Νευρωνικά Δίκτυα

Ρεφανίδης Γιάννης

Οκτώβριος 2011

Γενικά

- Σελίδα μαθήματος:
  - http://users.uom.gr/~yrefanid/Courses/NeuralNetworks/
- Συγγράμματα:
  - Πανεπιστημιακές παραδόσεις για Νευρωνικά Δίκτυα και Εξελικτικούς Αλγορίθμους, Γιάννης Ρεφανίδης.
  - Τεχνητά Νευρωνικά Δίκτυα. Κωνσταντίνος Διαμαντάρας. Εκδόσεις Κλειδάριθμος, 2007. Υπάρχουν τρία αντίτυπα στη βιβλιοθήκη του ΠαΜακ, με call no. QA76.87.D536 2007
- Εργασίες θα δοθούν κατά τη διάρκεια του εξαμήνου.

Αντικείμενο μαθήματος

- Μη-συμβολική Τεχνητή Νοημοσύνη
- Μηχανική μάθηση
- Ειδικότερα:
  - Νευρωνικά δίκτυα
  - Γενετικοί αλγόριθμοι
  - Εναλλακτικοί αλγόριθμοι μηχανικής μάθησης

Εφαρμογές Νευρωνικών Δικτύων (1/3)

- Αεροδιαστημική
  - Αυτόματος πιλότος
  - Έλεγχος αεροσκάφους
  - Ανίχνευση βλαβών
- Αυτοκίνητα
  - Αυτόματη οδήγηση
- Τραπεζικές εργασίες
  - Αξιολόγηση αιτήσεων για δάνειο
- Στρατιωτικές εφαρμογές
  - Καθοδήγηση όπλων
  - Αναγνώριση αντικειμένων (αντικείμενα, πρόσωπα κλπ)

Εφαρμογές Νευρωνικών Δικτύων (2/3)

- Ηλεκτρονική
  - Τεχνητή όραση
  - Σύνθεση ήχου
  - Διάγνωση βλαβών
- Οικονομικά
  - Αποτίμηση ακίνητης περιουσίας
  - Αξιολόγηση ομολογιακών δανείων επιχειρήσεων
  - Διαχείριση χαρτοφυλακίων
  - Πρόβλεψη συναλλαγματικών μεταβολών
- Ασφάλιση
  - Αξιολόγηση αιτήσεων
- Κατασκευές
  - Έλεγχος μονάδων, διάγνωση βλαβών
  - Σχεδιασμός και ανάλυση προϊόντων
  - Οπτικά συστήματα ελέγχου ποιότητας

Εφαρμογές Νευρωνικών Δικτύων (3/3)

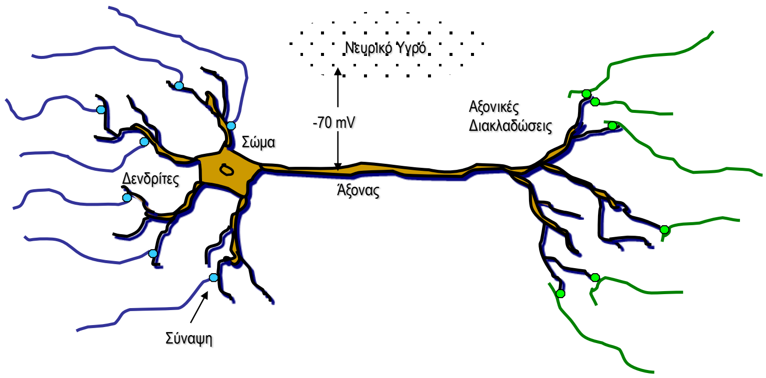
- Ιατρική
  - Ανάλυση συμπτωμάτων και διάγνωση για διάφορες ασθένειες
- Ορυκτός πλούτος
  - Αξιολόγηση περιοχών για γεώτρηση
- Ρομποτική
  - Έλεγχος κίνησης
  - Όραση
- Ομιλία
  - Αναγνώριση ομιλίας
  - Εκφορά ομιλίας
- Τηλεπικοινωνίες
  - Συμπίεση δεδομένων
  - Μετάφραση γλώσσας σε πραγματικό χρόνο
- Μεταφορές
  - Εύρεση δρομολογίων

Παρατήρηση

- Οι παραπάνω εφαρμογές απαιτούν σε μικρότερο ή μεγαλύτερο βαθμό τη διαδικασία της μηχανικής μάθησης.
- Τα νευρωνικά δίκτυα δεν αποτελούν το μοναδικό τρόπο αντιμετώπισης των παραπάνω εφαρμογών.
- Ωστόσο η χρήση νευρωνικών δικτύων προσφέρει:
  - Ευκολία υλοποίησης
  - Σχετικά αξιόπιστη λειτουργία
  - Ακαριαία απόκριση κατά την φάση πραγματικής λειτουργίας, εφόσον το νευρωνικό δίκτυο υλοποιηθεί σε hardware.

Βιολογικά Νευρωνικά Δίκτυα

Ο Φυσικός Νευρώνας (1/2)



Είσοδος
Αξονικές διακλαδώσεις άλλων νευρώνων

Έξοδος
Δενδρίτες άλλων νευρώνων

Ο Φυσικός Νευρώνας (2/2)

- **Δενδρίτες**: Σημεία εισόδου ηλεκτρικών σημάτων.
- **Άξονας**: Έξοδος ηλεκτρικών σημάτων.
- Οι δενδρίτες κάθε νευρώνα συνδέονται με τους άξονες άλλων νευρώνων.
- Τα σημεία σύνδεσης ονομάζονται **συνάψεις**.
- Η ταχύτητα μεταφοράς των ηλεκτρικών παλμών είναι από 10 μέχρι 100 m/sec.
- Οι νευρώνες καταναλώνουν την περισσότερη ενέργεια από όλα τα κύτταρα του οργανισμού.
- Υπολογίζεται ότι ο ανθρώπινος εγκέφαλος καταναλώνει 20 Watt ισχύος.

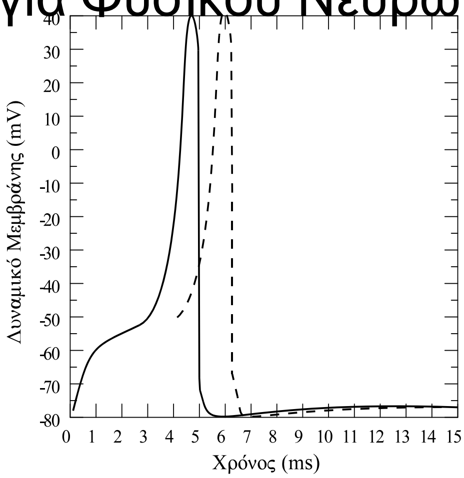
Ανθρώπινο Κεντρικό Νευρικό Σύστημα

- Ο ανθρώπινος εγκέφαλος έχει περίπου 10^{10} νευρώνες (10^9-10^{11}).
- Κάθε νευρώνας έχει περίπου 10^4 συνάψεις (οι οποίες όμως σε κάποιες περιπτώσεις φτάνουν και τις $2 \cdot 10^5$!)
- Ένας αριθμός νευρώνων, μαζί με τις συνδέσεις του, αποτελεί ένα νευρωνικό δίκτυο (Κεντρικό Νευρικό Σύστημα).
- Οι νευρώνες πιστεύεται ότι δεν πολλαπλασιάζονται.
- Ο ανθρώπινος εγκέφαλος ενός υγιούς ενήλικα χάνει περίπου 1000 νευρώνες την ημέρα.
  - Περισσότεροι για όσους πίνουν, καπνίζουν κλπ.
- Οι συνάψεις συνεχώς μεταβάλλονται.

Λειτουργία Φυσικού Νευρώνα (1/2)

- Κάθε νευρώνας έχει δύο <u>μόνο</u> δυνατές καταστάσεις: **Ενεργός** και **Μη-ενεργός**
- Όταν ο νευρώνας είναι μη-ενεργός, υπάρχει μια διαφορά δυναμικού περίπου -70mV μεταξύ της εξωτερικής επιφάνειάς του και του εσωτερικού του.
- Τα σήματα εισόδου στον νευρώνα αθροίζονται.
- Όταν τα σήματα εισόδου ξεπεράσουν ένα όριο (κατώφλι θ), ο νευρώνας γίνεται στιγμιαία ενεργός, παράγει έναν ηλεκτρικό παλμό, με τα ίδια πάντα χαρακτηριστικά, ο οποίος μέσω του άξονα μεταφέρεται σε άλλους νευρώνες.
- Μέγιστο 1000 παλμοί/sec ανά νευρώνα.

Λειτουργία Φυσικού Νευρώνα (2/2)



Το δυναμικό ενός νευρώνα κατά τη διάρκεια ενεργοποίησής του.

Σύγκριση Φυσικών Νευρωνικών Δικτύων και Υπολογιστή

Φυσικά Νευρωνικά Δίκτυα	**Υπολογιστής**
Ασύγχρονος τρόπος λειτουργίας	Σύγχρονος τρόπος λειτουργίας
Μαζική παράλληλη επεξεργασία	Σειριακή επεξεργασία
Εκπαιδεύονται με παραδείγματα	Προγραμματίζεται
Ανοχή στα σφάλματα	Συνήθως καμία ανόχη σε σφάλματα
Ανοχή σε βλάβες	Καμία ανοχή σε βλάβες
Χρόνος κύκλου της τάξης του msec	Χρόνος κύκλου της τάξης του nsec.
Συνύπαρξη μνήμης και μονάδων επεξεργασίας	Διαχωρισμός μνήμης και μονάδων επεξεργασίας

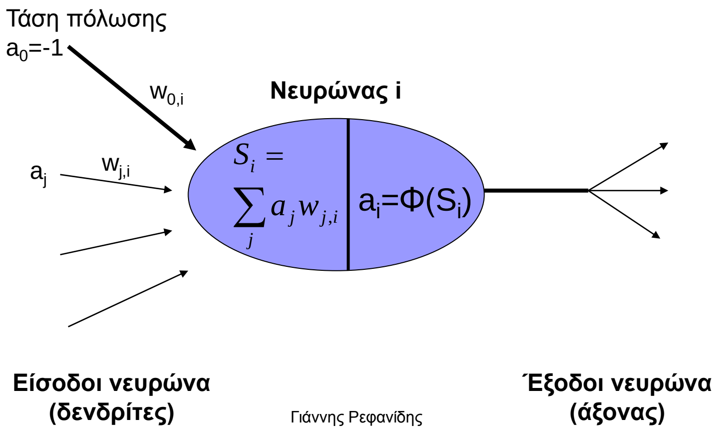
Ιστορική αναδρομή

- 1943: Το πρώτο μοντέλο νευρωνικού δικτύου (McCulloch & Pitts)
- 1949: Μοντέλο μάθησης του Hebb
  - Κάθε φορά που ενεργοποιείται μια σύναψη, αυτή ενισχύεται, με αποτέλεσμα το δίκτυο να μαθαίνει "λίγο περισσότερο" το πρότυπο που του παρουσιάζεται εκείνη τη στιγμή.
- 1957: Το μοντέλο του απλού αισθητήρα – perceptron (Rosenblatt).
- 1969: Οι Minsky & Papert απέδειξαν μαθηματικά ότι τα ΤΝΔ <u>ενός επιπέδου</u> δεν μπορούν να λύσουν συγκεκριμένα προβλήματα.
- 1982: Μαθηματική απόδειξη ότι ένα νευρωνικό δίκτυο <u>πολλών επιπέδων</u> μπορεί να αποθηκεύσει οποιαδήποτε πληροφορία.
- 1986: Μέθοδος οπισθοδιάδοσης για την εκπαίδευση ΤΝΔ (McClelland & Rumelhart).

Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα

Τεχνητός νευρώνας

- Το γενικό μοντέλο ενός νευρώνα:



Συνολική είσοδος τεχνητού νευρώνα

- a_i, a_j κλπ είναι οι έξοδοι των διαφόρων νευρώνων i, j κλπ, οι οποίοι γίνονται είσοδοι σε άλλους νευρώνες.
- Υπάρχει ένα "σήμα", το a_0, το οποίο έχει σταθερή τιμή (συνήθως -1 ή 1) και το οποίο αποτελεί είσοδο για όλους τους νευρώνες.
- Τα διάφορα σήματα a_j τα οποία αποτελούν είσοδο ενός νευρώνα i, πολλαπλασιάζονται με συντελεστές βαρύτητας $w_{j,i}$.
- Η συνολική είσοδος στον νευρώνα i είναι τελικά το άθροισμα όλων των επιμέρους εισόδων της, μετά τον πολλαπλασιασμό τους με τους συντελεστές βαρύτητας:

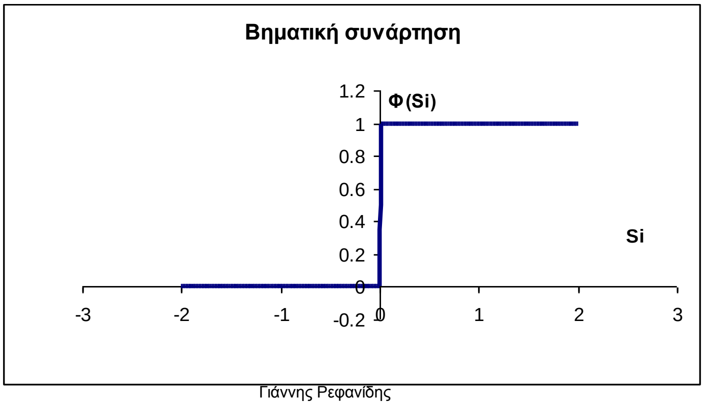
$$S_i = \sum_j a_j w_{j,i}$$

Έξοδος τεχνητού νευρώνα

- Η έξοδος του τεχνητού νευρώνα προκύπτει από την εφαρμογή της συνάρτησης ενεργοποίησης (activation function) στην συνολική του είσοδο S_i:
  - $a_i=\Phi(S_i)$
- Υπάρχουν διάφορες περιπτώσεις συναρτήσεων ενεργοποίησης:
  - Βηματική συνάρτηση (step function) ή συνάρτηση κατωφλίου (threshold function)
  - Συνάρτηση προσήμου (sign function)
  - Σιγμοειδής συνάρτηση (sigmoid ή logistics function)
  - Γραμμική συνάρτηση (linear function)
  - και άλλες...

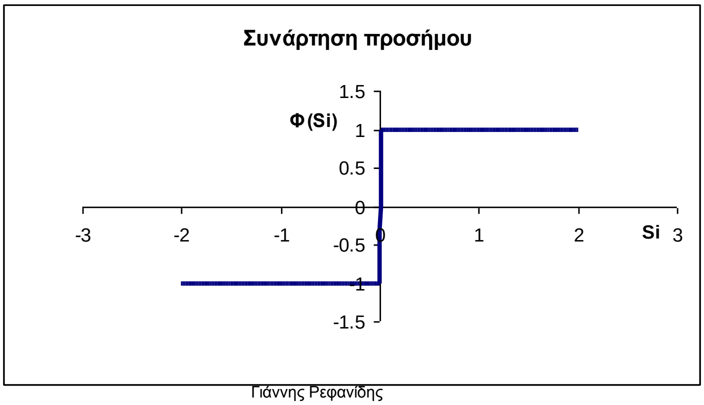
Βηματική Συνάρτηση Ενεργοποίησης

$$\Phi(S)=\begin{cases}1, \text{αν } S>0\\ 0, \text{αν } S\le 0\end{cases}$$



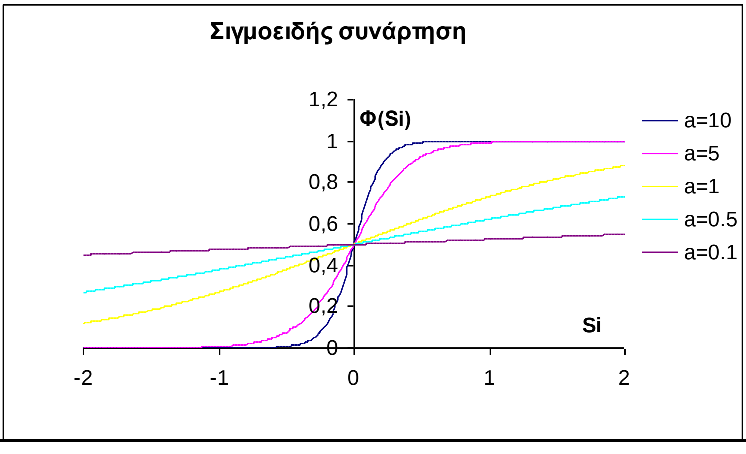
Συνάρτηση Προσήμου

$$\Phi(S) = \begin{cases} 1, \text{αν } S > 0 \\ -1, \text{αν } S \le 0 \end{cases}$$



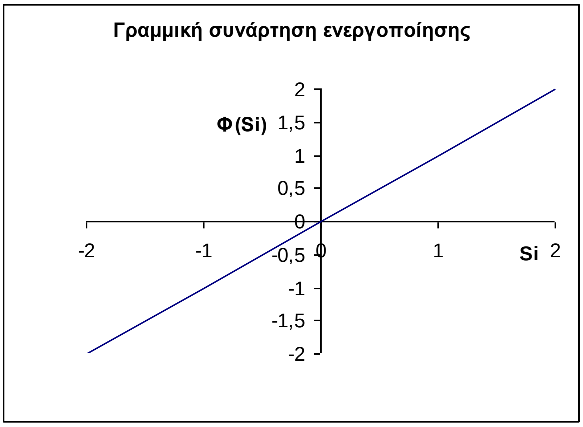
Σιγμοειδής συνάρτηση

$$\Phi(S_i) = \frac{1}{1 + e^{-a \cdot S_i}}$$



Γραμική συνάρτηση

$$\Phi(S_i) = \lambda \cdot S_i$$

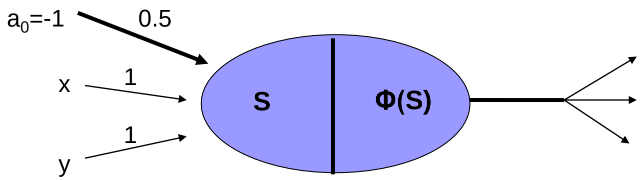


Παρατηρήσεις

- Οι τιμές που μπορούν να πάρουν τα σήματα εξόδου (άρα και τα σήματα εισόδου) a_i είναι πεπερασμένες:
  - Στους νευρώνες με βηματική συνάρτηση, η έξοδος μπορεί να είναι 0 ή 1.
  - Στους νευρώνες με συνάρτηση προσήμου, η έξοδος μπορεί να είναι -1 ή 1.
  - Στους νευρώνες με σιγμοειδή συνάρτηση, η έξοδος μπορεί να είναι οποιοσδήποτε πραγματικός αριθμός από το 0 έως το 1.
- Τα βάρη μπορούν να πάρουν οποιαδήποτε τιμή, ωστόσο πολλές φορές τα κανονικοποιούμε (π.χ. απαιτούμε το άθροισμα των βαρών στις εισόδους ενός νευρώνα να είναι ίσο με 1).
- Σε ένα νευρωνικό δίκτυο μπορεί να υπάρχουν νευρώνες διαφορετικού τύπου (δηλ. με διαφορετικές συναρτήσεις ενεργοποίησης).

Παράδειγμα: Συνάρτηση OR

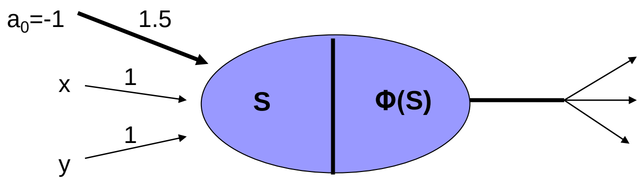
- Βηματική συνάρτηση ενεργοποίησης



x	y	S	Φ(S)
0	0	-0.5	0
0	1	0.5	1
1	0	0.5	1
1	1	1.5	1

Παράδειγμα: Συνάρτηση AND

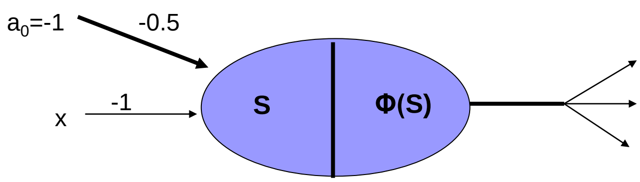
- Βηματική συνάρτηση ενεργοποίησης



x	y	S	Φ(S)
0	0	-1.5	0
0	1	-0.5	0
1	0	-0.5	0
1	1	0.5	1

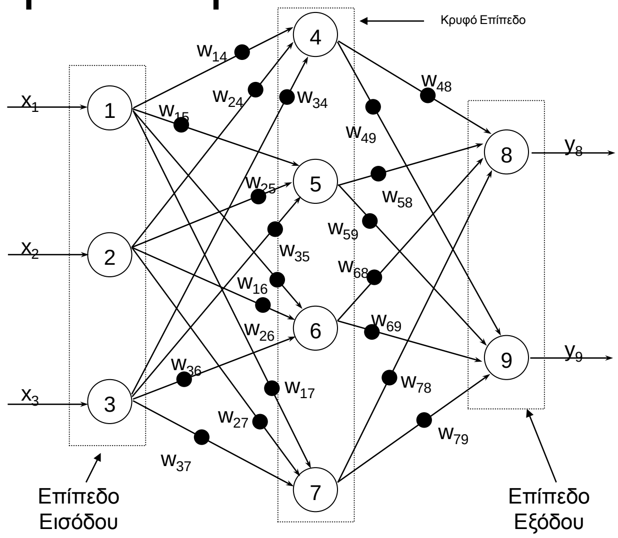
Παράδειγμα: Συνάρτηση NOT

- Βηματική συνάρτηση ενεργοποίησης



x	S	Φ(S)
0	0.5	1
1	-0.5	0

Τεχνητά Νευρωνικά Δίκτυα



Κατηγορίες Νευρωνικών Δικτύων

- Συνήθως οι νευρώνες είναι οργανωμένοι σε επίπεδα:
  - Επίπεδο εισόδου
  - Επίπεδο εξόδου
  - Κρυφά επίπεδα (κανένα, ένα ή και περισσότερα)
- Πλήρως συνδεδεμένα ΤΝΔ (fully connected): Κάθε νευρώνας συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου
- Μερικώς συνδεδεμένα ΤΝΔ (partially connected): Υπάρχουν νευρώνες που δεν συνδέονται με όλους τους νευρώνες του επόμενου επιπέδου.
- Δίκτυα με απλή τροφοδότηση (feedforward)
  - Δεν υπάρχουν συνδέσεις μεταξύ νευρώνων ενός επιπέδου και νευρώνων προηγούμενου επιπέδου.
- Δίκτυα με ανατροφοδότηση (feedback ή recurrent)
  - Υπάρχουν συνδέσεις μεταξύ νευρώνων ενός επιπέδου και νευρώνων προηγούμενου επιπέδου.

Είδη μάθησης

- Η μάθηση στα ΤΝΔ συνίσταται στην αλλαγή των βαρών των συνδέσεων μεταξύ των νευρώνων.
- Βασικές μέθοδοι μάθησης:
  - Μάθηση με επίβλεψη (supervised learning)
  - Μάθηση χωρίς επίβλεψη (unsupervised learning)

Μάθηση στον απλό τεχνητό νευρώνα

Perceptrons

- Τα Perceptrons είναι τα πιο απλά τεχνητά νευρωνικά δίκτυα.
- Αποτελούνται από ένα μόνο επίπεδο απλών νευρώνων, οι οποίοι λειτουργούν τόσο ως είσοδοι όσο και ως έξοδοι του δικτύου.
- Κάθε νευρώνας είναι ανεξάρτητος από τους υπόλοιπους, άρα και η μάθηση κάθε νευρώνα γίνεται ανεξάρτητα από τους υπόλοιπους νευρώνες.

Βασική ιδέα

- Εκτελούμε μάθηση με επίβλεψη, χρησιμοποιώντας ένα σύνολο από παραδείγματα εκπαίδευσης (training set).
- Για κάθε παράδειγμα, ελέγχουμε εάν η έξοδος που δίνει ο νευρώνας είναι η σωστή έξοδος.
  - Εάν είναι, προχωράμε στο επόμενο παράδειγμα.
  - Εάν όχι, τότε:
    - Εάν η σωστή έξοδος ήταν μεγαλύτερη από αυτήν που υπολόγισε ο νευρώνας, αυξάνουμε τα βάρη των εισόδων που ήταν θετικές και μειώνουμε τα βάρη των εισόδων που ήταν αρνητικές.
    - Εάν η σωστή έξοδος ήταν μικρότερη από αυτήν που υπολόγισε ο νευρώνας, μειώνουμε τα βάρη των εισόδων που ήταν θετικές και αυξάνουμε τα βάρη των εισόδων που ήταν αρνητικές.
  - Εκτελούμε τη διαδικασία αυτή, μέχρις ότου ο νευρώνας να απαντά σωστά σε όλα τα παραδείγματα, ή να μην βελτιώνει καθόλου την απόδοσή του.

Ο κανόνας Δέλτα (Delta rule) για βηματικές συναρτήσεις ενεργοποίησης

- $\Delta w_{j,i}=w_{j,i\text{-}new}-w_{j,i\text{-}old}=-d(a_i-o_i)a_j$
  - όπου:
    - a_i: Η τρέχουσα έξοδος του νευρώνα i
    - o_i: Η επιθυμητή έξοδος του νευρώνα i για το τρέχον παράδειγμα.
    - d: Ο ρυθμός μάθησης (d>0)
    - $w_{j,i\text{-}new}$: Το νέο βάρος εισόδου από τον νευρώνα j.
    - $w_{j,i\text{-}old}$: Το παλιό βάρος εισόδου από τον νευρώνα j.
  - Ο ρυθμός μάθησης d καθορίζει το πόσο γρήγορα συγκλίνει η μάθηση.
    - Μεγάλος ρυθμός μάθησης μπορεί να οδηγήσει σε γρηγορότερη σύγκλιση, αλλά και σε ταλάντωση γύρω από τις βέλτιστες τιμών βαρών.
    - Μικρός ρυθμός μάθησης έχουν ως αποτέλεσμα πιο αργή σύγκλιση, ενώ μπορεί να οδηγήσουν σε παγίδευση σε τοπικά ακρότατα.

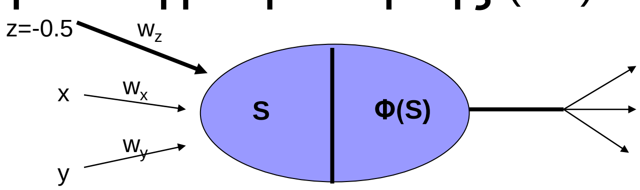
Παράδειγμα μάθησης (1/4)

- Έστω η εκπαίδευση ενός νευρώνα ώστε να λειτουργεί ως πύλη AND.
- Τα δεδομένα εκπαίδευσης είναι τα εξής:

x	**y**	**Επιθυμητή έξοδος**
0	0	0
0	1	0
1	0	0
1	1	1

- Έχουμε τρεις εισόδους, τις x, y και την σταθερή πόλωση (ίση με -0.5), έστω z.
- Αρχικοποιούμε λοιπόν τα τρία βάρη ως εξής: $w_x=w_y=w_z=0$
  - Προσοχή: Επειδή το δίκτυο μας έχει έναν μόνο νευρώνα, δεν χρησιμοποιούμε τον δείκτη i στα βάρη.

Παράδειγμα μάθησης (2/4)



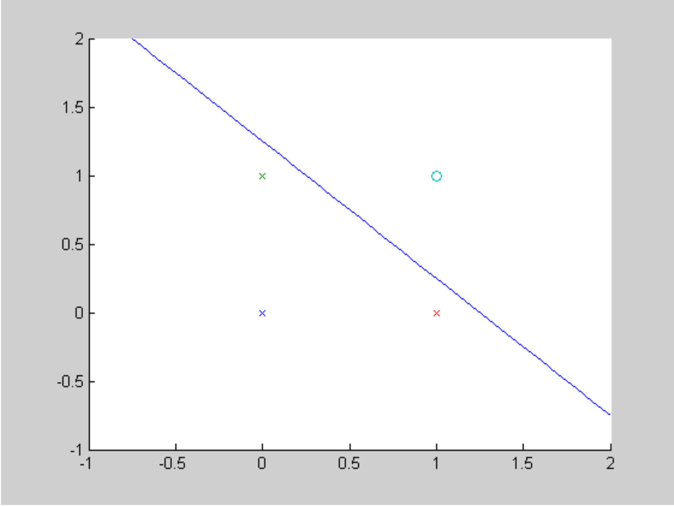
- Παρουσιάζουμε τα τέσσερα παραδείγματα διαδοχικά και κυκλικά στο δίκτυο, μέχρι αυτό να αρχίσει να απαντά σωστά σε όλα.
- Μετά από κάθε παρουσίαση, ενημερώνουμε τα τρία βάρη.
- Ο ρυθμός μάθησης τέθηκε στην τιμή d=0.1.
- Μετά από 28 επαναλήψεις τα βάρη πήραν τις τιμές:
  - $w_x=0.1$, $w_y=0.1$, $w_z=0.25$

Παράδειγμα μάθησης (3/4)

- Παρακάτω φαίνονται τα πρώτα βήματα του αλγορίθμου:

#	Παράδειγμα	Έξοδος	Επιθυμητή έξοδος	w_x	w_y	w_z
1	[0,0,-0.5]	0	0	0	0	0
2	[0,1,-0.5]	0	0	0	0	0
3	[1,0,-0.5]	0	0	0	0	0
4	[1,1,-0.5]	0	1	0.1	0.1	-0.05
5	[0,0,-0.5]	1	0	0.1	0.1	0
6	[0,1,-0.5]	1	0	0.1	0	0.05
7	[1,0,-0.5]	1	0	0	0	0.1
8	[1,1,-0.5]	0	1	0.1	0.1	0.05
…	…	…	…	…	…	…

Παράδειγμα μάθησης (4/4)



Ο κανόνας Δέλτα για συνεχείς συναρτήσεις ενεργοποίησης (1/2)

- Για συνεχείς συναρτήσεις ενεργοποίησης, θα θέλαμε να αλλάξουμε περισσότερο τα βάρη εκείνα, στις μεταβολές των οποίων το συνολικό σφάλμα είναι πιο «ευαίσθητο».
  - Προσπαθούμε να αποφύγουμε βάρη, μικρές μεταβολές των οποίων προκαλούν μεγάλες μεταβολές στο σφάλμα, γιατί αυτά δημιουργούν ένα ασταθές νευρωνικό δίκτυο.
- Έστω $E=Err^2=(a_i-o_i)^2$ το τετράγωνο του σφάλματος για έναν νευρώνα εξόδου και ένα συγκεκριμένο παράδειγμα.
- Ο ρυθμός αύξησης του σφάλματος E σε σχέση με ένα βάρος w_j δίνεται από τη σχέση:

$$\frac{\partial E}{\partial w_{ji}} = 2\cdot Err\frac{\partial Err}{\partial w_{ji}} = 2\cdot Err\frac{\partial}{\partial w_{ji}}\left[\Phi(\sum_j w_{ji}a_j)-o_i\right] = 2\cdot Err\frac{\partial}{\partial w_{ji}}\Phi(\sum_j w_{ji}a_j) =$$

$$= 2\cdot Err\frac{\partial \Phi(S_i)}{\partial S_i}\cdot\frac{\partial S_i}{\partial w_{ji}} = 2\cdot(\alpha_i - o_i)\cdot\Phi'(S_i)\cdot a_j$$

Ο κανόνας Δέλτα για συνεχείς συναρτήσεις ενεργοποίησης (2/2)

- Προφανώς τα βάρη πρέπει να αλλάξουν προς κατεύθυνση αντίθετη του ρυθμού αύξησης του σφάλματος.
- Άρα μια λογική αλλαγή του βάρους w_{ij} θα ήταν η εξής:

$$\Delta w_{ji} = -d\cdot\frac{\partial E}{\partial w_{ji}} = -d\cdot(\alpha_i - o_i)\cdot\Phi'(S_i)\cdot a_j$$

  - όπου η σταθερά 0.5 ενσωματώθηκε στο ρυθμό μάθησης d.
- Ο παραπάνω κανόνας μπορεί να εφαρμοστεί μόνο για συνεχείς συναρτήσεις ενεργοποίησης.
- Για τη σιγμοειδή συνάρτηση, ισχύει $\Phi'(S_i)=\Phi(S_i)\cdot(1-\Phi(S_i))$.
- Άρα, για τη σιγμοειδή συνάρτηση ο κανόνας δέλτα γίνεται:

$$\Delta w_{ji} = -d\cdot\frac{\partial E}{\partial w_{ji}} = -d\cdot(\alpha_i - o_i)\cdot a_i[1-a_i]\cdot a_j$$

Παρατηρήσεις (1/2)

- Από τη σχέση $\Delta w_{ji}=-d\cdot(a_i-o_i)\cdot\Phi'(S_i)\cdot a_j$ παρατηρούμε τα εξής:
- Τα βάρη στις εισόδους ενός νευρώνα επηρεάζονται το ίδιο από την τιμή της παραγώγου $\Phi'(S_i)$.
  - Κατά κάποιο τρόπο η παράγωγος $\Phi'(S_i)$ τροποποιεί τον ρυθμό μάθησης του νευρώνα.
- Τα βάρη διαφορετικών νευρώνων επηρεάζονται διαφορετικά από τις αντίστοιχες παραγώγους.
  - Άρα, η παράγωγος της συνάρτησης ενεργοποίησης ουσιαστικά δημιουργεί ξεχωριστό ρυθμό μάθησης για κάθε νευρώνα και για κάθε παράδειγμα εκπαίδευσης.

Παρατηρήσεις (2/2)

- Για τη σιγμοειδή συνάρτηση, η παράγωγος Φ'(S) είναι μεγαλύτερη όταν το S τείνει στο μηδέν και μικρότερη όταν το S απομακρύνεται από το μηδέν.
- Άρα νευρώνες οι οποίοι έχουν είσοδο κοντά στο μηδέν και έξοδο κοντά στο 0.5 (οι «αναποφάσιστοι» ή «ασταθείς» νευρώνες) είναι αυτοί που εμφανίζουν τον υψηλότερο ρυθμό μάθησης, δηλαδή έχουν τις μεγαλύτερες μεταβολές βαρών.

Σφάλματα

- Ως σφάλμα αναφορικά με ένα παράδειγμα μάθησης p και έναν νευρώνα k ορίζεται η ποσότητα:
  - $Err=(a_{k,p}-o_{k,p})$
- Το συνολικό σφάλμα για όλα τα παραδείγματα εκπαίδευσης είναι για τον νευρώνα k είναι:

$$E = \sum_{p} (a_{k,p} - o_{k,p})^2$$

- Με παρόμοιο τρόπο μπορούμε να ορίσουμε και το μέσο σφάλμα για όλους τους νευρώνες εξόδου k και όλα τα παραδείγματα p:

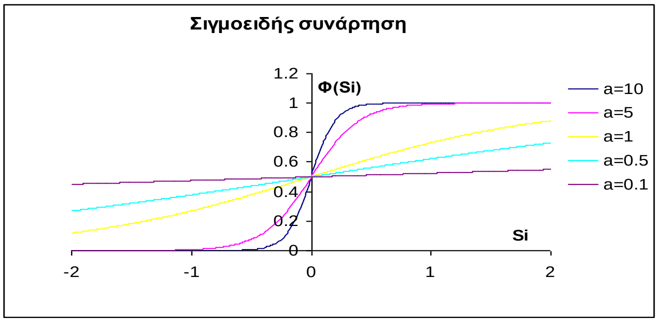
$$E = \frac{1}{P \cdot K} \sum_{p=1}^{P} \sum_{k=1}^{K} (a_{k,p} - o_{k,p})^2$$

Παρατηρήσεις

- Για συνεχείς συναρτήσεις όπως η σιγμοειδής, το αποτέλεσμα εξόδου δεν γίνεται ποτέ 1 ή 0, αλλά η τιμή της εξόδου τείνει ασυμπτωτικά στις τιμές αυτές (αναλόγως το παράδειγμα κάθε φορά) με την πρόοδο της μάθησης.
- Άρα ο τερματισμός της διαδικασίας μάθησης πραγματοποιείται όταν το συνολικό σφάλμα E για όλα τα παραδείγματα και για όλους τους νευρώνες εξόδου πέσει κάτω από μια μικρή τιμή.

Παράδειγμα μάθησης με σιγμοειδή συνάρτηση ενεργοποίησης (1/3)

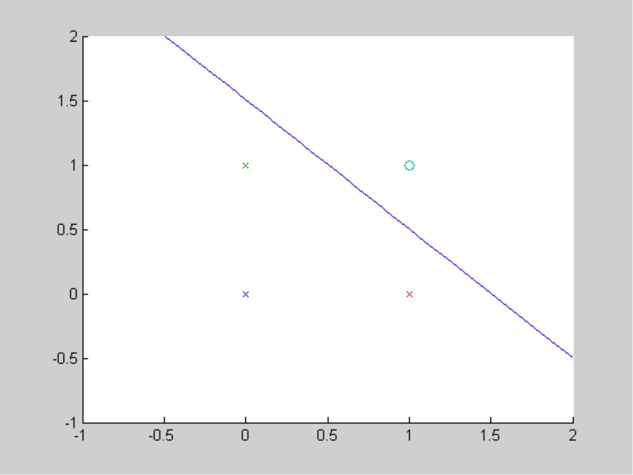
- Προσπαθούμε να προσομειώσουμε την πύλη AND με νευρώνα σιγμοειδούς συνάρτησης ενεργοποίησης.
- Έστω a=1 η παράμετρος της σιγμοειδούς συνάρτησης.



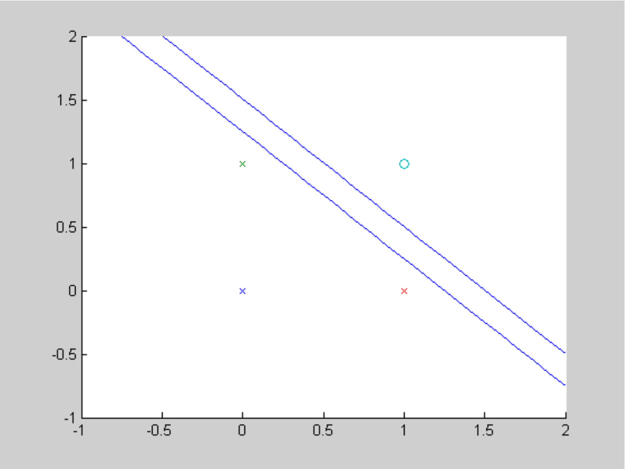
Παράδειγμα μάθησης με σιγμοειδή συνάρτηση ενεργοποίησης (2/3)

- Έστω 0.1 το όριο για το άθροισμα των τετραγωνικών σφαλμάτων για όλα τα παραδείγματα.
- Έστω d=1 ο ρυθμός μάθησης (αυξημένος σε σχέση με τις βηματικές συναρτήσεις).
- Μετά από 1268 επαναλήψεις (317 φορές για κάθε δείγμα) το νευρωνικό δίκτυο πέτυχε το στόχο του μέγιστου σφάλματος που του θέσαμε, καταλήγοντας στα παρακάτω βάρη:
  - w_x=2.9990, w_y=2.9918, w_z=9.0085
- Πράγματι για τις τιμές αυτές παίρνουμε:
  - 0 AND 0 → 0.0109
  - 0 AND 1 → 0.1806
  - 1 AND 0 → 0.1817
  - 1 AND 1 → 0.8156

Παράδειγμα μάθησης με σιγμοειδή συνάρτηση ενεργοποίησης (3/3)

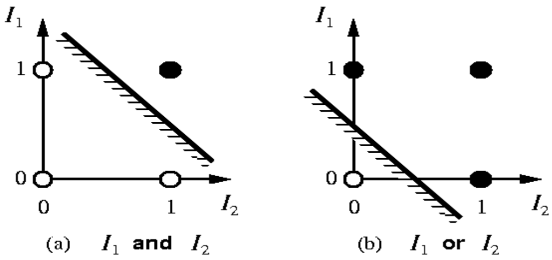


Σύγκριση βηματικής με σιγμοειδή συνάρτηση ενεργοποίησης



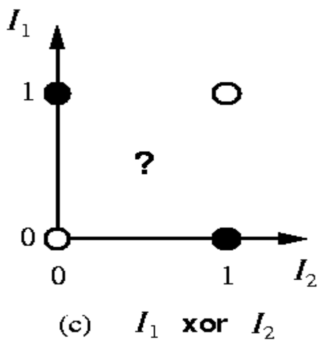
Γραμμικώς διαχωρίσιμες συναρτήσεις (1/2)

- Οι απλοί νευρώνες (perceptrons) μπορούν, μετά την εκπαίδευση, να "μαθαίνουν" γραμμικώς διαχωρίσιμες συναρτήσεις, όπως οι AND και OR.



Γραμμικώς διαχωρίσιμες συναρτήσεις (2/2)

- Δεν μπορούν όμως να "μάθουν" μη-γραμμικώς διαχωρίσιμες συναρτήσεις, όπως η XOR.
- Πράγματι, εάν προσπαθήσουμε να εκπαιδεύσουμε έναν απλό νευρώνα (όπως κάναμε με τη συνάρτηση AND), η διαδικασία της εκπαίδευσης δεν θα συγκλίνει ποτέ.
- Για μη-γραμμικές συναρτήσεις απαιτούνται νευρωνικά δίκτυα περισσοτέρων επιπέδων.



Νευρωνικά Δίκτυα Πολλών Επιπέδων

Δίκτυα πολλών επιπέδων

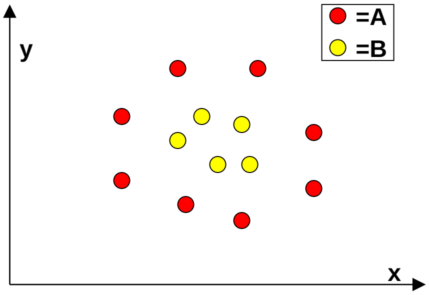
- Προβλήματα που δεν είναι γραμμικά διαχωρίσιμα μπορούν να λυθούν από τεχνητά νευρωνικά δίκτυα πολλών επιπέδων.
- Συνήθως έχουμε τα εξής επίπεδα:
  - Το επίπεδο εισόδου, το οποίο απλά στέλνει τα σήματα εισόδου σε όλους τους νευρώνες του κρυφού επιπέδου.
    - Το επίπεδο εισόδου δεν έχει νευρώνες. Λειτουργεί περίπου ως hub.
  - Το κρυφό επίπεδο, που αποτελείται από μη-γραμμικούς νευρώνες.
  - Το επίπεδο εξόδου, το οποίο μπορεί να αποτελείται είτε από γραμμικούς ή από μη γραμμικούς νευρώνες (ή και από συνδυασμό τους).
    - Μη-γραμμικούς (π.χ. σιγμοειδείς) νευρώνες έχουμε για τις εξόδους που πρέπει να έχουν διακριτές τιμές.
    - Γραμμικούς νευρώνες έχουμε για τις εξόδους που πρέπει να έχουν συνεχείς τιμές.
- Το επίπεδο εισόδου συνήθως δεν το μετράμε στο συνολικό αριθμό επιπέδων ενός δικτύου.

Δυνατότητες δικτύων

- Χρειάζεται τουλάχιστον ένα επίπεδο με μη-γραμμικούς νευρώνες.
- Ένα δίκτυο οσωνδήποτε επιπέδων που αποτελείται μόνο από γραμμικούς νευρώνες μπορεί να αντικατασταθεί από ένα ισοδύναμο δίκτυο ενός επιπέδου γραμμικών νευρώνων.
- Ένα δίκτυο με δύο επίπεδα νευρώνων (κρυφό-εξόδου) με μη-γραμμικούς νευρώνες στο κρυφό επίπεδο μπορεί να παραστήσει οποιαδήποτε συνάρτηση (με πεπερασμένο αριθμό ασυνεχειών) με οσηδήποτε ακρίβεια, η οποία καθορίζεται από τον αριθμό των νευρώνων του κρυφού επιπέδου.

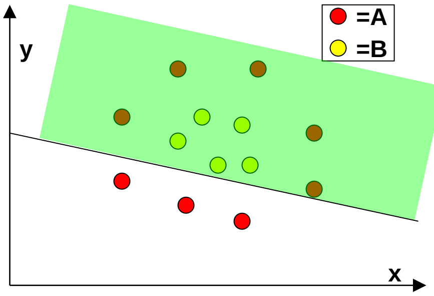
Παράδειγμα (1/4)

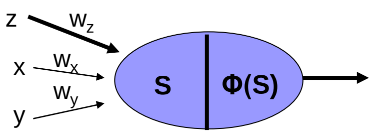
- Έστω το παρακάτω πρόβλημα κατηγοριοποίησης, με δύο βαθμούς ελευθερίας (x, y) και δύο κατηγορίες (A, B), το οποίο δεν είναι γραμμικά διαχωρίσιμο, άρα δεν μπορεί να "λυθεί" από έναν απλό νευρώνα.
  - Για παράδειγμα, οι άξονες θα μπορούσαν να είναι η ηλικία (x) και το εισόδημα (y) ενός ανθρώπου και οι δύο κατηγορίες να αντιπροσωπεύουν το είδος του αυτοκινήτου (A=σπορ, B=οικογενειακό) που προτιμά.



