

Πολλαπλής επιλογής

1. Έστω ότι εκτελείται ο αλγόριθμος αναζήτησης λύσης A^* και σε κάποιο βήμα το μέγιστο αναζήτησης έχει n μονοπάτια. Επιλέγεται ένα από αυτά, το τελευταίο στοιχείο του οποίου έχει k επόμενους κόμβους:

- A. Στο επόμενο βήμα, συνολικά προστίθενται στο μέγιστο αναζήτησης k νέα μονοπάτια.
 B. Στο επόμενο βήμα, συνολικά προστίθενται στο μέγιστο αναζήτησης ένα νέο μονοπάτι.
 Γ. Στο επόμενο βήμα, συνολικά προστίθενται στο μέγιστο αναζήτησης $k \cdot n$ νέα μονοπάτια.
 Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

(d)

μπορεί να προσεχθούν από 0 έως k μονοπάτια

2. Ο αλγόριθμος αναζήτησης A^* :

- A. Βρίσκει πάντα τη βέλτιστη λύση, δηλαδή το καλύτερο μονοπάτι προς το στόχο.
 B. Δεν είμαστε πάντα σίγουροι ότι θα βρει τη βέλτιστη λύση, δηλαδή το καλύτερο μονοπάτι προς το στόχο, όμως όταν έχει ταματιστεί ξέρουμε πάντα αν η λύση που βρήκε είναι βέλτιστη.
 Γ. Δεν μπορούμε ποτέ να είμαστε σίγουροι ότι θα βρει τη βέλτιστη λύση, δηλαδή το καλύτερο μονοπάτι προς το στόχο.
 Δ. Καμία από τις άλλες προτάσεις δεν είναι αληθής.

(d)

βρίσκει την βέλτιστη λύση για κάθε αποδεκτή ευρετική

3. Ποια από τις παρακάτω προτάσεις που αφορούν τον αλγόριθμο Alpha-Beta είναι ορθή:

- A. Είναι όμοιος με τον Minimax αλλά δεν ελέγχει όλους τους κόμβους, αποφεύγοντας αχρείαστες συγκρίσεις, με βάση δύο συνθήκες που εφαρμόζονται ταυτόχρονα σε όλα τα επίπεδα.
 B. Είναι όμοιος με τον Minimax αλλά δεν ελέγχει όλους τους κόμβους, αποφεύγοντας αχρείαστες συγκρίσεις, με βάση δύο συνθήκες που εφαρμόζονται εναλλακτικά στο min και max επίπεδο.
 Γ. Είναι όμοιος με τον Minimax αλλά δεν ελέγχει όλους τους κόμβους, αποφεύγοντας συγκρίσεις, με βάση δύο ευριστικά κριτήρια που εφαρμόζονται εναλλακτικά στο min και max επίπεδο, και υπό συνθήκες μπορούν να οδηγήσουν στη βέλτιστη λύση.
 Δ. Είναι όμοιος με τον Minimax αλλά δεν ελέγχει όλους τους κόμβους, αποφεύγοντας συγκρίσεις, με βάση δύο ευριστικά κριτήρια που εφαρμόζονται ταυτόχρονα σε όλα τα επίπεδα και υπό συνθήκες μπορούν να οδηγήσουν στη βέλτιστη λύση.

(c)

4. Δίνεται η γνώση Προτασιακής Λογικής K με υπογραφή $\text{Sig}(K) = \{p, q, r\}$. Γνωρίζουμε ότι η γνώση αυτή περιέχει (πιθανά μεταξύ άλλων) την πρόταση $p \leftrightarrow \neg q$. Για τη γνώση K ισχύουν τα εξής:

- A. Έχει 8 ερμηνείες και μπορεί να έχει από κανένα έως 8 μοντέλα, εξαρτάται από τις άλλες προτάσεις.
 B. Έχει 8 ερμηνείες και μπορεί να έχει από κανένα έως 4 μοντέλα, εξαρτάται από τις άλλες προτάσεις.
 Γ. Έχει άπειρες ερμηνείες και μπορεί να έχει από κανένα έως 8 μοντέλα, εξαρτάται από τις άλλες προτάσεις.
 Δ. Έχει άπειρες ερμηνείες και μπορεί να έχει από κανένα έως 4 μοντέλα, εξαρτάται από τις άλλες προτάσεις.

