Υλοποιώντας λογικές πύλες χρησιμοποιώντας perceptrons

Ένας μικρός οδηγός

Λευτέρης Ασλάνογλου
Προπτυχιακός Φοιτητής Μηχανικών Η/Υ & Πληροφορικής Πάτρας

Τρίτη, 5 Ιουνίου 2012

Το παρακάτω είναι ένα tutorial για όσους δυσκολεύονται να καταλάβουν τη διαδικασία υλοποίησης μιας λογικής πύλης διαμορφώνοντας κατάλληλα ένα perceptron. Τα perceptron ανήκουν στα Νευρωνικά Δίκτυα, στα οποία γίνεται μια εισαγωγή στις Ευρετικές, και αποτελούν το αντικείμενο του μαθήματος επιλογής Υπολογιστική Νοημοσύνη 1.

Τα perceptrons μπορούν να χρησιμοποιηθούν όχι μόνο για λογικές πύλες, και υπάρχουν και τρόποι ώστε τα νευρωνικά δίκτυα δεδομένου ενός συνόλου εκπαίδευσης (διάφορες είσοδοι και οι διάφορες ανάλογες αναμενόμενες έξοδοι) να "μάθει από μόνο του" να υλοποιεί την οποιαδήποτε συνάρτηση, διαμορφώνοντας κατάλληλα τα βάρη του και τα κατώφλιά του. Αυτή η χρησιμότητα μας επιτρέπει να προβλέψουμε αποτελέσματα για φαινόμενα τα οποία δεν ξέρουμε ακριβώς "πώς δουλεύουν", αλλά βάση παρατήρησης έχουμε φτιάξει ένα αρκετά μεγάλο σύνολο δεδομένων, το οποίο μπορεί να εκπαιδεύσει ένα νευρωνικό δίκτυο στη συμπεριφορά του φαινομένου που θέλουμε να μελετήσουμε, και το νευρωνικό δίκτυο να μας απαντήσει (με σχετική ακρίβεια) την επιρροή που θα είχε το φαινόμενο σε μια είσοδο για την οποία δεν έχουμε παρατηρήσει ακόμα έξοδο από το φαινόμενο.

Αυτά βέβαια υπάρχουν για να κάνουν το μάθημα της ΥΝ1 ακόμα πιο ενδιαφέρον, αλλά είναι πέρα από την ύλη τόσο θεωρητικά όσο και πρακτικά της ΕΕΜ.

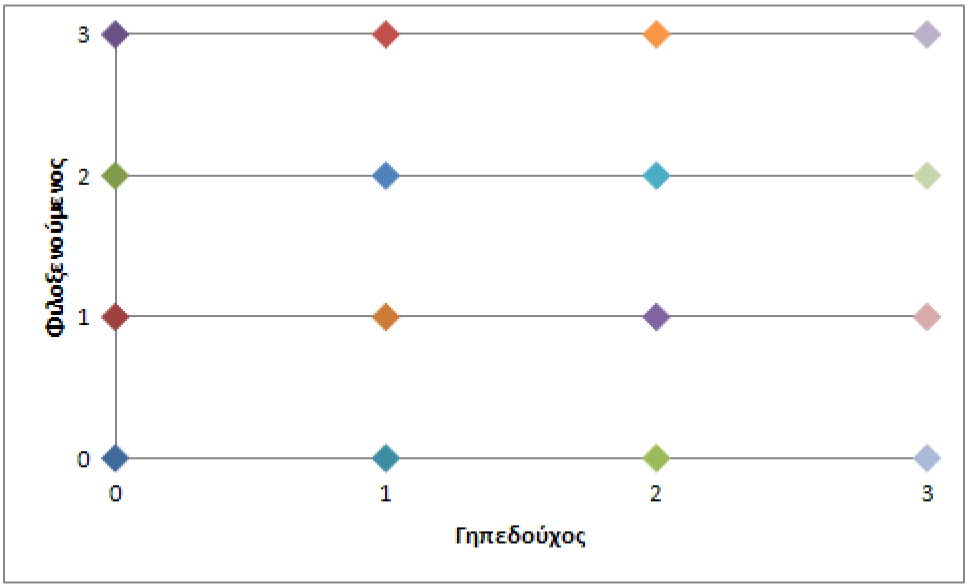
Στις ασκήσεις μας ζητάται η υλοποίηση λογικών πυλών, που ουσιαστικά ανήκει στην ευρύτερη κατηγορία της κατηγοριοποίησης σε κλάσεις. Η κατηγοριοποίηση σε κλάσεις ουσιαστικά είναι φαινόμενα/συστήματα που για οποιαδήποτε είσοδο, δίνουν μία εκ των εξόδων σε ένα σύνολο.