Παράδειγμα (2/4)

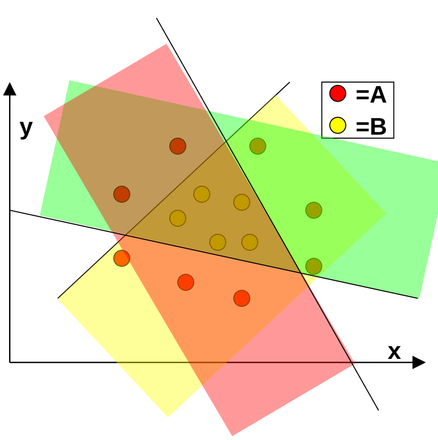
- Χρησιμοποιώντας έναν σιγμοειδή νευρώνα μπορούμε μπορούμε να χωρίσουμε το επίπεδο σε δύο ημιεπίπεδα.

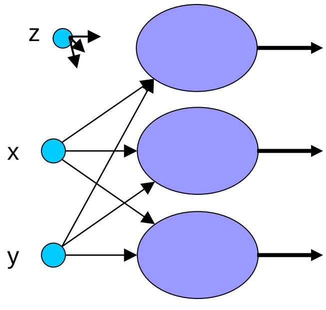




Παράδειγμα (3/4)

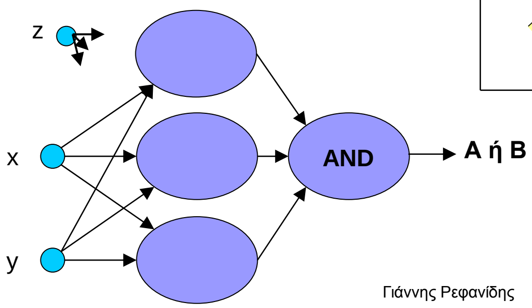
- Χρησιμοποιώντας τρεις σιγμοειδείς νευρώνες μπορούμε μπορούμε να χωρίσουμε το επίπεδο σε ημιεπίπεδα με τρεις διαφορετικούς τρόπους.

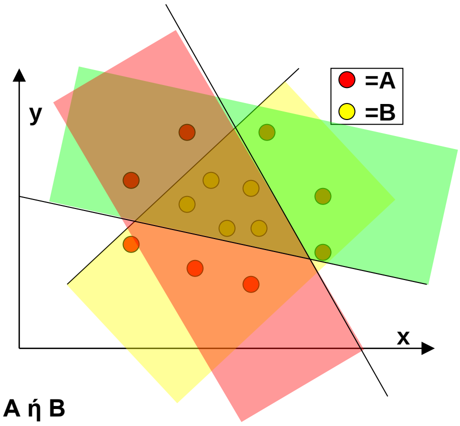




Παράδειγμα (4/4)

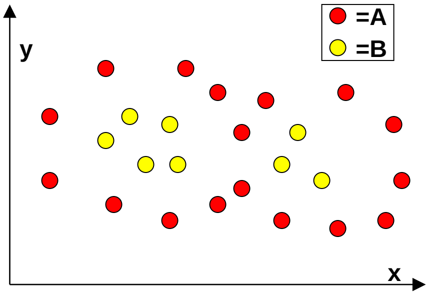
- Η τομή των τριών ημι-επιπέδων μπορεί να ληφθεί από έναν σιγμοειδή νευρώνα με τρεις εισόδους, που λειτουργεί ως πύλη AND.





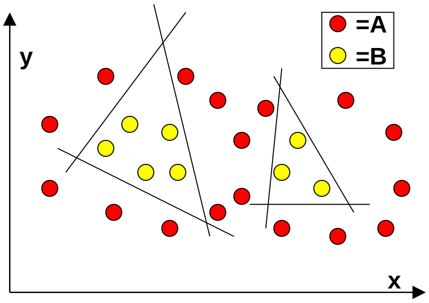
Συμπέρασμα

- Με την προηγούμενη μέθοδο μπορούμε να εκπαιδεύσουμε ένα νευρωνικό να αναγνωρίζει οποιοδήποτε σύνολο περιβάλλεται από κυρτό πολύγωνο στο επίπεδο (ή γενικότερα από κυρτή υπερ-επιφάνεια σε οποιονδήποτε υπερ-χώρο).
- Τι γίνεται με προβλήματα όπου οι κατηγορίες δεν περιβάλλονται από κυρτό πολύγωνο; (όπως στο διπλανό σχήμα)

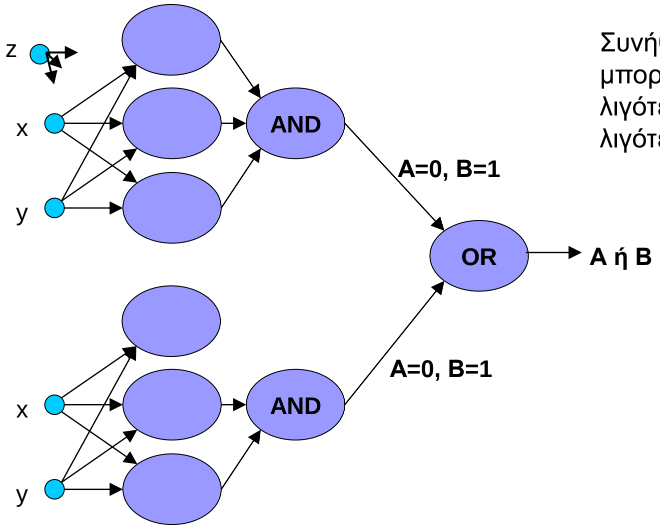


Μη κυρτές περιοχές (1/2)

- Για το παράδειγμα του διπλανού σχήματος μπορούμε να εκπαιδεύσουμε δύο νευρωνικά δίκτυα να αναγνωρίζουν τις δύο κυρτές περιοχές (όπως στο παράδειγμα που προηγήθηκε) και στη συνέχεια να συνδυάσουμε τις επιμέρους εξόδους τους με έναν σιγμοειδή νευρώνα που υλοποιεί τη συνάρτηση OR.



Μη κυρτές περιοχές (2/2)



Συνήθως η ίδια λειτουργία μπορεί να επιτευχθεί με λιγότερα επίπεδα και λιγότερους νευρώνες.

Μάθηση με Οπισθοδιάδοση Σφάλματος

Backpropagation Learning

Κεντρική ιδέα

- Τα παραδείγματα μάθησης παρουσιάζονται στο μη-εκπαιδευμένο δίκτυο και υπολογίζονται οι έξοδοι.
- Για κάθε νευρώνα εξόδου υπολογίζεται το σφάλμα και γίνεται η σχετική αλλαγή των βαρών εισόδου.
- Με κατεύθυνση από το επίπεδο εξόδου προς το επίπεδο εισόδου, για κάθε εσωτερικό νευρώνα υπολογίζεται η συμμετοχή του στα σφάλματα των νευρώνων εξόδου και γίνεται η αλλαγή των βαρών στην είσοδό του.
  - Η συμμετοχή ενός νευρώνα στα σφάλματα των νευρώνων του επόμενου επιπέδου του είναι ανάλογη της τρέχουσας εισόδου του και των συντελεστών βαρύτητας που τον συνδέουν με τους νευρώνες του επόμενου επιπέδου.

Προσαρμοσμένα σφάλματα (1/2)

- Έστω k ένας νευρώνας του επιπέδου εξόδου.
- Έστω a_k η έξοδος του νευρώνα k για ένα συγκεκριμένο παράδειγμα εισόδου και o_k η επιθυμητή έξοδος για το ίδιο παράδειγμα.
- Προφανώς το σφάλμα στη συγκεκριμένη περίπτωση για τον νευρώνα k είναι:
  - Σφάλμα = a_k-o_k
- Το πραγματικό σφάλμα κάθε νευρώνα το πολλαπλασιάζουμε επί την παράγωγο της συνάρτησης ενεργοποίησης, σύμφωνα με το γενικευμένο κανόνα δέλτα:
  - $\delta_k=(a_k-o_k)\Phi'(S_k)$
- Η παραπάνω τιμή ονομάζεται *προσαρμοσμένο σφάλμα νευρώνα*.

Προσαρμοσμένα σφάλματα (2/2)

- Ο πολλαπλασιασμός των σφαλμάτων επί την παράγωγο της συνάρτησης ενεργοποίησης έχει ως αποτέλεσμα:
  - Να μειώνεται αυξάνεται η βαρύτητα του σφάλματος για τα παραδείγματα εκείνα που η έξοδός τους $\Phi(S_k)$ φαίνεται να έχει σταθεροποιηθεί είτε στο 0 ή στο 1 (για σιγμοειδή συνάρτηση ενεργοποίησης).
  - η κλίση του διανύσματος $\Delta\mathbf{W}=(\Delta w_{i1,k}, \Delta w_{i2,k}, ..., \Delta w_{in,k})$ να έχει την κατεύθυνση της πιο απότομης μεταβολής (πτώσης) της επιφάνειας του σφάλματος E=E(**W**), όπου **W** το σύνολο των βαρών του νευρωνικού δικτύου.
- Τα προσαρμοσμένα σφάλματα είναι σε αντιστοιχία με τον γενικευμένο κανόνα δέλτα για συνεχείς συναρτήσεις ενεργοποίησης (διαφάνεια 40).
  - Ουσιαστικά μεταφέρουμε την παράγωγο της συνάρτησης ενεργοποίησης μέσα στο σφάλμα.

Σφάλματα κρυφών επιπέδων

- Τα σφάλματα των νευρώνων των κρυφών επιπέδων υπολογίζονται από τα σφάλματα των νευρώνων του αμέσως επόμενου επιπέδου, ως εξής:

$$\delta_i = \Phi'(S_i) \cdot \sum w_{ik} \delta_k$$

- όπου:
  - i ένας νευρώνας του κρυφού επιπέδου.
  - Το άθροισμα αναφέρεται σε όλους τους νευρώνες k του επόμενου επιπέδου (ή των επόμενων επιπέδων) με τους οποίους ο νευρώνας i συνδέεται με βάρη w_{ik}.
- Με τον παραπάνω τρόπο υπολογίζονται τα σφάλματα για όλους τους νευρώνες του δικτύου, μέχρι και το επίπεδο εισόδου.

Αλλαγές στα βάρη

- Έχοντας υπολογίσει για κάθε νευρώνα i το σφάλμα δ_i, η αλλαγή στα βάρη εισόδου σε όλους τους νευρώνες γίνεται ως εξής:

$$\Delta w_{ji} = -d \cdot \delta_i \cdot \alpha_j$$

- Δηλαδή, η αλλαγή στο βάρος από τον νευρώνα j στον νευρώνα i εξαρτάται από το σφάλμα του νευρώνα i, την έξοδο του νευρώνα j και την σταθερά μάθησης (learning rate) d.
  - Προσοχή: Η παράγωγος της συνάρτησης ενεργοποίησης έχει συμπεριληφθεί μέσα στα προσαρμοσμένα σφάλματα.

Επαναλήψεις

- Υπάρχουν δύο τρόποι παρουσίασης των παραδειγμάτων και αλλαγής των βαρών:
  - Για κάθε παράδειγμα προβαίνουμε αμέσως σε αλλαγές των βαρών (*αυξητική εκπαίδευση* - incremental training).
    - Μπορούμε να παρουσιάζουμε τα παραδείγματα και με τυχαία σειρά.
  - Περιμένουμε να παρουσιαστούν όλα τα παραδείγματα μια φορά, υπολογίζουμε τις αλλαγές των βαρών για κάθε παράδειγμα και τις εφαρμόζουμε ταυτόχρονα αφού παρουσιαστούν όλα τα παραδείγματα (*μαζική εκπαίδευση* - batch training).
- Η παρουσίαση όλων των παραδειγμάτων μια φορά (ανεξαρτήτως του τρόπου αλλαγής των βαρών) ονομάζεται *εποχή εκπαίδευσης* (epoch).

Συνθήκες τερματισμού εκπαίδευσης

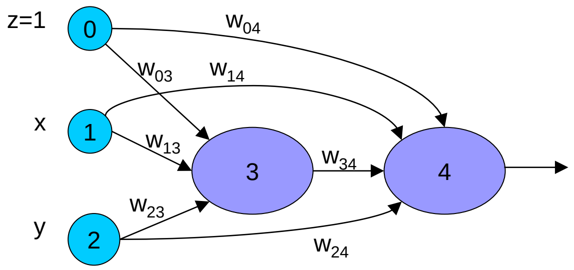
- Το συνολικό σφάλμα για όλα τα παραδείγματα ορίζεται σαν το άθροισμα των τετραγώνων των σφαλμάτων των νευρώνων εξόδου:

$$E = \frac{1}{P \cdot K} \sum_{p=1}^{P} \sum_{k=1}^{K} \delta_{k,p}^2$$

- Ως συνθήκη τερματισμού μπορεί να οριστεί η πτώση του παραπάνω σφάλματος κάτω από ένα όριο.
  - Εναλλακτικά, ως συνθήκη τερματισμού μπορεί να θεωρηθεί η πραγματοποίηση ενός συγκεκριμένου αριθμού εποχών εκπαίδευσης ή η πάροδος ενός συγκεκριμένου χρονικού διαστήματος.

Παράδειγμα: Συνάρτηση XOR (1/6)

- Έστω το παρακάτω δίκτυο με δύο σιγμοειδείς νευρώνες (3 και 4).



- Το κρυφό επίπεδο αποτελείται μόνο από τον νευρώνα 3, ενώ οι είσοδοι 1 και 2 συνδέονται απευθείας και με τον νευρώνα εξόδου 4.
- Φυσικά θα μπορούσαν να χρησιμοποιηθούν διαφορετικές τοπολογίες δικτύου για την ίδια συνάρτηση.

Παράδειγμα: Συνάρτηση XOR (2/6)

- Έστω ότι αρχικά όλα τα βάρη έχουν τιμή ίση με μηδέν.
- Έστω ότι το πρώτο παράδειγμα είναι το x=0, y=0, το οποίο αναμένουμε να βγάλει έξοδο $o_4=0$.
- Η συνολική είσοδος στον νευρώνα 3 είναι $S_3=0*1+0*0+0*0=0$, οπότε η έξοδός του είναι $\Phi(S_3)=1/(1+e^0)=0.5$.
- Παρόμοια, η συνολική είσοδος στον νευρώνα 4 είναι $S_4=0*1+0*0+0*0.0+0*0.5=0$, οπότε η έξοδός του είναι: $\Phi(S_4)=1/(1+e^0)=0.5$.

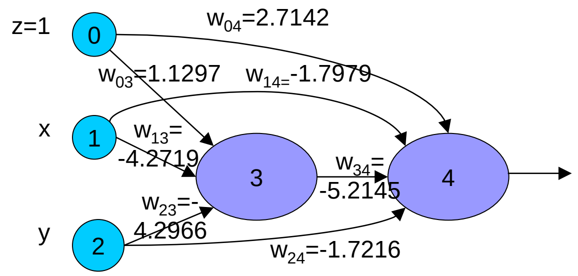
Παράδειγμα: Συνάρτηση XOR (3/6)

- Το σφάλμα στον νευρώνα εξόδου 4 είναι:
  - $\delta_4=(0.5-0)*\Phi'(S_4)=(0.5-0)*\Phi(0)*(1-\Phi(0))=(0.5-0)*0.5*0.5=0.125$
- Το σφάλμα στον κρυφό νευρώνα 3 είναι:
  - $\delta_3=\Phi'(S_3)*w_{34}*\delta 4=0$
    - (το βάρος w_{34} είναι ακόμη 0)
- Τα βάρη που αλλάζουν είναι τα w_{04}, w_{14}, w_{24} και w_{34}.
- Για παράδειγμα, το w_{04} γίνεται:
  - $w_{04\text{-}new}=w_{04\text{-}old}-d*\delta_4*a_0=0-1*(0.125)*1=-0.125$
    θεωρώντας ότι ο ρυθμός μάθησης είναι d=1.

Παράδειγμα: Συνάρτηση XOR (4/6)

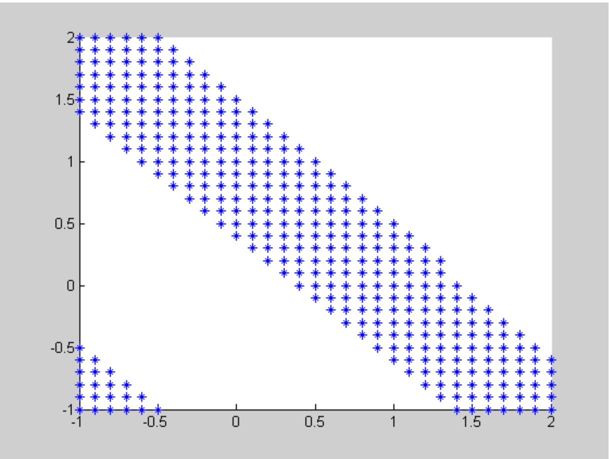
- Με το επόμενο παράδειγμα θα αλλάξουν όλα τα βάρη.
- Θέτοντας όριο στο σφάλμα E=0.1, μετά από 563 εποχές εκπαίδευσης τα βάρη παίρνουν τις παρακάτω τιμές:
  - $w_{03}=1.1297$, $w_{13}=-4.2719$, $w_{23}=-4.2966$
  - $w_{04}=2.7142$, $w_{14}=-1.7979$, $w_{24}=-1.7216$, $w_{34}=-5.2145$

Παράδειγμα: Συνάρτηση XOR (5/6)



x	y	a_3	a_4	$(a_4-o_4)^2$
0	0	0.7558	0.2267	0.0514
0	1	0.0404	0.6861	0.0985
1	0	0.0414	0.6683	0.1100
1	1	0.0005	0.3083	0.0950

Παράδειγμα: Συνάρτηση XOR (6/6)



Παρατηρήσεις στη μέθοδο

- Η μέθοδος της μάθησης με οπισθοδιάδοση λάθους δεν συγκλίνει πάντα στη βέλτιστη λύση.
- Ουσιαστικά εκτελεί μια αναζήτηση τύπου "αναρρίχησης λόφων" στο χώρο των συντελεστών βαρύτητας και με ευρετική συνάρτηση την κλίση του συνολικού σφάλματος, προσπαθώντας να βρει το ολικό ελάχιστο του συνολικού σφάλματος.
- Υπάρχει λοιπόν περίπτωση να παγιδευτεί σε τοπικά ελάχιστα και έτσι να μην βρει τα βέλτιστα βάρη.
- Αντιμετώπιση του προβλήματος:
  - Επανεκκινήσεις με τυχαία αρχικά βάρη.
  - Στοχαστική μεταβολή των βαρών
  - Αύξηση του αριθμού των νευρώνων του κρυφού επιπέδου.
  - κλπ

Matlab Neural Network Toobox

Τρέχουσα έκδοση: 4.0
(από το Matlab 6.0 και μετά)

Γενικά

- Εκκίνηση με:

  >>nntool
- Δυνατότητα για:
  - Κατασκευή νευρωνικών δικτύων διαφόρων τύπων
  - Εκπαίδευση
  - Προσομοίωση λειτουργίας
  - Δημιουργία/Αποθήκευση/Ανάκτηση συνόλων δεδομένων

Perceptrons

- Ένα στρώμα από νευρώνες
- Συνάρτηση ενεργοποίησης:
  - HARDLIM (βηματική 0/1)
  - HARDLIMS (προσήμου -1/1)
- Κανόνας μάθησης
  - LEARNP
    - Ο γνωστός κανόνας Δέλτα: $\Delta w_{j,i}=w_{j,i\text{-new}}-w_{j,i\text{-old}}=-d(a_i-o_i)a_j$
  - LEARNPN
    - Κανονικοποιημένος κανόνας Δέλτα: $\Delta w_{j,i}=w_{j,i\text{-new}}-w_{j,i\text{-old}}=-d(a_i-o_i)a_j/||a_j||$
    - Αποφεύγει προβλήματα με μεγάλες διαφορές στα μέτρα των διανυσμάτων εισόδου.

Δίκτυα πολλών επιπέδων απλής τροφοδότησης (1/3)

- Network type: **feed forward backprop**
- Συνάρτηση ενεργοποίησης
  - Μπορεί να είναι διαφορετική για κάθε επίπεδο
  - Διαθέσιμες:
    - LOGSIG (σιγμοειδής 0/1)
    - TANSIG (σιγμοειδής -1/1)
    - PURELIN (γραμμική)

Στο Matlab η τάση πόλωσης ισούται πάντα με 1

Δίκτυα πολλών επιπέδων απλής τροφοδότησης (2/3)

- Δύο μέθοδοι εκπαίδευσης:
  - TRAIN: Μαζική εκπαίδευση (batch training)
  - ADAPT: Αυξητική εκπαίδευση (incremental training)
- Θα χρησιμοποιούμε την μέθοδο **TRAIN**, η οποία μας δίνει περισσότερες επιλογές.
- Επιλογές εκπαίδευσης για τη μέθοδο TRAIN:
  - TRAINGD (train gradient descent): Η μάθηση με οπισθοδιάδοση λάθους στην απλή της μορφή.
    - ...και πολλές άλλες (θα αναφερθούν παρακάτω)

Δίκτυα πολλών επιπέδων απλής τροφοδότησης (3/3)

- Συνάρτηση μέτρησης του σφάλματος (Performance function):
  - SSE (Sum squared error): Άθροισμα τετραγώνων σφαλμάτων
  - MSE (Mean squared error): Μέσος όρος τετραγώνων σφαλμάτων για όλα τα παραδείγματα και όλες τις εξόδους.
  - MSEREG (Mean squared error with regularization):

$$SSE = \sum_{i=1}^{N} (a_i - o_i)^2$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (a_i - o_i)^2$$

$$MSEREG = \gamma \frac{1}{N} \sum_{i=1}^{N} (a_i - o_i)^2 + (1-\gamma) \frac{1}{N} \sum_{j=1}^{n} w_j^2$$

adapt vs train (1/2)

- Το Matlab διαθέτει δύο εντολές για εκπαίδευση, την adapt και την train.
- Η train εκτελεί πάντα batch training, έχει περισσότερες επιλογές εκπαίδευσης και συνήθως είναι αποτελεσματικότερη/αποδοτικότερη.
- Η adapt εκτελεί τόσο batch, όσο και incremental training, ανάλογα με τα δεδομένα.

adapt vs train (2/2)

- Εάν τα δεδομένα είναι απλοί πίνακες, η adapt εκτελεί batch training.
  - in=[1 2 3 4 5]; out=[1 4 9 16 25]
- Εάν τα δεδομένα είναι πίνακες κελιών, η adapt εκτελεί incremental training
  - >> in={1 2 3 4 5}; out={1 3 5 7 9}
  - in = [1] [2] [3] [4] [5]
  - out = [1] [3] [5] [7] [9]

Wizards

- nnstart
  - nftool για δίκτυα προσέγγισης συνάρτησης
  - nprtool για δίκτυα ταξινόμησης
  - nctool για ομαδοποίηση
  - ntstool για χρονοσειρές

Παραλλαγές της μάθησης με οπισθοδιάδοση σφάλματος

Ορμή (momentum)

- Με τον όρο 'ορμή' στην μεταβολή των βαρών εννοούμε την επίδραση της προηγούμενης μεταβολής ενός βάρους στην επόμενη:
  - $\Delta w_{ji}(t+1)=-(1-mc)d \cdot \delta_i \cdot a_j+mc \cdot \Delta w_{ij}(t)$
  - όπου mc (momentum constant) είναι η σταθερά της ορμής και παίρνει τιμές από το 0 έως το 1
- Χρησιμοποιείται σε ομαδική εκπαίδευση (batch training).
- Βοηθά στην αποφυγή μικρών τοπικών ελαχίστων.
- Συνήθως επιταχύνει την σύγκλιση.

(Matlab)

- Μέθοδος εκπαίδευσης: **TRAINGDM** (train gradient descent momentum)
  - Τυπική τιμή mc=0.9
- Εάν το συνολικό σφάλμα μιας νέας εποχής εκπαίδευσης ξεπερνά το σφάλμα της προηγούμενης εποχής εκπαίδευσης κατά ένα προκαθορισμένο ποσοστό (π.χ.1.04) τότε η ορμή για τη συγκεκριμένη εποχή δεν λαμβάνεται υπόψη.
  - net.trainparam.max_perf_inc=1.04

Μεταβλητός ρυθμός μάθησης (1/2) (Variable learning rate)

- Ένας μικρός ρυθμός μάθησης καθυστερεί την σύγκλιση.
- Ένας μεγάλος ρυθμός μάθησης μπορεί να προκαλέσει 'ταλαντώσεις' στην αναζήτηση και αδυναμία σταθεροποίησης στη λύση.
- Μεταβλητός ρυθμός μάθησης (με βάση κάποιο κριτήριο) μπορεί να αποφύγει τα παραπάνω δύο προβλήματα.

Μεταβλητός ρυθμός μάθησης (2/2)

- Εάν μετά από μια εποχή εκπαίδευσης το συνολικό σφάλμα ξεπερνά το σφάλμα της προηγούμενης εποχής κατά κάποιο ποσοστό (π.χ. επί 1.04), ο ρυθμός μάθησης μειώνεται (π.χ. επί 0.7).
- Εάν μετά από μια εποχή εκπαίδευσης το συνολικό σφάλμα είναι μικρότερο από το σφάλμα της προηγούμενης εποχής, ο ρυθμός μάθησης αυξάνεται (π.χ. επί 1.05).
- Η τεχνική του μεταβλητού ρυθμού μάθησης και της ορμής μπορούν να συνδυαστούν για ακόμη καλύτερα αποτελέσματα (πάντα σε ομαδική μάθηση - batch training).

(Matlab)

- Μέθοδος μάθησης: **TRAINGDA** (train gradient descent adaptive)
- Βασικές παράμετροι δικτύου:
  - net.trainparam.max_perf_inc=1.04
  - net.trainparam.lr_dec=0.7
  - net.trainparam.lr_inc=1.05
- Μέθοδος μάθησης: **TRAINGDX**
  - Συνδυάζει μεταβλητό ρυθμό μάθησης και ορμή.

Ευπροσάρμοστη οπισθοδιάδοση (Resilient backpropagation)

- Παραλλαγή του μεταβλητού ρυθμού μάθησης, που δεν λαμβάνει υπόψη την παράγωγο της συνάρτησης ενεργοποίησης αλλά μόνο το πρόσημο του σφάλματος.
- Η μεταβολή των βαρών δίνεται από τη σχέση:
  - $\Delta w_{ji}=\mp d$
    αναλόγως το πρόσημο του σφάλματος και τη σχετική είσοδο.
  - Η θετική σταθερά d πολλαπλασιάζεται με delta_inc>1 κάθε φορά που για δύο συνεχόμενες εποχές η μεταβολή έχει το ίδιο πρόσημο ενώ κατά delta_dec<1 κάθε φορά που η μεταβολή αλλάζει πρόσημο.
  - Η μέθοδος αυτή αντιμετωπίζει το πρόβλημα των πολύ μικρών παραγώγων, που εμφανίζεται έντονα σε μεγάλα δίκτυα.
  - Επίσης αντιμετωπίζει το πρόβλημα της ταλάντωσης γύρω από τοπικά ελάχιστα, μειώνοντας την τιμή του d.

(Matlab)

- Μέθοδος μάθησης: **TRAINRP** (train resilient backpropagation)
- Βασικές παράμετροι:
  - net.trainparams.delta0 = 0.07
  - net.trainparams.delta_inc = 1.2
  - net.trainparams.delta_dec = 0.5
  - net.trainparams.delta_max = 50

Αλγόριθμοι συζυγών κλίσεων
(Conjugate gradient algorithms)

- Πρόκειται για μια οικογένεια αλγορίθμων, οι οποίοι σε κάθε βήμα εκτελούν τοπική αναζήτηση προσπαθώντας να βρουν την βέλτιστη τιμή αλλά και την βέλτιστη κατεύθυνση μεταβολής των βαρών.
- Συνήθως πετυχαίνουν πολύ καλά αποτελέσματα, επιταχύνοντας κατά 10-100 φορές τη διαδικασία της εκπαίδευσης.
  - Λειτουργούν με αναλυτικό υπολογισμό δεύτερων παραγώγων
    - Fletcher-Reeves update (**TRAINCGF**)
    - Polak-Ribiere update (**TRAINCGP**)
    - Powell-Beale restarts (**TRAINCGB**)
    - Scaled conjugate gradient (**TRAINSCG**)
  - Λειτουργούν με υπολογισμό πρώτων παραγώγων και αριθμητική προσέγγιση δεύτερων παραγώγων
    - BFGS algorithm (**TRAINBGF**)
    - One step secant algorithm (**TRAINOSS**)
    - Levenberg-Marquardt (**TRAINLM**) ← **Προτεινόμενη από το Matlab μέθοδος**

Σύγκριση μεθόδων (1/2)

- Δύο ειδών προβλήματα:
  - Προσέγγιση συναρτήσεων (function approximation): Γραμμικοί νευρώνες εξόδου
  - Αναγνώριση προτύπων (pattern recognition): Σιγμοειδείς νευρώνες στην έξοδο
- TRAINLM: Προσέγγιση συναρτήσεων
  - Η πιο γρήγορη
  - Πολλές φορές δίνει τα μικρότερα σφάλματα
  - Προβλήματα με μεγάλα δίκτυα
  - Μέτρια σε προβλήματα αναγνώρισης προτύπων

Σύγκριση μεθόδων (2/2)

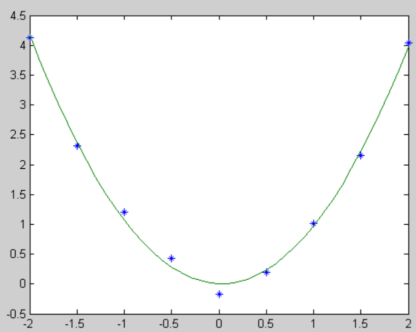
- TRAINRP: Αναγνώριση προτύπων
  - Η γρηγορότερη
  - Γενικά δεν δίνει τα μικρότερα σφάλματα
- TRAINSCG
  - Γρήγορη σε όλα τα προβλήματα
  - Καλή συμπεριφορά ιδιαίτερα στα μεγάλα δίκτυα.
- TRAINGDX
  - Μέτρια αποτελέσματα σε σχέση με τις υπόλοιπες μεθόδους.

Γενίκευση

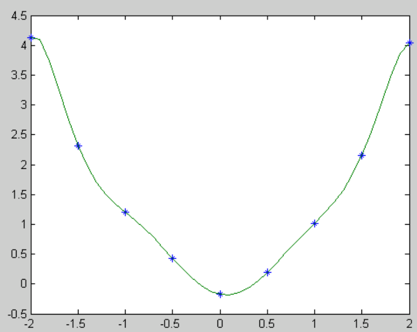
Υπερπροσαρμογή
(Overfitting)

- Ένα από τα βασικά προβλήματα που μπορεί να εμφανιστεί κατά την εκπαίδευση νευρωνικών δικτύων είναι αυτό της υπερβολικής εκπαίδευσης.
  - Το συνολικό σφάλμα για τα παραδείγματα εκπαίδευσης γίνεται πάρα πολύ μικρό, αλλά γίνεται υπεροβολικά μεγάλο σε άλλα παραδείγματα.
- Λέμε τότε ότι το νευρωνικό δίκτυο δεν γενικεύει (generalizes) καλά.
- Αντίστοιχο παράδειγμα έχουμε κατά την προσέγγιση δεδομένων με πολυώνυμα ελαχίστων τετραγώνων (βλέπε παράδειγμα επόμενης διαφάνειας).

Παράδειγμα



Προσέγγιση με πολυώνυμο 2ου βαθμού



Προσέγγιση με πολυώνυμο 8ου βαθμού

Μέγεθος νευρωνικού δικτύου

- Το πρόβλημα της υπερ-εκπαίδευσης παρουσιάζεται σε νευρωνικά δίκτυα με πολύ μεγάλο αριθμό νευρώνων στο κρυφό επίπεδο, σε σχέση πάντα με το πλήθος των παραδειγμάτων.
- Από την άλλη, μικρός αριθμός νευρώνων οδηγεί σε αδυναμία μάθησης.
- Η επιλογή του βέλτιστου αριθμού νευρώνων στο κρυφό επίπεδο απαιτεί συνήθως πειραματισμό.
  - Ένας εμπειρικός κανόνας προτείνει: $N > \frac{W}{\varepsilon}$
  - όπου:
    - W είναι ο αριθμός των βαρών
    - ε είναι η επιθυμητή % τελική τιμή σφάλματος (π.χ. 10% λάθος κατηγοριοποίηση σε νέα δεδομένα).
    - N είναι ο αριθμός των παραδειγμάτων εκπαίδευσης που, για το δεδομένο δίκτυο, απαιτούνται για να επιτευχθεί τιμή σφάλματος σε νέα παραδείγματα ίση με ε.

Γιάννης Ρεφανίδης 99

Τροποποίηση της συνάρτησης σφάλματος

- Τροποποιούμε τη συνάρτηση σφάλματος ως εξής:

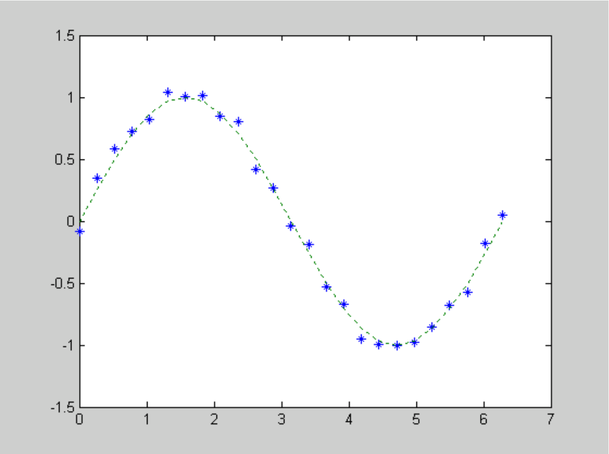
$$MSEREG = \gamma \frac{1}{N}\sum_{i=1}^{N}(a_i - o_i)^2 + (1-\gamma)\frac{1}{N}\sum_{j=1}^{n} w_j^2$$

  όπου γ είναι ο συντελεστής σφάλματος (π.χ. γ=0.5).
- Η παραπάνω συνάρτηση σφάλματος ευνοεί τους μικρούς συντελεστές βαρύτητας.
- Δυσκολία στον καθορισμό του συντελεστή γ.
- Η συνάρτηση σφάλματος MSEREG μπορεί να χρησιμοποιηθεί με όλες τις μεθόδους εκπαίδευσης που έχουν παρουσιαστεί.
- (MATLAB) net.trainparam.ratio = 0.5

Γιάννης Ρεφανίδης 100

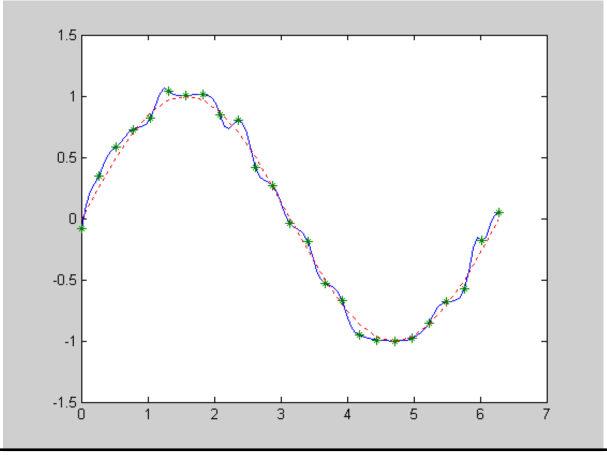
Παράδειγμα (1/5)

- Το σχήμα δείχνει 24 σημεία γύρω από τη συνάρτηση του ημιτόνου (με διακεκομμένη γραμμή είναι η πραγματική συνάρτηση).



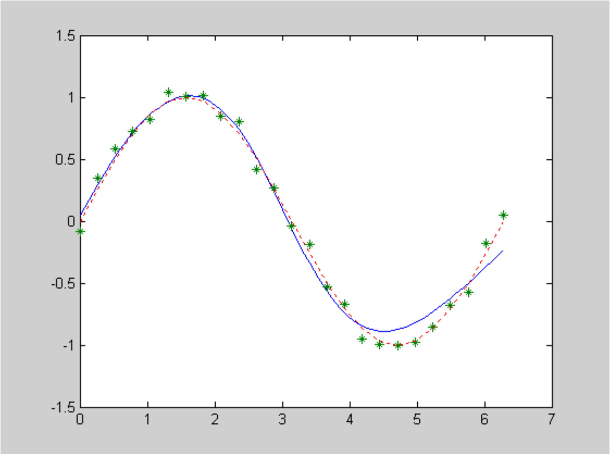
Παράδειγμα (2/5)

- Δίκτυο με 20 LOGSIG - 1 PURELIN, εκπαίδευση με τη μέθοδο TRAINSCG, σφάλμα MSE, για 1000 εποχές.



Παράδειγμα (3/5)

- Δίκτυο με 20 LOGSIG - 1 PURELIN, εκπαίδευση με τη μέθοδο TRAINSCG, σφάλμα MSEREG, για 1000 εποχές.

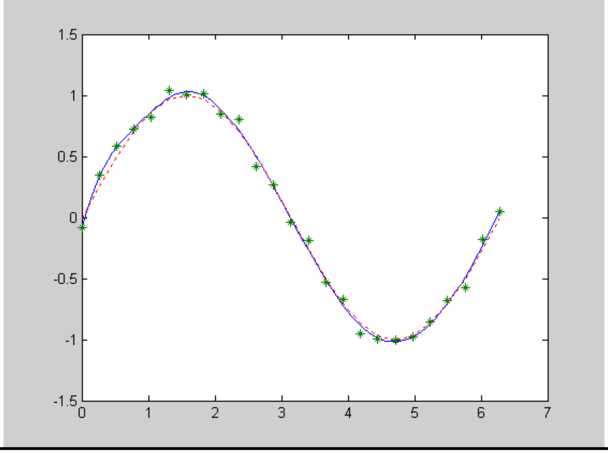


Αυτόματη ομαλοποίηση
(Automated regularization)

- Η μέθοδος εκπαίδευσης της αυτόματης ομαλοποίησης είναι παραλλαγή της μεθόδου TRAINLM, με στόχο την ομαλοποίηση των αποτελεσμάτων.
- Βασίζεται σε στατιστικές μεθόδους.
- Μετρά πόσες παράμετροι του δικτύου (βάρη) χρειάζονται πραγματικά.
- Υλοποιείται στο Matlab με τη μέθοδο **TRAINBR**.

Παράδειγμα (4/5)

- Δίκτυο με 20 LOGSIG-1 PURELIN, εκπαίδευση με τη μέθοδο TRAINBR, για 1000 εποχές.



Πρόωρη διακοπή (1/2) (Early Stopping)

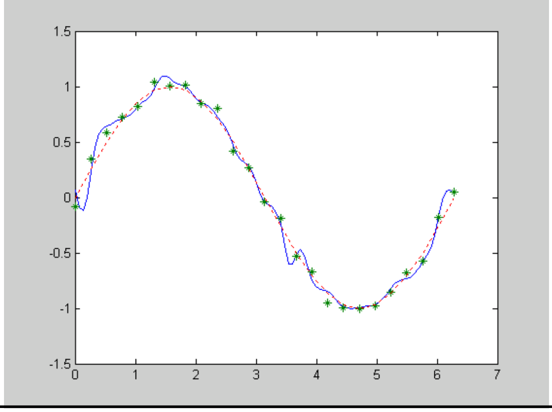
- Τα διαθέσιμα δεδομένα διαιρούνται σε τρεις ομάδες:
  - Τα δεδομένα εκπαίδευσης (training data)
  - Τα δεδομένα επαλήθευσης (validation data)
  - Τα δεδομένα ελέγχου (test data) - *προαιρετικά*
- Ένας συνήθης ποσοστιαίος διαχωρισμός είναι 50-25-25.
- Η εκπαίδευση γίνεται <u>μόνο</u> με τα δεδομένα εκπαίδευσης.
- Κατά την εκπαίδευση παρακολουθείται και το σφάλμα στα δεδομένα επαλήθευσης.
- Εάν κατά την εκπαίδευση συμβεί το σφάλμα στα δεδομένα επαλήθευσης να αυξηθεί για συγκεκριμένο αριθμό συνεχόμενων εποχών, η εκπαίδευση διακόπτεται και επιστρέφονται οι παράμετροι που αντιστοιχούσαν στο μικρότερο σφάλμα των δεδομένων επαλήθευσης.

Πρόωρη διακοπή (2/2)

- Τα δεδομένα ελέγχου χρησιμοποιούνται για να ελέγχουν την ισοκατανομή των δεδομένων επαλήθευσης.
  - Εάν τα δεδομένα ελέγχου παρουσιάσουν ελάχιστο σφάλμα σε αρκετά διαφορετικό σημείο της εκπαίδευσης σε σχέση με τα δεδομένα επαλήθευσης, πραγματοποιείται νέος χωρισμός των δεδομένων σε κατηγορίες.
- Η τεχνική πρόωρης διακοπής δεν έχει ως αποτέλεσμα την κατασκευή καλύτερου νευρωνικού δικτύου, αλλά την ανίχνευση προβλημάτων γενίκευσης κατά την εκπαίδευση με οποιαδήποτε μέθοδο, άρα μπορεί να αποτελέσει ένδειξη για την επιλογή μιας άλλης μεθόδου (π.χ. της TRAINBR).

Παράδειγμα (5/5)

- Δίκτυο με 20 LOGSIG - 1 PURELIN, εκπαίδευση με τη μέθοδο TRAINSCG, σφάλμα MSE, χρήση δεδομένων επαλήθευσης ομοιόμορφα κατανεμημένων ανάμεσα στα δεδομένα εκπαίδευσης, για 35 εποχές (μετά σταμάτησε...).



Κ-πλή Διασταυρωμένη επικύρωση
(K-fold cross validation)

- Χωρίζουμε τυχαία τα δεδομένα σε Κ ίσα υποσύνολα (συνήθως Κ=10)
- Κάνουμε εκπαίδευση με τα Κ-1 υποσύνολα και το τελευταίο χρησιμοποιείται για επαλήθευση.
- Η διαδικασία επαναλαμβάνεται Κ φόρες, κάθε φορά και με διαφορετικό σύνολο επαλήθευσης.
- Τα Κ εκπαιδευμένα δίκτυα χρησιμοποιούνται για την τελική πρόβλεψη (π.χ. πλειοψηφία ή μέσος όρος).