(b)

$$p \leftrightarrow q = (p \Rightarrow q) \vee (q \Rightarrow p) = (\neg p \vee q) \wedge (p \vee \neg q)$$

p	q	r	$\neg p \vee q$	$p \vee \neg q$	$p \leftrightarrow q$
0	0	0	1	0	0
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	1	0	1	0

#ερμηνειων = $2^3 = 8$
 4 μοντέλα

5. Δίνεται η γνώση Λογικής Πρώτης Τάξης $\mathcal{K}$ με υπογραφή $\text{Sig}(\mathcal{K}) = \{a, b, c, P, Q\}$ όπου a, b, c σταθερές και P, Q κατηγορήματα τάξης 1. Πόσες ερμηνείες έχει η γνώση αυτή;

A. 18 B. 32 Γ. Άπειρες. **Δ. Εξαρτάται από τις προτάσεις της γνώσης.**

Ⓐ

c: θέλει αναφορές

6. Δίνεται μία γνώση Λογικής Πρώτης Τάξης $\mathcal{K}$ η οποία περιέχει προτάσεις τύπου Horn. Για τον έλεγχο της συνέπειας της τελικής γνώσης εφαρμόζουμε τον αλγόριθμο ανάλυσης SLD. Ποια από τις επόμενες προτάσεις είναι ορθή;

- A. Ο αλγόριθμος ανάλυσης SLD πιθανά δεν θα τερματίσει, αλλά αν τερματίσει θα δώσει το σωστό αποτέλεσμα.
B. Ο αλγόριθμος ανάλυσης SLD πιθανά δεν θα τερματίσει, και ακόμα και αν τερματίσει μπορεί να μην δώσει το σωστό αποτέλεσμα.
Γ. Ο αλγόριθμος ανάλυσης SLD θα τερματίσει σίγουρα, αλλά μπορεί να μην δώσει το σωστό αποτέλεσμα.
Δ. Ο αλγόριθμος ανάλυσης SLD θα τερματίσει σίγουρα, και θα δώσει το σωστό αποτέλεσμα.

Εστω S ένα σύνολο προτάσεων.

Αναλυτική παραγωγή SLD (SLD resolution - Selected literals, Linear pattern over Definite clauses) μίας πρότασης p από την S , την οποία συμβολίζουμε ως $S \vdash p$, είναι μια ακολουθία προτάσεων $p_1, p_2, \dots, p_n$ τέτοια ώστε $p_1 \in S, p_n = p$ και p_{i+1} αναλύθөн του p_i και κάποιας πρότασης του S .

Αποδεικνύεται ότι για τις προτάσεις Horn ισχύει:

$$S \vdash p \text{ αν και μόνο αν } S \models p$$

- Ο αλγόριθμος SLD backward chaining (με βελτιστοποιήσεις) αποδεικνύεται στην πράξη ιδιαίτερα αποτελεσματικός για συλλογιστική σε οριστικά λογικά προγράμματα (χρησιμοποιείται στην Prolog), αν και στη χειρότερη περίπτωση απαιτεί εκθετικό χρόνο, ενώ σε κάποιες περιπτώσεις δεν τερματίζει.
- Ο αλγόριθμος SLD forward chaining αποδεικνύεται στην πράξη αποτελεσματικός μόνο αν έχουμε κάποιο πολύ καλό (ευριστικό) τρόπο να ελέγχουμε την παραγωγή νέων γεγονότων. Έχει αποδειχθεί ότι λειτουργεί σε πολυωνμικό χρόνο αν δεν χρησιμοποιούμε μεταβλητές και συναρτήσεις.

σίγουρα σωστό αποτέλεσμα τερματισμός?