Μέρος 1 – Δείξε μου το σκορ σου, να σου πω τι έγινε

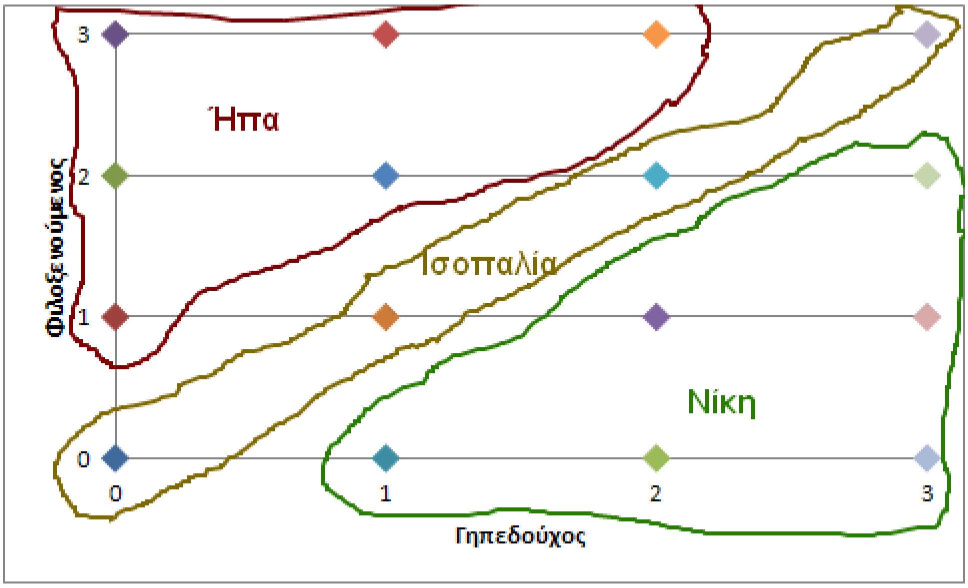
Ένα απλό πρόβλημα κατηγοριοποίησης σε κλάσεις είναι το κατά πόσο ένας ποδοσφαρικός αγώνας έληξε με νίκη, ισοπαλία ή ήττα του γηπεδούχου. Το σύνολο των κατηγοριών είναι το {ΝΙΚΗ, ΙΣΟΠΑΛΙΑ, ΗΤΤΑ}, και έχουμε δύο εισόδους, το σκορ του γηπεδούχου, και το σκορ του φιλοξενούμενου.

Αν το σκορ του γηπεδούχου είναι μεγαλύτερο από το σκορ του φιλοξενούμενου, τότε μιλάμε για νίκη. Αν το σκορ του γηπεδούχου είναι το ίδιο με το σκορ του φιλοξενούμενου, τότε μιλάμε για ισοπαλία. Αν το σκορ του γηπεδούχου είναι μικρότερο, τότε μιλάμε για ήττα.