Κανονικοποίηση των δεδομένων

- Πριν την εκπαίδευση είναι χρήσιμο να μετασχηματίζουμε τα δεδομένα εκπαίδευσης, ιδιαίτερα όταν αυτά είναι συνεχείς τιμές, ώστε να είναι ομοιόμορφα τα πεδία τιμών τους.
- Δύο είδη μετασχηματισμών:
  - Ελαχίστης και Μεγίστης τιμής (mapminmax)
    - Γραμμικός μετασχηματισμός του αρχικού συνόλου δεδομένων, έτσι ώστε η νέα ελάχιστη τιμή να είναι -1 και η νέα μέγιστη τιμή να είναι +1.
  - Μέσης τιμής και διασποράς (prestf, poststd, trastd)
    - Γραμμικός μετασχηματισμός του αρχικού συνόλου δεδομένων, έτσι ώστε η νέα μέση τιμή να είναι 0 και η διασπορά να είναι ίση με 1.
- Προσοχή χρειάζεται:
  - στην μετατροπή της εξόδου του νευρωνικού δικτύου στην αρχική κατανομή (Matlab: postmnmx, poststd),
  - στην περίπτωση επέκτασης του συνόλου εκπαίδευσης με νέα παραδείγματα (Matlab: tramnmx, trastd)

(Matlab) Μετασχηματισμός Ελαχίστης και Μεγίστης τιμής (1/2)

- >> X = [1 2 3 4 5
- 2 4 5 7 14] ;
- >> [Y, Settings]=mapminmax(X)
  - Y =
  - -1.0000 -0.5000 0 0.5000 1.0000
  - -1.0000 -0.6667 -0.5000 -0.1667 1.0000
  - Settings =
  - name: 'mapminmax'
  - xrows: 2
  - xmax: [2x1 double]
  - xmin: [2x1 double]
  - xrange: [2x1 double]
  - yrows: 2
  - ymax: 1
  - ymin: -1
  - yrange: 2
  - no_change: 0

(Matlab) Μετασχηματισμός Ελαχίστης και Μεγίστης τιμής (2/2)

- >> Z=mapminmax('reverse',Y,Settings)
- Z =
- 1.0000 2.0000 3.0000 4.0000 5.0000
- 2.0000 4.0000 5.0000 7.0000 14.0000
- >> XX = 3 4 5
- 6 7 8
- >> mapminmax('apply',XX, Settings)
- ans = 0 0.5000 1.0000
- -0.3333 -0.1667 0

(Matlab) Μετασχηματισμός Μέσης τιμής και Διασποράς (1/2)

```
>> X = [ 3 4 7 2 14
         4 11 -8 2 7]  ;
>> [Y,Settings]=mapstd(X)
Y =
  -0.6189   -0.4126    0.2063   -0.8251    1.6503
   0.1124    1.0954   -1.5729   -0.1685    0.5337
Settings =
        name: 'mapstd'
       xrows: 2
       yrows: 2
       xmean: [2x1 double]
        xstd: [2x1 double]
       ymean: 0
        ystd: 1
   no_change: 0
```


(Matlab) Μετασχηματισμός Μέσης τιμής και Διασποράς (2/2)

```
>> Z=mapstd('reverse',Y,Settings)
Z =
    3.0000    4.0000    7.0000    2.0000   14.0000
    4.0000   11.0000   -8.0000    2.0000    7.0000
>> XX = [ 1 2 8
          3 1 7 ] ;
>> W = mapstd('apply',XX,Settings)
W =
   -1.0314   -0.8251    0.4126
   -0.0281   -0.3090    0.5337
```

Άλλες συναρτήσεις

- removeconstantrows
  - Διαγράφει σειρές (εισόδους, attributes) από τον πίνακα εισόδου που έχουν σταθερή τιμή, άρα δεν προσφέρουν καμία πληροφορία.
- fixunknowns
  - Εάν οι τιμές που λείπουν είναι σημειωμένες ως NaN, τις αντικαθιστά με
  - Χρησιμοποιείται μόνο στις εισόδους

(Matlab) fixunknowns (1/2)

```
>> [Y, Settings]=fixunknowns([ 1 3 2 NaN 7] )
Y =
    1.0000    3.0000    2.0000    3.2500    7.0000
    1.0000    1.0000    1.0000         0    1.0000
Settings =
         name: 'fixunknowns'
        xrows: 1
        yrows: 2
      unknown: 1
        known: []
        shift: 0
       xmeans: 3.2500
    no_change: 0
```

Αντικατάσταση με μέσο όρο γραμμής

Τα μηδενικά δηλώνουν τιμή που προέκυψε από αντικατάσταση NaN

(Matlab) fixunknowns (2/2)

- >> X_again=fixunknowns('reverse',Y,Settings)
- X_again =
-      1   3   2   NaN   7

- >> Z=fixunknowns('apply', [1 NaN 2],Settings)
- Z =
-     1.0000   3.2500   2.0000
-     1.0000   0   1.0000

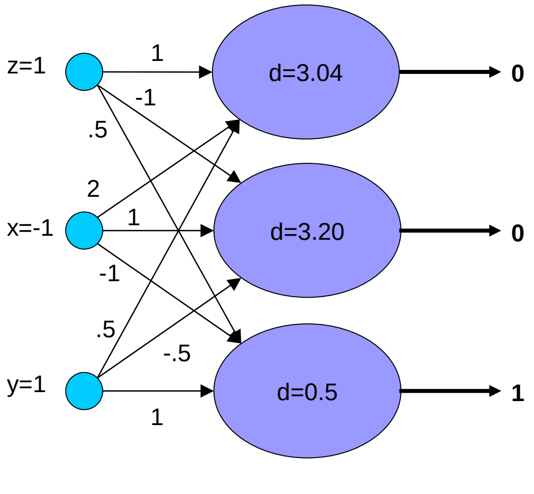
Ομαδοποίηση με Ανταγωνιστική Μάθηση

Δίκτυα Kohonen

Ανταγωνιστικοί Νευρώνες

- Ένα στρώμα με ανταγωνιστικούς νευρώνες λειτουργεί ως εξής:
  - Όλοι οι νευρώνες δέχονται το σήμα από τους νευρώνες του προηγούμενου επιπέδου (συνήθως το επίπεδο εισόδου).
  - Κάθε νευρώνας υπολογίζει την ευκλείδια απόσταση του διανύσματος εισόδου **X** από το διάνυσμα των βαρών των εισόδων του $\mathbf{W}_i$:
    - $d=||\mathbf{X}-\mathbf{W}_i||$
  - Ο νευρώνας με τη μικρότερη απόσταση παράγει έξοδο 1, ενώ όλοι οι υπόλοιποι παράγουν έξοδο 0.
  - Γενικά δεν υπάρχει επόμενο επίπεδο μετά το επίπεδο με ανταγωνιστικούς νευρώνες.

Παράδειγμα



Ανταγωνιστική Μάθηση

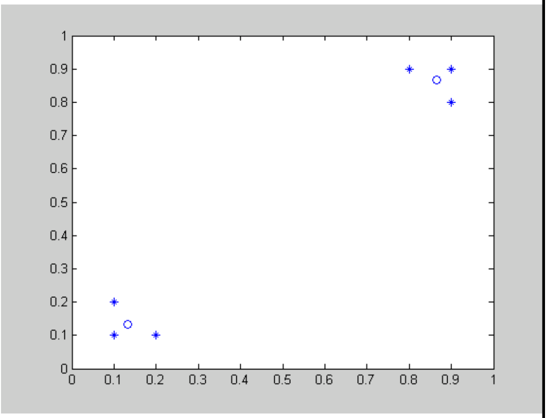
- Η μάθηση σε ένα επίπεδο ανταγωνιστικών νευρώνων γίνεται χωρίς επίβλεψη.
- Τα παραδείγματα εκπαίδευσης (μόνο είσοδοι) παρουσιάζονται στο δίκτυο διαδοχικά (και με τυχαία σειρά).
- Για κάθε παράδειγμα εκπαίδευσης βρίσκεται ο νευρώνας i με την μικρότερη ευκλείδια απόσταση d=||**Wi**-**X**|| και τα βάρη του αλλάζουν ώστε να πλησιάσουν περισσότερο το τρέχον παράδειγμα, σύμφωνα με τη σχέση:
  - **Wi'**=**Wi**+a(**X**-**Wi**) **ΚΑΝΟΝΑΣ ΜΑΘΗΣΗΣ KOHONEN**
- όπου:
  - i: Ο νευρώνας που κέρδισε.
  - **Wi**: Το προηγούμενο διάνυσμα βαρών του νευρώνα i.
  - **Wi'**: Το νέο διάνυσμα βαρών του νευρώνα i.
  - **X**: Το τρέχον διάνυσμα εισόδου
  - a: Ο ρυθμός εκπαίδευσης
- Τα βάρη των υπολοίπων νευρώνων παραμένουν αμετάβλητα.

Παράδειγμα (1/3)

- Έστω ότι έχουμε ένα δίκτυο με δύο εισόδους και 6 διανύσματα εκπαίδευσης:
  - p=[.1 .8 .1 .2 .9 .9
    .2 .9 .1 .1 .8 .9]
- Τα διανύσματα εισόδου ομαδοποιούνται σε δύο ομάδες:
  - Αυτά που είναι κοντά στο (0,0)
  - Αυτά που είναι κοντά στο (1,1)
- Κατασκευάζουμε ένα νευρωνικό δίκτυο ενός επίπεδου με δύο ανταγωνιστικούς νευρώνες.
- Μετά από την εκπαίδευση τα βάρη των δύο νευρώνων έχουν γίνει:
  - 1ος νευρώνας: **W1**=[0.8660 0.8660]'
  - 2ος νευρώνας: **W2**=[0.1325 0.1337]'

Παράδειγμα (2/3)

- Στο διπλανό παράδειγμα με αστεράκι φαίνονται τα 6 παραδείγματα εκπαίδευσης και με κύκλο φαίνονται τα βάρη των δύο νευρώνων.



Παράδειγμα (3/3)

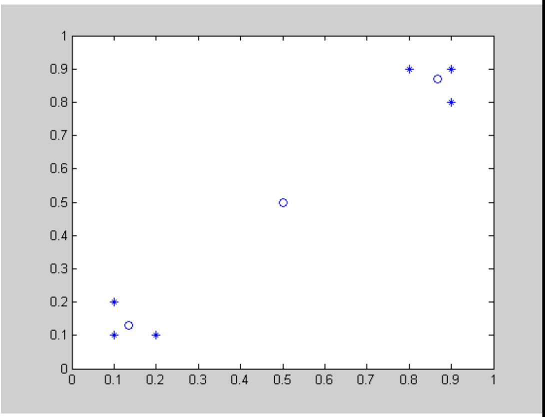
- Είδαμε στο παράδειγμα ότι είχαμε δύο ομάδες (clusters) παραδειγμάτων εκπαίδευσης.
- Με αυτό το σκεπτικό επιλέξαμε ένα δίκτυο με 2 νευρώνες.
- Θεωρώντας τα βάρη των νευρώνων ως διανύσματα στο χώρο των διανυσμάτων εκπαίδευσης (στην προκειμένη περίπτωση ο χώρος δύο διαστάσεων), κάθε ένα από τα διανύσματα βαρών μετατοπίσθηκε στο "κέντρο βάρους" κάθε μιας από τις ομάδες παραδειγμάτων.
- Εάν εκτελέσουμε ξανά την εκπαίδευση, ενδέχεται κάθε νευρώνας να "μάθει" την άλλη ομάδα από αυτή που έμαθε την πρώτη φορά.

Βάρη τάσης πόλωσης (1/4)

- Ένα πρόβλημα που εμφανίζουν τα δίκτυα ανταγωνιστικής μάθησης είναι ότι κάποιοι νευρώνες ενδέχεται να μην κερδίσουν ποτέ! (ειδικά εάν έχουμε πολλούς νευρώνες).
  - Νεκροί νευρώνες (dead neurons)
- Οι νευρώνες αυτοί θα κρατήσουν τα αρχικά τους βάρη (τα οποία τους έχουν αποδοθεί τυχαία).

Βάρη τάσης πόλωσης (2/4)

- Στο διπλανό παράδειγμα χρησιμοποιήθηκε ένα ανταγωνιστικό δίκτυο με τρεις νευρώνες.
- Ο τρίτος νευρώνας κράτησε τα αρχικά του βάρη, μη έχοντας κερδίσει καμία φορά.

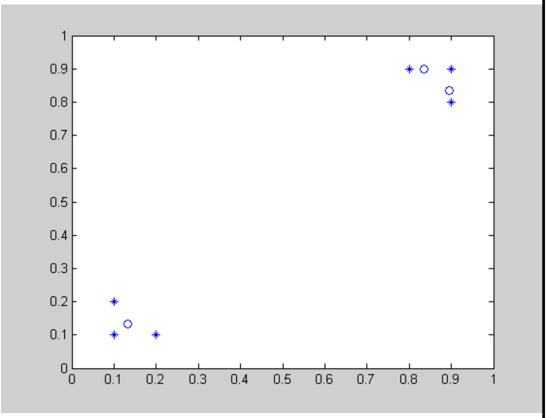


Βάρη τάσης πόλωσης (3/4)

- Για να αποφύγουμε νεκρούς νευρώνες χρησιμοποιούμε τα βάρη εισόδου της σταθερής τάσης πόλωσης (bias).
- Η τάση πόλωση έχει τιμή 1 για όλους τους νευρώνες (στο Matlab).
- Τα αρχικά βάρη της τάσης πόλωσης παίρνουν μια μεγάλη τιμή, ίδια για όλους τους νευρώνες, δημιουργώντας έτσι μια μεγάλη απόσταση από την τιμή της τάσης πόλωσης (1).
- Η αλλαγή των βαρών της τάσης πόλωσης γίνεται με τέτοιον τρόπο, ώστε να δίνονται μικρότερα βάρη στην τάση πόλωσης στους νευρώνες που δεν κερδίζουν συχνά.

Βάρη τάσης πόλωσης (4/4)

- Στο διπλανό παράδειγμα επιτρέψαμε την αλλαγή των βαρών της τάσης πόλωσης.
- Οι δύο νευρώνες "μοιραστήκαν" τη μία ομάδα παραδειγμάτων.
  - Συντελεστής conscience learning rate=0.001



(Matlab)

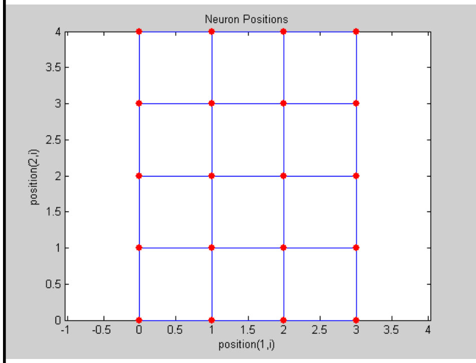
- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: Competitive
  - Kohonen learning rate: Ο ρυθμός μάθησης
  - Conscience learning rate: Παράμετρος που επηρεάζει το βαθμό προσαρμογής των βαρών για την τάση πόλωσης (τιμή 0 ισοδυναμεί με καμία προσαρμογή).
- Κατά την εκπαίδευση δεν χρειάζεται να δώσουμε Targets, παρά μόνο Inputs.
- Τα ανταγωνιστικά δίκτυα στο Matlab έχουν ένα μόνο επίπεδο.

Δίκτυα Αυτοοργάνωσης
(Self-Organizing Feature Maps, SOFM)

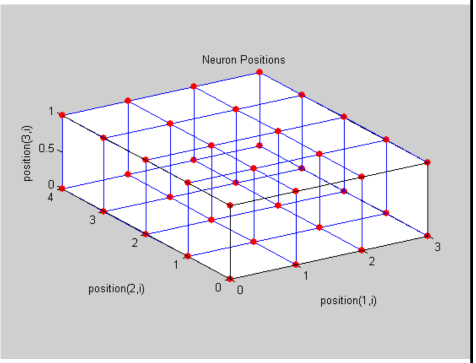
- Τα δίκτυα αυτοοργάνωσης είναι μια παραλλαγή των ανταγωνιστικών δικτύων.
- Οι νευρώνες του ανταγωνιστικού επιπέδου θεωρείται ότι σχηματίζουν ένα νοητό πλέγμα.
- Συνήθη πλέγματα:
  - Ορθογώνιο
  - Εξαγωνικό
  - Τυχαίο
- Για κάθε παράδειγμα εκπαίδευσης, εκτός από τα βάρη του νευρώνα-νικητή, αλλάζουν τα βάρη και για τους νευρώνες της "γειτονιάς" στο πλέγμα του νευρώνα-νικητή.
- Έτσι τα δίκτυα αυτοοργάνωσης καταφέρνουν να μάθουν, εκτός από τις ομάδες, και την τοπολογία αυτών, δηλαδή ποιες ομάδες είναι πιο πολυπληθείς.

Ορθογώνια πλέγματα

- Πλέγμα 4x5



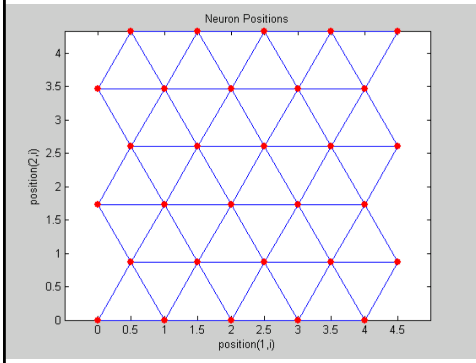
- Πλέγμα 4x5x2



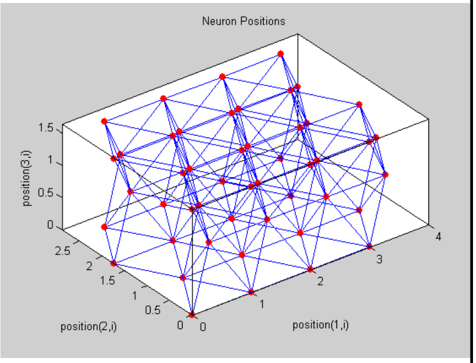
Μπορούν να υπάρξουν και πλέγματα περισσοτέρων των τριών διαστάσεων, αλλά δεν μπορούν να απεικονιστούν γραφικά.

Εξαγωνικά πλέγματα

- Πλέγμα 5x6

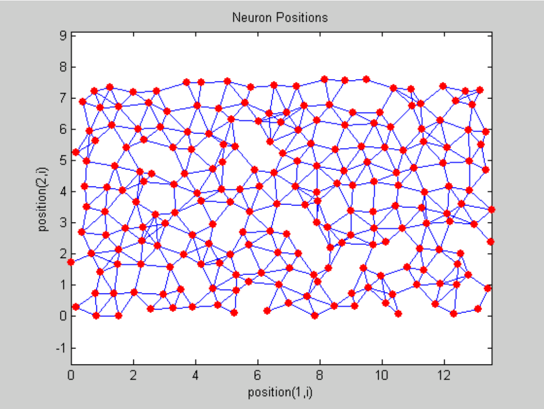


- Πλέγμα 4x4x3



Τυχαία πλέγματα

- Τυχαίο πλέγμα 18x12



Λογική λειτουργίας (1/2)

- Τα πλέγματα είναι "νοητά", δηλαδή δεν υπάρχει σύνδεση των νευρώνων του ανταγωνιστικού δικτύου με οποιονδήποτε τρόπο πάνω στο δίκτυο.
- Αρχικά τα βάρη για κάθε νευρώνα στο δίκτυο είναι τυχαία, ενώ τα βάρη "γειτονικών" νευρώνων στο πλέγμα μπορεί να είναι εντελώς άσχετα μεταξύ τους.
- Κατά την εκπαίδευση, για κάθε παράδειγμα επιλέγεται ο νευρώνας εκείνος τα βάρη του οποίου βρίσκονται κοντινότερα στο παράδειγμα.
  - Τα βάρη του νευρώνα αυτού αλλάζουν, σύμφωνα με τον κανόνα μάθησης Kohonen.
- Επιπλέον όμως επιλέγονται και οι γειτονικοί του νευρώνες στο πλέγμα, τα βάρη των οποίων επίσης αλλάζουν, χρησιμοποιώντας όμως το μισό του ρυθμού μάθησης (κανόνας μάθησης Kohonen).

Λογική λειτουργίας (2/2)

- Ως "γειτονιά" ενός νευρώνα θεωρούνται όλοι οι "γειτονικοί" του νευρώνες που βρίσκονται πάνω στο πλέγμα σε απόσταση το πολύ d από αυτόν.
  - Υπάρχουν διάφοροι τρόποι ορισμού της απόστασης (θα τους δούμε παρακάτω)
- Η απόσταση d που ορίζει το εύρος κάθε γειτονιάς ξεκινά συνήθως από μια μεγάλη τιμή και προοδευτικά μειώνεται.
- Η εμπειρία έχει δείξει ότι στο τέλος τα διανύσματα βαρών των νευρώνων έχουν ισοκατανεμηθεί μεταξύ των παραδειγμάτων εκπαίδευσης, καλύπτοντας τον χώρο κατά τρόπο ομοιόμορφο προς αυτόν των διανυσμάτων εκπαίδευσης.
- Επιπλέον, νευρώνες γειτονικοί στο πλέγμα έχουν "γειτονικά" διανύσματα βαρών, δημιουργώντας έτσι μια τοπολογία αντίστοιχη αυτής των διανυσμάτων εκπαίδευσης.

Συντεταγμένες νευρώνων

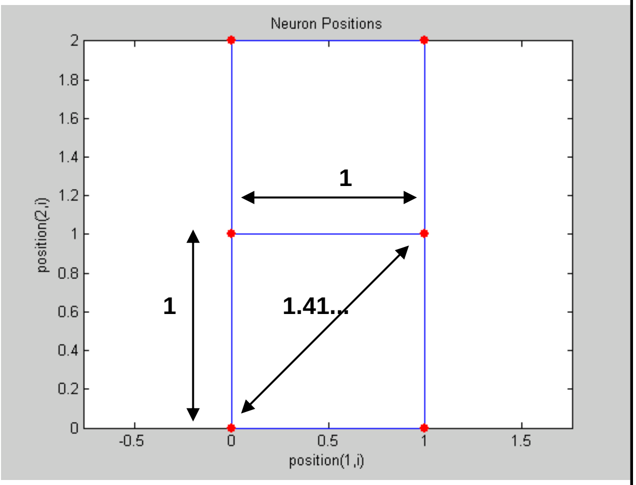
- Κάθε νευρώνας στο πλέγμα έχει "συντεταγμένες", οι οποίες αντιστοιχούν στους αύξοντες αριθμούς του στις διάφορες διαστάσεις του πλέγματος.
  - Π.χ., για ορθογώνιο πλέγμα δύο διαστάσεων υπάρχουν οι νευρώνες (1,1), (1,2), (4,3) κλπ.
- Οι συντεταγμένες αυτές δεν αλλάζουν (δεν έχουν δηλαδή σχέση με την εκπαίδευση του δικτύου).

Αποστάσεις μεταξύ νευρώνων

- Οι αποστάσεις των νευρώνων πάνω στο πλέγμα έχουν να κάνουν με τις συντεταγμένες τους.
- Υπάρχουν τέσσερις συναρτήσεις για τη μέτρηση των αποστάσεων των νευρώνων.
  - Ευκλείδια απόσταση (Matlab: dist)
  - Απόσταση κουτιού (Matlab: boxdist)
  - Απόσταση Manhatan (Matlab: mandist)
  - Απόσταση δεσμών (Matlab: linkdist)

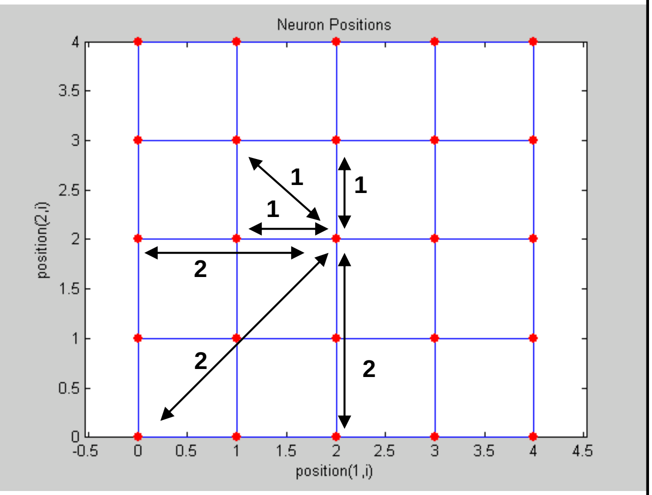
Ευκλείδια απόσταση

- Η γνωστή ευκλείδια απόσταση...



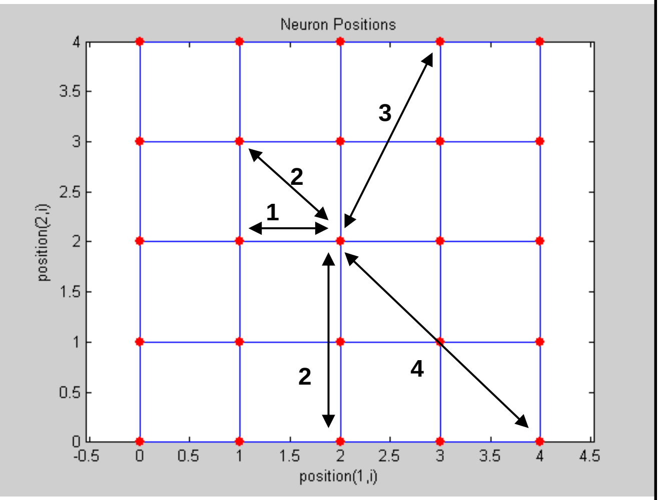
Απόσταση κουτιού

- Η απόσταση κουτιού επιστρέφει τη μέγιστη απόσταση των δύο νευρώνων στις διάφορες διαστάσεις.



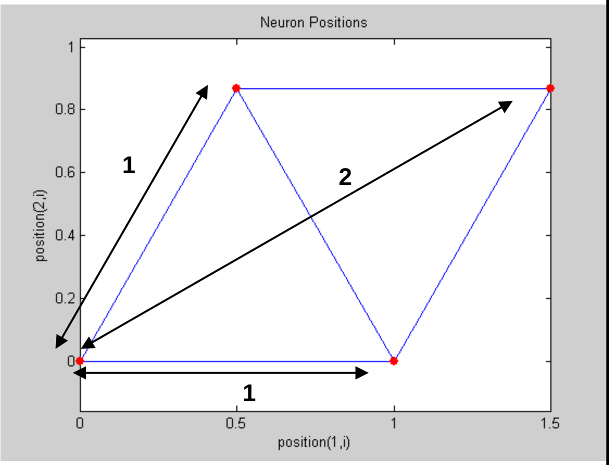
Απόσταση Manhatan

- Η απόσταση Mahnatan επιστρέφει το άθροισμα των απολύτων τιμών των αποστάσεων των δύο νευρώνων στις διάφορες διαστάσεις.



Απόσταση δεσμών

- Η απόσταση δεσμών μετρά το πλήθος των δεσμών που πρέπει να ακολουθήσουμε για να φθάσουμε από τον ένα νευρώνα στον άλλο.
- Σε ορθογώνια πλέγματα η απόσταση δεσμών ταυτίζεται με την απόσταση Mahnatan.



Εκπαίδευση δικτύων αυτοοργάνωσης

- Η εκπαίδευση των δικτύων αυτοοργάνωσης γίνεται σε δύο φάσεις:
  - Φάση διάταξης (ordering phase)
  - Φάση συντονισμού (tuning phase)
- Στη φάση διάταξης τα διανύσματα βαρών των νευρώνων αλλάζουν έντονα, ώστε γειτονικοί νευρώνες να αποκτήσουν "γειτονικά" διανύσματα βαρών.
- Στη φάση συντονισμού τα διανύσματα βαρών μεταβάλλονται αργά, ώστε να πάρουν προσαρμοστούν με τον καλύτερο τρόπο ανάμεσα στα παραδείγματα εκπαίδευσης.

Φάση διάταξης

- Η φάση διάταξης ξεκινά με τη "διάμετρο" της γειτονιάς ίση με τη μέγιστη απόσταση μεταξύ των νευρώνων στο πλέγμα.
- Η διάμετρος της γειτονιάς μειώνεται σταδιακά μέχρι την τιμή που θα έχει στην επόμενη φάση (Matlab: Neighborhood distance).
- Ο ρυθμός μάθησης ξεκινά από μία πολύ υψηλή τιμή (Matlab: Ordering phase learning rate) και πέφτει σταδιακά μέχρι την τιμή που θα έχει στην επόμενη φάση (Matlab: Tuning phase learning rate).
- Η φάση διάταξης διαρκεί ένα προκαθορισμένο πλήθος εποχών (Matlab: Ordering phase steps = 1000).

Φάση συντονισμού

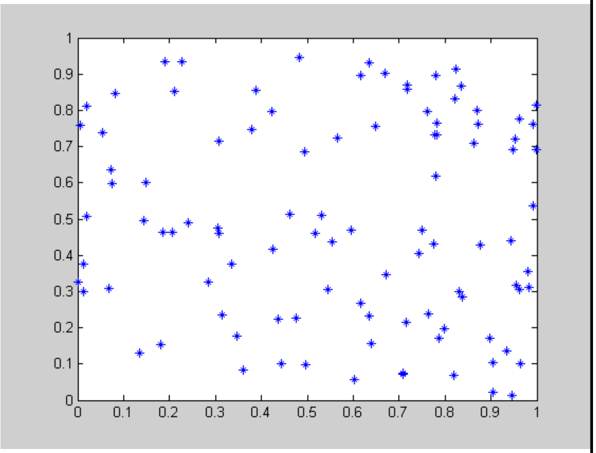
- Στη φάση συντονισμού η διάμετρος της γειτονιάς παραμένει σταθερή (Matlab: Neighborhood distance=1).
- Ο ρυθμός μάθησης μειώνεται πολύ αργά (Matlab: Tuning phase learning distance=0.02).
- Ο αριθμός εποχών εκπαίδευσης σε αυτή τη φάση πρέπει να είναι πολύ μεγαλύτερος από τις εποχές εκπαίδευσης της προηγούμενης φάσης.

(Matlab)

- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: Self Organizing Map
- Άλλες παράμετροι:
  - Dimensions of map
  - Topology function
  - Distance function
  - Ordering phase learning rate
  - Ordering phase steps
  - Tuning phase learning rate
  - Neighborhood distance
- Στα Self-Organizing maps δεν παίζει ρόλο (και δεν χρειάζεται) η τάση πόλωσης (αν και το Matlab την εμφανίζει σαν να υπάρχει στο παράθυρο View).

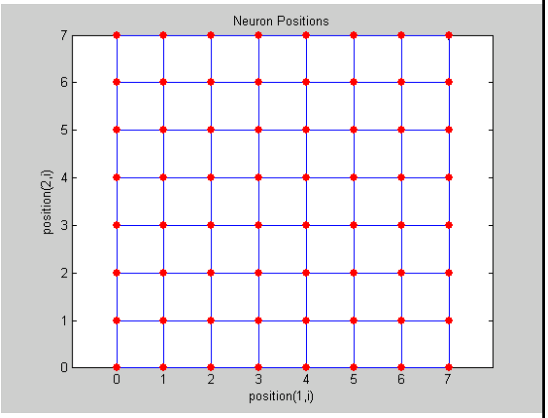
Παράδειγμα ομοιόμορφης κατανομής (1/6)

- Έστω σε χώρο δύο διαστάσεων η διπλανή ομοιόμορφη κατανομή 100 παραδειγμάτων:



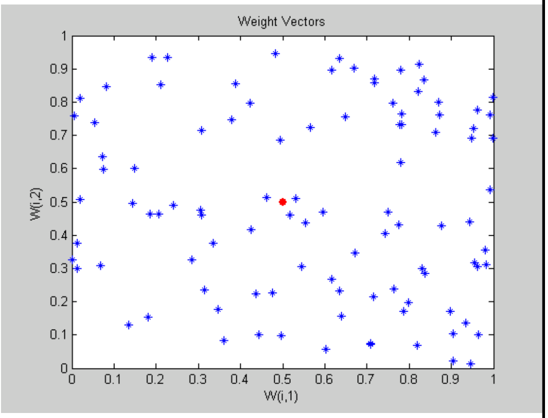
Παράδειγμα ομοιόμορφης κατανομής (2/6)

- Χρησιμοποιούμε ένα δίκτυο αυτοοργάνωσης με πλέγμα 8x8.



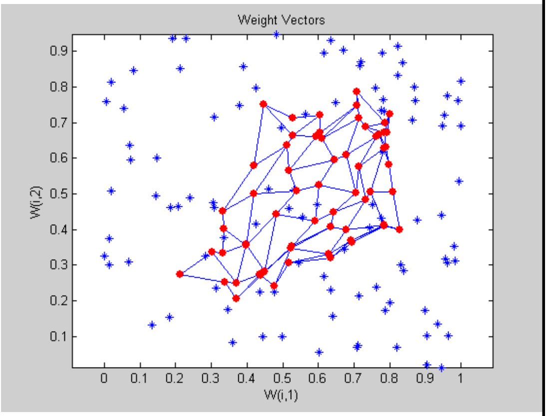
Παράδειγμα ομοιόμορφης κατανομής (3/6)

- Τα διανύσματα βαρών των νευρώνων πριν την έναρξη της εκπαίδευσης έχουν όλα την ίδια τιμή που είναι ίση με το μέσο όρο των διανυσμάτων εκπαίδευσης.



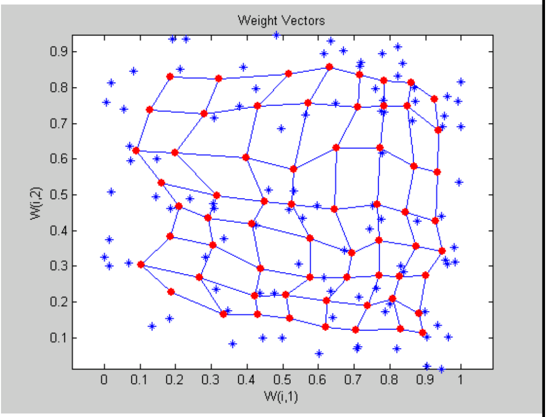
Παράδειγμα ομοιόμορφης κατανομής (4/6)

- Τα διανύσματα βαρών των νευρώνων μετά από 25 εποχές εκπαίδευσης στη φάση της διάταξης.



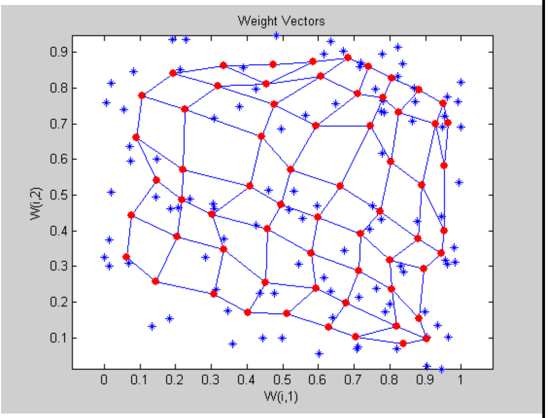
Παράδειγμα ομοιόμορφης κατανομής (5/6)

- Τα διανύσματα των βαρών των νευρώνων μετά από 300 εποχές εκπαίδευσης στη φάση της διάταξης και 100 εποχές εκπαίδευσης στη φάση του συντονισμού.



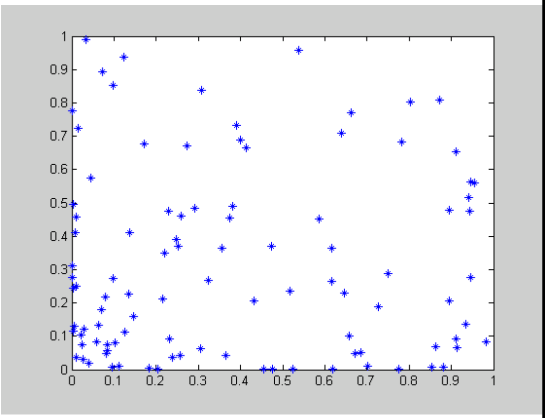
Παράδειγμα ομοιόμορφης κατανομής (6/6)

- Τα διανύσματα των βαρών των νευρώνων μετά από 300 εποχές εκπαίδευσης στη φάση της διάταξης και 1000 εποχές εκπαίδευσης στη φάση του συντονισμού.



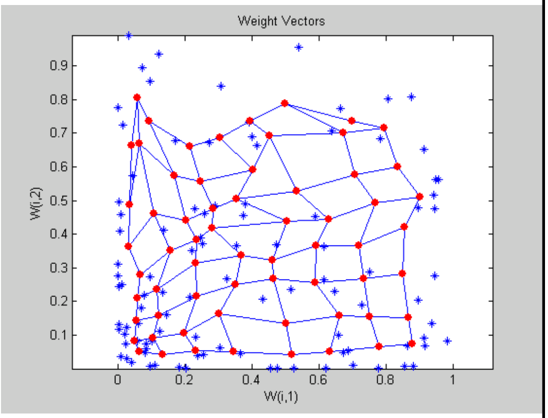
Παράδειγμα μη-ομοιόμορφης κατανομής (1/2)

- Έστω σε χώρο δύο διαστάσεων η διπλανή μη-ομοιόμορφη κατανομή 100 παραδειγμάτων, τα οποία είναι συγκεντρωμένα περισσότερο στην αρχή των αξόνων:



Παράδειγμα μη-ομοιόμορφης κατανομής (2/2)

- Χρησιμοποιήθηκε και πάλι ένα δίκτυο αυτοοργάνωσης 8x8.
- Στο διπλανό σχήμα φαίνονται τα διανύσματα των βαρών των νευρώνων μετά από 300 εποχές εκπαίδευσης στη φάση της διάταξης και 100 εποχές εκπαίδευσης στη φάση του συντονισμού.
- Παρατηρείται μια συγκέντρωση διανυσμάτων στην αρχή των αξόνων.



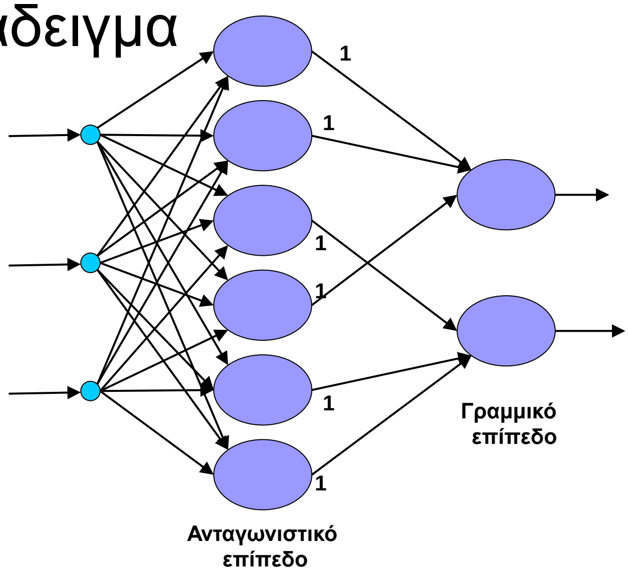
Επιβλεπόμενη κατηγοριοποίηση με ανταγωνιστικά δίκτυα (1/2)

- Τα ανταγωνιστικά δίκτυα μπορούν να χρησιμοποιηθούν και για επιβλεπόμενη κατηγοριοποίηση.
- Τα αντίστοιχα δίκτυα ονομάζονται Learning Vector Quantization (LVQ) ή Counter-propagation networks (CPN).
- Αποτελούνται από ένα (κρυφό) επίπεδο ανταγωνιστικών νευρώνων (χωρίς πλέγμα) και ένα δεύτερο επίπεδο με γραμμικούς νευρώνες (επίπεδο εξόδου).
- Οι νευρώνες του γραμμικού επιπέδου είναι σημαντικά λιγότεροι αυτών του ανταγωνιστικού επιπέδου και ίσοι με τον αριθμό των προκαθορισμένων κατηγοριών.

Επιβλεπόμενη κατηγοριοποίηση με ανταγωνιστικά δίκτυα (2/2)

- Τα διανύσματα εισόδου κατηγοριοποιούνται δύο φορές.
  - Το ανταγωνιστικό επίπεδο τα κατανέμει σε υπο-κατηγορίες.
  - Οι υπο-κατηγορίες του ανταγωνιστικού επιπέδου αντιστοιχίζονται στις κατηγορίες του γραμμικού επιπέδου.
- Κάθε νευρώνας του κρυφού επιπέδου ενεργοποιεί έναν και μόνο έναν νευρώνα του επιπέδου εξόδου (βάρος σύνδεσης προς αυτόν τον νευρώνα=1, προς τους υπόλοιπους νευρώνες=0).
- Η αντιστοίχιση των νευρώνων του κρυφού επιπέδου στους νευρώνες του επιπέδου εξόδου είναι σταθερή, καθορίζεται από την κατασκευή του δικτύου LVQ και δεν αλλάζει κατά τη διάρκεια της εκπαίδευσης.

Παράδειγμα



Μάθηση στα δίκτυα LVQ (1/2)

- Αρχικά καθορίζεται μια αντιστοίχιση των νευρώνων του κρυφού επιπέδου στους νευρώνες εξόδου, βάσει ποσοστών που δίνονται από τον χρήστη.
  - Π.χ., 50% των νευρώνων του κρυφού επιπέδου θα ενεργοποιούν τον πρώτο νευρώνα εξόδου και 50% θα ενεργοποιούν τον δεύτερο νευρώνα εξόδου.
- Τα ποσοστά πρέπει να ανταποκρίνονται στην κατανομή των παραδειγμάτων στις προκαθορισμένες κατηγορίες.
- Στη συνέχεια ξεκινά η εκπαίδευση, η οποία σκοπό έχει να αντιστοιχίσει τα παραδείγματα εκπαίδευσης στους "σωστούς" νευρώνες του κρυφού επιπέδου.
  - Για κάθε παράδειγμα εκπαίδευσης υπάρχουν πολλοί νευρώνες του κρυφού επιπέδου στους οποίους μπορεί να αντιστοιχηθεί.
  - Οι νευρώνες αυτοί ανταγωνίζονται μεταξύ τους.

Μάθηση στα δίκτυα LVQ (2/2)

- Για κάθε παράδειγμα εκπαίδευσης βρίσκεται ο νευρώνας-νικητής μεταξύ όλων των νευρώνων.
- Εάν ο νευρώνας-νικητής αντιστοιχίζει το παράδειγμα στη σωστή κατηγορία εξόδου, τα βάρη εισόδου του νευρώνα αλλάζουν ώστε να πλησιάσουν το παράδειγμα.
  - $\mathbf{W}_i' = \mathbf{W}_i + a(\mathbf{X} - \mathbf{W}_i)$ (δες και διαφάνεια 112)
- Εάν ο νευρώνας-νικητής αντιστοιχίζει το παράδειγμα σε λάθος κατηγορία εξόδου, τα βάρη εισόδου του νευρώνα αλλάζουν ώστε να απομακρυνθούν από το παράδειγμα.
  - $\mathbf{W}_i' = \mathbf{W}_i - a(\mathbf{X} - \mathbf{W}_i)$

Κανόνας μάθησης LVQ2

- Επέκταση του κανόνα LVQ1:
  - Εκτός από τον κοντινότερο νευρώνα, λαμβάνουμε υπόψη και τον δεύτερο κοντινότερο νευρώνα, εφόσον:
    - Ο ένας από τους δυο ανήκει στη σωστή κατηγορία και ο άλλος σε λάθος κατηγορία.
    - Το παράδειγμα βρίσκεται σε μια περιοχή κοντά στο μεσο-επίπεδο των δυο νευρώνων:

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s \qquad s \equiv \frac{1-w}{1+w}$$

    - όπου d_i, d_j, οι αποστάσεις του παραδείγματος από τους δύο νευρώνες.

(Matlab)

- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: LVQ
- Άλλες παράμετροι:
  - Number of hidden neurons: Αριθμός νευρώνων του ανταγωνιστικού επιπέδου
  - Output class percentages: Ποσοστιαία αντιστοίχιση των νευρώνων του ανταγωνιστικού επιπέδου στους νευρώνες εξόδου (από εδώ προκύπτει και ο αριθμός των κατηγοριών εξόδου).
  - Learning rate
  - Learning function = LEARNLV1
- Στα δίκτυα LVQ δεν παίζει ρόλο (και δεν χρειάζεται) η τάση πόλωσης (αν και το Matlab την εμφανίζει σαν να υπάρχει στο παράθυρο View).

Αναδρομικά Δίκτυα

Recurrent Networks

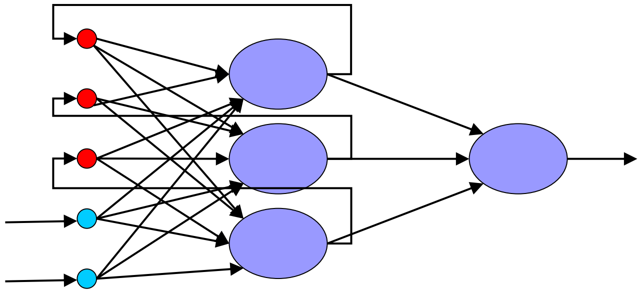
Γενικά

- Ένα νευρωνικό δίκτυο λέγεται αναδρομικό, εάν υπάρχει έστω και μια σύνδεση από έναν νευρώνα επιπέδου i προς έναν νευρώνα επιπέδου j, όπου j≤i.
- Οι αναδρομικές συνδέσεις καθιστούν τα δίκτυα αυτά ικανά να αναγνωρίζουν χρονικές ή και χωρικές συσχετίσεις μεταξύ των δεδομένων, χωρίς να χρειάζεται τα δεδομένα που αναφέρονται σε διαφορετικές χρονικές στιγμές ή χωρικές θέσεις να εμφανιστούν στο δίκτυο ταυτόχρονα.
- Η ύπαρξη αναδρομικών συνδέσεων δημιουργεί την ανάγκη συγχρονισμού στο δίκτυο, όσον αφορά τη μετάδοση των σημάτων. Έτσι εισάγονται "καθυστερήσεις" (delays) στις αναδρομικές συνδέσεις.
- Θα δούμε δύο περιπτώσεις τέτοιων δικτύων:
  - Δίκτυα Elman
  - Δίκτυα Hopfield

Δίκτυο Elman

- Το δίκτυο Elman αποτελείται από δύο επίπεδα νευρώνων.
  - Το κρυφό επίπεδο, το οποίο συνήθως έχει νευρώνες με σιγμοειδείς συναρτήσεις ενεργοποίησης -1/1 (tansig).
  - Το επίπεδο εξόδου, το οποίο συνήθως έχει γραμμικούς νευρώνες.
- Υπάρχει αναδρομική σύνδεση από τις εξόδους των νευρώνων του κρυφού επιπέδου στις εισόδους των ίδιων νευρώνων.
- Το κρυφό επίπεδο έχει συνήθως πολλούς νευρώνες.

Παράδειγμα: Δίκτυο Elman



Λειτουργία δικτύου Elman

- Η έξοδος των νευρώνων του κρυφού επιπέδου παρουσιάζεται ως είσοδος στους ίδιους νευρώνες την επόμενη χρονική στιγμή.
- Άρα η έξοδος του δικτύου Elman εξαρτάται τόσο από την τρέχουσα είσοδό του, όσο και από τις προηγούμενες εισόδους του (μάλιστα όχι μόνο από την αμέσως προηγούμενη).
- Είναι λοιπόν αναμενόμενο για δεδομένο διάνυσμα εισόδου να προκύψει σε δύο διαφορετικές δοκιμές διαφορετική έξοδος, εάν τα προηγούμενα διανύσματα που παρουσιάστηκαν στην είσοδο δεν ήταν ίδια.

Εκπαίδευση του δικτύου Elman

- Με δεδομένο ότι η απόκριση του δικτύου Elman εξαρτάται όχι μόνο από την τρέχουσα είσοδο, αλλά και από τις προηγούμενες, κατά την εκπαίδευση πρέπει τα παραδείγματα μάθησης να παρουσιάζονται πάντα με τη "σωστή" σειρά.
- Η εκπαίδευση του δικτύου Elman γίνεται με τη μέθοδο της οπισθοδιάδοσης του σφάλματος, όπου όμως αγνοείται η συνεισφορά στο σφάλμα της εσφαλμένης εισόδου από τις αναδρομικές συνδέσεις.
  - Αυτή η "παράβλεψη" έχει ως αποτέλεσμα να απαιτούνται περισσότερες εποχές εκπαίδευσης για τα δίκτυα Elman, από ό,τι για αντίστοιχα μη-αναδρομικά back-propagation δίκτυα.
- Προτεινόμενη μέθοδος εκπαίδευσης: TRAINBFG ή TRAINGDX

Παράδειγμα (1/3)

- Έστω:
  - p= [1 0 1 1 1 0 1 1]
  - t = [0 0 0 1 1 0 0 1]
- οι είσοδοι και οι έξοδοι σε ένα δίκτυο Elman.
- Η έξοδος είναι 1 όταν η τρέχουσα και η προηγούμενη είσοδος ήταν 1.
- Κατασκευάζουμε ένα δίκτυο Elman με μία είσοδο, 5 νευρώνες στο κρυφό επίπεδο και έναν νευρώνα εξόδου.