7. Δίνεται η γνώση Λογικής Πρώτης Τάξης $\mathcal{K}$ η οποία γνωρίζουμε ότι είναι συνεπής. Από την $\mathcal{K}$ διαγράφουμε μια πρόταση p και παίρνουμε την γνώση $\mathcal{K}'$. Ποια από τις παρακάτω προτάσεις που αφορούν την $\mathcal{K}'$ είναι ορθή;

- A. Η $\mathcal{K}'$ θα είναι σίγουρα συνεπής και μάλιστα θα έχει σίγουρα περισσότερα μοντέλα από την $\mathcal{K}$.
Β. Η $\mathcal{K}'$ θα είναι σίγουρα συνεπής και μάλιστα, αν ισχύει ότι η p δεν είναι λογικό συμπέρασμα της $\mathcal{K}'$ θα έχει σίγουρα περισσότερα μοντέλα από την $\mathcal{K}$.
Γ. Η $\mathcal{K}'$ θα είναι σίγουρα συνεπής, και θα έχει σίγουρα λιγότερα μοντέλα από την $\mathcal{K}$.
Δ. Η $\mathcal{K}'$ θα είναι σίγουρα συνεπής και μάλιστα, αν ισχύει ότι η p δεν είναι λογικό συμπέρασμα της $\mathcal{K}'$ θα έχει σίγουρα λιγότερα μοντέλα από την $\mathcal{K}$.

Ⓑ

$K \text{ συνεπής} \Rightarrow K' \text{ συνεπής}$

$I \text{ μοντέλο } K \Rightarrow I \text{ μοντέλο } K'$
 $I \text{ μοντέλο } K' \not\Rightarrow I \text{ μοντέλο } K$

8. Δίνεται ένα σύνολο δεδομένων D (πιθανά μη γραμμικά διαχωρίσιμων) και οι παρακάτω ταξινομητές:

1. Κοινότερου γείτονα
2. S-κοινότερων γειτόνων
3. Perceptron
4. SVM

οι οποίοι προσαρμόζονται κατά τη διαδικασία εκμάθησης στα δεδομένα (πιθανά με κάποιο σφάλμα). Κατά τη φάση της ταξινόμησης δίνεται ένα από τα στοιχεία του συνόλου δεδομένων στην είσοδο. Ποιος από τους ταξινομητές θα ταξινομήσει σίγουρα στη σωστή κλάση;

- Α. Μόνο ο 1.**
B. Ο 1 και ο 2.
Γ. Ο 1, ο 2 και ο 3.
Δ. Όλοι.

(εκτός)

SVM: μη γραμμικό mapping ώστε να μεγιστοποιεί απόσταση κλάσεων

kNN: είναι ευαίσθητο σε outliers για $k=1$, όμως δεν κάνει noise λάθος για τα δεδομένα

εκπαίδευσης, αφού ο κοινότερος γείτονας είναι ο εαυτός του

perceptron: δεν συζητάει για μη γραμμικά διαχωρίσιμα δεδομένα

9. Δίνεται ένα σύνολο δεδομένων D (γραμμικά διαχωρίσιμων) και οι παρακάτω ταξινομητές:

1. Perceptron
2. SVM

οι οποίοι προσαρμόζονται κατά τη διαδικασία εκμάθησης στα δεδομένα (χωρίς σφάλμα). Στη συνέχεια δίνεται ένα ακόμα δεδομένο στο οποίο πρέπει να προσαρμοστούν, το οποίο όμως και οι δύο ταξινομητές στη σωστή κλάση. Ποια από τις παρακάτω προτάσεις είναι ορθή;

- A. Εφόσον το στοιχείο ταξινομείται στη σωστή κλάση, οι ταξινομητές δεν θα τροποποιηθούν.
Β. Ο ταξινομητής 1 δεν θα τροποποιηθεί, όμως ο ταξινομητής 2 μπορεί να τροποποιηθεί, εξαρτάται από τη θέση του νέου στοιχείου.
Γ. Ο ταξινομητής 2 δεν θα τροποποιηθεί, όμως ο ταξινομητής 1 μπορεί να τροποποιηθεί, εξαρτάται από τη θέση του νέου στοιχείου.
Δ. Και οι δύο ταξινομητές μπορεί να τροποποιηθούν, εξαρτάται από τη θέση του νέου στοιχείου.