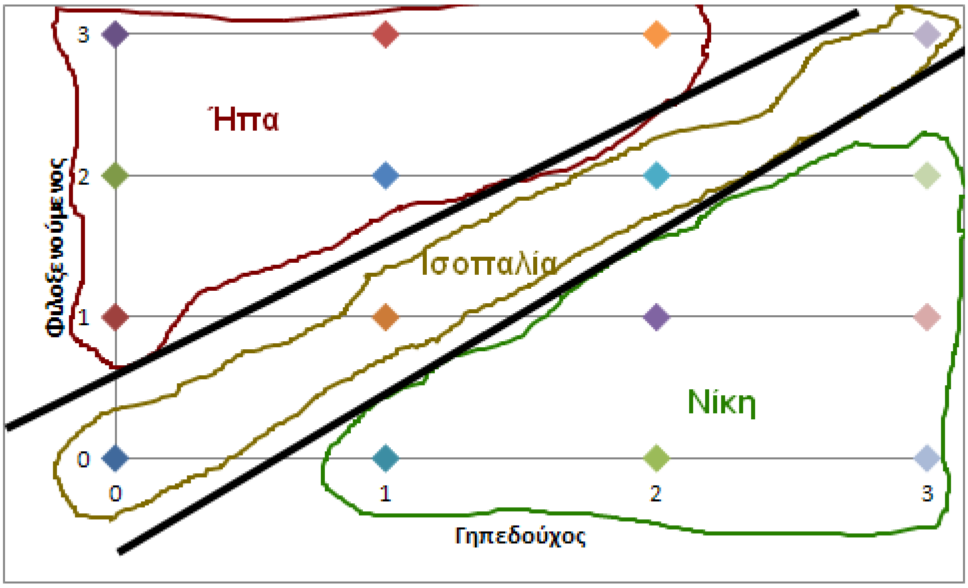
Το πρώτο πράγμα που πρέπει να σκεφτούμε πριν αποφασίσουμε ότι κάτι μπορεί να υλοποιηθεί με perceptron, είναι κατά πόσο οι κλάσεις είναι γραμμικά διαχωρίσιμες. Πρακτικά, αυτό σημαίνει ότι αν τοποθετήσω όλες τις πιθανές εισόδους σε ένα χώρο, κατά πόσο μπορώ να χωρίσω γραμμικά τις εισόδους στις ανάλογες εξόδους. Ας δούμε λοιπόν πώς θα μπορούσαμε να αναπαραστήσουμε κάτι τέτοιο.



Όπως βλέπουμε, όλα τα πιθανά αποτελέσματα μέχρι 3 γκολ αναπαριστώνται σε αυτό το διάγραμμα. Κάθε αποτέλεσμα έχει ως συντεταγμένη x τα γκολ του γηπεδούχου, και ως συντεταγμένη y τα γκολ του φιλοξενούμενου. Ας δούμε λοιπόν, ποιοι συνδυασμοί εισόδου δίνουν νίκη, ισοπαλία και ήττα για τον γηπεδούχο.



Δημιουργούνται 3 ξεκάθαρες περιοχές, μία για κάθε έξοδο/κλάση. Το ερώτημα τώρα είναι, μπορώ να τραβήξω μία ευθεία μεταξύ κάθε δύο γειτονικών περιοχών ώστε να τις χωρίσω; Αν ναι, μπορώ να υλοποιήσω αυτή την κατηγοριοποίηση με perceptron. Αν όχι, πρέπει να βρω άλλο τρόπο να την υλοποιήσω. Όπως όμως θα δούμε, στην προκειμένη περίπτωση, μπορούμε.



Μια χαρά ως εδώ. Βλέπω ότι είναι εφικτό αυτό που ζητάω. Τα perceptron λοιπόν, σε κάθε είσοδο εφαρμόζουν ένα βάρος. Η χρησιμότητά τους δε μας είναι ξεκάθαρη ακόμα, αλλά ας υποθέσουμε το πλέον απλό για αρχή. Έστω ότι το πρόβλημα λύνεται αν θέσουμε και τα δύο βάρη στο 1.

Γ	Φ	Αποτέλεσμα
0	0	Ισοπαλία
0	1	Ήττα
0	2	Ήττα
0	3	Ήττα
1	0	Νίκη
1	1	Ισοπαλία
1	2	Ήττα
1	3	Ήττα
2	0	Νίκη
2	1	Νίκη
2	2	Ισοπαλία
2	3	Ήττα
3	0	Νίκη
3	1	Νίκη
3	2	Νίκη
3	3	Ισοπαλία

Όλες οι πιθανές είσοδοι, και ποιο ήταν το αποτέλεσμα του αγώνα

Γ	Φ	Αποτέλεσμα	Γ*1	Φ*1	Γ*1+Φ*1
0	0	Ισοπαλία	0	0	0
0	1	Ήττα	0	1	1
0	2	Ήττα	0	2	2
0	3	Ήττα	0	3	3
1	0	Νίκη	1	0	1
1	1	Ισοπαλία	1	1	2
1	2	Ήττα	1	2	3
1	3	Ήττα	1	3	4
2	0	Νίκη	2	0	2
2	1	Νίκη	2	1	3
2	2	Ισοπαλία	2	2	4
2	3	Ήττα	2	3	5
3	0	Νίκη	3	0	3
3	1	Νίκη	3	1	4
3	2	Νίκη	3	2	5
3	3	Ισοπαλία	3	3	6