Πίνακες κελιών (1/3)

- Το Matlab υποστηρίζει δύο είδη πινάκων, τους απλούς πίνακες και τους πίνακες κελιών (cell arrays).
- Τα στοιχεία ενός απλού πίνακα είναι απλοί αριθμοί.
  - Για παράδειγμα ένας πίνακας 3x3 είναι ένας πίνακας με 9 απλά στοιχεία (αριθμούς).
- Τα στοιχεία ενός πίνακα κελιών είναι άλλοι πίνακες (απλοί ή και πίνακες κελιών), διαφορετικών ενδεχομένως διαστάσεων.
  - Για παράδειγμα, ένας πίνακας κελιών 3x3 συνήθως περιέχει 9 άλλους πίνακες, διαφόρων διαστάσεων.

Πίνακες κελιών (2/3)

- Για τους πίνακες κελιών χρησιμοποιούμε άγκιστρα αντί για αγκύλες, ενώ το Matlab εμφανίζει τα απλά στοιχεία τους ως πίνακες.
  - >> x={1 2 3}
    - x = [1]   [2]   [3]
  - >> x{1}
    - ans = 1
- Ουσιαστικά οι πίνακες κελιών σε μια κλασσική γλώσσα προγραμματισμού θα αντιστοιχούσαν σε πίνακες δεικτών.

Πίνακες κελιών (3/3)

- Το Neural Network Toolbox του Matlab χρησιμοποιεί τα δύο είδη πινάκων με διαφορετικό τρόπο.
  - Τους απλούς πίνακες τους ονομάζει ταυτόχρονους (concurrent) και θεωρεί ότι τα στοιχεία τους (παραδείγματα εκπαίδευσης) μπορούν να παρουσιασθούν στο δίκτυο με οποιαδήποτε σειρά.
  - Τους πίνακες κελιών τους ονομάζει σειριακούς πίνακες (sequential) και θεωρεί ότι η σειρά των στοιχεία στον πίνακα είναι η χρονική σειρά με την οποία πρέπει να παρουσιαστούν τα δεδομένα στο δίκτυο.
- Για αναδρομικά δίκτυα τα δεδομένα εκπαίδευσης αλλά και προσομοίωσης πρέπει να παρέχονται σε πίνακες κελιών.
- Το Neural Network Toolbox παρέχει δύο συναρτήσεις για τη μετατροπή απλών πινάκων σε πίνακες κελιών και αντίστροφα: con2seq και seq2con.

Παράδειγμα (2/3)

- Μετατρέπουμε λοιπόν τους δύο απλούς πίνακες σε πίνακες κελιών:
  - >> p_sec=con2seq(p)
    - p_sec= [1] [0] [1] [1] [1] [0] [1] [1]
  - >> t_sec=con2seq(t)
    - t_sec= [0] [0] [0] [1] [1] [0] [0] [1]
- Εκπαιδεύουμε το δίκτυο Elman για 1000 εποχές εκπαίδευσης και με στόχο σφάλματος 0.001.

Παράδειγμα (3/3)

- Τα αποτελέσματα αποθηκεύονται στη μεταβλητή Elman_outputs, την οποία την κάνουμε export στο workspace του Matlab.
- Με τη συνάρτηση seq2con μετατρέπουμε τον πίνακα κελιών 1x8 Elman_outputs σε έναν πίνακα κελιών 1x1, του οποίου το μοναδικό στοιχείο είναι ένας κανονικός πίνακας 1x8:
  - t0=seq2con(Elman_outputs);
- Εμφανίζουμε τον πίνακα των αποτελεσμάτων της προσομοίωσης:
  - >> t0{1,1}
    - ans = 0.0008 0.0004 0.0001 0.9991 0.9994 0.0000 0.0006 0.9999
- Συγκρίνουμε τα αποτελέσματα αυτά με τον αρχικό πίνακα t:
  - >> t
    - t = 0 0 0 1 1 0 0 1
  - >> t-t0{1,1}
    - ans = 1.0e-003 *
      -0.8376 -0.4106 -0.0841 0.8574 0.6351 -0.0402 -0.6384 0.1295

(Matlab)

- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: Elman
- Όλες οι υπόλοιποι παράμετροι είναι ίδιες με τα απλά feed-forward backprop δίκτυα.
- Γενικά δεν επιλέγουμε μεθόδους μάθησης οι οποίες μεταβάλλουν τα βάρη με μεγάλα βήματα, όπως είναι οι TRAINLM, TRAINRP κλπ., λόγω της προσέγγισης με την οποία υπολογίζονται τα σφάλματα, άρα και η κατεύθυνση μεταβολής των βαρών.
  - Συνιστώμενες μέθοδοι: TRAINGDX, TRAINBFG

Παρατήρηση

- Το παράδειγμα που προηγήθηκε θα μπορούσε πολύ εύκολα να λυθεί από ένα μη-αναδρομικό δίκτυο οπισθοδιάδοσης λάθους <u>με δύο εισόδους</u>, εφόσον μετασχηματίζαμε τα παραδείγματα εισόδου έτσι ώστε κάθε παράδειγμα να έχει δύο συνεχόμενες τιμές της ακολουθίας εισόδου, ενώ η έξοδος να έχει την τιμή που αντιστοιχεί στη δεύτερη από τις δύο τιμές εισόδου.
- Κάτι τέτοιο όμως σε πιο πολύπλοκες καταστάσεις δεν είναι δυνατόν να συμβεί, γιατί δεν είναι γνωστή η εξάρτηση της τωρινής εξόδου από τις προηγούμενες εισόδους.
  - π.χ. Χρηματιστήριο

Συσχετιστικά δίκτυα
Associative networks

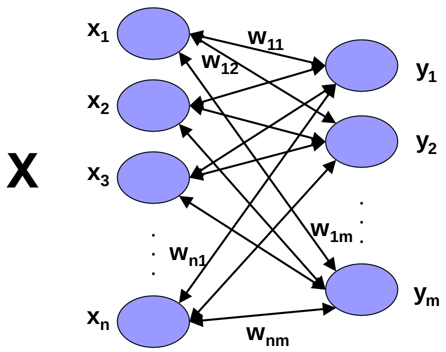
Δίκτυα Hopfield

Γενικά

- Τα συσχετιστικά δίκτυα (associative networks) ή μνήμες (memories) είναι αναδρομικά (feedback ή recurrent) νευρωνικά δίκτυα που έχουν την ικανότητα να θυμούνται ένα σύνολο από συσχετίσεις (X_k,Y_k).
- Έτσι, όταν στην είσοδο παρουσιάζεται το διάνυσμα X_k ή ένα "κοντινό" του διάνυσμα, στην έξοδο παρουσιάζεται το διάνυσμα Y_k.
- Τα συσχετιστικά δίκτυα λειτουργούν ως μνήμες, οι οποίες μάλιστα είναι ανθεκτικές στον θόρυβο.
- Επιπλέον, πρόκειται για μνήμες διπλής κατεύθυνσης, μιας και μπορούν να χρησιμοποιηθούν τα Y_k ως είσοδος και τα X_k ως έξοδος (Bidirectional Associative Memories, BAM).
- Τέλος, τα συσχετιστικά δίκτυα δεν χρειάζονται εκπαίδευση, αλλά τα βάρη τους μπορούν να υπολογιστούν με απλές μαθηματικές πράξεις από τα διανύσματα που πρέπει να θυμούνται.

Συνδεσμολογία

- Στη γενική τους μορφή αποτελούνται από δύο επίπεδα νευρώνων, οι οποίοι συνδέονται με αμφίδρομες συνδέσεις.



Υπολογισμός βαρών

- Έστω n οι νευρώνες εισόδου και m οι νευρώνες εξόδου.
- Έστω $\{(X_k,Y_k): k=1..K\}$ ένα σύνολο K ζευγών που πρέπει να "θυμάται" το δίκτυο.
- Κάθε βάρος w_{ij} από τον νευρώνα εισόδου i στον νευρώνα εξόδου j ισούται με το βάρος w_{ji} και δίνεται από τη σχέση (κανόνας του Hebb):

$$w_{ij} = \sum_{k=1}^{K} X_{k,i} Y_{k,j} = w_{ji}$$

- όπου $X_{k,i}$ το i στοιχείο του διανύσματος εισόδου X_k και $Y_{k,j}$ το j στοιχείο του αντίστοιχου διανύσματος εξόδου Y_k.
- Ισοδύναμα, εάν θεωρήσουμε ότι τα διανύσματα X_k και Y_k είναι διανύσματα στήλη, ο πίνακας των βαρών μπορεί να δοθεί από το παρακάτω γινόμενο πινάκων:

$$W = \sum_{k=1}^{K} X_k Y_k^T$$

Ενεργοποίηση νευρώνων

- Τα συσχετιστικά δίκτυα είναι **δίτιμα**, δηλαδή οι είσοδοι/έξοδοι των νευρώνων παίρνουν δύο μόνο τιμές, **έστω -1 και 1**.
- Η συνολική είσοδος ενός νευρώνα i, έστω S_i, ισούται με το γινόμενο των τιμών των εισόδων του επί τα βάρη αυτών:

$$S_i = \sum_j a_j w_{ji}$$

- όπου το j αναφέρεται σε άλλους νευρώνες, η έξοδος των οποίων αποτελεί είσοδο για τον νευρώνα i.
- Η έξοδος του νευρώνα i μπορεί να δοθεί από τον τύπο:

$$a_i = \Phi(S_i) = \begin{cases} +1, \text{αν } S_i > 0 \\ a_i, \text{αν } S_i = 0 \\ -1, \text{αν } S_i < 0 \end{cases}$$

Λειτουργία (1/2)

- Έστω ότι παρουσιάζεται στην είσοδο ένα παράδειγμα X_k.
- Οι τιμές του X_k θεωρούνται έξοδοι των νευρώνων εισόδου.
- Με βάση τον τύπο της διαφάνειας 169 υπολογίζονται οι έξοδοι των νευρώνων εξόδου.
- Οι έξοδοι των νευρώνων εξόδου γίνονται ξανά είσοδοι στους νευρώνες εισόδου (η αρχική είσοδος X_k πλέον αγνοείται).
- Με βάση τον τύπο της διαφάνειας 169 υπολογίζονται οι έξοδοι των νευρώνων εισόδου.
- Οι έξοδοι των νευρώνων εισόδου μεταβιβάζονται στους νευρώνες εξόδου και υπολογίζεται η έξοδός τους.
- κ.ο.κ.

Λειτουργία (2/2)

- Η διαδικασία που περιγράφηκε στην προηγούμενη διαφάνεια επαναλαμβάνεται μέχρις ότου οι νευρώνες ενός επιπέδου να εμφανίσουν σε δύο διαδοχικές επαναλήψεις την ίδια έξοδο.
- Στο σημείο αυτό η έξοδος των νευρώνων εξόδου αποτελεί την έξοδο του δικτύου.
- Εάν η αρχική είσοδος X_k ήταν ένα από τα αρχικά παραδείγματα (βάσει των οποίων υπολογίστηκαν τα βάρη του δικτύου), τότε αναμένεται να προκύψει στην έξοδο το αντίστοιχο παράδειγμα Y_k.
- Εάν η αρχική είσοδος δεν ταυτίζεται με κάποιο από τα αρχικά X_k, αλλά έστω X_k το "πλησιέστερο" προς αυτήν από τα αρχικά διανύσματα εισόδου, τότε αναμένεται ύστερα από μερικές επαναλήψεις η έξοδος να σταθεροποιηθεί στο διάνυσμα Y_k (και η έξοδος του επιπέδου εισόδου στο X_k).

Παράδειγμα (1/4)

- Έστω ότι θέλουμε να αντιστοιχίσουμε τα παρακάτω ζεύγη διανυσμάτων:
  - $X_1=[1, -1, -1, -1]^T \leftrightarrow Y_1=[1,1,1]^T$
  - $X_2=[-1, -1, -1, 1]^T \leftrightarrow Y_2=[1,-1,1]^T$
- Χρειαζόμαστε ένα δίκτυο με 4 νευρώνες εισόδου και 3 νευρώνες εξόδου.
- Πρέπει να υπολογίσουμε 12 βάρη, από τους νευρώνες εισόδου στους νευρώνες εξόδου.
  - Τα αντίστροφα βάρη, από τους νευρώνες εξόδου προς τους νευρώνες εισόδου, ταυτίζονται με τα αντίστοιχα βάρη από τους νευρώνες εισόδου προς τους νευρώνες εξόδου.

Παράδειγμα (2/4)

$$W = X_1 Y_1^T + X_2 Y_2^T = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & -2 \\ -2 & 0 & -2 \\ 0 & -2 & 0 \end{bmatrix}$$

Παράδειγμα (3/4)

- Έστω ότι παρουσιάζεται ως είσοδος το διάνυσμα X_1=[1, -1, -1, -1].
- Το διάνυσμα θεωρείται έξοδος των νευρώνων εισόδου και βάσει αυτού υπολογίζονται οι έξοδοι των νευρώνων εξόδου.
  - y_1=Φ(1*0 + -1*-2 + -1*-2 + -1*0)=Φ(4)=1
  - y_2=Φ(1*2 + -1*0 + -1*0 + -1*-2)=Φ(4)=1
  - y_3=Φ(1*0 + -1*-2 + -1*-2 + -1*0)=Φ(4)=1
- δηλαδή Y=[1,1,1], που είναι το αντίστοιχο διάνυσμα εξόδου.
- Εάν εφαρμόσουμε τις τιμές αυτές στο επίπεδο εισόδου, θα πάρουμε πάλι το διάνυσμα X_1, οπότε η διαδικασία θα σταματήσει.

Παράδειγμα (4/4)

- Έστω ότι παρουσιάζεται ως είσοδος το διάνυσμα X=[-1, -1, 1, 1], το οποίο είναι πιο κοντά στο X_2.
- Υπολογίζονται οι νέες έξοδοι των νευρώνων εξόδου
  - y_1=Φ(-1*0 + -1*-2 + 1*-2 + 1*0) =Φ(0)= 1 (προηγούμενη τιμή)
  - y_2=Φ(-1*2 + -1*0 + 1*0 + 1*-2) =Φ(-4)= -1
  - y_3=Φ(-1*0 + -1*-2 + 1*-2 + 1*0) =Φ(0)= 1 (προηγούμενη τιμή)
- δηλαδή το διάνυσμα Y_2=[1,-1,1].
- Οι τιμές Y εφαρμόζονται στους νευρώνες εισόδου.
  - x_1=Φ(1*0 + -1*2 + 1*0) = Φ(-2) = -1
  - x_2=Φ(1*-2 + -1*0 + 1*-2) = Φ(-4) = -1
  - x_3=Φ(1*-2 + -1*0 + 1*-2) = Φ(-4) = -1
  - x_4=Φ(1*0 + -1*-2 + 1*0) = Φ(2) = 1
- δηλαδή το διάνυσμα X_2=[-1, -1, -1, 1].
- Εφαρμόζοντας τις τιμές του X_2 στους νευρώνες εξόδου παράγεται και πάλι το Y_2, οπότε η διαδικασία σταματά και το Y_2 επιστρέφεται ως έξοδος.

Παρατήρηση

- Σε ένα συσχετιστικό δίκτυο, για κάθε ζεύγος διανυσμάτων X↔Y που μαθαίνει το δίκτυο, μαθαίνει επίσης και το ζεύγος των συμπληρωματικών τους διανυσμάτων, δηλαδή αυτών που προκύπτουν εάν αντικαταστήσουμε τα 1 με -1 και αντίστροφα.
- Έτσι στο προηγούμενο παράδειγμα το δίκτυο γνώριζε τα παρακάτω ζεύγη:
  - $X_1=[1, -1, -1, -1]^T \leftrightarrow Y_1=[1,1,1]^T$
  - $X_2=[-1, -1, -1, 1]^T \leftrightarrow Y_2=[1,-1,1]^T$
  - $X_3=[-1, 1, 1, 1]^T \leftrightarrow Y_3=[-1,-1,-1]^T$
  - $X_4=[1, 1, 1, -1]^T \leftrightarrow Y_4=[-1,1,-1]^T$

Χωρητικότητα συσχετιστικών δικτύων (1/2)

- Ένας εμπειρικός κανόνας λέει ότι σε ένα δίκτυο με N νευρώνες, ο μέγιστος αριθμός ζευγών που μπορούν να αποθηκευτούν είναι:

$$p_{\max} = \frac{N}{2\ln N}$$

- Στο προηγούμενο παράδειγμα είχαμε 7 νευρώνες, άρα μπορούσαμε να αποθηκεύσουμε περίπου 1.8 ζεύγη (αντί για 4 που αποθηκεύτηκαν, μαζί με τα συμπληρωματικά).
- Άρα το προηγούμενο δίκτυο μπορούσε να μην "συγκλίνει" σωστά, ακόμη και αν η αρχική είσοδος διέφερε κατά ένα μόνο στοιχείο από κάποιο παράδειγμα.
  - Αυτό είναι φανερό από το ότι μπορούν να βρεθούν είσοδοι που να διαφέρουν το ίδιο από 2 αρχικά παραδείγματα:
    - $X_1=[1, -1, -1, -1]^T$
    - $X_2=[-1, -1, -1, 1]^T$
    - $X=[-1, -1, -1, -1]$

Χωρητικότητα συσχετιστικών δικτύων (2/2)

- Σε προβληματικές περιπτώσεις σαν και αυτήν, πρέπει να αυξηθούν οι διαστάσεις των διανυσμάτων.
- Για παράδειγμα, εάν τα διανύσματα X παριστάνουν την εικόνα ενός γράμματος και τα Y τον κωδικό του γράμματος, θα πρέπει να αυξηθεί η διάσταση των διανυσμάτων X ψηφιοποιώντας τα γράμματα σε μεγαλύτερη ανάλυση.

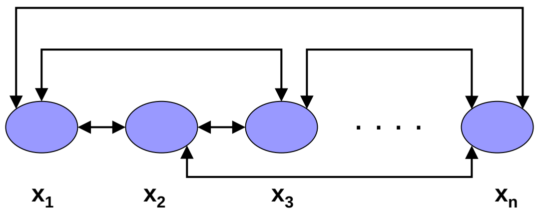
Αυτοσυσχετιστικά δίκτυα

- Ένα συσχετιστικό δίκτυο για το οποίο ισχύει $X_k=Y_k$ για όλα τα διανύσματα που θυμάται ονομάζεται αυτοσυσχετιστικό δίκτυο (autoassociative network).
- Σε ένα αυτοσυσχετιστικό δίκτυο τα επίπεδα εισόδου και εξόδου έχουν τον ίδιο αριθμό νευρώνων.
- Ο προγραμματισμός και η λειτουργία των αυτοσυσχετιστικών δικτύων δεν διαφέρει από ό,τι είδαμε στις προηγούμενες διαφάνειες.
  - Τα δίκτυα που είδαμε ως τώρα ονομάζονται ετεροσυσχετιστικά δίκτυα (heteroassociative networks) ή συσχετιστικές μνήμες διπλής κατεύθυνσης (Bidirectional Associative Memories, BAM).
- Τα αυτοσυσχετιστικά δίκτυα λειτουργούν ως φίλτρα για την αφαίρεση θορύβου από παραμορφωμένες εισόδους.

Δίκτυα Hopfield

- Τα δίκτυα Hopfield είναι μια ειδική περίπτωση αυτοσυσχετιστικών δικτύων.
- Έχουν ένα επίπεδο νευρώνων, με συνδέσεις μεταξύ όλων των νευρώνων.
  - Παραλείπονται οι συνδέσεις από κάθε νευρώνα στον εαυτό του.
- Κατά τη λειτουργία του δικτύου, οι τιμές εξόδου των νευρώνων δεν ανανεώνονται όλες μαζί, αλλά μία-μία, και μάλιστα με τυχαία σειρά.
  - Θα πρέπει να υπάρχει ισορροπία στη συχνότητα με την οποία ενημερώνονται οι έξοδοι των νευρώνων.

Συνδεσμολογία Hopfield



Παράδειγμα (1/5)

- Έστω ότι θέλουμε να αποθηκεύσουμε σε ένα δίκτυο Hopfield το ακόλουθο διάνυσμα:
  - X_1=[1 -1 1 1]
- Πρώτα υπολογίζουμε τα βάρη: $W = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix}$
- Επειδή όμως δεν υπάρχουν συνδέσεις από κάθε νευρώνα στον εαυτό του, τα διαγώνια στοιχεία του παραπάνω πίνακα μηδενίζονται:

$$W = \begin{bmatrix} 0 & -1 & 1 & 1 \\ -1 & 0 & -1 & -1 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 1 & 0 \end{bmatrix}$$

Παράδειγμα (2/5)

- Εάν στο δίκτυο παρουσιαστεί ως είσοδος το αρχικό διάνυσμα X_1=[1 -1 1 1], όλοι οι νευρώνες διατηρούν την έξοδό τους.
- Πράγματι, έστω ο πρώτος νευρώνας. Η νέα του είσοδος είναι:

$$S_1 = [1 \quad -1 \quad 1 \quad 1] \cdot \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} = 3$$

- Οπότε η νέα του έξοδος είναι $a_1=\Phi(S_1)=\Phi(3)=1$, ίδια δηλαδή με την παλιά.
- Το ίδιο θα συμβεί εάν δοκιμάσουμε να υπολογίσουμε νέα έξοδο για τους υπόλοιπους τρεις νευρώνες.

Παράδειγμα (3/5)

- Έστω ότι στο δίκτυο παρουσιάζεται η είσοδος X=[-1 -1 1 1], η οποία διαφέρει από το αρχικό διάνυσμα σε ένα στοιχείο.
- Θα υπολογίσουμε τις νέες εξόδους των νευρώνων με τυχαία σειρά, έστω με τη σειρά 3, 4, 1, 2.
- Νέα έξοδος 3ου νευρώνα:

$$S_3 = [-1 \quad -1 \quad 1 \quad 1] \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \\ 1 \end{bmatrix} = 1, \quad \Phi(S_3) = 1$$

Το διάνυσμα παραμένει X=[-1 -1 1 1]

Παράδειγμα (4/5)

- Νέα έξοδος 4ου νευρώνα:

$$S_4 = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} = 1, \quad \Phi(S_4) = 1$$

- Το διάνυσμα παραμένει X=[-1 -1 1 1].
- Νέα έξοδος του 1ου νευρώνα:

$$S_1 = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} = 3, \quad \Phi(S_1) = 1$$

Το διάνυσμα αλλάζει σε X=[1 -1 1 1]

Παράδειγμα (5/5)

- Τέλος ενημέρωση του 2ου νευρώνα:

$$S_2 = \begin{bmatrix} 1 & -1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 0 \\ -1 \\ -1 \end{bmatrix} = -3, \quad \Phi(S_2) = -1$$

- Έχει λοιπόν παραχθεί το αρχικό διάνυσμα. Αν συνεχίσουμε τις "ενημερώσεις", η κατάσταση αυτή δεν θα αλλάξει.

Παρατήρηση

- Αποδεικνύεται ότι ένα δίκτυο Hopfield πάντα συγκλίνει σε ένα από τα διανύσματα βάσει των οποίων "προγραμματίστηκε" το δίκτυο.
- Ωστόσο το διάνυσμα αυτό ενδέχεται να μην είναι το κοντινότερο προς το διάνυσμα που παρουσιάστηκε στην είσοδο του δικτύου.
  - Η υλοποίηση των δικτύων Hopfield στο Matlab προσπαθεί να ελαχιστοποιήσει την πιθανότητα να συμβεί κάτι τέτοιο.

(Matlab)

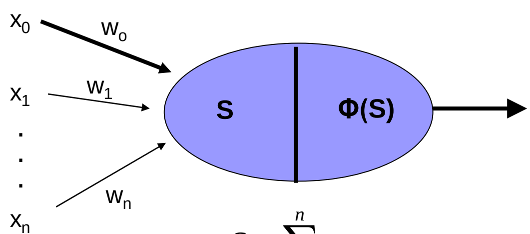
- Το Matlab υποστηρίζει μόνο τα δίκτυα Hopfield (αυτοσυσχετιστικά δίκτυα) και όχι τα ετεροσυσχετιστικά δίκτυα δύο επιπέδων νευρώνων.
- Υλοποιεί μια παραλλαγή τους (διαφορετικός τρόπος υπολογισμού των βαρών και χρήση τάσης πόλωσης), η οποία πετυχαίνει καλύτερη διάκριση των διανυσμάτων που αποθηκεύονται στο δίκτυο.
- Το μόνο που χρειάζεται να δώσουμε στο δίκτυο είναι τα διανύσματα που πρέπει να θυμάται.

Προσαρμόσιμα γραμμικά δίκτυα
Adaptive Linear Networks

Γενικά

- Τα προσαρμόσιμα γραμμικά δίκτυα (ADALINEs, Adaptive Linear Neuron networks) είναι πάρα πολύ απλά δίκτυα.
- Μοιάζουν πολύ με τα perceptrons.
- Αποτελούνται από έναν μόνο στρώμα γραμμικών νευρώνων.
  - Η μόνη διαφορά από τα perceptrons είναι δηλαδή ότι αντί για την βηματική συνάρτηση ενεργοποίησης έχουν την γραμμική.
- Μπορούμε να βάλουμε πολλούς γραμμικούς νευρώνες τον ένα δίπλα στον άλλο.
  - Το σχετικό δίκτυο είναι γνωστό ως M-ADALINE.
  - Οι νευρώνες (όπως και στην περίπτωση των perceptrons) δεν αλληλεπιδρούν μεταξύ τους και άρα εκπαιδεύονται ανεξάρτητα.
  - Αρκεί λοιπόν να δούμε ένα απλό ADALINE δίκτυο.

Γραμμικός νευρώνας



$$S = \sum_{i=0}^{n} x_i w_i$$

$$\Phi(S) = S$$

Παρατηρήσεις

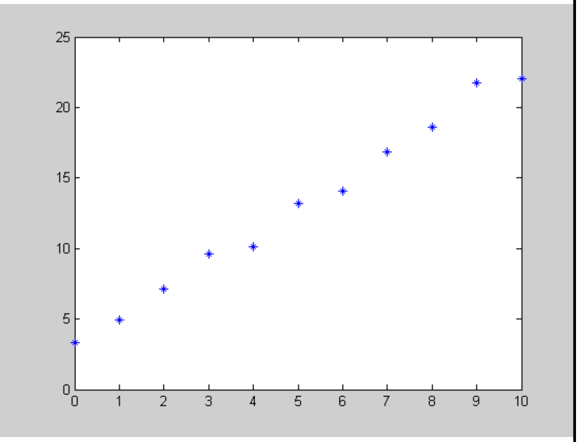
- Η έξοδος του νευρώνα είναι μια γραμμική συνάρτηση των εισόδων του.
- Εάν χρησιμοποιηθεί για να προσεγγίσει μια συνάρτηση, μπορεί να κάνει μόνο γραμμική προσέγγιση.
  - Ένα perceptron δεν μπορεί να χρησιμοποιηθεί για την προσέγγιση συνάρτησης (γραμμικής ή μη), μιας και βγάζει έξοδο μόνο 0 ή 1.
- Εάν χρησιμοποιηθεί για να κατηγοριοποιήσει τις εισόδους του, μπορεί να το επιτύχει πλήρως, μόνο εφόσον αυτές είναι γραμμικώς διαχωρίσιμες.
  - Ακριβώς όπως και σε ένα απλό perceptron.

Εκπαίδευση του γραμμικού νευρώνα

- Η εκπαίδευση του γραμμικού νευρώνα γίνεται ακριβώς όπως και στο perceptron:
  - $\Delta w_i = w_i - w_{i\text{-old}} = -d(y-o)x_i$
- όπου:
  - d : ο ρυθμός μάθησης
  - y : η τρέχουσα έξοδος
  - o : η επιθυμητή έξοδος
  - x_i : η i είσοδος
  - w_i : Το βάρος της i εισόδου
- Στόχος είναι να ελαχιστοποιηθεί το μέσο τετραγωνικό σφάλμα για όλα τα παραδείγματα.

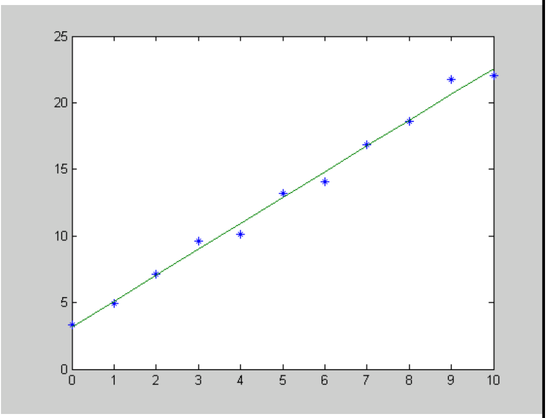
Παράδειγμα (1/2)

- Στο διπλανό σχήμα έχουμε 11 σημεία τα οποία βρίσκονται γύρω από την ευθεία y=2*x+3.



Παράδειγμα (2/2)

- Χρησιμοποιήθηκε ένας γραμμικός νευρώνας με 1 είσοδο (και προφανώς 1 έξοδο) και εκπαιδεύτηκε βάσει των προηγούμενων παραδειγμάτων.
- Στο διπλανό σχήμα φαίνονται τα αρχικά παραδείγματα (αστερίσκοι) και η έξοδος του δικτύου για διάφορες τιμές του x.



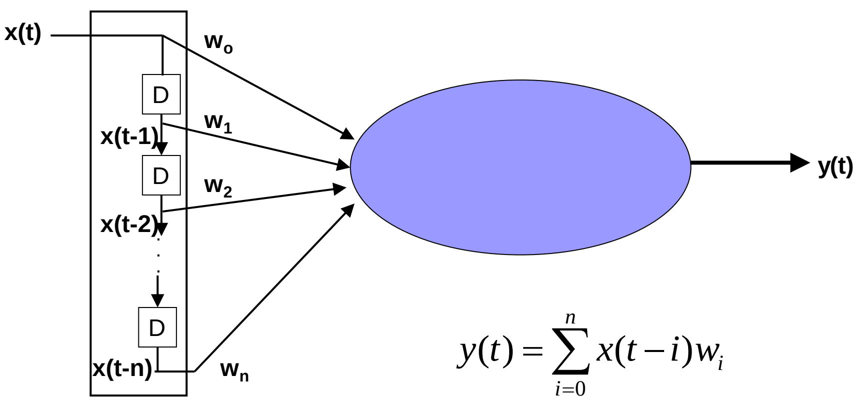
Παρατήρηση

- Από το τελευταίο παράδειγμα παρατηρεί κανείς ότι η λειτουργία του γραμμικού νευρώνα δεν διαφέρει σε τίποτα από την τεχνική των ελαχίστων τετραγώνων.
- Είναι ακριβώς έτσι!
  - Το πολύ-πολύ εάν έχουμε περισσότερες από μία εισόδους, πρέπει να εφαρμόσουμε την τεχνική ελαχίστων τετραγώνων σε χώρο περισσότερων διαστάσεων (π.χ. επίπεδο ελαχίστων τετραγώνων ή υπερ-επίπεδο ελ.τετρ. στη γενική περίπτωση).
- Μάλιστα το Matlab παρέχει δύο διαφορετικούς τρόπους κατασκευής ενός γραμμικού δικτύου:
  - Με εκπαίδευση: **Linear layer (train)**
  - Με σχεδίαση (απευθείας υπολογισμός των βαρών): **Linear layer (design)**

Γραμμή καθυστέρησης

- Τα γραμμικά δίκτυα αρχίζουν να γίνονται σημαντικά, όταν συνδυαστούν με ένα ειδικό στοιχείο, το στοιχείο καθυστέρησης (delay element).
  - Πολλά στοιχεία καθυστέρησης στη σειρά ορίζουν μια γραμμή καθυστέρησης (tapped delay line).
- Ένα στοιχείο καθυστέρησης προκαλεί την καθυστέρηση στη διάδοση του σήματος κατά το χρόνο που απαιτείται μέχρι να εμφανιστεί στην είσοδο η επόμενη είσοδος.

Σχηματική αναπαράσταση



$$y(t) = \sum_{i=0}^{n} x(t-i)w_i$$

Παρατηρήσεις

- Το "δίκτυο" στην προηγούμενη διαφάνεια φαίνεται εξωτερικά να έχει μία μόνο είσοδο (την x).
- Όμως εσωτερικά το δίκτυο θυμάται τις n προηγούμενες εισόδους και βάσει αυτών υπολογίζει τη νέα του έξοδο.
- Ειδικότερα:
  - Δεν είναι απαραίτητο οι προηγούμενες είσοδοι να υπάρχουν (μέχρι το βάθος χρόνου που υπάρχουν) όλες.
    - Για παράδειγμα, μπορεί να υπάρχουν οι είσοδος x(t-1) και x(t-3) και να μην υπάρχει η είσοδος x(t-2).
    - Πολλές φορές μάλιστα δεν υπάρχει η είσοδος x(t).
  - Φυσικά ο ίδιος νευρώνας μπορεί να έχει και άλλες εισόδους (εκτός από την x), με ή χωρίς γραμμές καθυστέρησης.

Θεωρητικό Παράδειγμα (1/2)

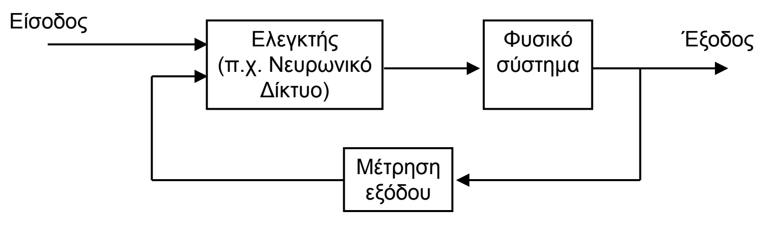
- Ας υποθέσουμε ότι το x(t) εκφράζει τη θέση ενός κινητού σώματος πάνω στον άξονα της κίνησής του.
- Γνωρίζοντας την τιμή x(t-1) μπορούμε να βγάλουμε συμπεράσματα για την ταχύτητά του, δηλαδή για την πρώτη παράγωγο του x.
- Γνωρίζοντας επιπλέον και την τιμή x(t-2), μπορούμε να βγάλουμε συμπεράσματα για την επιτάχυνσή του, δηλαδή για την δεύτερη παράγωγο του x, κ.ο.κ.
- Έστω τώρα ένα πρόβλημα όπου με βάση την θέση, την ταχύτητα και την επιτάχυνση ενός σώματος πρέπει να πάρουμε μια απόφαση, π.χ. να αλλάξουμε τη δύναμη ώθησης του οχήματος.

Θεωρητικό Παράδειγμα (2/2)

- Χρειαζόμαστε λοιπόν ένα νευρωνικό δίκτυο με τρεις εισόδους: για τη θέση, την ταχύτητα και την επιτάχυνση.
- Χρησιμοποιώντας μία μόνο είσοδο, σε συνδυασμό με μια γραμμή καθυστέρησης, αποφεύγουμε την χρήση πολλών οργάνων μέτρησης (στην προκειμένη περίπτωση για ταχύτητα και επιτάχυνση).
- Φυσικά το νευρωνικό θα πρέπει πρώτα να εκπαιδευτεί, κάτι που απαιτεί συλλογή πολλών παραδειγμάτων με ακολουθίες x(t) και τις αντίστοιχες ενδεδειγμένες ενέργειες y(t).
  - Παράδειγμα: Τα συστήματα αυτόματης σταθεροποίησης συχνότητας (automatic fine tuning, AFT) που υπάρχουν σε όλους τους ραδιοφωνικούς δέκτες FM, φροντίζουν τον διαρκή συντονισμό με την κύρια συχνότητα.

Συστήματα ελέγχου (1/2)

- Τα συστήματα ελέγχου κλειστού κύκλου (Closed-loop) έχουν γενικά τη μορφή που φαίνεται στο παρακάτω σχήμα:



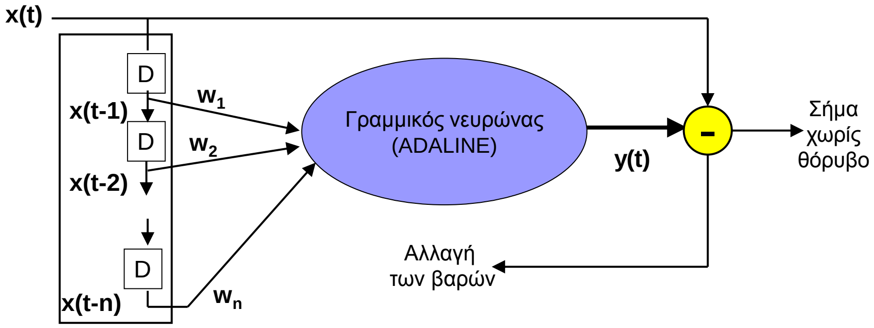
Συστήματα ελέγχου (2/2)

- Για παράδειγμα, έστω ένα σύστημα αυτόματης σταθεροποίησης συχνότητας (automatic fine tuning) για ραδιοφωνικούς δέκτες FM.
- Στην περίπτωση αυτή το φυσικό σύστημα είναι ο δέκτης FM.
- Η εξωτερική είσοδος είναι η συχνότητα που επιλέγουμε να ακούσουμε.
- Έξοδος είναι η συχνότητα που τελικά ακούμε.
- Το υποσύστημα μέτρησης μας λέει ποια είναι η συχνότητα που ακούμε, με τι ρυθμό μεταβάλλεται κλπ.
- Το νευρωνικό σύστημα συγκρίνει την επιθυμητή συχνότητα με την πραγματική συχνότητα, ενδεχομένως το ρυθμό μεταβολής της απόκλισης από την επιθυμητή συχνότητα, και προβαίνει στις κατάλληλες διορθωτικές κινήσεις (μετατοπίσεις) ώστε να συντονιστεί καλύτερα.
  - Φυσικά το Νευρωνικό Δίκτυο πρέπει αρχικά να εκπαιδευτεί βάσει επιλεγμένων παραδειγμάτων.

Πρόβλεψη σήματος (1/3)

- Μια από τις κυριώτερες εφαρμογές των ADALINES είναι στην πρόβλεψη σήματος.
- Έστω για παράδειγμα ένα σήμα θορύβου (π.χ. ηχητικό, ηλεκτρικό κλπ).
- Γνωρίζοντας τις προηγούμενες τιμές του σήματος, χρησιμοποιούμε ένα δίκτυο για να προβλέψουμε τη νέα τιμή του σήματος, πριν αυτή εμφανιστεί.
- Αυτό μας δίνει τη δυνατότητα, εφαρμόζοντας ένα αντίθετο από το προβλεπόμενο σήμα, να ακυρώνουμε το ίδιο το σήμα, δηλαδή να αφαιρούμε το θόρυβο (**γραμμικά φίλτρα**).
- Το σημαντικότερο είναι ότι το δίκτυο που εκτελεί αυτή τη λειτουργία αυτο-εκπαιδεύεται την ώρα της λειτουργίας του, έχοντας τη δυνατότητα να προσαρμόζεται σε κάθε νέο σήμα.

Πρόβλεψη σήματος (2/3)



Demo nnd10nc στο Matlab

Πρόβλεψη σήματος (3/3)

- Βλέπουμε στο διάγραμμα της προηγούμενης διαφάνειας ότι στην είσοδο του νευρωνικού δεν εφαρμόζεται το σήμα x(t), παρά μόνο τα σήματα x(t-1), x(t-2) κλπ.
- Το σήμα x(t) συγκρίνεται με την έξοδο y(t).
- Η διαφορά y(t)-x(t) αποτελεί το σφάλμα του δικτύου και βάσει αυτής αλλάζουν τα βάρη στις εισόδους x(t-1), x(t-2) κλπ, βάσει της σχέσης:
  - $\Delta w_i = w_i - w_{i\text{-}old} = -d(y-o)x_i$
- Άρα το δίκτυο μαθαίνει να προβλέπει την τιμή του σήματος x(t) από τις προηγούμενές τιμές του.
  - Με μικρές τροποποιήσεις το δίκτυο θα ήταν σε θέση να προβλέψει την τιμή x(t) από τη χρονική στιγμή t-1.

Μαθηματική ερμηνεία (1/2)

- Έστω μια συνάρτηση y=f(x).
- Εάν γνωρίζουμε την τιμή της συνάρτησης $f(x_0)$ καθώς και όλων των παραγώγων της $f^{(n)}(x_0)$ σε κάποιο σημείο x_0, μπορούμε να προσεγγίσουμε τη συνάρτηση σε οποιοδήποτε άλλο σημείο χρησιμοποιώντας μια **σειρά Taylor**:

$$f(x) = \sum_{n=0}^{\infty} f^{(n)}(x_o) \frac{(x - x_0)^n}{n!}$$

- Στην προκειμένη περίπτωση το γραμμικό δίκτυο μιμείται τον παραπάνω τύπο.

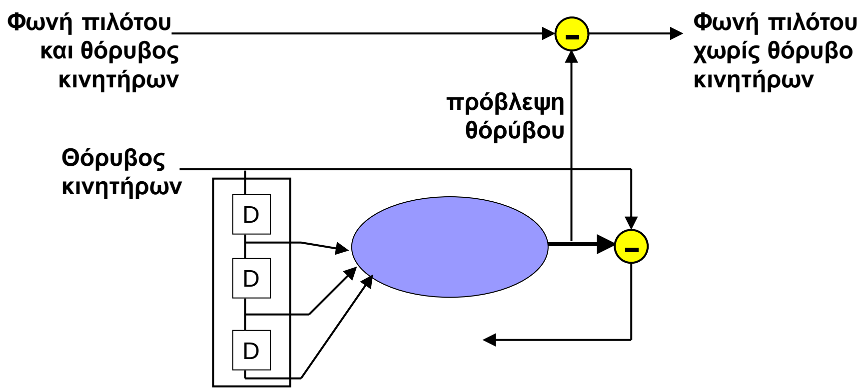
Μαθηματική ερμηνεία (2/2)

- Ως x_0 μπορούμε να θεωρήσουμε το t-1.
- Η τιμή x-x_0 ισούται με 1, οπότε όλες οι δυνάμεις "φεύγουν".
- Οι παράγωγοι μπορούν να υπολογιστούν από συνδυασμό των τιμών x(t-1), x(t-2) κλπ.
- Τελικά το νευρωνικό δίκτυο "μαθαίνει" τον τύπο προσαρμόζοντας τα βάρη του.

Θεωρητικό παράδειγμα (1/2)

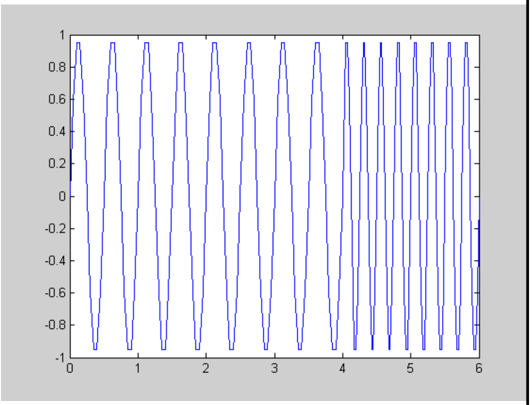
- Έστω ο πιλότος στο πιλοτήριο του αεροπλάνου.
- Όταν μιλάει στους επιβάτες μέσα από το μικρόφωνο, ο θόρυβος του αεροπλάνου θα έπρεπε να διέρχεται μέσα από το μικρόφωνο και να ακούγεται μαζί με τη φωνή του από τα μεγάφωνα, δημιουργώντας πολύ άσχημο ηχητικό αποτέλεσμα.
- Ωστόσο, κάτι τέτοιο δεν συμβαίνει.
- Στο σήμα που ξεκινά από το μικρόφωνο, το οποίο περιέχει τόσο τη φωνή όσο και το θόρυβο, παρεμβάλλεται ένα φίλτρο αφαίρεσης του θορύβου.

Θεωρητικό παράδειγμα (2/2)



Παράδειγμα (1/3)

- Έστω ένα ημιτονοειδές σήμα διάρκειας 6sec, με κυκλική συχνότητα 4π για τα πρώτα 4sec και 8π για τα επόμενα 2sec.