10. Θέλετε να εκτιμήσετε αν ένας ασθενής που παρουσιάζει τα συμπτώματα A και B πάσχει από την ασθένεια M ή N , χρησιμοποιώντας τον απλοϊκό ταξινομητή Bayes. Για να το επιτύχετε

- A. Πρέπει να γνωρίζετε όλες τις πιθανότητες $P(A, B|M)$, $P(A, B|N)$, $P(M)$, $P(N)$ και $P(A, B)$.
 B. Πρέπει να γνωρίζετε όλες τις πιθανότητες $P(A|M)$, $P(B|M)$, $P(A|N)$, $P(B|N)$, $P(M)$, $P(N)$ και $P(A, B)$.
 Γ. Αρκεί να γνωρίζετε μόνο τις πιθανότητες $P(M|A)$, $P(M|B)$, $P(N|A)$, $P(N|B)$, $P(A)$ και $P(B)$.
 Δ. Αρκεί να γνωρίζετε μόνο τις πιθανότητες $P(A|M)$, $P(B|M)$, $P(A|N)$, $P(B|N)$, $P(M)$ και $P(N)$.

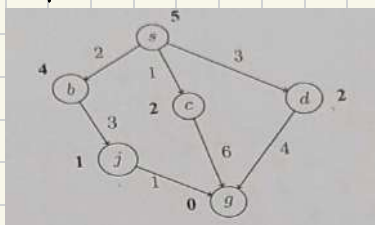
①

$$P(M|AB) = \frac{P(AB|M)P(M)}{P(AB)} = \frac{P(A|M)P(B|M)P(M)}{P(A)P(B)} \quad (\text{υποθέτω ότι } A \perp B)$$

$$P(N|AB) = \frac{P(AB|N)P(N)}{P(AB)} = \frac{P(A|N)P(B|N)P(N)}{P(A)P(B)}$$

$$\Rightarrow P(M|AB) \geq P(N|AB) \Rightarrow \frac{P(A|M)P(B|M)P(M)}{P(A|N)P(B|N)P(N)} \geq 1$$

Θέμα 1



1. Εκτελέστε τον αλγόριθμο αναρίχησης λόφου.

	frontier	closed	current	child nodes
1.	$(s, 5)^s$	$\{\}$	s	$(b, 4), (c, 2), (d, 2)$
2.	$(c, 2)^{sc}$	$\{s\}$	c	$(g, 0)$
3.	$(g, 0)^{sg}$	$\{s, c\}$	g	-

2. Εκτελέστε τον αλγόριθμο A^* με χρήση κλειστού συνόλου.

	frontier	closed	current	child nodes
1.	$(s, 0, s)^s$	$\{\}$	s	$(b, 2, 6)$ $(c, 1, 3)$ $(d, 3, 7)$
2.	$(c, 1, 3)^{sc}$ $(b, 2, 6)^{sb}$ $(d, 3, 7)^{sd}$	$\{(s, 0)\}$	c	$(g, 7, 0)$
3.	$(b, 2, 6)^{sb}$ $(d, 3, 7)^{sd}$ $(g, 7, 0)^{sg}$	$\{(s, 0), (c, 1)\}$	b	$(j, 5, 6)$
4.	$(j, 5, 6)^{bj}$ $(d, 3, 7)^{sd}$ $(g, 7, 0)^{sg}$	$\{(s, 0), (c, 1), (b, 2)\}$	j	$(g, 6, 0)$
5.	$(g, 6, 0)^{bjg}$ $(d, 3, 7)^{sd}$ $(g, 7, 0)^{sg}$	$\{(s, 0), (c, 1), (b, 2), (j, 5)\}$	g	-

hill climbing: scg (κόστος 7)
 A^* algorithm: sbjg (κόστος 6)

Θέμα 2

Δίνονται οι παρακάτω προτάσεις διατυπωμένες σε φυσική γλώσσα:

1. Κάθε παιδί που βρίσκει ένα παιχνίδι ή που βρίσκει ένα γλυκό είναι χαρούμενο.
2. Κάθε παιδί που δέχεται βοήθεια βρίσκει ένα παιχνίδι.
3. Κάθε παιδί που δεν είναι μικρό παιδί ή που προσπαθεί βρίσκει ένα γλυκό.
4. Αν κάθε μικρό παιδί προσπαθεί ή δέχεται βοήθεια, τότε κάθε παιδί είναι χαρούμενο.