Έστω ότι τα βάρη είναι και τα δύο ίσα με 1

Γ	Φ	Αποτέλεσμα	Γ*1	Φ*1	Γ*1+Φ*1
0	1	Ήττα	0	1	1
0	2	Ήττα	0	2	2
0	3	Ήττα	0	3	3
1	2	Ήττα	1	2	3
1	3	Ήττα	1	3	4
2	3	Ήττα	2	3	5
0	0	Ισοπαλία	0	0	0
1	1	Ισοπαλία	1	1	2
2	2	Ισοπαλία	2	2	4
3	3	Ισοπαλία	3	3	6
1	0	Νίκη	1	0	1
2	0	Νίκη	2	0	2
2	1	Νίκη	2	1	3
3	0	Νίκη	3	0	3
3	1	Νίκη	3	1	4
3	2	Νίκη	3	2	5

Ο ίδιος πίνακας, αλλά με ταξινομημένα τα ματς ανάλογα το αποτέλεσμα

Βλέπουμε ότι κάτι πάει στραβά. Οι κλάσεις μας δεν είναι και πολύ διαχωρισμένες, καθώς τα αθροίσματα $\sum_{i=1}^{n} x_i w_i$ (μη σας μπερδεύει το x_i, εδώ ας πούμε ότι $x_1 = Σκόρ\_Γηπεδούχου$ και $x_2 = Σκόρ\_Φιλοξενούμενου$) δεν κάνουν και πολύ καλή δουλειά στο να διαχωρίσουν τις ήττες από τις ισοπαλίες και τις νίκες. Γιατί θέλουμε να συμβαίνει αυτό; Γιατί θέλουμε να μπορούμε να εφαρμόσουμε ένα ξεκάθαρο κριτήριο διαχωρισμού ως μια συνάρτηση με **μία** είσοδο. Μια πιθανή τέτοια συνάρτηση, που στην περίπτωση των perceptrons ονομάζονται συναρτήσεις ενεργοποίησης, θα μπορούσε να είναι:

$$\varphi(u) = \begin{cases} NIKH, & u > 0 \\ I\Sigma O\Pi A\Lambda IA, & u = 0 \\ HTTA, & u < 0 \end{cases}$$

Αρχίζουμε και σκεφτόμαστε λοιπόν, πώς θα μπορούσαμε να πειράξουμε τα βάρη ώστε να γίνει πιο ξεκάθαρος ο διαχωρισμός. Ας μην ξεχνάμε ότι το u είναι και αυτό μια συνάρτηση που πατάει στο άθροισμα $\sum_{i=1}^{n} x_i w_i$. Συγκεκριμένα, $u = \sum_{i=1}^{n} x_i w_i + \theta w_\theta$. Βλέπουμε ότι το θ, που ονομάζεται και κατώφλι, δε θα κάνει τίποτα άλλο παρά να «σπρώξει» το άθροισμα γινομένων είσοδος-επί-βάρος πάνω ή κάτω, άρα δε μας ενδιαφέρει προς το παρόν. Προς το παρόν πρέπει να φτιάξουμε τα βάρη μας.

Σκεφτόμαστε λοιπόν λογικά τι είναι αυτό που ψάχνουμε, και πώς η κάθε παράμετρος/είσοδος του προβλήματος επηρεάζει το αποτέλεσμα. Σκεφτόμαστε λοιπόν ότι όσο ευεργετική επιρροή έχει κάθε γκολ του γηπεδούχου στο αποτέλεσμά του, άλλο τόσο αρνητική επιρροή έχει κάθε γκολ του φιλοξενούμενου. Τα βάρη αντιπροσωπεύουν ακριβώς αυτό, το τι επιρροή έχει η κάθε είσοδος στο αποτέλεσμα. Άρα προφανώς, αφού κάθε φορά που δεχόμαστε γκολ από το φιλοξενούμενο το αποτέλεσμα δεν είναι και τόσο καλό, δε μπορούμε να δώσουμε θετικό βάρος στην είσοδο των γκολ του!