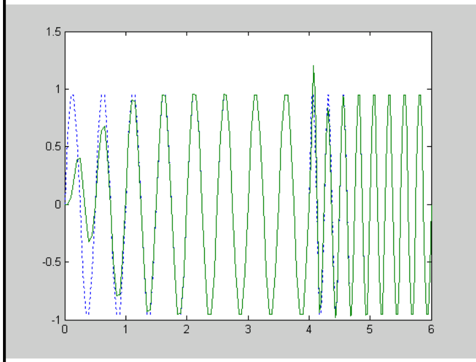


Παράδειγμα (2/3)

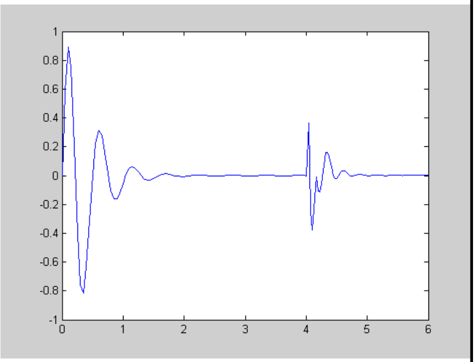
- Κατασκευάζουμε ένα γραμμικό φίλτρο ενός γραμμικού νευρώνα με είσοδο τις τιμές x(t-1), x(t-2), ..., x(t-5).
- Η τιμή x(t) χρησιμοποιείται για σύγκριση με την έξοδο y(t) του νευρώνα.
- Έτσι ο νευρώνας μαθαίνει να προβλέπει την τιμή x(t) από τις τιμές x(t-1), x(t-2), ..., x(t-5).
- Εκπαιδεύουμε το δίκτυο για μία μόνο εποχή (δηλαδή εκπαίδευση σε πραγματικό χρόνο), παρουσιάζοντας τις τιμές του σήματος με τη σειρά.
- Παίρνουμε την έξοδο του νευρώνα και το σφάλμα της εξόδου κατά τη διάρκεια της εκπαίδευσης.

Παράδειγμα (3/3)

- Έξοδος δικτύου



- Σφάλμα εξόδου (ή σήμα μετά την αφαίρεση του θορύβου)



(Matlab) (1/2)

- Το Matlab παρέχει δύο τρόπους κατασκευής γραμμικών δικτύων:
  - Με εκπαίδευση: **Linear layer (train)**
  - Με σχεδίαση (απευθείας υπολογισμός των βαρών): **Linear layer (design)**
- Στην περίπτωση ύπαρξης γραμμών καθυστέρησης, η μέθοδος με σχεδίαση δεν μπορεί να εφαρμοστεί.
- Επιπλέον, η εκπαίδευση πρέπει να γίνει αυξητικά. Κάτι τέτοιο απαιτεί τη μετατροπή των πινάκων εισόδου/εξόδου σε πίνακες κελιών (βλέπε διαφάνειες 156-160).

(Matlab) (2/2)

- Ειδικά στην περίπτωση γραμμικών φίλτρων, δεν έχει νόημα να εκτελέσουμε περισσότερες από μια εποχές εκπαίδευσης.
  - Ουσιαστικά τα δίκτυα αυτά μαθαίνουν καλύτερα τα τελευταία παραδείγματα, ενώ ξεχνούν γρήγορα τα προηγούμενα.
- Αντίθετα, στην περίπτωση απλών συστημάτων ελέγχου (διαφάνειες 232-235), όπως π.χ. αυτό του ραδιοφωνικού δέκτη FM, μπορούμε να εκτελέσουμε πολλές εποχές εκπαίδευσης, φροντίζοντας όμως τα δεδομένα μας να παρουσιάζονται πάντα με την ίδια σειρά.

Χρήσιμα στατιστικά μεγέθη

Statistics

Αυτοσυσχέτιση

AUTOCORRELATION

Γενικά

- Η συνάρτηση αυτοσυσχέτισης (autocorrelation function, Box and Jenkins, 1976) μπορεί να χρησιμοποιηθεί για δύο στόχους:
  - Ανίχνευση μη τυχαιότητας (non-randomness) στα δεδομένα
  - Ανίχνευση συγκεκριμένης συσχέτισης στα δεδομένα.

Ορισμός

- Έστω μια σειρά από τυχαίους μετρήσεις Y_1, Y_2, …, Y_N.
- Ορίζουμε την αυτοσυσχέτιση με υστέρηση (lag) ίση με k ως εξής:

$$r_k = \frac{\sum_{i=1}^{N-k}\left(Y_i - \bar{Y}\right)\left(Y_{i+k} - \bar{Y}\right)}{\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2} = \frac{\sum_{i=1}^{N-k}\left(Y_i - \bar{Y}\right)\left(Y_{i+k} - \bar{Y}\right)}{\sigma_y^2}$$

- Κανονικοποιημένη μορφή:

$$r_k = \frac{1}{N-1}\frac{\sum_{i=1}^{N-k}\left(Y_i - \bar{Y}\right)\left(Y_{i+k} - \bar{Y}\right)}{\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2} = \frac{1}{N-1}\frac{\sum_{i=1}^{N-k}\left(Y_i - \bar{Y}\right)\left(Y_{i+k} - \bar{Y}\right)}{\sigma_y^2}$$

Παρατηρήσεις

- Όταν η αυτοσυσχέτιση χρησιμοποιείται για την ανίχνευση μη-τυχαιότητας, συνήθως χρησιμοποιείται μόνο υστέρηση k=1.
- Όταν χρησιμοποιείται για την ανίχνευση συγκεκριμένου προτύπου σε χρονοσειρές (οπότε δεν υπάρχει τυχαιότητα), γίνεται γραφική παράσταση της αυτοσυσχέτισης ως προς την υστέρηση.

Παράδειγμα

- Έστω οι παρακάτω 20 τυχαίοι αριθμοί:
  - 74, 50, 95, 78, 1, 46, 32, 0, 77, 39, 15, 86, 98, 75, 0, 58, 62, 44, 30, 86
- Μέσος όρος = 52.3

$$\sum_{i=1}^{N-1}\left(Y_i-\overline{Y}\right)\left(Y_{i+1}-\overline{Y}\right)=-737.09$$

$$\sum_{i=1}^{N}\left(Y_i-\overline{Y}\right)^2=18124,51$$

- r_1 = -0.04
- r_0 = 1

Συσχέτιση χρονοσειρών

TIME SERIES CROSS-CORRELATION

Γενικά

- Με παρόμοιο τρόπο μπορούμε να υπολογίσουμε τη συσχέτιση ενός σήματος με ένα άλλο.
- Η συσχέτιση χρονοσειρών είναι μια μέτρηση της ομοιότητάς τους ως συνάρτηση της διαφοράς φάσης τους (time lag).
  - Ονομάζεται επίσης sliding dot product ή sliding inner product.
- Χρησιμοποιείται για την αναζήτηση σε ένα μεγάλης διάρκειας σήμα ενός μικρότερου (pattern).

Ορισμός (για διακριτά σήματα)

- Έστω δύο σήματα x(t) και y(t):

$$r_k = \frac{\sum_{t=1}^{N-k} \left(y(t)-\bar{y}\right)\left(x(t+k)-\bar{x}\right)}{\sigma_y \sigma_x}$$

- Κανονικοποιημένη μορφή:

$$r_k = \frac{1}{N-1} \frac{\sum_{t=1}^{N-k} \left(y(t)-\bar{y}\right)\left(x(t+k)-\bar{x}\right)}{\sigma_y \sigma_x}$$

- Εάν η έξοδος y(t) εκφράζει κάποιο σφάλμα e(t), ψάχνουμε να βρούμε συσχέτιση του σφάλματος με την είσοδο.

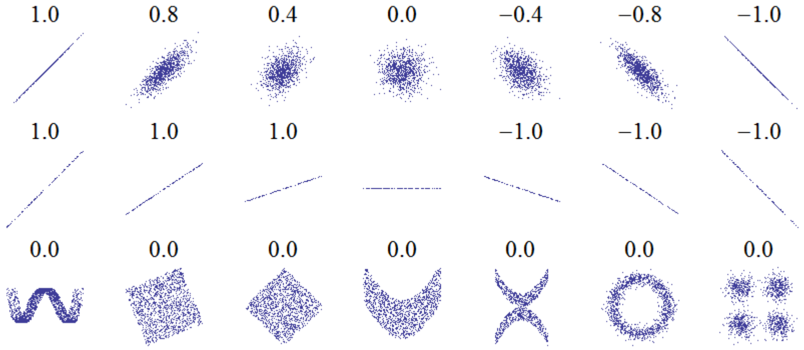
Συσχέτιση Pearson

PEARSON PRODUCT-MOMENT CORRELATION (PMCC)

Συντελεστής συσχέτισης Pearson

- Έστω δύο μεταβλητές X και Y και ένα σύνολο N ζευγών (X_1,Y_1), (X_2,Y_2), …, (X_n,Y_n).
- Η συσχέτιση Pearson μεταξύ των μεταβλητών ορίζεται ως εξής:
- $\rho_{X,Y} = r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{X})^2}}$
- Δεν αφορά χρονοσειρές.

Παραδείγματα συντελεστών συσχέτισης



http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient (Dec2010)

Συνδιακύμανση

COVARIANCE

Ορισμός

- Έστω ένα σύνολο ζευγών τιμών (X, Y) για τις τυχαίες μεταβλητές X και Y.
- Συνδιακύμανση:
  - Cov(X,Y) = E[(X-E[X]) (Y-E[Y])]
- Ιδιότητες
  - Cov(X,Y) = E[XY] – E[X]E[Y]
  - Cov(X,a)=0
  - Cov(X,X)=Var(X) Cov(X,Y)=Cov(Y,X)
  - Cov(aX,bY)=abCov(X,Y) Cov(X+a,Y+b)=Cov(X,Y)

Αυτοσυνδιακύμανση Autocovariance

- Έστω το σήμα X σε δύο διαφορετικές στιγμές t και s
  - Υποθέτουμε ένα παράθυρο συγκεκριμένου μεγέθους γύρω από αυτές τις στιγμές
- $K_{XX}(t,s) = E[(X_t-\mu_t)(X_s-\mu_s)] = E[X_tX_s] – \mu_t\mu_s$
- Για αυθαίρετα μεγάλες σειρές μπορούμε να υποθέσουμε $\mu_t = \mu_s = \mu$.

Δυναμικά Δίκτυα

Dynamic Networks

Γενικά

- Ως δυναμικά χαρακτηρίζονται εκείνα τα δίκτυα τα οποία έχουν είτε στοιχεία καθυστέρησης, είτε αναδρομικές συνδέσεις, είτε και τα δύο.
  - Feed-forward dynamic networks
  - Recurrent dynamic networks
- Πρόκειται για ακολουθιακά κυκλώματα (στην ορολογία των ψηφιακών συστημάτων), τα οποία διαθέτουν μνήμη και απαιτούν χρονισμό για τη λειτουργία τους.

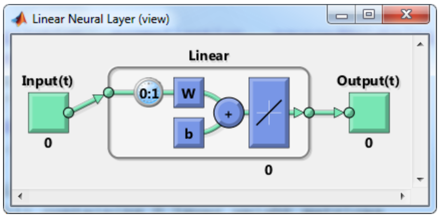
Παράδειγμα: linearlayer

TDLs

- net=linearlayer([0,1],0.01)

Ρυθμός μάθησης

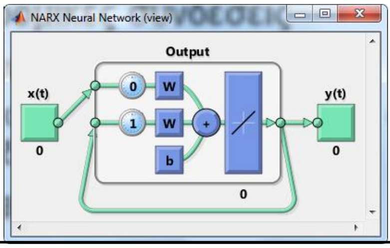
- Γραμμικό δίκτυο ενός επιπέδου με TDLs στην είσοδο (χωρίς αναδρομικές συνδέσεις)



- net.biasConnect=0 για να αφαιρέσουμε την τάση πόλωσης.

Παράδειγμα: narxnet

- Nonlinear autoregressive networks with an external (exogenous) input
- Αναδρομικές συνδέσεις
- Γραμμές καθυστέρησης
  - Τόσο στην είσοδο όσο και στις αναδρομικές συνδέσεις
- net=narxnet(0,1,[],'closed')



Παρατηρήσεις

- Τα linearlayer δίκτυα χαρακτηρίζονται ως Finite Impulse Response (FIR), γιατί η έξοδός τους μηδενίζεται κάποια στιγμή μετά τον τερματισμό της εισόδου.
- Τα narx δίκτυα χαρακτηρίζονται ως Infinite Impulse Response (IIR), γιατί η έξοδός τους δεν μηδενίζεται ποτέ μετά τον τερματισμό της εισόδου (ωστόσο εξασθενεί πολύ).

Εκπαίδευση

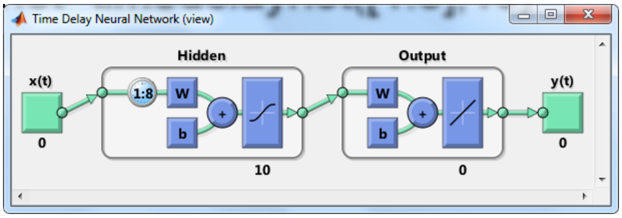
- Τα δυναμικά δίκτυα εκπαιδεύονται με τον ίδιο τρόπο που εκπαιδεύονται και τα στατικά δίκτυα.
  - Ισχύουν όλοι οι γνωστοί αλγόριθμοι.
  - Τα παραδείγματα παρουσιάζονται πάντα με συγκεκριμένη σειρά.
  - Η εκπαίδευση διαρκεί περισσότερο και είναι πιθανότερο να οδηγηθεί σε τοπικά ελάχιστα.

Focused Time-Delay NN (FTDNN) (timedelaynet)

- Απλό δίκτυο backpropagation με στοιχεία καθυστέρησης στην είσοδο
  - Focused networks: Τα στοιχεία καθυστέρησης εμφανίζονται μόνο στην είσοδο ενός στατικού κατά τα λοιπά δικτύου.
- Πρόβλεψη σε χρονοσειρές, με βάση προηγούμενες τιμές που δίνονται.

Παράδειγμα

- net=timedelaynet([1:8],10)



- net=train(net, x(9:end), y(9:end), x(1:8))

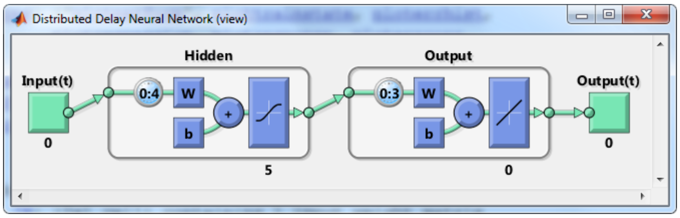
Αφού έχουμε 8 στοιχεία καθυστέρησης, η εκπαίδευση ξεκινά από την $9^{η}$ θέση των πινάκων εισόδου/εξόδου, ωστόσο δίνουμε και τις 8 πρώτες τιμές του πίνακα εισόδου για αρχικοποίηση των στοιχείων καθυστέρησης.

Προετοιμασία δεδομένων

- Στα δυναμικά δίκτυα τα δεδομένα χρεάζονται ιδιαίτερη προετοιμασία, έτσι ώστε να δοθούν αρχικές τιμές σε όλα τα στοιχεία καθυστέρησης.
- Κάποια παραδείγματα εκπαίδευσης χάνονται.
- [p, Pi, Ai, t] = preparets(net, in, out)
- net=train(net, p, t, Pi, Ai)
  - Pi: Initial input states　　Ai: Initial layer states
  - p: Shifted inputs　　t: Shifted targets

Distributed Time-Delay NN (DTDNN) (distdelaynet)

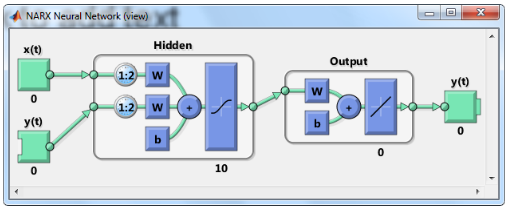
- Έχει στοιχεία καθυστέρησης τόσο στην είσοδο, όσο και μεταξύ των δύο στρωμάτων.
- net=distdelaynet({[0:4],[0:3]},5)



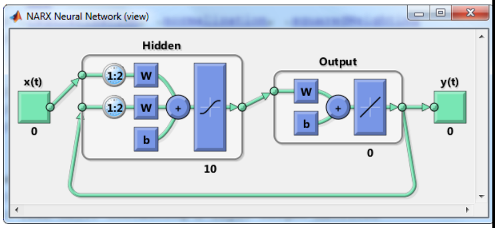
Non-Linear Auto-Regressive with Exogenous inputs (NARX)

- $y(t)=f(\quad y(t-1), y(t-2), \ldots, y(t-n_y),$
  $x(t-1), x(t-2), \ldots, x(t-n_x) \quad)$
- Εφαρμογές:
  - Πρόβλεψη της επόμενης τιμής του y
  - Μη γραμμικό φίλτρο (αφαίρεση θορύβου)
  - Μοντελοποίηση μη γραμμικών δυναμικών συστημάτων

Ανοιχτός / Κλειστός βρόχος (1/2)



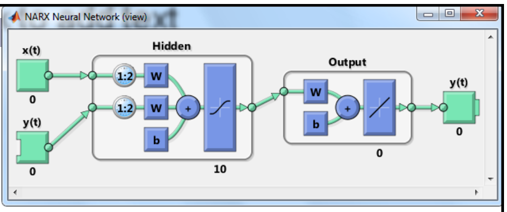
- Σχετικές εντολές:
  - net=closeloop(net)
  - net=openloop(net)



Ανοιχτός / Κλειστός βρόχος (2/2)

- Όταν μπορούμε να παρατηρούμε τις αμέσως προηγούμενες τιμές του y, τότε χρησιμοποιούμε την αρχιτεκτονική ανοιχτού βρόχου.
  - Στην εκπαίδευση που γνωρίζουμε όλες τις τιμές του y, χρησιμοποιούμε ανοιχτό βρόχο, κάτι που έχει ως αποτέλεσμα γρηγορότερη και καλύτερη εκπαίδευση.
- Όαν δεν είμαστε σε θέση να παρατηρήσουμε τις τιμές του y, αναγκαζόμαστε να τις προσεγγίσουμε μέσω της εξόδου του δικτύου (κλειστός βρόχος).
  - Π.χ., πρόβλεψη μεθεπόμενης τιμής της εξόδου, όταν δεν είναι ακόμα γνωστή η επόμενη τιμή.

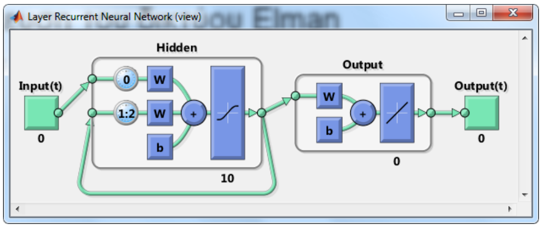
narxnet



- net=narxnet([1:2],[1:2],10)
- Προετοιμασία δεδομένων:
  - [p, Pi, Ai, t] = preparets(net, x, y1, y2)
    - x: Πίνακας εισόδου
    - y1: Πίνακας εξόδου (target) για εξόδους χωρίς ανάδραση
    - y2: Πίνακας εξόδου (target) για εξόδους με ανάδραση
  - net=train(net,p,t,Pi,Ai)
    - Pi: Initial input states — Ai: Initial layer states
    - p: Shifted inputs — t: Shifted targets
  - yp=net(p,Pi,Ai)

Layer-Recurrent Network (layrecnet)

- Γενίκευση του δικτύου Elman
- net = layrecnet(1:2,10);



Πολλαπλές χρονοσειρές (1/2)

- Ενδέχεται για ένα δυναμικό δίκτυο να διαθέτουμε πολλαπλές χρονοσειρές, όχι συνεχόμενες η μία με την άλλη.
- Για να μην κάνουμε ξεχωριστές εκπαιδεύσεις, μπορούμε να δημιουργήσουμε έναν πίνακα κελιών, όπου κάθε κελί περιέχει περισσότερα από ένα διανύσματα.
- Οι χρονοσειρές με λιγότερα στοιχεία συμπληρώνονται με NaN.

Πολλαπλές χρονοσειρές (2/2)

- Έτσι, για κάθε κελί η εκπαίδευση γίνεται παράλληλα (αλλά και ανεξάρτητα), δηλαδή:
  - Παίρνει τα πρώτα διανύσματα από όλα τα κελιά.
  - Μετά παίρνει τα δεύτερα διανύσματα από όλα τα κελιά.
  - κ.ο.κ.
- π.χ., για z=f(z-1,x-1,y-1), έστω τρεις σειρές:
  - Είσοδος: { [x11, x12, x13; y11, y12, y13],
  - [x21, x22, x23; y21, y22, y23] }

- { [2 3 4] [2 4 4] [1 5 4] [2 9 NaN] }
- 3 4 4 2 23 5 3 2 3 2 3 NaN

Στάθμιση σφαλμάτων (1/2)

- Στην εκπαίδευση ενός νευρωνικού δικτύου το συνολικό σφάλμα ορίζεται ως:

$$mse = \frac{1}{N}\sum_{i=1}^{N} e_i^2$$

- Τα σφάλματα όμως θα μπορούσαν να σταθμιστούν, έτσι ώστε:

$$mse = \frac{1}{N}\sum_{i=1}^{N} w_i^e e_i^2$$

Στάθμιση σφαλμάτων (2/2)

- Το βάρος ενός σφάλματος μπορεί να είναι συνάρτηση του χρόνου, του αριθμού του δείγματος κλπ.
  - Για παράδειγμα, σε μια χρονοσειρά μπορεί να θέλουμε μικρότερα σφάλματα στο τέλος.
- Χρειάζεται απλά να περάσουμε έναν πίνακα με βάρη ως τελευταίο όρισμα στην train.
  - Ο πίνακας αυτός πρέπει να έχει το ίδιο πλήθος στοιχείων, όπως και ο πίνακας των target data.

Ακτινικά Δίκτυα
Radial Basis Networks

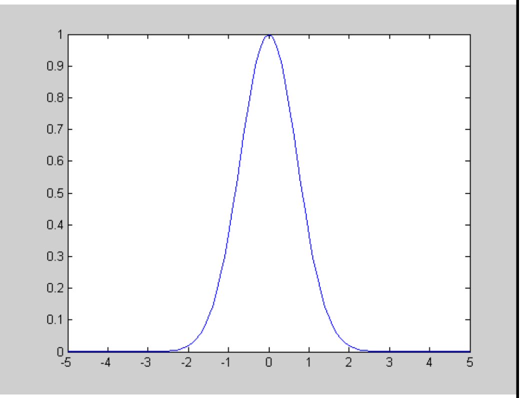
Γενικά

- Ένα ακτινικό δίκτυο (radial basis network) μοιάζει στη λειτουργία του με ένα ανταγωνιστικό δίκτυο.
- Ο βαθμός ενεργοποίησης ενός ακτινικού νευρώνα εξαρτάται από την απόσταση του σήματος εισόδου από τα βάρη εισόδου του νευρώνα:
  - Si=||**w**-**x**||
  - όπου **w** τα βάρη στην είσοδο του νευρώνα και **x** η τρέχουσα είσοδος.
- Ενεργοποιούνται όλοι οι νευρώνες, όμως ο βαθμός της ενεργοποίησης είναι μεγαλύτερος για τους νευρώνες εκείνους, τα βάρη εισόδου των οποίων βρίσκονται πιο κοντά στην τρέχουσα είσοδο.

Ακτινική συνάρτηση ενεργοποίησης

- Η ακτινική συνάρτηση ενεργοποίησης είναι η εξής:

$$\Phi(S_i) = e^{-(\frac{0.8326}{\sigma} \cdot S_i)^2}$$

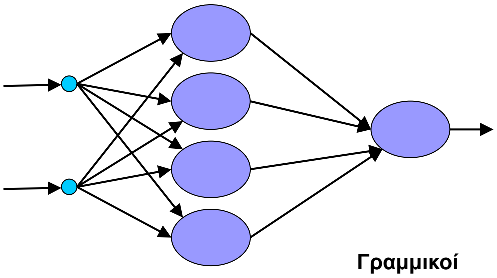


- όπου σ ένας παράγοντας που καθορίζει το πλάτος της διπλανής καμπύλης.
  - Για $S_i = \sigma$ ισχύει:

$$e^{-0.8326^2} = 0.5$$

Δομή ακτινικού δικτύου (1/2)

- Στη γενική του μορφή ένα ακτινικό δίκτυο έχει την παρακάτω δομή:



Δομή ακτινικού δικτύου (2/2)

- Οι νευρώνες του ακτινικού επιπέδου είναι τόσοι όσα και τα παραδείγματα εκπαίδευσης.
- Κάθε ακτινικός νευρώνας λοιπόν αντιστοιχεί σε ένα παράδειγμα εκπαίδευσης.
- Τα βάρη στις εισόδους κάθε νευρώνα του ακτινικού επιπέδου είναι ίδια με τις τιμές του αντίστοιχου παραδείγματος εκπαίδευσης.
- Οι έξοδοι των νευρώνων του ακτινικού επιπέδου πολλαπλασιάζονται με τα βάρη στις εισόδους των νευρώνων του γραμμικού επιπέδου και τελικά δίνουν τις εξόδους του δικτύου.

Εκπαίδευση ακτινικού δικτύου

- Το ακτινικό δίκτυο δεν χρειάζεται εκπαίδευση.
- Τα βάρη μπορούν να υπολογιστούν αυτόματα από τα παραδείγματα εκπαίδευσης.
- Ειδικότερα, τα βάρη στις εισόδους των ακτινικών νευρώνων αντιστοιχούν στις εισόδους των παραδειγμάτων εκπαίδευσης (ένα παράδειγμα για κάθε νευρώνα).
- Στη συνέχεια τα βάρη στις εξόδους των ακτινικών νευρώνων υπολογίζονται έτσι ώστε το ακτινικό δίκτυο να απαντά απολύτως σωστά σε όλα τα παραδείγματα εκπαίδευσης.
  - Ο υπολογισμός των βαρών αυτών βασίζεται στην επίλυση συστημάτων Ν γραμμικών εξισώσεων με Ν αγνώστους, όπου Ν το πλήθος των παραδειγμάτων (και άρα και των ακτινικών νευρώνων) αλλά και το πλήθος των βαρών στις εισόδους κάθε γραμμικού νευρώνα.

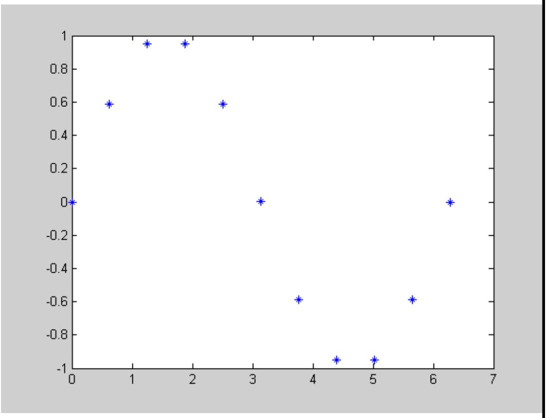
Σύγκριση με feed-forward δίκτυα

- Σε σχέση με ένα απλό νευρωνικό δίκτυο feed-forward, ένα ακτινικό δίκτυο πλεονεκτεί γιατί:
  - Δεν χρειάζεται εκπαίδευση
  - Παρουσιάζει καλύτερη συμπεριφορά σε προβλήματα με πάρα πολλά παραδείγματα
- Και μειονεκτεί γιατί:
  - Έχει περισσότερους νευρώνες
  - Έχει μεγαλύτερο χρόνο υπολογισμού κατά τη λειτουργία του (λόγω ακριβώς των περισσότερων νευρώνων).

Γιάννης Ρεφανίδης 267

Παράδειγμα (1/8)

- Έστω ότι χρησιμοποιούμε ένα ακτινικό δίκτυο που προσεγγίζει την συνάρτηση του ημιτόνου στο διάστημα [0,2π].
- Χρησιμοποιούμε 11 παραδείγματα εκπαίδευσης, ομοιόμορφα κατανεμημένα στο διάστημα [0,2π].



Γιάννης Ρεφανίδης 268

Παράδειγμα (2/8)

- Το ακτινικό νευρωνικό δίκτυο έχει 11 ακτινικούς νευρώνες στο κρυφό επίπεδο και 1 γραμμικό νευρώνα στο επίπεδο εξόδου.
- Οι 11 ακτινικοί νευρώνες αντιστοιχούν στα παραδείγματα εκπαίδευσης με είσοδο 0, π/5, 2π/5, 3π/5, ..., 9π/5, 2π.
- Κάθε ένας από τους 11 νευρώνες έχει στην είσοδό του ως βάρος μία από τις παραπάνω τιμές.

Παράδειγμα (3/8)

- Έστω ότι παρουσιάζεται στην είσοδο το παράδειγμα x=π.
- Ο παρακάτω πίνακας δείχνει τις εξόδους των 11 ακτινικών νευρώνων.
  - Θεωρούμε ότι η παράμετρος σ στον τύπο της διαφάνειας 191 ισούται με 1.

#	w	S	Φ(S)	#	w	S	Φ(S)
1	0	π	0.0011	7	6π/5	π/5	0.7606
2	π/5	4π/5	0.0125	8	7π/5	2π/5	0.3346
3	2π/5	3π/5	0.0852	9	8π/5	3π/5	0.0852
4	3π/5	2π/5	0.3346	10	9π/5	4π/5	0.0125
5	4π/5	π/5	0.7606	11	2π	π	0.0011
6	π	0	1				

Παράδειγμα (4/8)

- Οι έξοδοι από τους ακτινικούς νευρώνες ενεργοποιούν (μέσω των σχετικών βαρών) τους νευρώνες του επιπέδου εξόδου.
- Επειδή η τιμή x=π αντιστοιχεί σε ένα από τα παραδείγματα με τα οποία προγραμματίστηκε το δίκτυο, η έξοδος του δικτύου δεν θα έχει καθόλου σφάλμα.

Παράδειγμα (5/8)

- Έστω ότι παρουσιάζεται στην είσοδο το παράδειγμα x=π/4, το οποίο δεν ταυτίζεται με κανένα από τα παραδείγματα "εκπαίδευσης" του δικτύου.
- Ο παρακάτω πίνακας δείχνει τις εξόδους των 11 ακτινικών νευρώνων.
  - Θεωρούμε ότι η παράμετρος σ στον τύπο της διαφάνειας 191 ισούται με 1.

#	w	S	Φ(S)	#	w	S	Φ(S)
1	0	π/4	0.6521	7	6π/5	19π/20	0.0021
2	π/5	π/20	0.9830	8	7π/5	23π/20	0.00011
3	2π/5	3π/20	0.8573	9	8π/5	27π/20	0
4	3π/5	7π/20	0.4325	10	9π/5	31π/20	0
5	4π/5	11π/20	0.1262	11	2π	7π/4	0
6	π	3π/4	0.0213				

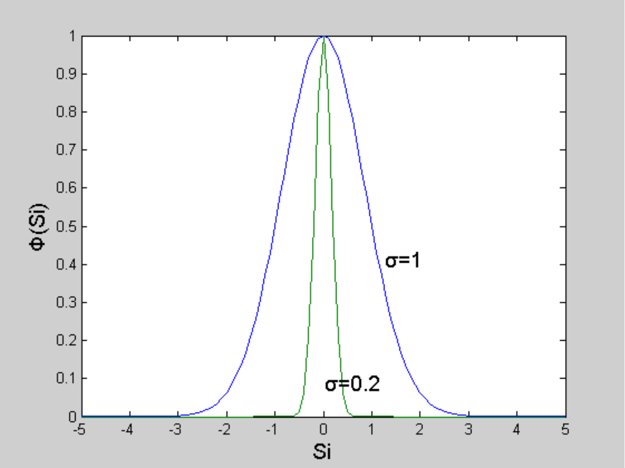
Παράδειγμα (6/8)

- Οι έξοδοι από τους ακτινικούς νευρώνες ενεργοποιούν (μέσω των σχετικών βαρών) τους νευρώνες του επιπέδου εξόδου.
- Επειδή η τιμή x=π/4 δεν ταυτίζεται με κανένα από τα παραδείγματα με τα οποία προγραμματίστηκε το δίκτυο, η έξοδος του δικτύου θα έχει μικρό σφάλμα.

Παρατήρηση (1/2)

- Η παράμετρος σ στον τύπο της διαφάνειας 191 παίζει μεγάλο ρόλο στη συμπεριφορά του δικτύου.
- Ουσιαστικά η παράμετρος σ κλιμακώνει τις αποστάσεις των διανυσμάτων εισόδου από τα διανύσματα βαρών για τους ακτινικούς νευρώνες.
  - Μικρή τιμή στην παράμετρο σ μεγεθύνει τις αποστάσεις.
  - Μεγάλη τιμή στην παράμετρο σ μικραίνει τις αποστάσεις.
    - Για παράδειγμα, όσο πιο μικρή είναι η τιμή της παραμέτρου σ, τόσο πιο απότομα πέφτει η έξοδος των νευρώνων καθώς αυξάνεται η απόσταση της εισόδου από τα βάρη εισόδου.

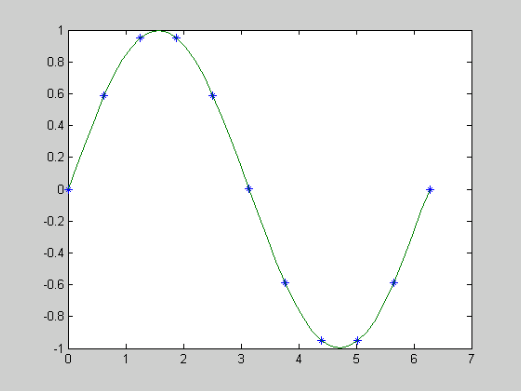
Παρατήρηση (2/2)



Γιάννης Ρεφανίδης 275

Παράδειγμα (7/8)

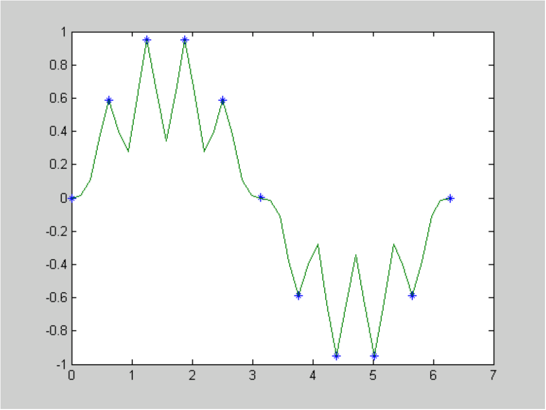
- Στην εικόνα φαίνεται η έξοδος του δικτύου για σ=1.



Γιάννης Ρεφανίδης 276

Παράδειγμα (8/8)

- Στην εικόνα φαίνεται η έξοδος του δικτύου για σ=0.2.



(Matlab)

- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: **Radial basis (exact fit)**
- Τα μόνα δεδομένα που χρειάζεται να δώσουμε είναι τα παραδείγματα εισόδου (όπως πάντα κάθε παράδειγμα είναι μία στήλη στον σχετικό πίνακα), τα αντίστοιχα παραδείγματα εξόδου και τη σταθερά σ (αναφέρεται ως Spread constant).
- Με την κατασκευή του το δίκτυο είναι έτοιμο για λειτουργία (simulation), αφού δεν χρειάζεται εκπαίδευση.

(Matlab)

- Το Matlab υποστηρίζει ακόμη δύο ακτινικά δίκτυα.
  - Το δίκτυο **Radial basis (fewer neurons)** διαφέρει από όσα είδαμε στο ότι γίνεται προσπάθεια να χρησιμοποιηθούν λιγότεροι ακτινικοί νευρώνες στο κρυφό επίπεδο.
  - Έτσι δεν υπάρχει αντιστοίχιση ανάμεσα στα παραδείγματα εκπαίδευσης και στους ακτινικούς νευρώνες.
  - Αυτό έχει ως αποτέλεσμα το δίκτυο να παρουσιάζει μικρά σφάλματα ακόμη και στα παραδείγματα εκπαίδευσης.
  - Χρειάζεται να ορίσουμε έναν στόχο συνολικού σφάλματος (performance goal), από τον οποίο θα εξαρτηθεί τελικά πόσοι ακτινικοί νευρώνες θα χρησιμοποιηθούν.
    - Οι νευρώνες προστίθενται ένας-ένας στο δίκτυο, μέχρι να καταστεί δυνατό το συνολικό σφάλμα να πέσει κάτω από τον στόχο.
    - Στην οριακή περίπτωση οι νευρώνες θα γίνουν ίσοι με τον αριθμό των παραδειγμάτων, οπότε όπως έχουμε δει το σφάλμα μηδενίζεται.

(Matlab)

- Το δίκτυο **Generalized Regression**:
  - Έχει τόσους ακτινικούς νευρώνες όσα είναι τα παραδείγματα εκπαίδευσης.
    - Έχει γραμμικούς νευρώνες εξόδου όσες οι απαιτούμενες έξοδοι.
  - Τα βάρη στις εισόδους των ακτινικών νευρώνων ισούνται με τις εισόδους των <u>αντίστοιχων</u> παραδειγμάτων εκπαίδευσης.
  - Τα βάρη στις εξόδους των ακτινικών νευρώνων ισούνται με τις εξόδους των <u>αντίστοιχων</u> παραδειγμάτων εκπαίδευσης (<u>πλεονέκτημα</u> αφού έτσι αποφεύγουμε τους υπολογισμούς τους με επίλυση συστημάτων γραμμικών εξισώσεων).
- Η έξοδος του δικτύου καθορίζεται ως εξής:
  - Υπολογίζονται όλες οι έξοδοι των ακτινικών νευρώνων.
  - Γίνεται κανονικοποίησή τους ώστε το άθροισμά τους να είναι 1.
  - Υπολογίζονται κανονικά οι έξοδοι των γραμμικών νευρώνων.
    - Για παράδειγμα, εάν μόνο ένας ακτινικός νευρώνας βγάλει μη-μηδενική έξοδο, τα βάρη στις εξόδους του θα εμφανιστούν στην έξοδο του δικτύου.
  - Το δίκτυο αυτό μπορεί και να μην δώσει απολύτως ακριβή έξοδο ακόμη και για τα παραδείγματα εκπαίδευσης (<u>μειονέκτημα</u> σε σχέση με τα RB exact fit).
- Η διαδικασία δημιουργίας ενός δικτύου Generalized regression είναι εντελώς ίδια με αυτή ενός δικτύου Radial basis (exact fit).

Κατηγοριοποίηση με ακτινικά δίκτυα – *Πιθανοτικά δίκτυα*

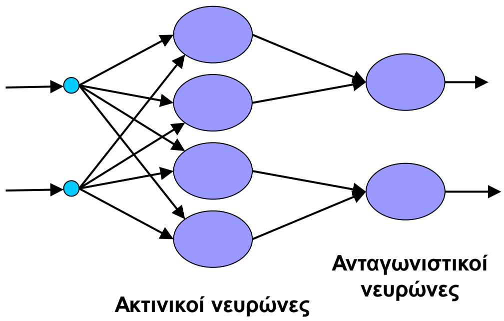
Probabilistic Neural Networks

Γενικά

- Μια παραλλαγή των ακτινικών δικτύων μπορεί να χρησιμοποιηθεί για εφαρμογές κατηγοριοποίησης και μάλιστα με πολύ καλά αποτελέσματα.
- Τα σχετικά δίκτυα λέγονται πιθανοτικά δίκτυα (probabilistic neural networks).
- Διαφέρουν από τα ακτινικά δίκτυα που είδαμε στις προηγούμενες διαφάνειες κατά το ότι το επίπεδο εξόδου είναι ανταγωνιστικό.
- Η λογική είναι η εξής:
  - Έστω Ν αρχικά παραδείγματα που κατατάσσονται σε 2 κατηγορίες Α και Β.
  - Έστω ένα νέο παράδειγμα Χ το οποίο θέλουμε να κατατάξουμε σε μια από τις δύο κατηγορίες.
  - Το Χ συγκρίνεται με κάθε ένα από τα Ν παραδείγματα και κάθε παράδειγμα συνεισφέρει στην πιθανότητα το Χ να ανήκει στην κατηγορία Α ή Β βάσει της απόστασης του Χ από το παράδειγμα και της σχέσης της διαφάνειας 191.
  - Οι επιμέρους πιθανότητες αθροίζονται και τελικά το Χ κατατάσσεται στην κατηγορία για την οποία προέκυψε μεγαλύτερο άθροισμα.

Δομή πιθανοτικού δικτύου (1/2)

- Ένα πιθανοτικό δίκτυο έχει την παρακάτω δομή:



Δομή πιθανοτικού δικτύου (2/2)

- Οι νευρώνες του ακτινικού επιπέδου (κρυφό επίπεδο) είναι τόσοι όσα και τα παραδείγματα εκπαίδευσης.
- Οι νευρώνες του ανταγωνιστικού επιπέδου (επίπεδο εξόδου) είναι τόσοι όσες και οι κατηγορίες.
  - Υπενθύμιση: Από τους νευρώνες αυτούς θα ενεργοποιείται κάθε φορά μόνο ένας.
- Τα βάρη στις εισόδους κάθε νευρώνα του ακτινικού επιπέδου είναι ίδια με τις τιμές του αντίστοιχου παραδείγματος εκπαίδευσης.
- Τα βάρη μεταξύ των νευρώνων του ακτινικού και του ανταγωνιστικού επιπέδου είναι όλα μηδέν, εκτός από αυτά που συνδέουν νευρώνες του ανταγωνιστικού επιπέδου με τις αντίστοιχές τους κατηγορίες και τα οποία είναι ίσα με 1.
  - Άρα στην έξοδο κάθε ακτινικού νευρώνα υπάρχει μόνο ένα βάρος ίσο με 1.
  - Υπενθύμιση: Κάθε νευρώνας του ακτινικού επιπέδου αντιστοιχεί σε ένα παράδειγμα εκπαίδευσης και άρα σε μία συγκεκριμένη κατηγορία εξόδου.

Λειτουργία πιθανοτικού δικτύου

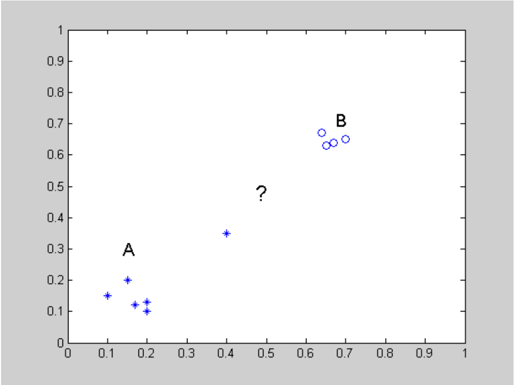
- Έστω Χ ένα παράδειγμα που εφαρμόζεται στην είσοδο του δικτύου.
- Για κάθε ακτινικό νευρώνα υπολογίζεται η απόσταση του Χ από τα βάρη του νευρώνα και τελικά ο νευρώνας παράγει μια έξοδο.
- Οι έξοδοι των ακτινικών νευρώνων μεταφέρονται στους κατάλληλους νευρώνες εξόδου.
- Ο νευρώνας εξόδου με τη μεγαλύτερη είσοδο "νικά" και δίνει έξοδο 1, ενώ όλοι οι υπόλοιποι νευρώνες εξόδου δίνουν έξοδο 0.

Παρατηρήσεις

- Η κατηγορία στην οποία κατατάσσεται μια νέα είσοδος εξαρτάται:
  - Από το πλήθος των αρχικών παραδειγμάτων που είναι "κοντά" στην είσοδο.
  - Από το πόσο κοντά σε κάθε τέτοιο παράδειγμα βρίσκεται η νέα είσοδος.
  - Από την τιμή της παραμέτρου σ (διαφάνεια 191).
- Οριακές περιπτώσεις:
  - Εάν η παράμετρος σ γίνει πολύ μικρή ($\sigma \to 0$), κάθε παράδειγμα κατατάσσεται στην κατηγορία του πλησιέστερου παραδείγματος.
  - Εάν η παράμετρος σ γίνει πολύ μεγάλη ($\sigma \to \infty$), το παράδειγμα κατατάσσεται σε εκείνη την κατηγορία που είχε τα περισσότερα παραδείγματα εκπαίδευσης.

Παράδειγμα (1/5)

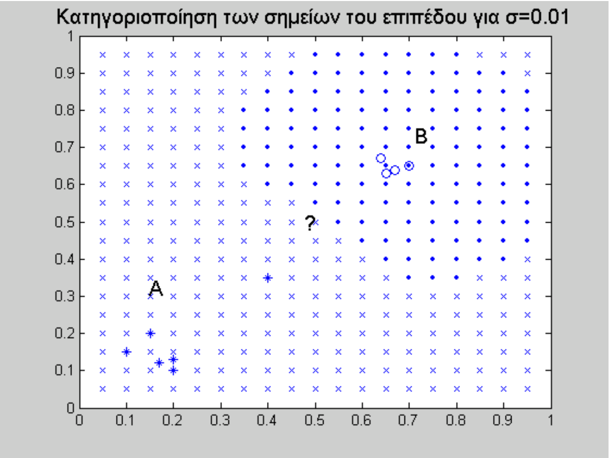
- Έστω δύο κατηγορίες Α και Β και 10 παραδείγματα (6 της κατηγορίας Α και 4 της κατηγορίας Β) σε χώρο δύο διαστάσεων, όπως φαίνεται στο παρακάτω σχήμα.
- Έστω ένα νέο παράδειγμα με συντεταγμένες (0.5,0.5), το οποίο μας ενδιαφέρει να κατάταξουμε σε μία από τις κατηγορίες Α και Β (σημειώνεται με ? στην εικόνα).