Σας ζητούνται τα εξής:

1. Μεταφράστε τις παραπάνω προτάσεις σε προτάσεις ΔΠΠ.
- Σας προτείνεται να χρησιμοποιήσετε τα καθιερωμένα παιδί(x), μικρόπαιδί(x), χαρούμενο(x), βρίσκει παιχνίδι(x), βρίσκει γλυκό(x), δέχεται βοήθεια(x) και προσπαθεί(x) ή ευκολονόητες συντρήσεις αυτών.
- Η χρήση του βρίσκειπαιχνίδι(x), αντί των βρίσκει(x, y) και παιχνίδι(y), και αντίστοιχα για το βρίσκειγλυκό(x), προτείνεται για ευκολία σας.

1. $\forall x ((\text{παιδί}(x) \wedge (\text{βρίσκειΠαιχνίδι}(x) \vee \text{βρίσκειΓλυκό}(x)) \Rightarrow \text{χαρούμενο}(x))$
2. $\forall x ((\text{παιδί}(x) \wedge \text{δέχεταιΒοήθεια}(x)) \Rightarrow \text{βρίσκειΠαιχνίδι}(x))$
3. $\forall x ((\text{παιδί}(x) \wedge (\neg \text{μικρόΠαιδί}(x) \vee \text{προσπαθεί}(x))) \Rightarrow \text{βρίσκειΓλυκό}(x))$
4. $\forall x ((\text{μικρόΠαιδί}(x) \wedge (\text{προσπαθεί}(x) \vee \text{δέχεταιΒοήθεια}(x))) \Rightarrow \forall y (\text{παιδί}(y) \wedge \text{χαρούμενο}(y)))$

2. Μεταφράστε τις προτάσεις που κατασκευάσατε σε κανονική συντακτική μορφή με στόχο να τις χρησιμοποιήσετε για να ελέγξετε με τον αλγόριθμο της ανάλυσης αν (1), (2), (3) $\models$ (4).
- Για διασφάλιση σας δεν χρειάζεται να γράψετε ξεχωριστά ένα ένα όλα τα βήματα κάθε μετατροπής. Με την προϋπόθεση ότι τα βήματα που κάνετε είναι σωστά, μπορείτε όσους είναι εύλογο να κάνετε περισσότερα από ένα βήματα ταυτόχρονα και με τη σειρά που εφάρμοζεται καλύτερα.

$$\begin{aligned}
 1. \quad & \forall x ((\text{παιδί}(x) \wedge \text{βρίσκει}(x)) \vee (\text{παιδί}(x) \wedge \text{βρίσκει}(x))) \Rightarrow \text{χαρούμενο}(x)) \\
 & \neg(\text{παιδί}(x) \wedge \text{βρίσκει}(x)) \vee (\text{παιδί}(x) \wedge \text{βρίσκει}(x)) \vee \text{χαρούμενο}(x) \\
 & (\neg(\text{παιδί}(x) \wedge \text{βρίσκει}(x)) \wedge \neg(\text{παιδί}(x) \wedge \text{βρίσκει}(x))) \vee \text{χαρούμενο}(x) \\
 & ((\neg \text{παιδί}(x) \vee \neg \text{βρίσκει}(x)) \wedge (\neg \text{παιδί}(x) \vee \neg \text{βρίσκει}(x))) \vee \text{χαρούμενο}(x) \\
 & (\neg \text{παιδί}(x) \vee \neg \text{βρίσκει}(x) \vee \text{χαρούμενο}(x)) \vee (\neg \text{παιδί}(x) \vee \neg \text{βρίσκει}(x) \vee \text{χαρούμενο}(x)) \\
 & \rightarrow \{ [\neg \text{παιδί}(x), \neg \text{βρίσκει}(x), \text{χαρούμενο}(x)]_1, [\neg \text{παιδί}(x), \neg \text{βρίσκει}(x), \text{χαρούμενο}(x)]_2 \}
 \end{aligned}$$

