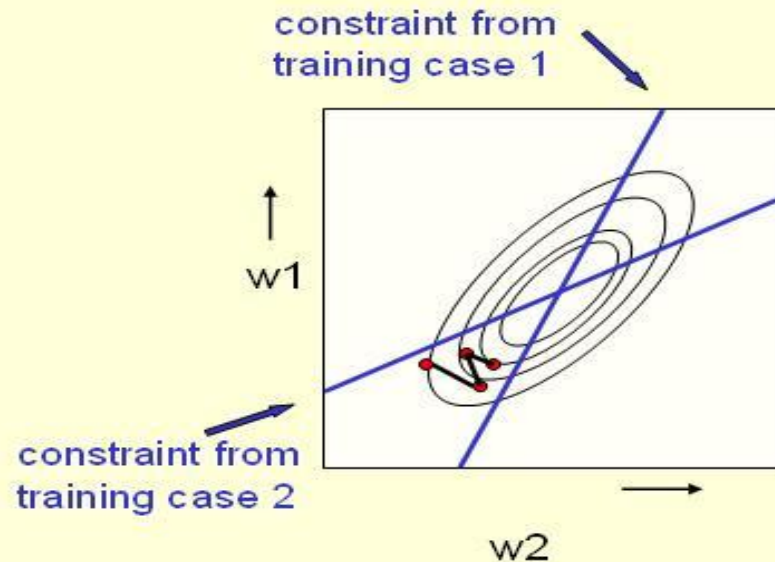
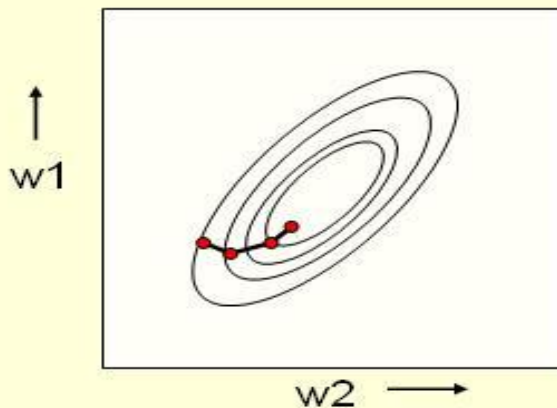
Training Strategies

- Online training:
 - Update weights after each sample
- Offline (batch training):
 - Compute error over all samples
 - Then update weights
- Online training “noisy”
 - Sensitive to individual instances
 - However, may escape local minima

Αλγόριθμος Κανόνα Δέλτα

Online versus batch learning

- Batch learning does steepest descent on the error surface
- Online learning zig-zags around the direction of steepest descent



Μηχανική Μάθηση & Εξόρυξη Γνώσης

Ερωτήσεις
?

Βιβλιογραφία

- ▶ Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, Τεχνητή Νοημοσύνη - Γ' Έκδοση, ISBN: 978-960-8396-64-7, Έκδοση/Διάθεση: Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011
- ▶ Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- ▶ Κ. Διαμαντάρας, Ι. Μπότσης, Μηχανική Μάθηση – Α' Έκδοση, ISBN: 978-960-461-955-5, Εκδόσεις Κλειδάριθμος, 2019
- ▶ P.-N. Tan, M. Steinbach, V. Kumar, «Introduction to Data Mining», Addison Wesley, 2006