Παράδειγμα (2/5)

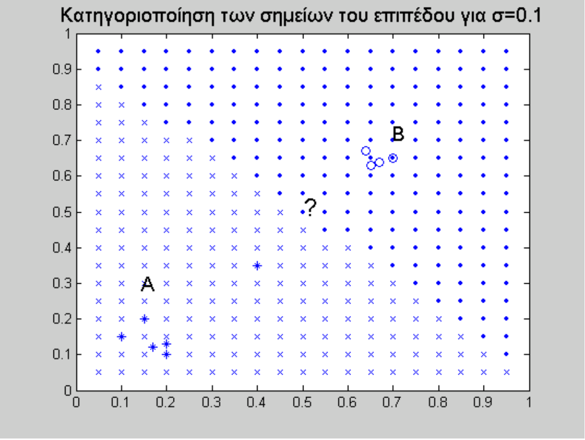
- Κατασκευάζουμε ένα πιθανοτικό δίκτυο με 10 ακτινικούς νευρώνες στο κρυφό επίπεδο και 2 ανταγωνιστικούς νευρώνες στο επίπεδο εξόδου.
- Δοκιμάζουμε διάφορους συνδυασμούς για τη σταθερά σ:
  - Για σ=0.01 το νέο παράδειγμα κατατάσσεται στην κατηγορία Α, λόγω του κοντινότερου παραδείγματος από την ομάδα Α.
  - Για σ=0.1 το νέο παράδειγμα κατατάσσεται στην κατηγορία Β, λόγω της κοντινότερης ομάδας παραδειγμάτων της ομάδας Β.
  - Για σ=1 το νέο παράδειγμα κατατάσσεται στην κατηγορία Α, λόγω της μεγαλύτερης ομάδας παραδειγμάτων της ομάδας Α.

Παράδειγμα (3/5)



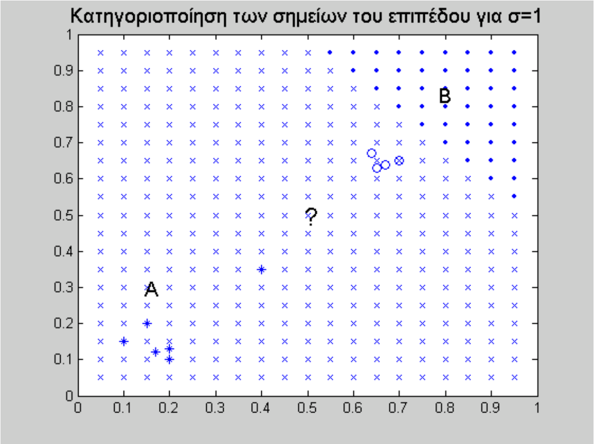
Γιάννης Ρεφανίδης

Παράδειγμα (4/5)



Γιάννης Ρεφανίδης

Παράδειγμα (5/5)



(Matlab)

- Στο παράθυρο Create New Network επιλέγουμε:
  - Network type: **Probabilistic**
- Τα μόνα δεδομένα που χρειάζεται να δώσουμε είναι τα παραδείγματα εισόδου (όπως πάντα κάθε παράδειγμα είναι μία στήλη στον σχετικό πίνακα), τα αντίστοιχα παραδείγματα εξόδου και τη σταθερά σ (αναφέρεται ως Spread constant).
- Με την κατασκευή του το δίκτυο είναι έτοιμο για λειτουργία (simulation), αφού δεν χρειάζεται εκπαίδευση.

Matlab
Εξειδικευμένα δίκτυα

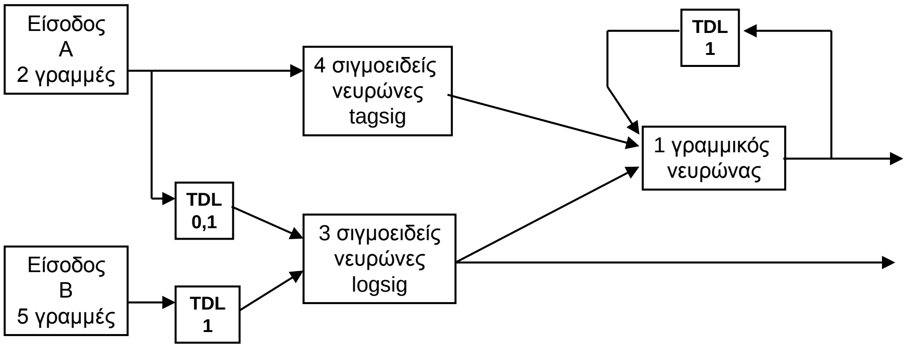
Custom networks

Γενικά

- Το Matlab Neural Network Toolbox μας δίνει τη δυνατότητα να φτιάξουμε παραλλαγές / συνδυασμούς των δικτύων που είδαμε ως τώρα.
- Για παράδειγμα, μπορούμε να φτιάξουμε δίκτυα όπου:
  - Ένα στρώμα συνδέεται με το μεθεπόμενό του.
  - Υπάρχουν αναδρομικές συνδέσεις οποιασδήποτε μορφής.
  - Όλες οι είσοδοι δεν συνδέονται με τους ίδιους νευρώνες.
  - Υπάρχουν είσοδοι με διαφορετικές ιδιότητες καθυστέρησης.
  - κλπ
- Για κάτι τέτοιο δεν μπορούμε να χρησιμοποιήσουμε το γραφικό interface (εντολή nntool), αλλά πρέπει να δώσουμε εντολές στην γραμμή εντολών του Matlab.

Παράδειγμα (1/2)

- Έστω ότι θέλουμε να κατασκευάσουμε το παρακάτω δίκτυο:



Παράδειγμα (2/2)

- Στο προηγούμενο παράδειγμα χρειάζεται να ορίσουμε 3 στρώματα και 2 ξεχωριστές εισόδους.
- Πρέπει επίσης να ορίσουμε τις συνδέσεις των εισόδων με τα στρώματα αλλά και τα στρώματα μεταξύ τους.
- Για κάθε σύνδεση πρέπει να ορίσουμε εάν υπάρχουν γραμμές καθυστέρησης και ποιες.
- Για κάθε στρώμα πρέπει να ορίσουμε:
  - τον αριθμό των νευρώνων του
  - τη συνάρτηση ενεργοποίησης
  - εάν έχει σταθερή τάση
  - κλπ.

Δημιουργία δικτύου και στρωμάτων

- Δημιουργία αφηρημένου δικτύου:
  - net=network
- Ορισμός στρωμάτων:
  - net.numInputs=2
  - net.numlayers=3
- Καθορισμός των στρωμάτων που έχουν τάση πόλωσης:
  - net.biasConnect=[1;0;1]
- ή
  - net.biasConnect(1)=1
  - net.biasConnect(3)=1

Καθορισμός συνδέσεων επιπέδων

- Ο πίνακας net.inputConnect καθορίζει ποιες είσοδοι συνδέονται με ποια στρώματα.
  - net.inputConnect(1,1)=1;
  - net.inputConnect(2,1)=1;
  - net.inputConnect(2,2)=1;
- Ο πίνακας net.layerConnect καθορίζει τις συνδέσεις των στρωμάτων:
  - net.layerconnect(3,1)=1
  - net.layerconnect(3,2)=1
  - net.layerconnect(3,3)=1
- Προσοχή: Οι δείκτες (i,j) παραπάνω σημαίνουν ότι η έξοδος του στρώματος j είναι είσοδος στο στρώμα i.

Καθορισμός εξόδων

- Ο πίνακας net.outputConnect καθορίζει ποια στρώματα δίνουν την έξοδό τους στο περιβάλλον.
  - net.outputConnect=[0 1 1];
- Ο πίνακας net.targetConnect καθορίζει ποιων στρωμάτων η έξοδος θα χρησιμοποιηθεί κατά την εκπαίδευση.
  - net.targetConnect=[0 0 1];
- Παρατήρηση: Δεν είναι απαραίτητο όλες οι έξοδοι να είναι έξοδοι-στόχοι, δηλαδή να χρησιμοποιηθούν για τον καθορισμό των σφαλμάτων κατά την εκπαίδευση.

Καθορισμός εισόδων

- Το Matlab ορίζει έναν πίνακα κελιών για κάθε είσοδο, όπως και για κάθε στρώμα.
  - net.inputs
- Κάθε κελί έχει ένα σύνολο από πληροφορίες για το αντίστοιχο αντικείμενο.
- Καθορισμός εύρους τιμών για τις δύο εισόδους:
  - net.inputs{1}.range=[0 10; 0 10]
  - net.inputs{2}.range=[-2 2; -2 2; -2 2; -2 2; -2 2]

Καθορισμός στρωμάτων

- Παρόμοια, ο πίνακας κελιών net.layers καθορίζει τις ιδιότητες των στρωμάτων.
- Ορισμός αριθμού νευρώνων:
  - net.layers{1}.size=4;
- Ορισμός συνάρτησης ενεργοποίησης:
  - net.layers{1}.transferFcn='tansig'
- Ορισμός συνάρτησης αρχικοποίησης βαρών:
  - net.layers{1}.initFcn='initnw'
- Παρόμοια και για τα υπόλοιπα στρώματα:
  - net.layers{2}.size=3;
  - net.layers{2}.transferFcn='logsig'
  - net.layers{2}.initFcn='initnw'
  - net.layers{3}.initFcn='initnw'

Γραμμές καθυστέρησης

- Ο πίνακας κελιών net.inputWeights περιέχει πληροφορίες για τις συνδέσεις των εισόδων με τα στρώματα.
- Ορισμός γραμμών καθυστέρησης:
  - net.inputWeights{2,1}.delays=[0 1];
  - net.inputWeights{2,2}.delays=[1];
- Παρόμοια, ο πίνακας κελιών net.layerWeights περιέχει πληροφορίες για τις συνδέσεις μεταξύ των στρωμάτων:
  - net.layerWeights{3,3}.delays=[1];

Καθορισμός παραμέτρων εκπαίδευσης

- Καθορισμός μεθόδου εκπαίδευσης:
  - net.trainFcn='trainlm'
- Καθορισμός μεθόδου υπολογισμού σφάλματος:
  - net.performFcn='mse'
- Καθορισμός εποχών, αποδεκτού σφάλματος κλπ.:
  - net.trainParam.epochs=1000
  - net.trainParam.goal=0.1
  - κλπ.

Καθορισμός συγκεκριμένων βαρών

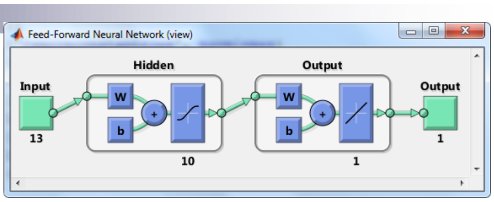
- Ο πίνακας κελιών net.IW περιέχει τους πίνακες βαρών των συνδέσεων από τις εισόδους προς τα στρώματα.
  - Για παράδειγμα, το στοιχείο net.IW{2,1} είναι ο πίνακας βαρών από την πρώτη είσοδο προς το δεύτερο στρώμα.
- Παρόμοια, ο πίνακας κελιών net.LW περιέχει τους πίνακες βαρών των συνδέσεων μεταξύ των στρωμάτων.
  - Για παράδειγμα, το στοιχείο net.LW{3,2} είναι ο πίνακας βαρών της σύνδεσης από το δεύτερο προς το τρίτο στρώμα.
- Τέλος ο πίνακας net.b περιέχει τους πίνακα βαρών των τάσεων σταθερής πόλωσης των στρωμάτων.
  - Για παράδειγμα, το στοιχείο net.b{1} είναι ο πίνακας βαρών της τάσης σταθερής πόλωσης του πρώτου στρώματος.

Εκπαίδευση του δικτύου

- Αρχικοποιούμε τα βάρη του δικτύου, σύμφωνα με τη συνάρτηση αρχικοποίησης που έχει οριστεί για κάθε στρώμα, με την εντολή:
  - init(net)
- Εάν έχουμε δύο πίνακες P και T με τις εισόδους και τις αντίστοιχες εξόδους ενός συνόλου παραδειγμάτων εκπαίδευσης, εκπαιδεύουμε το δίκτυο με την εντολή (πάντα σύμφωνα με τις παραμέτρους εκπαίδευσης που έχουμε ορίσει) :
  - [net, tr]=train(net,P,T)
- Τέλος, εάν έχουμε ένα σύνολο δεδομένων Q και θέλουμε να προσομοιώσουμε την έξοδο του δικτύου για τα δεδομένα αυτά, εκτελούμε την εντολή:
  - Y=net(Q)
- όπου net το όνομα του δικτύου (σαν να ήταν εντολή).

Άλλες εντολές του Matlab

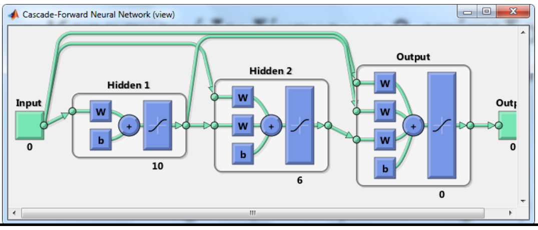
feedforwardnet



- net=feedforwardnet(10)
  - Κατασκευάζει δίκτυο 2 επιπέδων (tansig και linear) με 10 νευρώνες στο κρυφό.
- net=configure(net,Inputs,Targets)
  - Ρυθμίζει τις εισόδους και τις εξόδους του δικτύου, καθώς και συναρτήσεις προ- και μετά-επεξεργασίας των εισόδων/εξόδων.
  - Δείτε net.inputs{1}, net.outputs{2}

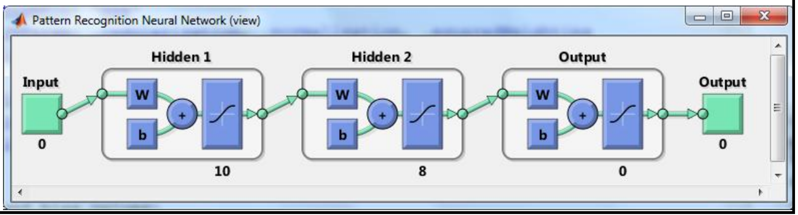
cascadeforwardnet

- Κατασκευάζει δίκτυο με 3 επίπεδα, με συνδέσεις κάθε επιπέδου (και των εισόδων) προς όλα τα επόμενα.
- net=cascadeforwardnet([10,6])
- net=configure(net,Inputs,Targets)



patternnet

- Για προβλήματα ταξινόμησης
- net=patternnet([10,8])
- net=configure(net,Inputs,Targets)



Διαχείριση μνήμης

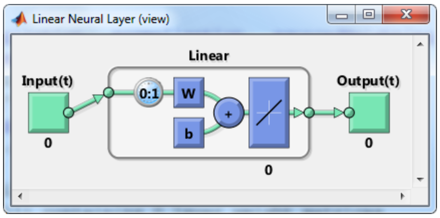
- net.efficiency.memoryReduction
- Η default τιμή είναι 1.
- Εάν θέσουμε:
  - net.efficiency.memoryReduction = k
- τότε τα δεδομένα χωρίζονται σε k ομάδες και στη συνέχεια κάθε εποχή σπάει σε k μέρη, άρα χρειάζεται το 1/k της μνήμης.
  - Αυξάνεται ο χρόνος της εκπαίδευσης.

linearlayer

TDLs

Ρυθμός μάθησης

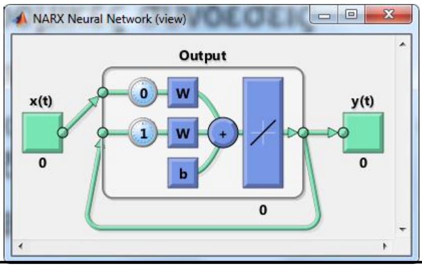
- net=linearlayer([0,1],0.01)
- Γραμμικό δίκτυο ενός επιπέδου με TDLs στην είσοδο (χωρίς αναδρομικές συνδέσεις)



- net.biasConnect=0 για να αφαιρέσουμε την τάση πόλωσης.

narxnet

- Αναδρομικές συνδέσεις
- Γραμμές καθυστέρησης
  - Τόσο στην είσοδο όσο και στις αναδρομικές συνδέσεις
- net=narxnet(0,1,[],'closed')



Παρατηρήσεις

- Τα linearlayer δίκτυα χαρακτηρίζονται ως Finite Impulse Response (FIR), γιατί η έξοδός τους μηδενίζεται κάποια στιγμή μετά τον τερματισμό της εισόδου.
- Τα narx δίκτυα χαρακτηρίζονται ως Infinite Impulse Response (IIR), γιατί η έξοδός τους δεν μηδενίζεται ποτέ μετά τον τερματισμό της εισόδου (ωστόσο εξασθενεί πολύ).

Παραδείγματα εφαρμογών των ΝΔ

Matlab NN toolbox – User stories (1/4)

- Στη σελίδα του Matlab Neural Network toolbox (http://www.mathworks.com/products/neuralnet/) μπορείτε να βρείτε δύο ιστορίες εφαρμογής του σε πραγματικά προβλήματα.
  - Πρόβλεψη οικονομικών κρίσεων σε αναδυόμενες αγορές
    - Dr. Paul McNelis, Georgetown University
  - Αφορμή στάθηκε η κρίση του 1997 στην Ινδονησία, όπου η τιμή του Ινδονησιακής ρούπιας έπεσε δραματικά.
  - Η εφαρμογή βασίστηκε στην ανάλυση των μηνιαίων αναγκών και για τα δύο νομίσματα για τα τελευταία 13 χρόνια, περιλαμβανομένης της κρίσης.

Matlab NN toolbox – User stories (2/4)

- Μετά από εξέταση διαφόρων αρχιτεκτονικών, υιοθετήθηκε ένα κλασσικό feed-forward μοντέλο με ένα κρυφό στρώμα.
- Για τους νευρώνες του κρυφού στρώματος χρησιμοποιήθηκε η σιγμοειδής συνάρτηση.
- Τέλος, για τους νευρώνες στην έξοδο χρησιμοποιήθηκε η γραμμική συνάρτηση.
- Αποτελέσματα:
  - Το μοντέλο που αναπτύχθηκε επιδεικνύει μεγαλύτερη ακρίβεια από τα παραδοσιακά μοντέλα.
  - Το πρόγραμμα χρησιμοποιείται από την κεντρική τράπεζα της Ινδονησίας για πρόβλεψη των αναγκών σε χρήματα και του δομικού πληθωρισμού.

Matlab NN toolbox – User stories (3/4)

- Η εταιρεία Halliburton Energy Services παρέχει προϊόντα και υπηρεσίες για εξερεύνηση και εξόρυξη πετρελαίου και φυσικού αερίου σε όλο τον κόσμο:
  - Αξιολόγηση ενδεχομένων κοιτασμάτων
  - Γεώτρηση και συντήρηση των πετρελαιοπηγών
- Στην αρχή μιας γεώτρησης ανοίγεται μια τρύπα. Στη συνέχεια εισάγονται ατσαλένια καλύμματα τα οποία στερεώνονται στα τοιχώματα της τρύπας.
- Για να μπορέσει να εισέλθει το πετρέλαιο στον σωλήνα, ο σωλήνας μετατρέπεται σε διάτρητο μέσω εκρηκτικών γομώσεων που τοποθετούνται στο σωλήνα από ένα διατρητικό πυροβόλο.
- Για λόγους ασφαλείας είναι ζωτικής σημασίας να γνωρίζουν εάν όλες οι εκρηκτικές γομώσεις εξεράγησαν, πριν ανέλθει το διατρητικό πυροβόλο στην επιφάνεια.

Matlab NN toolbox – User stories (4/4)

- Τα ηχητικά σήματα που δηλώνουν επιτυχή έκρηξη μιας γόμωσης είναι πολλές φορές δυσδιάκριτα, εξαιτίας του μεγάλου βάθους του πηγαδιού (2 με 3 μίλια) και των υπόλοιπων θορύβων της όλης γεώτρησης (αντλίες, γεννήτριες και άλλες συσκευές γύρω από την κεφαλή της γεώτρησης).
- Ο μηχανικός Roger Schultz ανέλαβε να λύσει το πρόβλημα.
- Χρησιμοποίησε ένα προσαρμόσιμο, predictive μη-γραμμικό NN.
- Αξιοποίησε το γεγονός ότι ο θόρυβος προέρχεται από μηχανές της επιφάνειας, οπότε μπορεί να απομονωθεί και να προβλεφθεί από ένα κατάλληλο NN.

Pacific Northwest National Laboratory (1/2)

- Πολλές εφαρμογές σε θέματα:
  - Περιβάλλοντος (ανίχνευση διαρροών)
  - Διαχείρισης Ενέργειας
  - Διάγνωσης Ασθενειών
  - Ανάλυσης Δεδομένων από πειράματα Φυσικής
- Αναλυτική περιγραφή:
  - http://www.emsl.pnl.gov:2080/proj/neuron/neural/projects.html
  - Το site έχει σύντομες περιγραφές των εφαρμογών και σε πολλές περιπτώσεις τα σχετικά technical papers.

Pacific Northwest National Laboratory (2/2)

- Ανίχνευση ισοτόπων με φασματογράφο ακτίνων γάμμα.
  - Χρησιμοποιήθηκαν συσχετιστικά δίκτυα (τύπου Hopfield).
- Μέτρηση του βαθμού της αναισθησίας του ασθενή κατά τη διάρκεια εγχειρήσεων
  - Δεδομένα εισόδου είναι εγκεφαλογράφημα και ηλεκτρομυόγραμμα του προσώπου.
  - Χρησιμοποιήθηκε feedforward δίκτυο.
- Διάγνωση καρδιοπνευμονικών παθήσεων.
  - Δεδομένα εισόδου: Πίεση αίματος, παλμοί, ρυθμός αναπνοής, αέρια αίματος κλπ.

Πρόβλεψη τιμής μετοχών (1/2)

- Παράδειγμα από την εργασία:
  - Mark Fishman, Dean Barr and Walter Loick: Using neural nets in market analysis. Technical Analysis of Stocks & Commodities, 9 (April): 18-21, 1991.
- Κατασκεύασαν ένα νευρωνικό δίκτυο feedforward backpropagation το οποίο προέβλεπε την τιμή που θα είχε μια μετοχή μετά από πέντε μέρες, βασιζόμενο σε 6 οικονομικούς δείκτες για αυτή τη μετοχή.
- Δεν έχουν δώσει πολλές λεπτομέρειες για το ΝΝ τους: Προτίμησαν να το πουλήσουν σε εταιρεία.

Πρόβλεψη τιμής μετοχών (2/2)

- Οι έξι δείκτες που χρησιμοποίησαν στην εκπαίδευση του δικτύου είναι:
  - Ένας αργός στοχαστικός δείκτης %D 9 ημερών.
  - Ένας αργός στοχαστικός δείκτης %K 9 ημερών.
  - Ο δείκτης ADX 18-ημερών (Average Directional Movement Index).
  - Ο δείκτης MACD (Moving average Convergence/Divergence)
  - Η τρέχουσα τιμή της μετοχής
  - Η μεταβολή της τιμής της μετοχής τις τελευταίες 5 μέρες

Αξιολόγηση Εταιρικών Ομολόγων (1/2)

- Σχετική αναφορά:
  - Soumitra Dutta and Shashi Shekhar. Bond rating: A non conservative application of neural networks. In Proc. of IEEE International Conference on Neural Networks,1990, pp. 443-450.
- Χρησιμοποίησαν ένα δίκτυο με ένα κρυφό επίπεδο με σιγμοειδείς νευρώνες και έναν γραμμικό νευρώνα στην έξοδο.
- Εκπαίδευσαν το δίκτυο χρησιμοποιώντας ένα δείγμα από 47 εταιρείες που εκδίδουν εταιρικά ομόλογα.

Αξιολόγηση Εταιρικών Ομολόγων (2/2)

- Χρησιμοποίησαν 9 κριτήρια:
  - Ύψος ομολογιακού δανείου προς περιουσία της εταιρείας.
  - Ποσοστό άλλων οφειλών
  - Πωλήσεις δια της καθαρής αξίας της επιχείρησης
  - Κέρδη δια των πωλήσεων
  - Οικονομική ευρρωστία
  - Κέρδη διά των παγίων εξόδων
  - Αύξηση εσόδων τα τελευταία 5 χρόνια
  - Εκτίμηση αύξησης εσόδων τα επόμενα 5 χρόνια
  - Κεφάλαιο κίνησης δια των πωλήσεων

Περισσότερες πληροφορίες

- NeuroNet-II, European Network of Excellence in Neural Networks
  - http://www.kcl.ac.uk/neuronet/

Γενετικοί Αλγόριθμοι

Γενικά

- Διάφορες ονομασίες:
  - Γενετικοί αλγόριθμοι (Genetic algorithms)
  - Εξελικτικοί αλγόριθμοί (Evolutionary algorithms)
  - Εξελικτικά προγράμματα (Evolutionary programs)
- Πρόκειται για περιοχή της Τεχνητής Νοημοσύνης
- Μιμείται τη βιολογική διεργασία της εξέλιξης:
  - Τα είδη του ζωικού και φυτικού βασιλείου μεταλλάσσονται και στις επόμενες γενεές επικρατούν τα ισχυρότερα.
- Εφαρμογές:
  - Βελτιστοποίηση
  - Μηχανική μάθηση

Κεντρική ιδέα

- Κανόνας της φυσικής επιλογής:
  - Οι οργανισμοί που δε μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι πολλαπλασιάζονται μέσω της αναπαραγωγής.
- Οι απόγονοι παρουσιάζουν μικρές διαφοροποιήσεις από τους προγόνους τους, ενώ συνήθως υπερισχύουν αυτοί που συγκεντρώνουν τα καλύτερα χαρακτηριστικά.
- Σποραδικά συμβαίνουν τυχαίες μεταλλάξεις, από τις οποίες οι περισσότερες οδηγούν τα μεταλλαγμένα άτομα στο θάνατο, αν και είναι πιθανό, πολύ σπάνια όμως, να οδηγήσουν στη δημιουργία νέων "καλύτερων" οργανισμών.

Ο βασικός αλγόριθμος

- Έστω το σύνολο P(t) των υποψηφίων λύσεων τη χρονική στιγμή t:
  - $P(t)=\{x_1^t, x_2^t, ..., x_n^t\}$
- Έστω t=0
- Αρχικοποίησε τον πληθυσμό P(0) με τυχαίες υποψήφιες λύσεις.
- Όσο δεν πληρείται κάποιο κριτήριο τερματισμού:
  - Αξιολόγησε όλα τα μέλη του πληθυσμού P(t) με μία συνάρτηση καταλληλότητας (fitness function)
  - Επέλεξε μερικά ζεύγη από τα μέλη του P(t) για αναπαραγωγή, λαμβάνοντας υπόψη και την καταλληλότητα των μελών του P(t).
  - Παρήγαγε τους απογόνους των παραπάνω ζευγών χρησιμοποιώντας διάφορους γενετικούς τελεστές.
  - Εισήγαγε τους απογόνους στον πληθυσμό, διαγράφοντας αντίστοιχο αριθμό υπαρχόντων μελών.
  - Θέσε t=t+1

Παρατηρήσεις

- Το σύνολο P(0) αποτελείται από υποψήφιες λύσεις, πολλές από τις οποίες είτε:
  - δεν είναι αποδεκτές, ή
  - είναι χαμηλής ποιότητας.
- Η λειτουργία των γενετικών αλγορίθμων έχει τη λογική του αλγορίθμου αναζήτησης αναρρίχησης λόφων.
  - Η ύπαρξη πολλών υποψηφίων λύσεων και η ταυτόχρονη εξέλιξή τους μειώνει τις πιθανότητες παγίδευσης σε τοπικά ακρότατα, αν και πάντα υπάρχει πιθανότητα να μην βρεθεί το ολικό ακρότατο.
- Υπάρχουν τέσσερα κρίσιμα θέματα σχετικά με τους γενετικούς αλγορίθμους, τα οποία θα εξεταστούν στις επόμενες διαφάνειες:
  - Αναπαράσταση υποψηφίων λύσεων
  - Συνάρτηση καταλληλότητας
  - Επιλογή γονέων
  - Αναπαραγωγή

Παράδειγμα (1/6)

- Θα μελετήσουμε την κλασσική μορφή των γενετικών αλγορίθμων χρησιμοποιώντας ένα παράδειγμα προβλήματος ικανοποίησης περιορισμών.
- Έστω ότι θέλουμε να βρούμε μια ανάθεση τιμών (αληθές/ψευδές) στις μεταβλητές a, b, c, d, e και f, ώστε να ικανοποιείται η έκφραση:
  - $(\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (\neg e \vee f)$
- Φυσικά το παραπάνω παράδειγμα μπορεί να λυθεί και με όλους τους γνωστούς αλγορίθμους αναζήτησης.

Αναπαράσταση υποψηφίων λύσεων (1/2)

- Στην κλασσική αναπαράσταση όλη η πληροφορία αναπαρίσταται σαν μια συμβολοσειρά από bits (bit-string).
- Στο πρόβλημα ικανοποίησης περιορισμών, μια φυσιολογική αναπαράσταση είναι να αντιστοιχίσουμε ένα bit σε κάθε μεταβλητή.
- Για παράδειγμα, μια υποψήφια λύση θα μπορούσε να είναι η:
  - 101010
- η οποία αποδίδει την τιμή Αληθές στις μεταβλητές a, c και e και την τιμή Ψευδές στις b, d και f.

Αναπαράσταση υποψηφίων λύσεων (2/2)

- Στην ορολογία των γενετικών αλγορίθμων κάθε bit των υποψηφίων λύσεων ονομάζεται *γονίδιο* (gene).
- Ολόκληρη η συμβολοσειρά αναφέρεται ως χρωμόσωμα (chromosome).
  - Σε επόμενες διαφάνειες θα δούμε αναπαραστάσεις όπου η βασική μονάδα πληροφορίας δεν είναι το bit, αλλά άλλες δομές δεδομένων όπως ακέραιοι, πραγματικοί αριθμοί κλπ.

Συνάρτηση καταλληλότητας (1/4)

- Η συνάρτηση καταλληλότητας (fitness function) πρέπει να βαθμολογεί τις υποψήφιες λύσεις, διακρίνοντας αυτές που είναι κοντά σε μια τελική λύση από αυτές που απέχουν πάρα πολύ.
- Η συνάρτηση καταλληλότητας παίζει σημαντικό ρόλο στην επιβίωση ή μη των διάφορων υποψηφίων λύσεων.
- Ο καθορισμός της συνάρτησης καταλληλότητας είναι στενά συνδεδεμένος με το εκάστοτε πρόβλημα.
- Συνήθως μεγαλύτερες τιμές στην συνάρτηση καταλληλότητας δηλώνουν καλύτερες υποψήφιες λύσεις.

Συνάρτηση καταλληλότητας (2/4)

- Για προβλήματα ικανοποίησης περιορισμών, συνηθίζεται ως συνάρτηση καταλληλότητας να θεωρείται το πλήθος των περιορισμών που ικανοποιεί κάθε υποψήφια λύση.
- Για παράδειγμα, η υποψήφια λύση 101010 βαθμολογείται με 4, αφού στην πρόταση
  - $(\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (\neg e \vee f)$
- ικανοποιεί 4 από τους 5 περιορισμούς.
- Ουσιαστικά η συνάρτηση καταλληλότητας είναι το αντίστοιχο της ευρετικής συνάρτησης των αλγορίθμων αναζήτησης, οπότε ισχύουν για αυτήν όλα τα σχόλια που έχουν γίνει για τις ευρετικές συναρτήσεις στο μάθημα της Τεχνητής Νοημοσύνης (π.χ. λεπτής/χοντρής υφής κλπ).

Συνάρτηση καταλληλότητας (3/4)

- Σύγκλιση (convergence) ονομάζεται το φαινόμενο της επικράτησης ενός χρωμοσώματος ή μικρών παραλλαγών του, σε μεγάλο ποσοστό στον πληθυσμό.
  - Ένα γονίδιο συγκλίνει, όταν έχει την ίδια τιμή στο 95% των χρωμοσωμάτων.
  - Ένας πληθυσμός συγκλίνει, όταν όλα τα γονίδιά του έχουν συγκλίνει.
- Πρόωρη σύγκλιση (premature convergence): Ο πληθυσμός συγκλίνει πολύ γρήγορα γύρω από ένα χρωμόσωμα, το οποίο όμως αποτελεί τοπικό (όχι ολικό) ακρότατο.
- Αργή σύγκλιση (slow convergence): Μετά από μεγάλο αριθμό επαναλήψεων ο πληθυσμός δεν έχει συγκλίνει.

Συνάρτηση καταλληλότητας (4/4)

- Το φαινόμενο της πρόωρης ή μη σύγκλισης έχει να κάνει με τη συνάρτηση καταλληλότητας.
- Εάν η συνάρτηση καταλληλότητας έχει απότομες κορυφές, είναι πολύ πιθανό να παρατηρηθεί το φαινόμενο της πρόωρης σύγκλισης.
- Εάν η συνάρτηση καταλληλότητας είναι αρκετά ομαλή και χωρίς μεγάλες διαφορές, είναι πολύ πιθανό να έχουμε αργή σύγκλιση.
- Το πρόβλημα λύνεται με το μετασχηματισμό της συνάρτησης καταλληλότητας.
  - Για παράδειγμα, η προσθήκη μιας θετικής σταθεράς στη συνάρτηση καταλληλότητας, π.χ. **eval'(S)=eval(S)+C**, εξομαλύνει τις διαφορές μεταξύ των χρωμοσωμάτων, όπου **eval** η συνάρτηση καταλληλότητας (από τη λέξη evaluate).

Επιλογή γονέων (1/4)

- Η διαδικασία επιλογής γονέων έχει να κάνει με τον τρόπο που επιλέγονται τα χρωμοσώματα για αναπαραγωγή.
- Η πιθανότητα να επιλεγεί ένα χρωμόσωμα προς αναπαραγωγή συνδέεται άμεσα με την καταλληλότητά του.
- Σε κάθε γενιά, ο αριθμός των χρωμοσωμάτων που επιλέγονται για αναπαραγωγή δεν είναι απαραίτητα ίσος με τον πληθυσμό τους.
- Είναι μάλιστα δυνατόν, μέσα στην ίδια γενιά, ένα χρωμόσωμα να επιλεγεί περισσότερες από μια φορές, ενώ κάποιο άλλο να μην επιλεγεί καθόλου.

Επιλογή γονέων (2/4)

- Μια τεχνική επιλογής χρωμοσωμάτων προς αναπαραγωγή, με πιθανότητα ανάλογη της καταλληλότητάς τους, είναι αυτή της ρουλέτας:
  1. Παράγεται το άθροισμα S όλων των τιμών αξιολόγησης των υποψηφίων λύσεων.
  2. Επιλέγεται ένας τυχαίος αριθμός n, από το 0 μέχρι το S, χρησιμοποιώντας συνάρτηση ομοιόμορφης κατανομής για τη δημιουργία των τυχαίων αριθμών. Θέτουμε K=0.
  3. Εξετάζεται η επόμενη υποψήφια λύση (ξεκινώντας από την πρώτη) και η τιμή της προστίθεται στον καταχωρητή K.
  4. Αν η τιμή του K γίνει μεγαλύτερη ή ίση του n, η λύση επιλέγεται. Στην αντίθετη περίπτωση εκτελείται πάλι το 3.
  5. Εάν συμπληρώθηκε ο επιθυμητός αριθμός επιλεγμένων υποψηφίων λύσεων ο αλγόριθμος τερματίζει, αλλιώς συνεχίσει με το βήμα 2.

Επιλογή γονέων (3/4)

- **Χάσμα γενεών**: Ονομάζουμε το ποσοστό των χρωμοσωμάτων που ανανεώνονται σε κάθε γενιά, προς το σύνολο των χρωμοσωμάτων της γενιάς.
- Στην αρχική εκδοχή των γενετικών αλγορίθμων το παραπάνω ποσοστό ισούται με τη μονάδα.
- Σε νέωτερες υλοποιήσεις συνηθίζεται σε κάθε γενιά να ανανεώνεται ένα μόνο μέρος των χρωμοσωμάτων (**μέθοδος μερικής ανανέωσης**).
- Σε αυτή την περίπτωση τίθεται πλέον και θέμα επιλογής των χρωμοσωμάτων που θα "πεθάνουν".

Επιλογή γονέων (4/4)

- Στην περίπτωση μερικής ανανέωσης του πληθυσμού των χρωμοσωμάτων, υπάρχουν οι εξής προσεγγίσεις:
  - Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητά τους και τυχαία επιλογή των γονέων που θα αποχωρήσουν.
  - Επιλογή των γονέων προς αναπαραγωγή τυχαία και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητά τους.
  - Επιλογή των γονέων προς αναπαραγωγή με πιθανότητα ανάλογη προς την καταλληλότητά τους και επιλογή των γονέων που θα αποχωρήσουν με πιθανότητα αντιστρόφως ανάλογη προς την καταλληλότητά τους.
- Όπως φαίνεται από τα παραπάνω, σε καμία περίπτωση δεν θεωρείται ότι τα χρωμοσώματα που θα αποχωρήσουν είναι αυτά που επελέγησαν για αναπαραγωγή.

Αναπαραγωγή (1/4)

- Αναπαραγωγή είναι η διαδικασία δημιουργίας δύο νέων χρωμοσωμάτων (απόγονοι) από δύο υπάρχοντα χρωμοσώματα (γονείς).
- Υπάρχουν διάφορες τεχνικές, οι οποίες εντάσσονται στις παρακάτω δύο γενικές κατηγορίες:
  - Διασταύρωση (crossover)
  - Μετάλλαξη (mutation)

Αναπαραγωγή (2/4)

- Κατά τη διασταύρωση:
  - Για κάθε ζευγάρι γονέων επιλέγεται τυχαία ένα σημείο διασταύρωσης (cross-point), δηλαδή μία θέση μέσα στα χρωμοσώματα (η ίδια και στα δύο).
  - Τα δύο χρωμοσώματα "σπάζουν" σε δύο κομμάτια το καθένα.
  - Το αριστερό κομμάτι του πρώτου χρωμοσώματος ενώνεται με το δεξιό κομμάτι του δεύτερου και αντίστροφα.

Αναπαραγωγή (3/4)

- Έστω για παράδειγμα τα παρακάτω δύο χρωμοσώματα από το πρόβλημα ικανοποίησης περιορισμών:
  - **011010**
  - **101001**
- Έστω ότι επιλέγεται ως σημείο διασταύρωσης η θέση 4.
- Κάθε χρωμόσωμα-γονέας σπάει σε 2 κομμάτια, στη θέση 4:
  - **0110-10**
  - **1010-01**
- Τα χρωμοσώματα-απόγονοι που προκύπτουν είναι τα ακόλουθα:
  - **011001**
  - **101010**

Αναπαραγωγή (4/4)

- Η μετάλλαξη είναι η τυχαία αλλαγή της τιμής ενός γονιδίου ενός χρωμοσώματος.
- Η μετάλλαξη, όπως και στους βιολογικούς οργανισμούς, εμφανίζεται με σχετικά πολύ μικρή πιθανότητα, π.χ. 10^{-3} για κάθε γονίδιο.
- Συνήθως εκτελείται στα γονίδια των χρωμοσωμάτων-απογόνων που προκύπτουν μετά τη διασταύρωση.
- Για παράδειγμα, έστω το χρωμόσωμα 011001 που προέκυψε μετά την πρώτη διασταύρωση στην προηγούμενη διασταύρωση.
- Μετά από μία μετάλλαξη, π.χ. στο 2ο γονίδιό του, το χρωμόσωμα γίνεται 001001.
  - Τις περισσότερες φορές δεν γίνεται καμία μετάλλαξη, ωστόσο είναι δυνατό να συμβούν περισσότερες από μία μεταλλάξεις στο ίδιο χρωμόσωμα.

Συνθήκες τερματισμού

- Υπάρχουν δύο κατηγορίες συνθηκών τερματισμού:
  - Η πρώτη κατηγορία βασίζεται στο ποσοστό των χρωμοσωμάτων που έχουν συγκλίνει και τερματίζει όταν αυτό ξεπεράσει ένα συγκεκριμένο όριο (π.χ. 95%).
  - Η δεύτερη κατηγορία μετρά τη βελτίωση του μέσου όρου της καταλληλότητας για όλα τα χρωμοσώματα ενός πληθυσμού και τερματίζει όταν για έναν αριθμό διαδοχικών πληθυσμών δεν παρατηρηθεί βελτίωση.

Παράδειγμα (2/6)

- Επανερχόμαστε στο παράδειγμα της διαφάνειας 6.
- Για να βρεθεί μια λύση στο πρόβλημα, χρειάζεται καταρχήν να δημιουργηθεί ένας αρχικός πληθυσμός.
- Έστω 4 το μέγεθος του πληθυσμού.
  - Σημειώστε ότι το συνολικό πλήθος των διαφορετικών διανυσμάτων είναι 2^5=64, ενώ υπάρχουν περισσότερες της μιας λύσεις στο πρόβλημα. Έτσι δεν έχει νόημα να έχουμε έναν πολύ μεγάλο πληθυσμό, στον οποίο εξαρχής θα υπάρχει το χρωμόσωμα-λύση.
- Έστω τα παρακάτω αρχικά χρωμοσώματα:
  - **010011** με βαθμό **3** και αθροιστική καταλληλότητα **3**
  - **101010** με βαθμό **4** και αθροιστική καταλληλότητα **7**
  - **010010** με βαθμό **2** και αθροιστική καταλληλότητα **9**
  - **110101** με βαθμό **4** και αθροιστική καταλληλότητα **13**
    - $(\neg a \vee c) \wedge (\neg a \vee c \vee \neg e) \wedge (\neg b \vee c \vee d \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (\neg e \vee f)$

Παράδειγμα (3/6)

- Ως συνθήκη τερματισμού του γενετικού αλγορίθμου ορίζουμε την εύρεση ενός χρωμοσώματος με βαθμό 5.
- Υποθέτουμε ότι από κάθε γενιά επιλέγονται 4 χρωμοσώματα προς αναπαραγωγή, ενώ όλα τα χρωμοσώματα της προηγούμενης γενιάς αποχωρούν.
- Επιλέγουμε ένα-ένα τα χρωμοσώματα προς αναπαραγωγή, χρησιμοποιώντας την τεχνική της ρουλέτας:
  - Επιλέγεται τυχαία και από ομοιόμορφη κατανομή ένας αριθμός από το 0 μέχρι το 13, π.χ. έστω Κ=5 ο αριθμός αυτός.
  - Βρίσκεται το χρωμόσωμα εκείνο με τη μικρότερη αθροιστική καταλληλότητα, η οποία όμως είναι μεγαλύτερη από την τιμή Κ (στην προκειμένη περίπτωση το 2ο χρωμόσωμα).

Παράδειγμα (4/6)

- Η διαδικασία αυτή επαναλαμβάνεται τόσες φορές, όσα χρωμοσώματα θέλουμε για αναπαραγωγή, στην προκειμένη περίπτωση 4 φορές.
- Έστω ότι επελέγησαν για αναπαραγωγή τα χρωμοσώματα 2,4,2,1.
  - Παρατηρούμε ότι το χρωμόσωμα 3 δεν επελέγη, ενώ το χρωμόσωμα 2 επελέγη 2 φορές.
    - Το παραπάνω αποτέλεσμα είναι σε συμφωνία με τις τιμές καταλληλότητας των χρωμοσωμάτων, μιας και η τεχνική της ρουλέτας ευνοεί τα χρωμοσώματα με υψηλές τιμές καταλληλότητας, ώστοσο θα μπορούσε να είχε συμβεί και το ακριβώς αντίθετο, δηλαδή να επιλεγεί μία ή περισσότερες φορές ένα χρωμόσωμα με χαμηλή καταλληλότητα και να μην επιλεγεί καθόλου ένα χρωμόσωμα με υψηλή καταλληλότητα.

Παράδειγμα (5/6)

- Έστω ότι επιλέγουμε να κάνουμε τις διασταυρώσεις 2-4 και 2-1.
- Για κάθε διασταύρωση πρέπει να καθορισθεί και το σημείο διασταύρωσης, έστω η θέση 3 για την 2-4 και η θέση 5 για την 2-1.
- Από τα χρωμοσώματα:
  - **1: 010011** **2: 101010**
  - **3: 010010** **4: 110101**
- και για τις παραπάνω διασταυρώσεις/σημεία διασταύρωσης προκύπτουν οι παρακάτω απόγονοι:
  - (2-4): 101101 με βαθμό 5 110010 με βαθμό 1
  - (2-1): 101011 με βαθμό 5 010010 με βαθμό 3
- Υποθέτοντας ότι δεν συμβαίνει καμία μετάλλαξη στα χρωμοσώματα-απογόνους, αυτά αποτελούν τον νέο πληθυσμό.

Παράδειγμα (6/6)

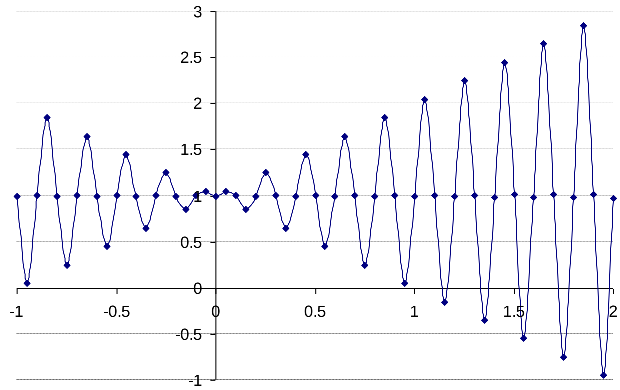
- Ο νέος πληθυσμός περιλαμβάνει χρωμοσώματα με βαθμό 5, άρα ικανοποιείται η συνθήκη τερματισμού και επιστρέφεται ως λύση ένα από αυτά.
  - Εάν δεν είχε ικανοποιηθεί η συνθήκη τερματισμού, η διαδικασία θα επαναλαμβανόταν με τα χρωμοσώματα του νέου πληθυσμού.
- Παρατηρούμε ότι ο μέσος όρος της καταλληλότητας των χρωμοσωμάτων του πρώτου πληθυσμού ήταν 13/4=3.25, ενώ ο μέσος όρος της καταλληλότητας των χρωμοσωμάτων του νέου πληθυσμού έχει αυξηθεί σε 16/4=4.
- Η αύξηση οφείλεται στον τρόπο επιλογής των χρωμοσωμάτων προς διασταύρωση από την τεχνική της ρουλέτας.