$$\begin{aligned}
 3. \quad & \forall x ((\text{παιδί}(x) \wedge (\neg \text{μικρόΠαιδί}(x) \vee \text{προσπαθεί}(x))) \Rightarrow \text{βρίσκειΓλυκό}(x)) \\
 & \neg(\text{παιδί}(x) \wedge (\neg \text{μικρόΠαιδί}(x) \vee \text{προσπαθεί}(x))) \vee \text{βρίσκειΓλυκό}(x) \\
 & (\neg \text{παιδί}(x) \vee \neg(\neg \text{μικρόΠαιδί}(x) \vee \text{προσπαθεί}(x))) \vee \text{βρίσκειΓλυκό}(x) \\
 & \neg \text{παιδί}(x) \vee (\text{μικρόΠαιδί}(x) \wedge \neg \text{προσπαθεί}(x)) \vee \text{βρίσκειΓλυκό}(x) \\
 & (\neg \text{παιδί}(x) \vee \text{μικρόΠαιδί}(x) \vee \text{βρίσκειΓλυκό}(x)) \wedge (\neg \text{παιδί}(x) \vee \neg \text{προσπαθεί}(x) \vee \text{βρίσκειΓλυκό}(x)) \\
 & \rightarrow \{ [\neg \text{παιδί}(x), \text{μικρόΠαιδί}(x), \text{βρίσκειΓλυκό}(x)]_4, [\neg \text{παιδί}(x), \neg \text{προσπαθεί}(x), \text{βρίσκειΓλυκό}(x)]_5 \}
 \end{aligned}$$

3. Εφαρμόζοντας τον αλγόριθμο της ανάλυσης ελέγξτε αν (1), (2), (3) $\models$ (4).

$$\begin{aligned}
 4. \quad & \forall x ((\text{μικρόΠαιδί}(x) \wedge \text{προσπαθεί}(x)) \vee (\text{μικρόΠαιδί}(x) \wedge \text{δέχεταιΒοήθεια}(x))) \Rightarrow \forall y (\text{παιδί}(y) \wedge \text{χαρούμενο}(y))) \\
 & \neg(\text{μικρόΠαιδί}(x) \wedge \text{προσπαθεί}(x)) \vee \neg(\text{μικρόΠαιδί}(x) \wedge \text{δέχεταιΒοήθεια}(x)) \vee \forall y (\text{παιδί}(y) \wedge \text{χαρούμενο}(y)) \\
 & (\neg \text{μικρόΠαιδί}(x) \vee \neg \text{προσπαθεί}(x)) \vee (\neg \text{μικρόΠαιδί}(x) \vee \neg \text{δέχεταιΒοήθεια}(x)) \vee (\text{παιδί}(y) \wedge \text{χαρούμενο}(y)) \\
 & \rightarrow \{ [\neg \text{μικρόΠαιδί}(x), \neg \text{προσπαθεί}(x), \text{παιδί}(y)]_6, [\neg \text{μικρόΠαιδί}(x), \neg \text{δέχεταιΒοήθεια}(x), \text{παιδί}(y)]_7, [\neg \text{μικρόΠαιδί}(x), \text{δέχεταιΒοήθεια}(x), \text{παιδί}(y)]_8, [\neg \text{μικρόΠαιδί}(x), \neg \text{προσπαθεί}(x), \text{παιδί}(y)]_9 \}
 \end{aligned}$$

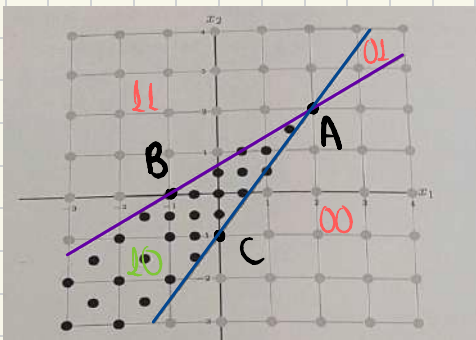
Θέμα 3

Θεωρούμε ότι έχουμε ένα σύνολο σημείων δεδομένων που αντιστοιχούν σε διδιάστατα διανύσματα, καθένα από τα οποία γνωρίζουμε ότι ανήκει ή στην κλάση A ή στην κλάση B . Τα σημεία που αντιστοιχούν στα διανύσματα αυτά έχουν σχεδιαστεί στο παρακάτω διάγραμμα. Τα σημεία που ανήκουν στην κατηγορία A έχουν σχεδιαστεί με μαύρο χρώμα, και τα σημεία που ανήκουν στην κατηγορία B έχουν σχεδιαστεί με γκρι χρώμα.

Έστω ότι διαθέτουμε νευρώνες τύπου perceptron με την εξής συνάρτηση ενεργοποίησης:

$$f(x) = \begin{cases} 1 & \text{αν } x \geq 0 \\ -1 & \text{αν } x < 0 \end{cases}$$

Επιπλέον, κάθε νευρώνας έχει δύο εισόδους x_1, x_2 με αντίστοιχα βάρη w_1, w_2 συν μία σταθερή είσοδο 1 με βάρος w_0 . Θέλουμε να κατασκευάσουμε με βάση αυτούς τους νευρώνες ένα νευρωνικό δίκτυο το οποίο να διαχωρίζει σωστά τις δύο κλάσεις. συγκεκριμένα, αν του δίνεται στην είσοδο ένα διάνυσμα της κλάσης A να παράγει έξοδο 1 και αν του δίνεται στην είσοδο ένα διάνυσμα της κλάσης B να παράγει έξοδο -1.



Ευθεία 1: $A(2,2), C(0,-1)$

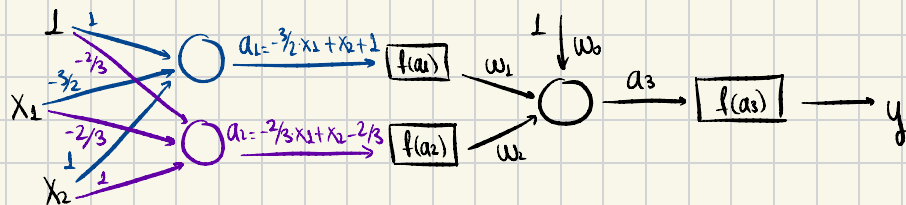
$$(ε_1): \frac{x_2 - (-1)}{x_1 - 0} = \frac{2 - (-1)}{2 - 0} \Rightarrow x_2 = \frac{3}{2}x_1 - 1$$

Ευθεία 2: $A(2,2), B(-1,0)$

$$(ε_2): \frac{x_2 - 0}{x_1 - (-1)} = \frac{2 - 0}{2 - (-1)} \Rightarrow x_2 = \frac{2}{3}x_1 + \frac{2}{3}$$

1. Είναι το πρόβλημα που πρέπει να λύσει το νευρωνικό δίκτυο γραμμικά διαχωρίσιμο; Ποιο είναι το ελάχιστο πλήθος επιπέδων νευρώνων που απαιτούνται για να λυθεί το συγκεκριμένο πρόβλημα; Πόσους πρέπει να περιέχει το κάθε επίπεδο και ποιος θα είναι ο ρόλος τους;

Ευθεία 1: $-3/2 x_1 + x_2 + 1 = 0 \rightarrow a_1(x_1, x_2) = -3/2 x_1 + x_2 + 1 \geq \text{πάνω/κάτω από } \epsilon_1$
 Ευθεία 2: $-2/3 x_1 + x_2 - 2/3 = 0 \rightarrow a_2(x_1, x_2) = -2/3 x_1 + x_2 - 2/3 \geq \text{πάνω/κάτω από } \epsilon_2$



περιοχή 00: $\text{sign}(a_1) = -1, \text{sign}(a_2) = -1$: κάτω από $\epsilon_1, \epsilon_2 \sim$ **κλάση B**
 περιοχή 01: $\text{sign}(a_1) = -1, \text{sign}(a_2) = 1$: κάτω από ϵ_1 , πάνω από $\epsilon_2 \sim$ **κλάση B**
 περιοχή 10: $\text{sign}(a_1) = 1, \text{sign}(a_2) = -1$: πάνω από ϵ_1 , κάτω από $\epsilon_2 \sim$ **κλάση A**
 περιοχή 11: $\text{sign}(a_1) = 1, \text{sign}(a_2) = 1$: πάνω από $\epsilon_1, \epsilon_2 \sim$ **κλάση B**

2. Με βάση την απάντησή σας στο προηγούμενο ερώτημα κατασκευάστε ένα νευρωνικό δίκτυο που να λύνει το πρόβλημα.

$\langle (-1, -1), B \rangle: f(a_3(-1, -1)) < 0: -w_1 - w_2 + w_0 < 0$
 $\langle (-1, 1), B \rangle: f(a_3(-1, 1)) < 0: -w_1 + w_2 + w_0 < 0$
 $\langle (1, -1), A \rangle: f(a_3(1, -1)) \geq 0: w_1 - w_2 + w_0 \geq 0$
 $\langle (1, 1), B \rangle: f(a_3(1, 1)) < 0: w_1 + w_2 + w_0 < 0$