Ειδικές περιπτώσεις

- Στις προηγούμενες διαφάνειες παρουσιάστηκε η γενική φιλοσοφία λειτουργίας των γενετικών αλγορίθμων και οι βασικές τεχνικές.
- Ωστόσο σε πολλά προβλήματα οι βασικές τεχνικές δεν δίνουν καλά αποτελέσματα.
- Χρειάζονται λοιπόν προσαρμογές τόσο στον τρόπο αναπαράστασης των υποψηφίων λύσεων, όσο και στο σχεδιασμό των γενετικών τελεστών.
  - Δυστυχώς οι ειδικές αυτές περιπτώσεις είναι ο κανόνας και όχι η εξαίρεση...
- Στις επόμενες διαφάνειες θα παρουσιαστούν και θα σχολιαστούν τέτοιες ειδικές περιπτώσεις.

Παράδειγμα: Εύρεση μεγίστου συνάρτησης μιας μεταβλητής (1/4)

- Έστω η αριθμητική συνάρτηση μιας μεταβλητής $f(x)=x\cdot\sin(10\cdot\pi\cdot x)+1$ της οποίας αναζητούμε το μέγιστο στο διάστημα [-1,2].



Παράδειγμα: Εύρεση μεγίστου συνάρτησης μιας μεταβλητής (2/4)

- Θα χρησιμοποιήσουμε γενετικούς αλγορίθμους για την επίλυση του προβλήματος.
- Μολονότι το πεδίο τιμών είναι συνεχές, η ανάγκη για δυαδική αναπαράσταση μας υποχρεώνει να το διακριτοποιήσουμε.
- Έστω ότι ζητούμε ακρίβεια 6 δεκαδικών ψηφίων.
- Χρειαζόμαστε 22 δυαδικά ψηφία για να αναπαραστήσουμε $3\cdot 10^6$ αριθμούς που βρίσκονται στο διάστημα [-1,2].
  - Πράγματι: $2^{21}<3\cdot 10^6<2^{22}$

Παράδειγμα: Εύρεση μεγίστου συνάρτησης μιας μεταβλητής (3/4)

- Έστω $b_{21}b_{20}...b_1b_0$ ένας δυαδικός αριθμός με 22 ψηφία.
- Έστω x' ο δεκαδικός αριθμός που ισούται με τον παραπάνω δυαδικό.
- Η αντιστοίχιση του x' σε έναν αριθμό x στο διάστημα [1,2] γίνεται με γραμμική παρεμβολή με τον παρακάτω τύπο:
  - $x=-1+x'\cdot 3/(2^{22}-1)$
- Η συνάρτηση καταλληλότητας είναι η ίδια η συνάρτηση f(x) εφαρμοζόμενη στον δεκαδικό αριθμό x που αντιστοιχεί στο χρωμόσωμα.

Παράδειγμα: Εύρεση μεγίστου συνάρτησης μιας μεταβλητής (4/4)

- Έστω αρχικός πληθυσμός 50 τυχαίων χρωμοσωμάτων, από τα οποία το 25% διασταυρώνεται σε κάθε γενεά ενώ υπάρχει πιθανότητα μετάλλαξης 10^{-2}.
- Σε κάθε γενεά εντοπίζεται το καλύτερο χρωμόσωμα. Μερικά αποτελέσματα είναι τα παρακάτω:
  - 1η γενιά: max(f(x))=1.44
  - 10η γενιά: max(f(x))=2.25
  - 50η γενιά: max(f(x))=2.73
  - 100η γενιά: max(f(x))=2.84
  - 150η γενιά: max(f(x))=2.85
- Ως συνθήκη τερματισμού θεωρήθηκε η μη-σημαντική βελτίωση του πληθυσμού για διαδοχικές γενεές.

Μετάλλαξη σε δυαδικές αριθμητικές κωδικοποιήσεις (1/2)

- Σε αριθμητικά προβλήματα στα οποία χρησιμοποιείται δυαδική κωδικοποίηση, η τεχνική της μετάλλαξης παρουσιάζει αδυναμίες.
- Το πρόβλημα οφείλεται στο ότι όλα τα bits του χρωμοσώματος δεν είναι εξίσου σημαντικά.
- Πράγματι, έστω ένα χρωμόσωμα το οποίο έχει πολύ καλή τιμή καταλληλότητας.
- Εάν γίνει μετάλλαξη σε ένα μη-σημαντικό bit (δεξιά bits) τότε προκύπτει μικρή αλλαγή στον αριθμό που αναπαριστά το χρωμόσωμα και άρα μικρή αλλαγή στην καταλληλότητα του νέου χρωμοσώματος.

Μετάλλαξη σε δυαδικές αριθμητικές κωδικοποιήσεις (2/2)

- Εάν γίνει μετάλλαξη σε ένα σημαντικό bit (αριστερά bits) τότε προκύπτει μεγάλη αλλαγή στον αριθμό που αναπαριστά το χρωμόσωμα και είναι δυνατόν το νέο χρωμόσωμα να έχει πολύ διαφορετική τιμή καταλληλότητας.
- Μια καλή λύση είναι να ορισθούν μικρότερες πιθανότητες μετάλλαξης για τα σημαντικά bits.
  - Για παράδειγμα, εάν το ψηφίο b_0 έχει πιθανότητα μετάλλαξης 10^{-2}, τότε το b_1 θα έχει $2^{-1}10^{-2}$ και το b_{21} θα έχει $2^{-21}10^{-2}$.
- Η παραπάνω τεχνική ονομάζεται ανομοιόμορφη μετάλλαξη (non-uniform mutation).

Διασταύρωση σε προβλήματα αριθμητικής βελτιστοποίησης (1/2)

- Η κλασσική διασταύρωση σε προβλήματα αριθμητικής βελτιστοποίησης παρουσιάζουν αδυναμία προσέγγισης με μεγάλη ακρίβεια των μεγίστων. Πράγματι:
  - Έστω x1 και x2 δύο χρωμοσώματα κοντά σε ένα μέγιστο.
  - Τα χρωμοσώματα κατά πάσα πιθανότητα είναι σχεδόν ίδια, με εξαίρεση τα τελευταία, λιγότερο σημαντικά τους bits.
  - Έστω ότι τα χρωμοσώματα αυτά επιλέγονται για διασταύρωση και το σημείο διασταύρωσης δεν είναι ανάμεσα στα λιγότερο σημαντικά bits.
  - Το αποτέλεσμα είναι ότι τα χρωμοσώματα-απόγονοι είναι ίδια με τα χρωμοσώματα-γονείς!

Διασταύρωση σε προβλήματα αριθμητικής βελτιστοποίησης (2/2)

- Για το λόγο αυτό έχουν προταθεί διάφορες παραλλαγές της διασταύρωσης, όπως η αριθμητική διασταύρωση (arithmetical crossover).
- Σύμφωνα με αυτή, από δύο χρωμοσώματα-γονείς x1 και x2 προκύπτουν τα δύο χρωμοσώματα-παιδιά σύμφωνα με τους τύπους:
  - $x1'=a\cdot x1+(1-a)\cdot x2$
  - $x2'=a\cdot x2+(1-a)\cdot x1$
- όπου a τυχαίος αριθμός που επιλέγεται για κάθε διασταύρωση από ομοιόμορφη κατανομή στο [-1,2].

Το πρόβλημα του πλανόδιου πωλητή (1/5)

- Θα προσπαθήσουμε να λύσουμε το γνωστό πρόβλημα του πλανόδιου πωλητή χρησιμοποιώντας γενετικούς αλγορίθμους.
- Στο πρόβλημα αυτό η χρήση δυαδικής κωδικοποίησης και η εφαρμογή των τεχνικών της διασταύρωσης και της μετάλλαξης στην αρχική τους μορφή θα δημιουργούσαν πολλά προβλήματα.
- Επιλέγεται λοιπόν καταρχήν μια αναπαράσταση με ακεραίους.
- Έτσι λοιπόν ένα χρωμόσωμα είναι ένα διάνυσμα ακεραίων της μορφής:
  - $<i_1, i_2, ..., i_N>$
- όπου
  - N ο αριθμός των πόλεων,
  - $i_1, i_2, ..., i_N \in 1..N$
  - $i_j \neq i_k$ για $j \neq k$

Το πρόβλημα του πλανόδιου πωλητή (2/5)

- Το μεγαλύτερο πρόβλημα που εμφανίζεται είναι αυτό της αναπαραγωγής.
- Τόσο η διασταύρωση, όσο και η μετάλλαξη οδηγούν σε άκυρα χρωμοσώματα.
- Υπάρχουν δύο λύσεις:
  - Ή διατηρούνται οι τεχνικές αναπαραγωγής ως έχουν και γίνεται επιδιόρθωση των άκυρων χρωμοσωμάτων.
  - Ή τροποποιούνται οι τεχνικές αναπαραγωγής ώστε να δίνουν πάντα έγκυρα χρωμοσώματα.
- Θα δούμε καταρχήν τη δεύτερη προσέγγιση.

Το πρόβλημα του πλανόδιου πωλητή (3/5)

- Οι βασικές ιδιότητες της διασταύρωσης και της μετάλλαξης, τις οποίες θέλουμε να διατηρήσουμε σε οποιαδήποτε τροποποίησή τους είναι οι εξής:
  - Η διασταύρωση απαιτεί πάντα δύο γονείς και παράγει δύο παιδιά.
  - Η μετάλλαξη απαιτεί ένα χρωμόσωμα και παράγει ένα μεταλλαγμένο χρωμόσωμα.
- Μια τροποποιημένη μετάλλαξη, η οποία διατηρεί την εγκυρότητα των χρωμοσωμάτων είναι η εξής:
  - Εάν μια πόλη ενός χρωμοσώματος επιλεγεί για μετάλλαξη, τότε ανταλλάσει θέση με τη διπλανή της (ή και με οποιαδήποτε άλλη).

Το πρόβλημα του πλανόδιου πωλητή (4/5)

- Θα περιγράψουμε τώρα τους τροποποιημένους τελεστές διασταύρωσης, χρησιμοποιώντας τα χρωμοσώματα:
  - p1=(1 2 3 4 5 6 7 8 9)
  - p2=(4 5 2 1 8 7 6 9 3)
- Αρχικά επιλέγονται τυχαία <u>δύο</u> σημεία διασταύρωσης σε κάθε χρωμόσωμα:
  - p1=(1 2 3 | 4 5 6 7 | 8 9)
  - p2=(4 5 2 | 1 8 7 6 | 9 3)
- Στη συνέχεια δημιουργείται η αρχική μορφή των απογόνων, διατηρώντας τα μεσσαία τμήματα των αρχικών χρωμοσωμάτων.
  - o1=(x x x | 4 5 6 7 | x x)
  - o2=(x x x | 1 8 7 6 | x x)

Το πρόβλημα του πλανόδιου πωλητή (5/5)

- p1=(1 2 3 | 4 5 6 7 | 8 9)
- p2=(4 5 2 | 1 8 7 6 | 9 3)
- o1=(x x x | 4 5 6 7 | x x)
- o2=(x x x | 1 8 7 6 | x x)

- Στη συνέχεια, κάθε απόγονος συμπληρώνει τις πόλεις που του λείπουν από τον δεύτερο γονέα του, διατηρώντας τη σειρά των πόλεων του δεύτερου γονέα και προσέχοντας να μην επαναλάβει καμία πόλη:
  - o1=(2 1 8 | 4 5 6 7 | 9 3)
  - o2=(3 4 5 | 1 8 7 6 | 9 2)

Ορισμός νέων τελεστών διασταύρωσης

- Σε πολλά προβλήματα χρειάζεται να ορίζονται νέοι τελεστές διασταύρωσης.
- Αυτό μπορεί να γίνει είτε για να παράγονται έγκυρα χρωμοσώματα-απόγονοι (όπως στην περίπτωση του προβλήματος του πλανόδιου πωλητή) ή για να παράγονται καλύτερα χρωμοσώματα-απόγονοι (όπως στην περίπτωση της αριθμητικής διασταύρωσης).
- Σε κάθε περίπτωση, ένας νέος τελεστής διασταύρωσης θα πρέπει να εξασφαλίζει την παρακάτω σημαντική ιδιότητα:
  - Από δύο καλά χρωμοσώματα-γονείς θα πρέπει να υπάρχει σχετικά μεγάλη πιθανότητα να προκύψουν εξίσου καλά χρωμοσώματα-απόγονοι.

Ταξινομημένη αναπαράσταση (1/3)

- Στο πρόβλημα του πλανόδιου πωλητή χρησιμοποιήθηκε η συνηθισμένη κωδικοποίηση (με τη μόνη διαφορά της χρήσης ακεραίων και όχι δυαδικών γονιδίων) και ορίστηκαν νέοι τελεστές αναπαραγωγής.
- Θα μπορούσε να γίνει και το αντίθετο, δηλαδή να οριστεί νέα κωδικοποίηση στην οποία οι κλασικοί τελεστές αναπαραγωγής να παράγουν έγκυρα χρωμοσώματα.
- Μια τέτοια είναι η **ταξινομημένη αναπαράσταση** (ordinal representation).

Ταξινομημένη αναπαράσταση (2/3)

- Στην ταξινομημένη αναπαράσταση ορίζουμε καταρχήν μια διάταξη των πόλεων, έστω 1,2,3,...9, εφόσον έχουμε 9 πόλεις.
- Στη συνέχεια, ένα χρωμόσωμα (δηλαδή μία διαδρομή του πλανόδιου πωλητή) αποτελείται από 9 αριθμούς:
  - $<i_1, i_2, \ldots i_9>$
- τέτοιους ώστε ο αριθμός i_j να βρίσκεται στο διάστημα 1..9-j+1.
- Για παράδειγμα, το παρακάτω είναι ένα έγκυρο χρωμόσωμα στην ταξινομημένη κωδικοποίηση:
  - <1 1 2 1 4 1 3 1 1>

Ταξινομημένη αναπαράσταση (3/3)

- Κάθε γονίδιο στην ταξινομημένη αναπαράσταση αντιστοιχεί στη θέση της πόλης στην αρχική διάταξη των πόλεων, εφόσον αφαιρεθούν οι πόλεις που δηλώνονται από τα προηγούμενα (προς τα αριστερά) γονίδια.
- Έτσι το χρωμόσωμα <1 1 2 1 4 1 3 1 1> αντιστοιχεί στη διαδρομή <1 2 4 3 8 5 9 6 7>.
- Το βασικό πλεονέκτημα της ταξινομημένης αναπαράστασης είναι ότι η κλασσική διασταύρωση λειτουργεί πολύ καλά και παράγει έγκυρα χρωμοσώματα-απογόνους.
  - Δεν συμβαίνει όμως το ίδιο και με την κλασσική μετάλλαξη.
- Δυστυχώς, πειραματικά αποτελέσματα έδειξαν ότι η ταξινομημένη αναπαράσταση δεν παράγει καλά αποτελέσματα για το πρόβλημα του πλανόδιου πωλητή.

Θεώρημα των Σχημάτων (1/6)

- Το "θεώρημα" των σχημάτων (J. Holland το 1975) είναι μια προσπάθεια ερμηνείας της σύγκλισης των γενετικών αλγορίθμων.
  - Το θεώρημα δεν καλύπτει όλες τις εκδοχές γενετικών αλγορίθμων. Εξάλλου δεν συγκλίνουν όλοι οι γενετικοί αλγόριθμοι!
- Ένα σχήμα είναι ένα χρωμόσωμα, του οποίου τα γονίδια αποτελούνται από το αλφάβητο του προβλήματος συν το σύμβολο #.
- Μερικά σχήματα (για διάφορα προβλήματα και αναπαραστάσεις) είναι τα ακόλουθα:
  - #10##
  - 1#00#
  - ###7561##
- Το σύμβολο # είναι χαρακτήρας "μπαλαντέρ" και μπορεί να "αντικατασταθεί" από οποιοδήποτε σύμβολο του αλφαβήτου.

Θεώρημα των Σχημάτων (2/6)

- **Τάξη** (order) ενός σχήματος S, έστω **o(S)**, ορίζεται το πλήθος των συμβόλων του σχήματος που είναι διάφορα του #.
- **Μήκος** (length) ενός σχήματος S, έστω **δ(S)**, ορίζεται η μέγιστη απόσταση μεταξύ δύο μη-# χαρακτήρων του σχήματος.
- Συμβολίζουμε με **ξ(S,t)** το πλήθος των χρωμοσωμάτων του πληθυσμού που ταιριάζουν με ένα σχήμα S τη χρονική στιγμή t.
- Η αξία ενός σχήματος, **eval(S,t)**, είναι ο μέσος όρος της καταλληλότητας όλων των χρωμοσωμάτων του συνόλου ξ(S,t).
- Συμβολίζουμε με **F(t)** το άθροισμα των τιμών καταλληλότητας όλων των χρωμοσωμάτων του πληθυσμού τη χρονική στιγμή t.
- Τέλος έστω **pop_size** το πλήθος των χρωμοσωμάτων του πληθυσμού.

Θεώρημα των Σχημάτων (3/6)

- Εάν θεωρήσουμε ότι τα σχήματα τα οποία επιλέγονται για αναπαραγωγή (μέσω των αντίστοιχων χρωμοσωμάτων του συνόλου ξ(S,t)) μεταφέρονται με μεγάλη πιθανότητα στον επόμενο πληθυσμό, μπορούμε να γράψουμε την εξής σχέση:
  - ξ(S,t+1)=ξ(S,t)·pop_size·eval(S,t)/F(t)
- Η παραπάνω σχέση λέει ότι για σχήματα με μέση τιμή καταλληλότητας eval(S,t) μεγαλύτερη από το μέσο όρο της τιμής καταλληλότητας όλου του πληθυσμού F(t)/pop_size, ο πληθυσμός τους ξ(S,t+1) στο σύνολο του πληθυσμού pop_size συνεχώς αυξάνει.

Θεώρημα των Σχημάτων (4/6)

- Στην ανάλυση που προηγήθηκε δεν ελήφθη υπόψη ο αριθμός των σχημάτων που επιλέγονται για αναπαραγωγή, αλλά καταστρέφονται λόγω των διαδικασιών της διασταύρωσης και της μετάλλαξης.
- Η πιθανότητα καταστροφής ενός σχήματος κατά τη διασταύρωση είναι ανάλογη του μήκους του σχήματος προς το μήκος των χρωμοσωμάτων:
  - $p_c \cdot \delta(S)/L$
    - L είναι το μήκος των χρωμοσωμάτων
    - p_c είναι το ποσοστό των χρωμοσωμάτων που επιλέγονται για διασταύρωση
- Η πιθανότητα καταστροφής ενός σχήματος λόγω της μετάλλαξης είναι ανάλογη της τάξης του:
  - $o(S) \cdot p_m$
    - p_m είναι η πιθανότητα μετάλλαξης

Θεώρημα των Σχημάτων (5/6)

- Με βάση τα παραπάνω, η σχέση της διαφάνειας 47 γίνεται:

$$\xi(S,t+1) \geq \xi(S,t) \cdot \frac{pop\_size}{F(t)} \cdot eval(S,t) \cdot (1-p_c \cdot \frac{\delta(S)}{L}) \cdot (1-o(S) \cdot p_m)$$

- Η ανισότητα στην παραπάνω σχέση προέκυψε από το γεγονός ότι είναι δυνατόν μια διασταύρωση που θα "κόψει" ένα σχήμα σε δύο κομμάτια, να το διατηρήσει στους απογόνους, επειδή έτυχε το σχήμα να υπήρχε και στους δύο γονείς!
- **Θεώρημα των σχημάτων:**
- *Σχήματα μικρού μήκους και χαμηλής τάξης με καταλληλότητα μεγαλύτερη της μέσης αυξάνουν τη συμμετοχή τους στον πληθυσμό με την πάροδο του χρόνου.*

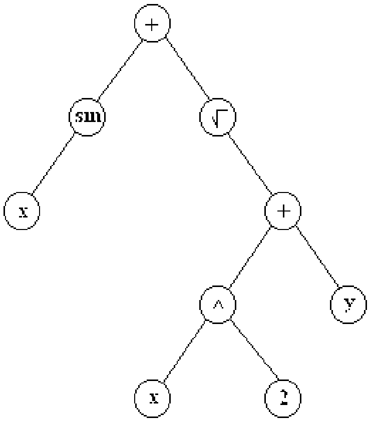
Θεώρημα των Σχημάτων (6/6)

- Το θεώρημα των σχημάτων δείχνει πολύ σημαντικές κατευθύνσεις όσον αφορά την επιλογή της αναπαράστασης των χρωμοσωμάτων.
- Καλές αναπαραστάσεις είναι αυτές που:
  - *Υπάρχει μικρή (αν όχι καθόλου) αλλεπίδραση μεταξύ των γονιδίων.*
    - Κάτι τέτοιο συνέβη με την αναπαράσταση στο πρόβλημα ικανοποίησης περιορισμών, όπου τα επιμέρους γονίδια ήταν ανεξάρτητα μεταξύ τους και η όποια διάταξή τους στο χρωμόσωμα δεν είχε καμία επίδραση στην αποτελεσματικότητα του αλγορίθμου.
  - *Γονίδια τα οποία είναι σχετικά πρέπει να είναι γειτονικά.*
    - Αυτό εξηγεί την αναποτελεσματικότητα της ταξινομημένης αναπαράστασης στο πρόβλημα του πλανόδιου πωλητή.

Γενετικός προγραμματισμός (1/4)

- Ο γενετικός προγραμματισμός είναι μια ειδική περίπτωση των γενετικών αλγορίθμων, όπου οι υποψήφιες λύσεις αντιστοιχούν σε προγράμματα ή συνηθέστερα σε αριθμητικές εκφράσεις.
- Τα προγράμματα στον γενετικό προγραμματισμό παριστάνωνται με τα δένδρα συντακτικής τους ανάλυσης, όπως φαίνεται στο διπλανό σχήμα για την έκφραση:

$$\sin(x)+\sqrt{x^2+y}$$



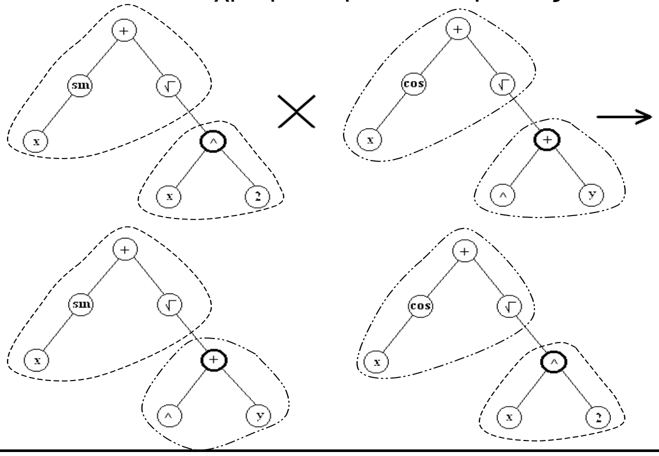
Γενετικός προγραμματισμός (2/4)

- Το αλφάβητο ενός προβλήματος γενετικού προγραμματισμού περιλαμβάνει:
  - Τις συναρτήσεις (π.χ. sin, cos, √, +, *, κλπ)
  - Τα τερματικά σύμβολα (x, y, πραγματικοί αριθμοί κλπ)
- Η συνάρτηση αξιολόγησης είναι η αποτίμηση των εκφράσεων που αντιστοιχούν στα διάφορα χρωμοσώματα ως προς ένα σύνολο δεδομένων εκπαίδευσης.
  - Υπό αυτή την έννοια ο γενετικός προγραμματισμός μπορεί να θεωρηθεί ως εναλλακτικός τρόπος μηχανικής μάθησης.

Γιάννης Ρεφανίδης

Γενετικός προγραμματισμός (3/4)

- Η διαδικασία της διασταύρωσης εκτελείται αφαιρώντας ένα κλαδί από κάθε χρωμόσωμα-γονέα και ανταλλάσσοντας τα κλαδιά αυτά στα χρωμοσώματα-απογόνους.



Γενετικός προγραμματισμός (4/4)

- Η τεχνική της μετάλλαξης πολλές φορές παραλείπεται.
- Εφόσον όμως εφαρμοστεί, αυτό μπορεί να γίνει με διάφορους τρόπους, όπως ανταλλάσοντας τη θέση δύο κλαδιών εντός του ίδιου χρωμοσώματος ή αντικαθιστώντας τερματικά σύμβολα κλπ.
- Γενικά, η απόδοση του γενετικού προγραμματισμού εξαρτάται πάρα πολύ από τον τρόπο αναπαράστασης των προγραμμάτων.
- Ήδη γίνονται προσπάθειες εφαρμογής τεχνικών γενετικού προγραμματισμού σε αυτόματη δημιουργία προγραμμάτων.

Παράδειγμα: Το πρόβλημα του σάκου

Προβλήματα βελτιστοποίησης με περιορισμούς

- Η εφαρμογή των γενετικών αλγορίθμων σε προβλήματα βελτιστοποίησης με περιορισμούς μπορεί να γίνει με τρεις τρόπους:
  - Η πρώτη προσέγγιση είναι η "τιμωρία" των χρωμοσωμάτων που παραβιάζουν κάποιους περιορισμούς μέσω της συνάρτησης καταλληλότητας. Η τιμωρία αυτή μάλιστα μπορεί να εξαρτάται και από το απόλυτο μέγεθος της παραβίασης.
  - Η δεύτερη προσέγγιση είναι η χρήση ειδικών τεχνικών που μετατρέπουν τα χρωμοσώματα που παραβιάζουν κάποιους περιορισμούς σε άλλα χρωμοσώματα που δεν παραβιάζουν κανέναν περιορισμό.
  - Η τρίτη προσέγγιση έχει είναι η χρήση ειδικών αναπαραστάσεων όπου όλα τα χρωμοσώματα είναι έγκυρα.

Το πρόβλημα του σάκου

- Θα εξετάσουμε τις τρεις προσεγγίσεις στο πρόβλημα του σάκου.
- Ο ορισμός του προβλήματος έχει ως εξής:
  - Έχουμε έναν σάκο χωρητικότητας C.
  - Έχουμε N αντικείμενα, κάθε ένα από τα οποία έχει όγκο V(i) και αξία P(i), i=1..N.
  - Όλα τα αντικείμενα δεν χωρούν στο σάκο.

$$\sum_{i=1}^{N} V(i) > C$$

  - Θέλουμε να επιλέξουμε και να τοποθετήσουμε ένα υποσύνολο των αντικειμένων στο σάκο, έτσι ώστε το άθροισμα των αξιών τους να είναι το μέγιστο δυνατό.

Αναπαράσταση

- Η πιο απλή αναπαράσταση για το πρόβλημα είναι η χρήση ενός διανύσματος μήκους N (όσα δηλαδή και τα αντικείμενα), με στοιχεία μηδενικά και άσσους:
  - $<x_1, x_2, \ldots, x_N>$
- Οι άσσοι δηλώνουν τη συμπερίληψη των αντίστοιχων αντικειμένων στο σάκο, ενώ τα μηδενικά τη μη-συμπερίληψη.
- Ένα χρωμόσωμα είναι έγκυρο, εάν ισχύει:

$$\sum_{i=1}^{N} x(i) \cdot V(i) \le C$$

- Η αξία ενός τέτοιου χρωμοσώματος είναι:

$$P(x) = \sum_{i=1}^{N} x(i) \cdot P(i)$$

Χρήση αρνητικής βαθμολόγησης

- Η πρώτη προσέγγιση είναι η χρήση αρνητικής βαθμολόγησης (penalties) για τα χρωμοσώματα εκείνα των οποίων ο συνολικός όγκος υπερβαίνει τη χωρητικότητα του σάκου:

$$eval(x) = \sum_{i=1}^{N} x(i)P(i) - Pen(x)$$

- όπου η τιμή της συνάρτησης Pen(x) εξαρτάται από το μέγεθος της παραβίασης:

$$Pen(x) = \begin{cases} 0, & \text{αν } \sum_{i=1}^{N} x(i)V(i) \le C \\ Pen(\sum_{i=1}^{N} x(i)V(i) - C) & \text{αλλιώς} \end{cases}$$

Διόρθωση μη-έγκυρων χρωμοσωμάτων

- Η δεύτερη προσέγγιση είναι η διόρθωση των μη-έγκυρων χρωμοσωμάτων.
- Κάτι τέτοιο μπορεί να γίνει με την τυχαία μετατροπή των άσσων σε μηδενικά, μέχρι το χρωμόσωμα να μην παραβιάζει τον περιορισμό του όγκου του σάκου.

Εναλλακτική κωδικοποίηση (1/2)

- Η τρίτη προσέγγιση είναι η χρήση εναλλακτικής κωδικοποίησης, και συγκεκριμένα της **ταξινομημένης αναπαράστασης** (την είδαμε στο πρόβλημα του πλανόδιου πωλητή).
- Για παράδειγμα, έστω ότι έχουμε 6 αντικείμενα, τα οποία τα αριθμούμε ως: (1 2 3 4 5 6)
- Το χρωμόσωμα:
  - (4 3 4 1 1 1)
- αντιστοιχεί στη διάταξη των αντικειμένων:
  - (4 3 6 1 2 5)

Εναλλακτική κωδικοποίηση (2/2)

- Η χρήση της ταξινομημένης αναπαράστασης στο πρόβλημα του σάκου πρέπει να ερμηνευτεί ως εξής:
  - Κάθε χρωμόσωμα ορίζει μια πλήρη διάταξη όλων των αντικειμένων.
  - Τα πρώτα αντικείμενα αυτής της διάταξης εισέρχονται στον σάκο, μέχρις ότου κάποιο αντικείμενο ξεπεράσει τη χωρητικότητα του σάκου.

Πειραματικά αποτελέσματα

- Εκτεταμένα πειραματικά αποτελέσματα έδειξαν ότι:
  - Σε προβλήματα όπου η χωρητικότητα του σάκου είναι περίπου το μισό του συνολικού όγκου των αντικειμένων, η πρώτη προσέγγιση (αρνητική βαθμολογία) έδωσε τα καλύτερα αποτελέσματα.
  - Σε προβλήματα με πολύ μικρή χωρητικότητα σάκου, η δεύτερη προσέγγιση (συναρτήσεις επιδιόρθωσης) έδωσε τα καλύτερα αποτελέσματα.

Μηχανική Μάθηση

Γενικά

- Μηχανική μάθηση είναι η ικανότητα ενός συστήματος:
  - Να προσακτά νέα γνώση κατά την αλληλεπίδρασή του με το περιβάλλον στο οποίο δραστηριοποιείται.
  - Να βελτιώνει με την επανάληψη τον τρόπο με τον οποίο δραστηριοποιείται.
- Συστήματα με ικανότητα μηχανικής μάθησης είναι σε θέση:
  - Να μεταβάλλονται διαρκώς προς το καλύτερο, αναφορικά με τις λειτουργίες που είναι σε θέση να εκτελέσουν.
  - Να μεταβάλλουν τη βάση γνώσης τους είτε μετασχηματίζοντας την εσωτερική τους δομή (π.χ. νευρωνικά δίκτυα) ή αποκτώντας επιπλέον γνώση (π.χ. νέες προτάσεις, κανόνες κλπ).
  - Να εκτελούν γενικεύσεις, δηλαδή να αγνοούν χαρακτηριστικά και ιδιότητες που δεν είναι αντιπροσωπευτικά της έννοιας/ενέργειας που πρέπει να μάθουν.

Γενικές κατηγορίες μάθησης

- Μπορούμε να διακρίνουμε τρεις γενικές περιπτώσεις προβλημάτων μάθησης:
  - **Μάθηση με επίβλεψη** (supervised learning) ή **επαγωγική μάθηση** (inductive learning): Μας δίνεται ένα σύνολο παραδειγμάτων εισόδου και ο χαρακτηρισμός τους με κάποια τιμή (διακριτή ή συνεχή). Το ζητούμενο είναι να κατασκευάσουμε μια "συνάρτηση" η οποία να χαρακτηρίζει σωστά τα παραδείγματα αλλά να γενικεύει ικανοποιητικά και σε νέα παραδείγματα.
  - **Μάθηση χωρίς επίβλεψη** (unsupervised learning): Μας δίνεται ένα σύνολο παραδειγμάτων εισόδου χωρίς κανέναν χαρακτηρισμό. Το ζητούμενο είναι να τα ομαδοποιήσουμε σε κατηγορίες.
  - **Ενισχυτική μάθηση** (Reinforcment learning): Δεν υπάρχει σύνολο παραδειγμάτων εισόδου. Το "πρόγραμμα" πρέπει να δημιουργήσει μόνο του τα παραδείγματα εκτελώντας ενέργειες και παρατώντας τα αποτελέσματά τους. Το ζητούμενο είναι το πρόγραμμα να μάθει μια "πολιτική" επιλογής ενεργειών.
    - Ένα πρόγραμμα που μαθαίνει μόνο του να παίζει σκάκι.

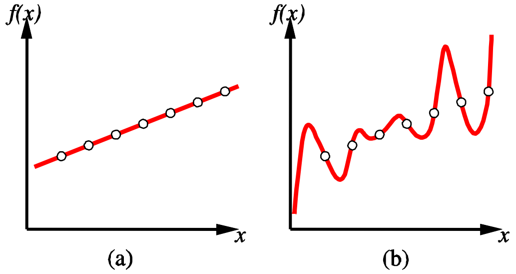
Επαγωγική μάθηση
Inductive learning

Γενικά

- Η επαγωγική μάθηση είναι μάθηση με επίβλεψη.
  - Το απλούστερο παράδειγμα είναι η εύρεση γραμμική παλινδρόμηση (ευθεία ελαχίστων τετραγώνων) σε ένα σύνολο δεδομένων της μορφής y=f(x).
- Ένας ορισμός της επαγωγικής μάθησης θα μπορούσε να είναι ο εξής:
  - Έστω f μια άγνωστη συνάρτηση και έστω ένα σύνολο γνωστών παραδειγμάτων της f. Ζητείται να βρεθεί μια συνάρτηση h που προσεγγίζει την f.
- Η συνάρτηση h ονομάζεται υπόθεση (hypothesis).
- Το βασικό πρόβλημα είναι ότι δεν μπορούμε να ξέρουμε εάν μια υπόθεση h προσεγγίζει καλά την f σε όλα τα σημεία της, μιας και εμείς γνωρίζουμε ένα πεπερασμένο σύνολο παραδειγμάτων.

Χώρος των υποθέσεων (1/5)

- Παρακάτω φαίνεται ένα σύνολο 7 σημείων (x,f(x)), τα οποία περιγράφονται ακριβώς από 2 υποθέσεις:
  - Μια ευθεία
  - Ένα πολυώνυμο 6^{ου} βαθμού

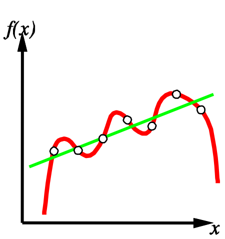


Χώρος των υποθέσεων (2/5)

- Προφανώς στο παράδειγμα της προηγούμενης σελίδας, η ευθεία είναι η καλύτερη υπόθεση.
- Γενικά, ισχύει το "ξυράφι" του Ockham (Ockham's razor: O Ockham είναι Άγγλος φιλόσοφος του 14ου αιώνα):
  - Επιλέγουμε πάντα την απλούστερη υπόθεση που είναι συμβατή (consistent) με τα δεδομένα.
- Ο ορισμός του της "απλότητας" δεν είναι πάντα εύκολος.
- Στην περίπτωση των πολυωνύμων, προφανώς ένα πολυώνυμο $1^{ου}$ βαθμού είναι απλούστερο από ένα πολυώνυμο $6^{ου}$ βαθμού.

Χώρος των υποθέσεων (3/5)

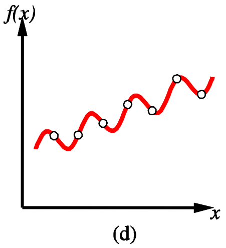
- Παρακάτω φαίνεται πάλι ένα σύνολο έξι παραδειγμάτων:
  - Η ευθεία δεν είναι απόλυτα συμβατή με τα παραδείγματα.
  - Το πολυώνυμο 6ου βαθμού είναι συμβατό.
- Ποια είναι η καλύτερη υπόθεση σε αυτή την περίπτωση;
- Αν θεωρήσουμε ότι τα δεδομένα περιέχουν θόρυβο, τότε πρέπει να κάνουμε έναν συμβιβασμό μεταξύ της πολυπλοκότητας της υπόθεσης και της συμβατότητάς της με τα δεδομένα.



(c)

Χώρος των υποθέσεων (4/5)

- Στο τελευταίο παράδειγμα τα δεδομένα θα μπορούσαν να προσεγγισθούν ακριβώς από μια ημιτονοειδή συνάρτηση.
- Έχοντας επιλέξει ως χώρο υποθέσεων τα πολυώνυμα, δεν θα καταλήγαμε ποτέ στην ημιτονοειδή υπόθεση.
- Επιλέγοντας όμως μεγάλους χώρους υποθέσεων (π.χ. το σύνολο όλων των συναρτήσεων) αυξάνουμε πολύ την πολυπλοκότητα του προβλήματος.



Χώρος των υποθέσεων (5/5)

- Πρέπει λοιπόν να υπάρχει ένας συμβιβασμός μεταξύ της γενικότητας ενός χώρου υποθέσεων και της πολυπλοκότητας εύρεσης υποθέσεων σε αυτό το χώρο.
- Ένας πρόβλημα μάθησης λέγεται εφικτό (realizable) εάν ο χώρος υποθέσεων περιλαμβάνει την σωστή υπόθεση.
- Στην αντίθετη περίπτωση λέγεται ανέφικτο (unrealizable).
  - Τις περισσότερες φορές δεν είμαστε σε θέση να γνωρίζουμε εάν το πρόβλημα είναι ή όχι εφικτό.

Δένδρα Αποφάσεων

Δένδρα αποφάσεων (1/2)

- Ένα δένδρο αποφάσεων παίρνει ως είσοδο ένα διάνυσμα τιμών σε κάποιες ιδιότητες και επιστρέφει μια απάντηση.
  - Οι τιμές εισόδου μπορεί να είναι συνεχείς ή διακριτές.
  - Η έξοδος μπορεί επίσης να είναι συνεχής ή διακριτή.
    - Εάν η έξοδος είναι διακριτή, έχουμε ένα πρόβλημα **κατηγοριοποίησης** (classification).
    - Εάν η έξοδος είναι συνεχής, έχουμε ένα πρόβλημα **παλινδρόμησης** (regression).
- Θα ασχοληθούμε με προβλήματα κατηγοριοποίησης όπου μάλιστα υπάρχουν δύο μόνο κατηγορίες, η θετική (positive) και η αρνητική (negative).

Δένδρα αποφάσεων (2/2)

- Σε ένα δένδρο αποφάσεων, κάθε κόμβος εκτελεί έναν έλεγχο (test) σε κάποια από τις μεταβλητές του προβλήματος.
- Ξεκινώντας από τη ρίζα του δένδρου, εκτελούμε τους ελέγχους στις τιμές των μεταβλητών του τρέχοντος παραδείγματος και ανάλογα με το αποτέλεσμα μετακινούμαστε σε κάποιον από τους κόμβους-παιδιά.
  - Σε κάθε κόμβο επιτρέπεται να γίνει έλεγχος σε **μία** μόνο μεταβλητή.
- Η διαδικασία αυτή επαναλαμβάνεται μέχρι να φθάσουμε σε ένα φύλλο του δένδρου.
- Κάθε φύλλο το δένδρου αντιστοιχεί σε κάποια κατηγορία.
  - Επιτρέπεται διαφορετικά φύλλα του δένδρου να αντιστοιχούν στην ίδια κατηγορία.

Παράδειγμα (1/9)

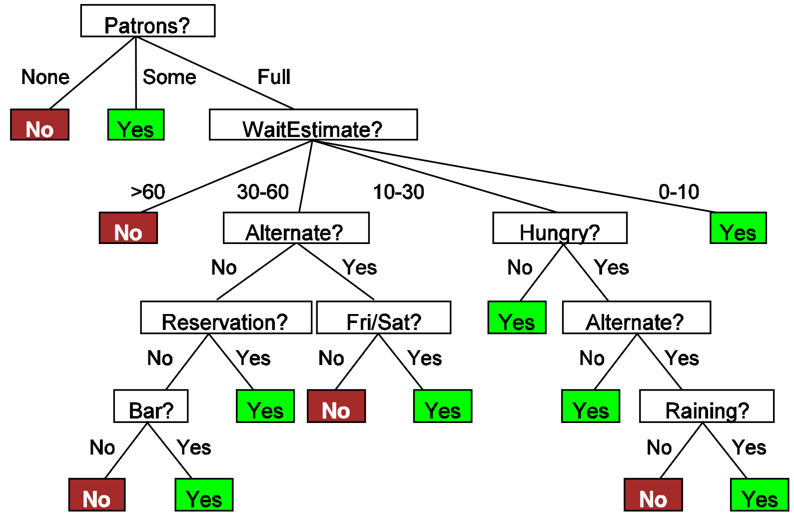
- Έστω το πρόβλημα εάν θα περιμένουμε ή όχι σε ένα εστιατόριο για τραπέζι.
  - Παράδειγμα από Artificial Intelligence: A Modern Approach, S. Russell & P. Norvig, 2003.
- Θεωρούμε ότι η απόφασή μας εξαρτάται από τους παρακάτω παράγοντες:
  - **Alternate**: Υπάρχει ή όχι εναλλακτικό εστιατόριο στην περιοχή.
  - **Bar**: Υπάρχει χώρος στο εστιατόριο για να περιμένουμε.
  - **Fri/Sat**: Είναι Παρασκευή/Σάββατο.
  - **Hungry**: Είμαστε πεινασμένοι.

Παράδειγμα (2/9)

- **Patrons**: Πελάτες που υπάρχουν ήδη στο εστιατόριο (None, Some, Full).
- **Price**: Το επίπεδο τιμών του εστιατορίου ($, $$, $$$).
- **Raining**: Βρέχει.
- **Reservation**: Έχουμε κάνει κράτηση.
- **Type**: Ο τύπος του εστιατορίου (French, Italian, Thai, burger)
- **WaitEstimate**: Αναμενόμενος χρόνος αναμονής.

- Ονομάζουμε την απόφασή μας, εάν θα περιμένουμε ή όχι, **WillWait**.
- Στην επόμενη διαφάνεια φαίνεται ένα δένδρο αποφάσεων που αντιστοιχεί στις προτιμήσεις ενός ανθρώπου.

Παράδειγμα (3/9)



Παράδειγμα (4/9)

- Εάν στο δένδρο αποφάσεων της προηγούμενης διαφάνειας απαντήσουμε Patrons=10 και WaitEstimate=(0-10), τότε η απόφαση είναι να περιμένουμε.

- Το δένδρο αποφάσεων στην προηγούμενη διαφάνεια δεν χρησιμοποιεί τα χαρακτηριστικά Price και Type.
- Αυτό σημαίνει ότι τα δύο αυτά χαρακτηριστικά έχουν ασήμαντη (ή καθόλου) επίδραση στην απόφαση του ανθρώπου εάν θα περιμένει ή όχι στο εστιατόριο.

Εκφραστική ικανότητα

- Ένα δένδρο αποφάσεων μπορεί να περιγράψει οποιαδήποτε συνάρτηση με διακριτό, πεπερασμένο πεδίο ορισμού και διακριτό, πεπερασμένο πεδίο τιμών.
  - Σε περίπτωση συνεχών μεταβλητών στο πεδίο ορισμού, συνήθως αυτές διακριτοποιούνται.
- Στη χειρότερη περίπτωση, ένα δένδρο αποφάσεων έχει τόσα φύλλα, όσα είναι και τα στοιχεία του πεδίου ορισμού της συνάρτησης.
  - Παράδειγμα συναρτήσεων ακατάλληλων για αναπαράσταση με δένδρα αποφάσεων είναι η συνάρτηση πλειοψηφίας (majority function) και η συνάρτηση ισοτιμίας (parity function).

Κατασκευή δένδρων αποφάσεων (1/3)

- Τα δένδρα αποφάσεων κατασκευάζονται βάσει ενός συνόλου παραδειγμάτων.
- Για κάθε πεπερασμένο σύνολο παραδειγμάτων χωρίς αντιφάσεις υπάρχει πάντα ένα δένδρο αποφάσεων, που ονομάζεται τετριμμένο (trivial). Αυτό περιέχει τόσα φύλλα, όσα και τα παραδείγματα και βάθος όσο το πλήθος των μεταβλητών του προβλήματος.
- Ωστόσο εμείς επιζητούμε το δένδρο αποφάσεων με το μικρότερο βάθος, το οποίο ερμηνεύει όλα τα παραδείγματα.
- Ο αλγόριθμος που θα παρουσιαστεί ονομάζεται **ID3** (Induction of Decision Trees), Quinlan 1986.

Κατασκευή δένδρων αποφάσεων (2/3)

- Η εύρεση του απλούστερου δένδρου αποφάσεων είναι πολύ δύσκολη.
- Χρησιμοποιούμε ευρετικούς μηχανισμούς για να βρούμε ένα απλό δένδρο, όχι εγγυημένα το απλούστερο.
- Ο κανόνας είναι ο εξής:
  - Επέλεξε στη ρίζα να κάνεις έλεγχο με το **πιο σημαντικό χαρακτηριστικό** (most important attribute), δηλαδή αυτό που προκαλεί τη μεγαλύτερη διαφοροποίηση στα παραδείγματα αναφορικά με τη μεταβλητή-στόχο.
  - Στη συνέχεια χώρισε τα παραδείγματα σε τόσα υποσύνολα, όσα είναι τα παιδιά του πιο σημαντικού χαρακτηριστικού, και εφάρμοσε αναδρομικά την ίδια διαδικασία στα νέα σύνολα και για τα εναπομείναντα χαρακτηριστικά.

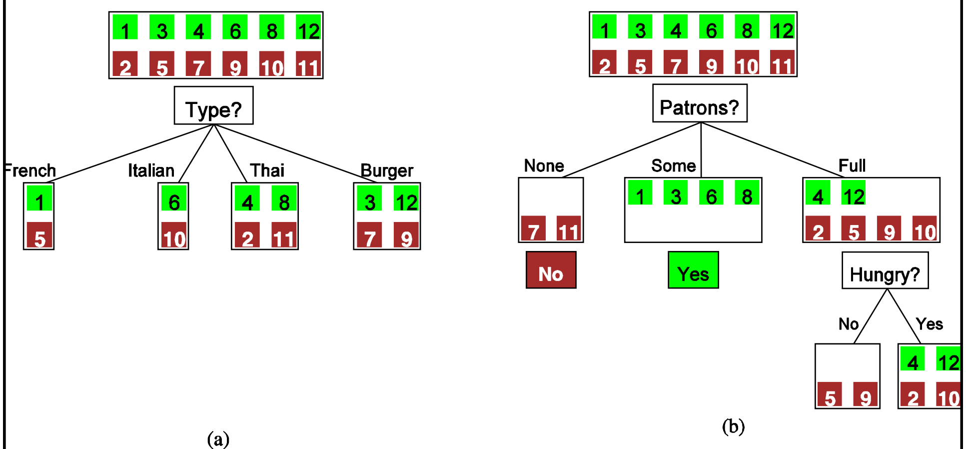
Κατασκευή δένδρων αποφάσεων (3/3)

- Όσο υπάρχουν κόμβοι με θετικά και αρνητικά παραδείγματα και με χαρακτηριστικά που δεν έχουν ελεγχθεί ακόμη, η διαδικασία συνεχίζεται.
- Όταν σε κάποιον κόμβο όλα τα παραδείγματα ανήκουν στην ίδια κατηγορία, ο κόμβος αυτός γίνεται φύλλο του δένδρου.
- Όταν κάποιος κόμβος δεν έχει παραδείγματα, τότε αυτός αντιστοιχίζεται σε μια κατηγορία, βάσει των παραδειγμάτων του κόμβου-γονέα.
- Εάν τέλος σε κάποιον κόμβο υπάρχουν θετικά και αρνητικά παραδείγματα, αλλά έχουν εξαντληθεί όλα τα χαρακτηριστικά ελέγχου, ο κόμβος αυτός είναι ένα διφορούμενος κόμβος.
  - Στην τελευταία περίπτωση ο κόμβος μπορεί να αντιστοιχηθεί είτε στην πλειοψηφούσα κατηγορία των παραδειγμάτων του, ή να επιστρέφει όλες τις κατηγορίες με τις αντίστοιχες συχνότητες εμφάνισής τους.
  - Το φαινόμενο αυτό μπορεί να οφείλεται είτε στην ύπαρξη θορύβου, είτε στην παράβλεψη σημαντικών χαρακτηριστικών.

Παράδειγμα (5/9)

#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
X_1	yes	no	no	yes	some	$$$	no	yes	French	0-10	**yes**
X_2	yes	no	no	yes	full	$	no	no	Thai	30-60	**no**
X_3	no	yes	no	no	some	$	no	no	Burger	0-10	**yes**
X_4	yes	no	yes	yes	full	$	yes	no	Thai	10-30	**yes**
X_5	yes	no	yes	no	full	$$$	no	yes	French	>60	**no**
X_6	no	yes	no	yes	some	$$	yes	yes	Italian	0-10	**yes**
X_7	no	yes	no	no	none	$	yes	no	Burger	0-10	**no**
X_8	no	no	no	yes	some	$$	yes	yes	Thai	0-10	**yes**
X_9	no	yes	yes	no	full	$	yes	no	Burger	>60	**no**
X_{10}	yes	yes	yes	yes	full	$$$	no	yes	Italian	10-30	**no**
X_{11}	no	no	no	no	none	$	no	no	Thai	0-10	**no**
X_{12}	yes	yes	yes	yes	full	$	no	no	Burger	30-60	**yes**

Παράδειγμα (6/9)

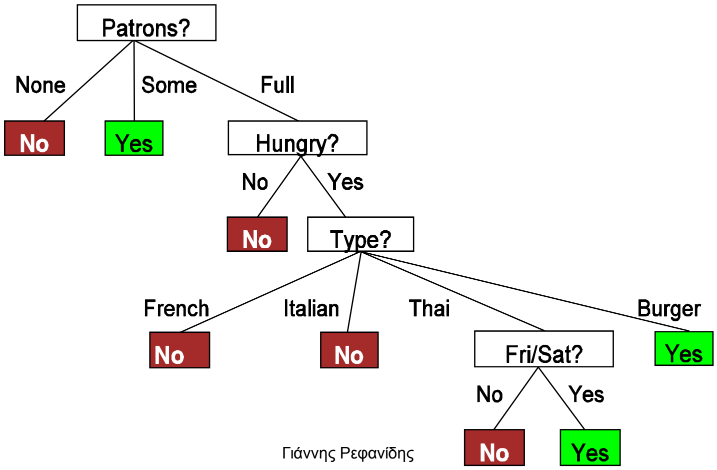


Παράδειγμα (7/9)

- Στην προηγούμενη διαφάνεια, με πράσινο σημειώνονται τα θετικά παραδείγματα (WillWait=yes) και με κόκκινο τα αρνητικά (WillWait=no).
- Στο σχήμα (a) η επιλογή του χαρακτηριστικού Type για έλεγχο στη ρίζα του δένδρου δεν έδωσε κανένα αποτέλεσμα, μιας και στους κόμβους-παιδιά εξακολουθεί να υπάρχει πλήρης "ισοπαλία" μεταξύ των δύο κατηγοριών.
- Στο σχήμα (b) το χαρακτηριστικό Patrons δίνει τρία παιδιά, από τα οποία το ένα έχει μόνο θετικά παραδείγματα και το άλλο μόνο αρνητικά.
- Στο τρίτο παιδί επιλέγεται για έλεγχο το χαρακτηριστικό Hungry, που οδηγεί σε ένα παιδί-φύλλο και ένα παιδί που χρειάζεται περαιτέρω ελέγχους.

Παράδειγμα (8/9)

- Το τελικό δένδρο αποφάσεων είναι το παρακάτω:



Παράδειγμα (9/9)

- Το τελικό δένδρο αποφάσεων είναι διαφορετικό (πιο μικρό) από αυτό που παρουσιάστηκε αρχικά (διαφάνεια 16), μολονότι και τα δύο εξηγούν τα ίδια παραδείγματα.
  - Ενδεχομένως το αρχικό δένδρο αποφάσεων κατασκευάστηκε βάσει περισσοτέρων παραδειγμάτων.
- Το τελικό δένδρο αποφάσεων δεν χρειάστηκε να συμπεριληφθούν πουθενά έλεγχοι για τα χαρακτηριστικά Raining και Reservation.

Επιλογή χαρακτηριστικών (1/8)

- Η επιλογή των χαρακτηριστικών στους κόμβους του δένδρου έγινε με βάση το αναμενόμενο κέρδος πληροφορίας (Shannon & Weaver, 1949).
- Η θεωρία της πληροφορίας (information theory) μετρά την πληροφορία σε bits.
  - Ένα bit πληροφορίας είναι όση πληροφορία χρειαζόμαστε για να απαντήσουμε μια ερώτηση με "ναι" ή "όχι".

Επιλογή χαρακτηριστικών (2/8)

- Γενικά, εάν μια ερώτηση έχει N απαντήσεις, v_1, v_2, ..., v_N, και κάθε μία από αυτές έχει πιθανότητα $P(v_i)$ να εμφανιστεί, τότε το περιεχόμενο πληροφορίας της πραγματικής απάντησης (όποια και αν είναι αυτή) δίνεται από τη σχέση:

$$I(P(v_1, v_2, ..., v_N)) = \sum_{i=1}^{N} -P(v_i)\log_2 P(v_i)$$

- Για παράδειγμα, έστω ότι ρίχνουμε ένα κέρμα κορώνα-γράμματα. Υπάρχουν δύο αποτελέσματα με πιθανότητες ½ το καθένα. Το περιεχόμενο της πληροφορίας στην απάντηση προκύπτει από την παραπάνω σχέση να είναι ίσο με 1 bit.
- Εάν το κέρμα είναι χαλασμένο και έρχεται κορώνα κατά 99% και γράμμα μόνο κατά 1%, το περιεχόμενο της πληροφορίας στην απάντηση προκύπτει ίσο με 0.08 bits.

Επιλογή χαρακτηριστικών (3/8)

- Θεωρώντας ότι οι πιθανότητες των διάφορων απαντήσεων είναι ίσες με τον αριθμό των παραδειγμάτων που αντιστοιχούν στην κάθε απάντηση προς το πλήθος των απαντήσεων, βρίσκουμε ότι η πληροφορία μιας απάντησης με δύο ενδεχόμενα ισούται με:

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right)=-\frac{p}{p+n}\log_2\frac{p}{p+n}-\frac{n}{p+n}\log_2\frac{n}{p+n}$$

Επιλογή χαρακτηριστικών (4/8)

- Στο παράδειγμα με το εστιατόριο, έχουμε δύο δυνατές απαντήσεις (WillWait=yes/no), κάθε μία με την ίδια αρχική πιθανότητα (τα αρχικά 12 παραδείγματα είναι μοιρασμένα σε έξι θετικά και 6 αρνητικά).
- Άρα η πληροφορία της τελικής απάντησης είναι ίση με 1 bit.
- Στην προσπάθειά μας να κατατάξουμε ένα νέο παράδειγμα, εκτελούμε ελέγχους, μέχρι να μην χρειάζεται καθόλου πληροφορία για την τελική απάντηση.

Επιλογή χαρακτηριστικών (5/8)

- Το κριτήριο με το οποίο επιλέγουμε το χαρακτηριστικό που θα ελέξουμε στον τρέχοντα κόμβο έχει να κάνει με το πόση πληροφορία απομένει να μαζέψουμε μετά από έναν έλεγχο.
- Έστω ότι ελέγχουμε ένα χαρακτηριστικό Α, το οποίο χωρίζει το σύνολο παραδειγμάτων του κόμβου Ε σε υποσύνολα E_1, E_2, ..., E_m, όπου m οι διαφορετικές τιμές του Α.
- Έστω p_i,n_i τα θετικά και αρνητικά παραδείγματα για κάθε μια από τις δυνατές απαντήσεις στο χαρακτηριστικό Α.
- Κάθε ένας από τους κόμβους παιδιά του τρέχοντος κόμβου θα χρειάζεται πληροφορία:

$$I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right)$$

Επιλογή χαρακτηριστικών (6/8)

- Επιλέγοντας λοιπόν το χαρακτηριστικό Α για έλεγχο, γνωρίζουμε ότι θα καταλήξουμε σε έναν από τους κόμβους-παιδιά του τρέχοντος κόμβου.
- Η πιθανότητα να πάμε σε κάθε έναν από τους κόμβους παιδιά για τις διάφορες τιμές του Α είναι:

$$\frac{p_i+n_i}{p+n}$$

- Άρα η αναμενόμενη πληροφορία που θα μας λείπει μετά τον έλεγχο στο Α είναι:

$$Remainder(A)=\sum_{i=1}^{m}\frac{p_i+n_i}{p+n}I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right)$$

Επιλογή χαρακτηριστικών (7/8)

- Το αναμενόμενο κέρδος σε πληροφορία μετά τον έλεγχο στο χαρακτηριστικό A ισούται με την πληροφορία που μας έλειπε πριν τον έλεγχο μείον την αναμενόμενη πληροφορία που θα μας λείπει μετά τον έλεγχο:

$$Gain(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - Remainder(A)$$

- Τελικά επιλέγουμε για έλεγχο το χαρακτηριστικό εκείνο που έχει τη μικρότερη αναμενόμενη τιμή στην υπολειπόμενη πληροφορία ή ισοδύναμα το μεγαλύτερο αναμενόμενο κέρδος.

Επιλογή χαρακτηριστικών (8/8)

- Στην περίπτωση του παραδείγματος με το εστιατόριο, ο έλεγχος των δύο χαρακτηριστικών Patrons και Type στη ρίζα του δένδρου (διαφάνεια 23) έδωσε τις παρακάτω τιμές για το αναμενόμενο κέρδος:

$$Gain(Patrons) = 1 - \left[\frac{2}{12} I(0,1) + \frac{4}{12} I(1,0) + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right)\right] \approx 0.541 \text{bits}$$

$$Gain(Type) = 1 - \left[\frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right)\right] \approx 0 \text{ bits}$$

- Φυσικά, εκτός από τα δύο παραπάνω χαρακτηριστικά, ελέγχθηκαν και όλα τα υπόλοιπα.

Κλάδεμα δένδρων αποφάσεων (1/2)

- Μια βάση δεδομένων μπορεί να περιέχει πολλά χαρακτηριστικά (πεδία), περισσότερα από αυτά που χρειάζονται για την κατασκευή του δένδρου αποφάσεων.
- Εάν ξέρουμε ποια ακριβώς χαρακτηριστικά καθορίζουν την τιμή ενός πεδίου-στόχου, μπορούμε να απομονώσουμε μόνο αυτά τα χαρακτηριστικά και να κατασκευάσουμε το δένδρο αποφάσεων.
- Είναι σύνηθες όμως κατά την κατασκευή δένδρων αποφάσεων να χρησιμοποιούνται όλα τα χαρακτηριστικά και να αφήνεται στη διαδικασία κατασκευής να επιλέξει ποια χαρακτηριστικά θα ελέγξει, σε μια προσπάθεια ανίχνευσης κρυφών συσχετίσεων.
- Αυτό ενδέχεται να οδηγήσει στη χρήση κάποιου μη-σχετικού χαρακτηριστικού για τον διαχωρισμό αντιφατικών παραδειγμάτων (ίδιες τιμές σε όλα τα σχετικά χαρακτηριστικά και διαφορετική κατηγορία).

Κλάδεμα δένδρων αποφάσεων (2/2)

- Το πρόβλημα της χρήσης μη-σχετικών χαρακτηριστικών για σωστό διαχωρισμό όλων των παραδειγμάτων λέγεται υπερπροσαρμογή (overfitting).
- Η τεχνική του κλαδέματος των δένδρων αποφάσεων (decision tree pruning) ανακαλύπτει αυτόματα μη-σχετικά χαρακτηριστικά και τα αγνοεί από όλα τα στάδια κατασκευής του δένδρου αποφάσεων.
- Η τεχνική βασίζεται στον υπολογισμό του αναμενόμενου κέρδους πληροφορίας, εάν το μη-σχετικό χαρακτηριστικό επιλεγεί στη ρίζα του δένδρου αποφάσεων.
  - Χαρακτηριστικά με μικρό αναμενόμενο κέρδος πληροφορίας δεν λαμβάνονται υπόψη.

Ελλειπή δεδομένα

- Σε πολλά προβλήματα δεν είναι γνωστές οι τιμές όλων των χαρακτηριστικών για όλα τα παραδείγματα.
- Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί ως εξής:
  - Κάθε παράδειγμα συνοδεύεται με έναν συντελεστή βαρύτητας.
  - Όλα τα αρχικά παραδείγματα έχουν συντελεστή βαρύτητας 1.
  - Κάθε φορά που ένα παράδειγμα δεν έχει τιμή στο χαρακτηριστικό που ελέγχεται, αυτό αντιγράφεται σε όλους τους κόμβους παιδιά με την τιμή που αντιστοιχεί σε κάθε παιδί και συντελεστή βαρύτητας ίσο με το λόγο των παραδειγμάτων κάθε κόμβου παιδιού προς το σύνολο των παραδειγμάτων του κόμβου-γονέα.
  - Κατά τον υπολογισμό των πιθανοτήτων, ο αριθμός των παραδειγμάτων κάθε κόμβου ισούται με το άθροισμα των συντελεστών βαρύτητάς τους.

Χαρακτηριστικά με πολλές τιμές (1/3)

- Τα πεδία που έχουν πάρα πολλές διακριτές τιμές οδηγούν συνήθως σε μεγάλο αναμενόμενο κέρδος πληροφορίας.
- Ωστόσο τέτοια πεδία ενδέχεται να είναι εντελώς άσχετα με ένα πρόβλημα.
  - Για παράδειγμα, στο πρόβλημα επιλογής εστιατορίου, έστω ότι υπάρχει ένα χαρακτηριστικό Month που δηλώνει το μήνα κατά τον οποίο συνέβη το εν λόγω περιστατικό.
  - Έχοντας 12 μόνο παραδείγματα, ενδέχεται κάθε ένα από αυτά να έχει συμβεί σε διαφορετικό μήνα, με αποτέλεσμα να αρκεί το χαρακτηριστικό του Μήνα για να αποφασίσουμε μια τιμή για το πεδίο WillWait.
  - Ωστόσο το χαρακτηριστικό Month είναι άσχετο με το WillWait.

Χαρακτηριστικά με πολλές τιμές (2/3)

- Ένας τρόπος να απομονώνονται τέτοια πεδία είναι να υπολογίσουμε το λόγο των κερδών (**gain ratio**) ως εξής:
  - Έστω I_1 το αναμενόμενο κέρδος πληροφορίας αναφορικά με το πεδίο WillWait, εάν ελέξουμε το μήνα.
    - Στην περίπτωση των 12 παραδειγμάτων, αυτό μπορεί να είναι ακόμη και 1 (εάν όλα τα παραδείγματα συνέβησαν σε διαφορετικό μήνα)
  - Έστω I_2 το κέρδος πληροφορίας, εάν μάθουμε την τιμή του μήνα.
    - Η πληροφορία του να μάθουμε το μήνα, με 12 ισοπίθανες απαντήσεις, έχει μέγεθος 3.585 bits.
  - Κατά την κατασκευή του δένδρου επιλέγουμε τα χαρακτηριστικά όχι βάσει του I_1 αλλά βάσει του λόγου I_1/I_2.

Χαρακτηριστικά με πολλές τιμές (3/3)

- Χαρακτηριστικά με πάρα πολλές τιμές που επιπλέον είναι μη-σχετικά με το χαρακτηριστικό-στόχο, ανιχνεύονται και από τις τεχνικές κλαδέματος δένδρων αποφάσεων, όταν όμως ο αριθμός των παραδειγμάτων γίνει σημαντικά μεγαλύτερος από το πλήθος των διακριτών τιμών του μη-σχετικού χαρακτηριστικού.

Αριθμητικά χαρακτηριστικά (1/2)

- Αριθμητικά χαρακτηριστικά, όπως πεδία πραγματικών αριθμών ή μη-φραγμένα πεδία ακέραιων αριθμών απαιτούν προσαρμογή της διαδικασίας επαγωγής δένδρων αποφάσεων, μιας και αυτά δεν παράγουν πεπερασμένο αριθμό παιδιών.
- Υπάρχουν 2 προσεγγίσεις στο πρόβλημα:
  - Διακριτοποιούμε τα πεδία αυτά, επιλέγοντας κατηγορίες που αντιστοιχούν σε συγκεκριμένες περιοχές τιμών, και στη συνέχεια εφαρμόζουμε τη συνήθη διαδικασία κατασκευής δένδρων αποφάσεων.
  - Τροποποιούμε τον αλγόριθμο ώστε να αποφασίζει σε κάθε κόμβο εάν θα χρησιμοποιήσει ένα τέτοιο πεδίο, χωρίζοντάς τις τιμές του σε δύο υποδιαστήματα στο σημείο εκείνο που δίνει το μέγιστο αναμενόμενο κέρδος.
    - Σε αυτή την περίπτωση, το αριθμητικό πεδίο που χρησιμοποιήθηκε σε κάποιο κόμβο, μπορεί να ξαναχρησιμοποιηθεί παρακάτω.

Αριθμητικά χαρακτηριστικά (2/2)

- Και οι δύο προσεγγίσεις εμφανίζουν προβλήματα:
  - Εάν επιλέξουμε να διακριτοποιήσουμε μόνοι μας τα αριθμητικά πεδία, εισάγουμε μεγάλη υποκειμενικότητα στα παραγόμενα δένδρα αποφάσεων.
  - Εάν αφήσουμε τη διαδικασία κατασκευής του δένδρου αποφάσεων να επιλέξει εάν και πώς θα χρησιμοποιήσει τα πεδία αυτά, αυξάνουμε υπερβολικά το κόστος κατασκευής του δένδρου αποφάσεων.
    - Πράγματι, για κάθε κόμβο πρέπει να ελέγχονται όλα τα αριθμητικά πεδία για κάθε "πιθανή" θέση μέσα στο πεδίο ορισμού τους (πιθανές είναι οι θέσεις που οδηγούν σε διαφορετική κατηγοριοποίηση τουλάχιστον ενός παραδείγματος).

Έξοδος με συνεχείς τιμές

- Εάν η έξοδος που θέλουμε να πάρουμε από το δένδρο αποφάσεων έχει συνεχείς τιμές, τότε η διαδικασία προσαρμόζεται ως εξής:
  - Κάποια από τα χαρακτηριστικά χρησιμοποιούνται για ελέγχους και κάποια όχι.
  - Τα χαρακτηριστικά που δεν χρησιμοποιήθηκαν σε ελέγχους χρησιμοποιούνται στα φύλλα για να οριστεί μια συνάρτηση ελαχίστων τετραγώνων.
- Το παραγόμενο δένδρο λέγεται δένδρο παλινδρόμησης (regression tree).
  - Το πρόβλημα είναι πότε σταματάμε τους ελέγχους και αποφασίζουμε ότι τα υπόλοιπα χαρακτηριστικά θα χρησιμοποιηθούν σε μια διαδικασία παλινδρόμησης.

Κανόνες κατηγοριοποίησης

Classification Rules

Γενικά

- Οι κανόνες κατηγοριοποίησης αποτελούν την εναλλακτική λύση στα δένδρα αποφάσεων.
- Πρόκειται για κανόνες της μορφής:
  - IF συνθήκες THEN κατηγορία
- όπου οι συνθήκες μπορεί να είναι σύζευξη πολλών χαρακτηριστικών.
- Στο παράδειγμα του εστιατορίου, ένας κανόνας κατηγοριοποίησης θα μπορούσε να είναι ο εξής:
  - IF Patrons=Full AND Hungry=no THEN WillWait=no
- Οι κανόνες κατηγοριοποίησης μπορούν να χρησιμοποιηθούν σε έμπειρα συστήματα (π.χ. Clips).

Δένδρα αποφάσεων και κανόνες κατηγοριοποίησης

- Είναι πολύ εύκολο από ένα δένδρο αποφάσεων να κατασκευάσουμε ένα σύνολο κανόνων κατηγοριοποίησης ως εξής:
  - Για κάθε φύλλο του δένδρου αποφάσεων κατασκευάζουμε έναν κανόνα, του οποίου οι συνθήκες είναι όλοι οι έλεγχοι και οι αντίστοιχες τιμές στους κόμβους από τη ρίζα μέχρι το φύλλο.
- Συνήθως ωστόσο υπάρχει δυνατότητα κατασκευής απλούστερων συνόλων κανόνων κατηγοριοποίησης, για τα ίδια δεδομένα εκπαίδευσης.
- Η αντίστροφη εργασία, δηλαδή της κατασκευής ενός δένδρου αποφάσεων από ένα σύνολο κανόνων κατηγοριοποίησης, είναι πιο δύσκολη διαδικασία.
  - Δοκιμάστε να φτιάξετε ένα δένδρο αποφάσεων από τους κανόνες:
    - if a and b then x
    - if c and d then x

Κατασκευή κανόνων κατηγοριοποίησης (1/7)

- Ο απλούστερος τρόπος κατασκευής ενός συνόλου κανόνων κατηγοριοποίησης είναι ο εξής:
  - Επιλέγουμε μια κατηγορία για την οποία θέλουμε να κατασκευάσουμε έναν κανόνα, έστω WillWait=yes.
  - Ξεκινάμε με έναν κανόνα χωρίς συνθήκες:
    - **IF ? THEN WillWait=yes**
- Το κρίσιμο σημείο είναι η επιλογή των συνθηκών που θα συμπεριληφθούν στις προϋποθέσεις του κανόνα.
- Το κριτήριο επιλογής είναι οι κανόνες να καλύπτουν όσο το δυνατόν περισσότερα σωστά παραδείγματα και λιγότερα λάθος.

Παράδειγμα (1/4)

#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
X_1	yes	no	no	yes	some	$$$	no	yes	French	0-10	**yes**
X_2	yes	no	no	yes	full	$	no	no	Thai	30-60	**no**
X_3	no	yes	no	no	some	$	no	no	Burger	0-10	**yes**
X_4	yes	no	yes	yes	full	$	yes	no	Thai	10-30	**yes**
X_5	yes	no	yes	no	full	$$$	no	yes	French	>60	**no**
X_6	no	yes	no	yes	some	$$	yes	yes	Italian	0-10	**yes**
X_7	no	yes	no	no	none	$	yes	no	Burger	0-10	**no**
X_8	no	no	no	yes	some	$$	yes	yes	Thai	0-10	**yes**
X_9	no	yes	yes	no	full	$	yes	no	Burger	>60	**no**
X_{10}	yes	yes	yes	yes	full	$$$	no	yes	Italian	10-30	**no**
X_{11}	no	no	no	no	none	$	no	no	Thai	0-10	**no**
X_{12}	yes	yes	yes	yes	full	$	no	no	Burger	30-60	**yes**

Κατασκευή κανόνων κατηγοριοποίησης (2/7)

- Έχουμε 10 χαρακτηριστικά με διάφορες τιμές το καθένα.
- Για τον κανόνα: **IF ? THEN WillWait=yes** βρίσκουμε ότι:

Alt=yes	3/6	Rain=yes	3/5
Alt=no	3/6	Rain=no	3/7
Bar=yes	3/6	Res=yes	3/5
Bar=no	3/6	Res=no	3/7
Fri=yes	2/5	Type=French	1/2
Fri=no	4/7	Type=Thai	2/4
Hun=yes	5/7	Type=Italian	1/2
Hun=no	1/5	Type=Burger	2/4
Pat=full	2/6	Est=0-10	4/6
Pat=some	4/4	Est=10-30	1/2
Pat=none	0/2	Est=30-60	1/2
Price=$	3/7	Est=>60	0/2
Price=$$	2/2		
Price=$$$	1/3		

Κατασκευή κανόνων κατηγοριοποίησης (3/7)

- Κάθε υποψήφια συνθήκη στην προηγούμενη διαφάνεια χαρακτηρίζεται από έναν λόγο p/t, όπου:
  - t είναι το πλήθος των παραδειγμάτων που καλύπτει η συνθήκη.
  - p είναι το πλήθος των παραδειγμάτων μεταξύ των t τα οποία κατηγοριοποιούνται σωστά.
- Από όλες τις υποψήφιες συνθήκες επιλέγουμε εκείνη που μεγιστοποιεί το λόγο p/t.
- Άρα ο κανόνας γίνεται:
  - IF Pat=some THEN WillWait=yes
- Ο κανόνας αυτός δίνει σωστά αποτελέσματα σε όλα τα παραδείγματα που κατηγοριοποιεί (p/t=4/4), ωστόσο υπάρχουν ακόμη δύο παραδείγματα με WillWait=yes, άρα χρειάζεται να κατασκευαστούν και άλλοι κανόνες.

Παράδειγμα (2/4)

#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
X_2	yes	no	no	yes	full	$	no	no	Thai	30-60	**no**
X_4	yes	no	yes	yes	full	$	yes	no	Thai	10-30	**yes**
X_5	yes	no	yes	no	full	$$$	no	yes	French	>60	**no**
X_7	no	yes	no	no	none	$	yes	no	Burger	0-10	**no**
X_9	no	yes	yes	no	full	$	yes	no	Burger	>60	**no**
X_{10}	yes	yes	yes	yes	full	$$$	no	yes	Italian	10-30	**no**
X_{11}	no	no	no	no	none	$	no	no	Thai	0-10	**no**
X_{12}	yes	yes	yes	yes	full	$	no	no	Burger	30-60	**yes**

Κατασκευή κανόνων κατηγοριοποίησης (4/7)

- Για να κατασκευάσουμε τους υπόλοιπους κανόνες που αντιστοιχούν στην κατηγορία WillWait=yes, αφαιρούμε προσωρινά από τα 12 παραδείγματα τα 4 που κατηγοριοποιούνται από τον πρώτο κανόνα και εκτελούμε την ίδια διαδικασία (διαφάνεια 51).
  - Ουσιαστικά έχουμε 2 παραδείγματα με WillWait=yes μεταξύ των 8 που απέμειναν.
- Εκτελώντας την ίδια διαδικασία βρίσκουμε ότι οι καλύτερες συνθήκες είναι οι:
  - Hungry=yes 2/4
  - Est=10-30 1/2
  - Est=30-60 1/2
- Από τις παραπάνω συνθήκες που ισοβαθμούν, επιλέγεται η 1η, γιατί αντιστοιχεί σε μεγαλύτερο σύνολο παραδειγμάτων.

Παράδειγμα (3/4)

#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
X_2	yes	no	no	yes	full	$	no	no	Thai	30-60	**no**
X_4	yes	no	yes	yes	full	$	yes	no	Thai	10-30	**yes**
X_{10}	yes	yes	yes	yes	full	$$$	no	yes	Italian	10-30	**no**
X_{12}	yes	yes	yes	yes	full	$	no	no	Burger	30-60	**yes**

Κατασκευή κανόνων κατηγοριοποίησης (5/7)

- Ο κανόνας γίνεται προσωρινά:
  - IF Hungry=yes then WillWait=yes
- Ο κανόνας βρίσκει τα 2 σωστά παραδείγματα, αλλά βρίσκει και 2 λάθος.
- Πρέπει να προσθέσουμε επιπλέον συνθήκες για να αφαιρέσουμε τα 2 λάθος παραδείγματα.
- Εκτελούμε την ίδια διαδικασία στα 4 παραδείγματα του παραπάνω κανόνα (διαφάνεια 53) και βρίσκουμε ότι οι καλύτερες συνθήκες είναι οι εξής:
  - Rain=yes 1/1
  - Type=Burger 1/1
- Αν και μπορούμε να επιλέξουμε οποιονδήποτε από τους παραπάνω δύο ελέγχους, επιλέγεται ο πρώτος γιατί αντιστοιχεί σε απλούστερη μεταβλητή:
  - IF Hungry=yes AND Rain=yes then WillWait=yes

Κατασκευή κανόνων κατηγοριοποίησης (6/7)

- Έχουμε κατηγοριοποιήσει σωστά 5 από τα 6 παραδείγματα με WillWait=yes.
- Εκτελώντας ξανά την ίδια διαδικασία για να κατηγοριοποιήσουμε το τελευταίο παράδειγμα μεταξύ των 7 υπολοίπων παραδειγμάτων (διαφάνεια 56), καταλήγουμε στον κανόνα:
  - IF Est=20-60 AND Bar=yes THEN WillWait=yes
    - Να σημειωθεί ότι για η δεύτερη συνθήκη του παραπάνω κανόνα επελέγη από πολλές ισόβαθμες.
- Η ίδια διαδικασία, ξεκινώντας με το αρχικό σύνολο των 12 παραδειγμάτων, εφαρμόζεται για την παραγωγή κανόνων για την τιμή WillWait=no.

Παράδειγμα (4/4)

#	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	Wait
X_2	yes	no	no	yes	full	$	no	no	Thai	30-60	**no**
X_5	yes	no	yes	no	full	$$$	no	yes	French	>60	**no**
X_7	no	yes	no	no	none	$	yes	no	Burger	0-10	**no**
X_9	no	yes	yes	no	full	$	yes	no	Burger	>60	**no**
X_{10}	yes	yes	yes	yes	full	$$$	no	yes	Italian	10-30	**no**
X_{11}	no	no	no	no	none	$	no	no	Thai	0-10	**no**
X_{12}	yes	yes	yes	yes	full	$	no	no	Burger	30-60	**yes**

Κατασκευή κανόνων κατηγοριοποίησης (7/7)

- Παρατηρούμε ότι οι κανόνες που προέκυψαν είναι πολύ διαφορετικοί από το δένδρο αποφάσεων.
- Η διαφορά οφείλεται στο ότι οι δύο διαδικασίες κατασκευής προσπαθούν να βελτιστοποιήσουν διαφορετικά κριτήρια:
  - Στο δένδρο αποφάσεων προσπαθούμε να ελαχιστοποιήσουμε τον μέσο αριθμό ελέγχων που κάνουμε για όλες τις κατηγορίες.
  - Στους κανόνες κατηγοριοποίησης προσπαθούμε να ελαχιστοποιήσουμε τον αριθμό ελέγχων που κάνουμε για μία μόνο κατηγορία.
- Έτσι εάν μας ενδιαφέρει να κατατάξουμε ένα άγνωστο παράδειγμα σε μια κατηγορία, με ένα δένδρο αποφάσεων εκτελούμε πολύ λιγότερους ελέγχους. Αντίθετα, εάν μας ενδιαφέρει να ελέξουμε εάν ένα παράδειγμα ανήκει σε μια συγκεκριμένη κατηγορία, με τους κανόνες κατηγοριοποίησης εκτελούμε λιγότερους ελέγχους.

Λίστες αποφάσεων

- Οι κανόνες κατηγοριοποίησης που είδαμε μπορούν να ελέγχονται με οποιαδήποτε σειρά για την κατηγοριοποίηση ενός παραδείγματος.
- Μια εναλλακτική προσέγγιση είναι να απαιτήσουμε ο έλεγχος των κανόνων να γίνεται με τη σειρά που αυτοί δημιουργήθηκαν.
- Κάτι τέτοιο δίνει τη δυνατότητα δημιουργίας απλούστερων κανόνων, μιας και για τη δημιουργία ενός κανόνα δεν λαμβάνονται υπόψη τα παραδείγματα που κατηγοριοποιούνται από υπάρχοντες κανόνες.
- Επιπλέον αυτή η προσέγγιση αποκλείει περιπτώσεις αντίφασης, όπου σε νέα παραδείγματα ενεργοποιηθούνται περισσότεροι από έναν κανόνες ή ακόμη και κανένας.
- Τέτοιοι κανόνες ονομάζονται **λίστες αποφάσεων** (decision lists).

Κανόνες Συσχέτισης
Association Rules

Γενικά

- Οι κανόνες συσχέτισης είναι παρόμοιοι με τους κανόνες κατηγοριοποίησης, αλλά κατασκευάζονται και χρησιμοποιούνται με διαφορετικό τρόπο.
- Όπως και με τους κανόνες κατηγοριοποίησης, προσπαθούμε να προβλέψουμε την τιμή ενός χαρακτηριστικού από τις τιμές άλλων χαρακτηριστικών.
- Ωστόσο, δεν μας ενδιαφέρει εάν ο κανόνας δίνει πάντα σωστά αποτελέσματα. Αντίθετα μας ενδιαφέρει:
  - Ο κανόνας να δίνει σωστά αποτελέσματα με μεγάλη πιθανότητα.
  - Ο κανόνας να καλύπτει μεγάλο ποσοστό παραδειγμάτων.

Παράδειγμα (1/3)

- Έστω το πρόβλημα του καιρού (Data Mining, Witten & Frank, 1999):
  - Έχουμε ένα σύνολο από δεδομένα καιρού, βάσει των οποίων αποφασίζεται εάν θα διεξαχθεί ένας αγώνας ποδοσφαίρου ή όχι.
  - Τα χαρακτηριστικά (πεδία) του προβλήματος είναι τα εξής:
    - outlook (Τιμές: sunny, overcast, rainy)
    - temperature (Τιμές: hot, mild, cool)
    - humidity (Τιμές: high, normal)
    - windy (Τιμές: false, true)
    - play (Τιμές: yes, no}

Παράδειγμα (2/3)

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Παράδειγμα (3/3)

- **if** humidity=normal and windy=false **then** play=yes
- **if** windy=false and play=no **then** outlook=sunny and humidity=high
  - Αυτός ο κανόνας έχει δύο αποτελέσματα

Υποστήριξη και Εμπιστοσύνη (1/2)

- Κάθε κανόνας συσχέτισης χαρακτηρίζεται από δύο νούμερα:
  - **Υποστήριξη (support)**: Είναι το πλήθος των παραδειγμάτων για τα οποία ο κανόνας εφαρμόζεται επιτυχώς (ισχύουν τόσο οι προϋποθέσεις όσο και τα αποτελέσματά του).
  - **Εμπιστοσύνη (confidence)**: Είναι ο λόγος του πλήθους των παραδειγμάτων που υποστηρίζουν τον κανόνα προς το πλήθος των παραδειγμάτων που υποστηρίζουν τις προϋποθέσεις του κανόνα (ανεξαρτήτως τι γίνεται με τα αποτελέσματά του).
- Για παράδειγμα, ο κανόνας:
  - **if** temperature=hot and humidity=high **then** play=no
- έχει:
  - Υποστήριξη: 2
  - Αξιοπιστία: 66.6%

Υποστήριξη και Εμπιστοσύνη (2/2)

- Αντίθετα ο κανόνας:
  - **if** windy=false and play=no **then** outlook=sunny and humidity=high
- έχει:
  - Υποστήριξη: 2
  - Αξιοπιστία: 100%
- Προφανώς αναζητούμε κανόνες με όσο το δυνατόν μεγαλύτερη υποστήριξη και αξιοπιστία.
- Στο πρόβλημα του καιρού υπάρχουν 58 κανόνες με ελάχιστη υποστήριξη 2 και ελάχιστη αξιοπιστία 95%.

Παρατήρηση

- Από τον κανόνα:
  - **if** A **then** B and C
    - Υποστήριξη = X
    - Αξιοπιστία = Y
- προκύπτουν οι κανόνες:
  - **if** A **then** B
    - Υποστήριξη ≥ X
    - Αξιοπιστία ≥ Y
  - **if** A **then** C
    - Υποστήριξη ≥ X
    - Αξιοπιστία ≥ Y

Κατασκευή κανόνων συσχέτισης (1/6)

- Ένας κανόνας συσχέτισης έχει μία ή περισσότερες προϋποθέσεις και ένα ή περισσότερα αποτελέσματα.
- Τόσο οι προϋποθέσεις όσο και τα αποτελέσματα μπορούν να πάρουν οποιαδήποτε τιμή, έτσι ο αριθμός των συνδυασμών και άρα των κανόνων καθίσταται τεράστιος.
- Από όλους τους κανόνες μας ενδιαφέρουν μόνο αυτοί που ξεπερνούν κάποιες ελάχιστες τιμές υποστήριξης και αξιοπιστίας.
- Χρειαζόμαστε έναν αποδοτικό αλγόριθμο δημιουργίας μόνο καλών κανόνων.
  - Δεν θα θέλαμε δηλαδή έναν αλγόριθμο τύπου παραγωγής και δοκιμής (generate and test), ο οποίος πρώτα δημιουργεί όλους τους υποψήφιους κανόνες και στη συνέχεια απορρίπτει κάποιους.

Κατασκευή κανόνων συσχέτισης (2/6)

- Ο αλγόριθμος κατασκευής κανόνων συσχέτισης βασίζεται στον υπολογισμό **συνόλων στοιχείων** (item sets):
  - Ένα σύνολο στοιχείων είναι ένα σύνολο χαρακτηριστικών, σε κάθε ένα από τα οποία έχει αντιστοιχηθεί μία τιμή.
  - Για κάθε σύνολο στοιχείων μετρούμε τον αριθμό των παραδειγμάτων που το υποστηρίζουν (support).
- Για παράδειγμα, το σύνολο στοιχείων:
  - humidity=normal, windy=false, play=yes
- υποστηρίζεται από 4 παραδείγματα.

Κατασκευή κανόνων συσχέτισης (3/6)

- Ξεκινάμε υπολογίζοντας τα σύνολα με ένα στοιχείο, και κρατάμε αυτά που έχουν υποστήριξη μεγαλύτερη από κάποιο όριο.
- Για να δημιουργήσουμε σύνολα 2 στοιχείων, συνδυάζουμε τα σύνολα ενός στοιχείου που βρήκαμε στο πρώτο βήμα. Τα νέα σύνολα τα ελέγχουμε ως προς την υποστήριξή τους και διαγράφουμε όσα είναι κάτω από το όριο.
- Με παρόμοιο τρόπο βρίσκουμε σύνολα 3 στοιχείων, συνδυάζοντας τα σύνολα ενός και δύο στοιχείων.
- <u>ΠΡΟΣΟΧΗ</u>: Κάθε ένα από τα παραπάνω βήματα απαιτεί **<u>μία</u>** ανάγνωση όλης της βάσης δεδομένων.

Κατασκευή κανόνων συσχέτισης (4/6)

- Έστω ότι το σύνολο στοιχείων {A,B,Γ} έχει υποστήριξη X, ενώ το σύνολο {A,B} έχει υποστήριξη Y≥X.
- Τότε ο κανόνας:
  - if A and B then Γ
- έχει υποστήριξη X και αξιοπιστία X/Y.
- Με παρόμοι τρόπο ελέγχουμε την υποστήριξη και την αξιοπιστία των κανόνων:
  - if A and Γ then B
  - if B and Γ then A

Παράδειγμα

- Για παράδειγμα, από το σύνολο στοιχείων:
  - {humidity=normal, windy=false, play=yes}
- προκύπτουν οι παρακάτω κανόνες
  - if humidity=normal and windy=false then play=yes (4/4)
  - if humidity=normal and play=yes then windy=false (4/6)
  - if windy=false and play=yes then humidity=normal (4/6)
  - if humidity=normal then windy=false and play=yes (4/7)
  - if windy=false then humidity=normal and play=yes (4/8)
  - if play=yes then humidity=normal and windy=false (4/9)
  - if – then humidity=normal and windy=false and play=yes (4/12)

Κατασκευή κανόνων συσχέτισης (5/6)

- Για την κατασκευή κανόνων με πολλαπλά συμπεράσματα υπάρχει ένας πιο αποδοτικός τρόπος.
- Απλοί κανόνες που προέκυψαν από το ίδιο σύνολο στοιχείων συνδυάζονται για να δώσουν κανόνες με πολλά συμπεράσματα.
- Για παράδειγμα, οι κανόνες :
  - if A and B then Γ
    - Υποστήριξη: Χ, Αξιοπιστία: Χ/Υ1
  - if A and Γ then B
    - Υποστήριξη: Χ, Αξιοπιστία: Χ/Υ2
- μπορούν να δώσουν τον κανόνα:
  - if A then B and Γ
    - Υποστήριξη: Χ, Αξιοπιστία: Χ/Υ, όπου Υ≥max(Υ1,Υ2)

Κατασκευή κανόνων συσχέτισης (6/6)

- Ο κανόνας if A then B and Γ έχει υποστήριξη εντός ορίων, ενώ η αξιοπιστία του είναι μικρότερη από αυτή των δύο απλών κανόνων.
- Το σημαντικό όμως είναι ότι η αξιοπιστία του νέου κανόνα μπορεί να βρεθεί χωρίς αναζήτηση στη βάση δεδομένων, παρά μόνο κοιτάζοντας στις τιμές υποστήριξης των συνόλων στοιχείων.

Παρατηρήσεις

- Ο αλγόριθμος που παρουσιάστηκε έχει το χαρακτηριστικό να μειώνει στο ελάχιστο τις σαρώσεις της βάσης δεδομένων.
- Σε τεράστιες βάσεις δεδομένων, οι οποίες δεν χωρούν στη μνήμη του υπολογιστή, η ελάττωση των προσπελάσεων σε αυτές έχει σημαντικό αποτέλεσμα στην ταχύτητα ολοκλήρωσης της διαδικασίας εξαγωγής κανόνων.
- Ο χρόνος ολοκλήρωσης της διαδικασίας επηρρεάζεται πάρα πολύ από το κατώτερο όριο που θέτουμε στην υποστήριξη.
- Μικρές μειώσεις του ορίου αυτού προκαλούν μεγάλη αύξηση στο χρόνο ολοκλήρωσης της διαδικασίας.
  - Συνήθως ξεκινούμε με ένα υψηλό όριο υποστήριξης, το οποίο σταδιακά το μειώνουμε (εκτελώντας ολόκληρη την διαδικασία από την αρχή) μέχρι να εξάγουμε το πλήθος των κανόνων συσχέτισης που επιθυμούμε.

Εμπορικά προγράμματα

- Πολλά εμπορικά προγράμματα βάσεων δεδομένων υλοποιούν δένδρα αποφάσεων:
  - Microsoft SQL Server Analysis Services
    - Decision Trees
    - Clustering
      - http://www.microsoft.com/sql/evaluation/compare/AnalysisDMWP.asp
  - ORACLE 9i Data Mining
    - Naive Bayes for Classification
    - Adaptive Bayes for Classification
    - K-Means for Clustering
    - O-Cluster for Clustering
    - Association Rules
    - Attribute Importance
      - http://otn.oracle.com/products/bi/9idmining.html
      - http://otn.oracle.com/products/bi/htdocs/o9idm_aq.html