Γρήγορα λοιπόν προκύπτει ότι το βάρος του κάθε γκολ του φιλοξενούμενου είναι αρνητικό. Τι τιμή όμως πρέπει να έχει; Ίση κατ' απόλυτη τιμή με το βάρος του γηπεδούχου, ή μήπως μικρότερη ή μεγαλύτερη; Οι κανόνες του ποδοσφαίρου λένε ότι κάθε γκολ, ανεξαρτήτως από ποια ομάδα μπαίνει, έχει την ίδια αξία (εκτός αν παίζεις μέσα-έξω, όπως γίνεται στο Champions League και άλλες διοργανώσεις, αλλά ας πάρουμε την απλή περίπτωση εδώ). Συνεπώς, κατ'απόλυτη τιμή τα δύο βάρη πρέπει να είναι ίδια. Ποια θα είναι αυτή η απόλυτη τιμή; Οποιαδήποτε! Θα μπορούσε να είναι το 1, το 0,000854, ή το 1.354.327.879. Δεν έχει καμία διαφορά. Απλά επειδή το 1 κάνει τους υπολογισμούς (και τη ζωή μας) πιο εύκολη, χρησιμοποιούμε αυτό συνήθως. Ας δούμε λοιπόν πώς διαμορφώνεται ο πίνακας αν εφαρμόσουμε αρνητικό βάρος στα γκολ του φιλοξενούμενου.

Γ	Φ	Αποτέλεσμα	Γ*1	Φ*(-1)	Γ*1+Φ*1
0	1	Ήττα	0	-1	-1
0	2	Ήττα	0	-2	-2
0	3	Ήττα	0	-3	-3
1	2	Ήττα	1	-2	-1
1	3	Ήττα	1	-3	-2
2	3	Ήττα	2	-3	-1
0	0	Ισοπαλία	0	0	0
1	1	Ισοπαλία	1	-1	0
2	2	Ισοπαλία	2	-2	0
3	3	Ισοπαλία	3	-3	0
1	0	Νίκη	1	0	1
2	0	Νίκη	2	0	2
2	1	Νίκη	2	-1	1
3	0	Νίκη	3	0	3
3	1	Νίκη	3	-1	2
3	2	Νίκη	3	-2	1

Παρατηρείτε κάποιο μοτίβο; Ναι, πράγματι, τώρα το άθροισμα στις ήττες είναι πάντα αρνητικό, στις ισοπαλίες 0, και στις νίκες θετικό. Θυμηθείτε τη συνάρτηση ενεργοποίησης που είπαμε θα χρησιμοποιήσουμε:

$$\varphi(u) = \begin{cases} NIKH, & u > 0 \\ I\Sigma O\Pi A\Lambda IA, & u = 0 \\ HTTA, & u < 0 \end{cases}$$

Βλέπουμε λοιπόν ότι αυτή η αλλαγή στο βάρος του φιλοξενούμενου ουσιαστικά μας έδωσε το u! Θα μπορούσαμε εύκολα να πούμε ότι πλέον το άθροισμα δε χρειάζεται κανένα «σπρώξιμο» και να βάλουμε το κατώφλι στο 0, αλλά αυτό θα προϋπόθετε να γνωρίζουμε τι ακριβώς κάνει το κατώφλι, και πώς το βρίσκουμε. Ας πούμε λοιπόν ότι δεν ξέρουμε.

Ας θυμηθούμε τον ορισμό του u.

$$u = \sum_{i=1}^{n} x_i w_i + \theta w_\theta$$

Σε συνδυασμό με τη συνάρτηση ενεργοποίησης, βλέπουμε ότι θέλουμε το u να είναι θετικό για κάθε τιμή του αθροίσματος που συμπεριλαμβάνεται στις νίκες, ίσο με το μηδέν για τις ισοπαλίες, και μικρότερο του μηδέν για τις ήττες.

Σύμφωνα με τον τελευταίο πίνακα, είδαμε ότι η ελάχιστη τιμή που παίρνει το άθροισμα όταν έχουμε νίκη, είναι η 1. Άρα ουσιαστικά θέλουμε

$$\begin{aligned} & u > 0 \Leftrightarrow \\ & \sum_{i=1}^{n} x_i w_i + \theta w_\theta > 0 \\ & 1 + \theta w_\theta > 0 \end{aligned}$$

Το βάρος του κατωφλιού w_θ συνήθως θα μας δίνεται, και έστω ότι είναι +1.

$$\begin{aligned} & 1 + \theta > 0 \\ & \theta > -1 \end{aligned}$$

Η μέγιστη τιμή που παίρνει το άθροισμα όταν έχουμε ήττα, είναι -1, και άρα έχουμε αντίστοιχα

$$\begin{aligned} & (-1) + \theta < 0 \\ & \theta < 1 \end{aligned}$$

Η μόνη τιμή που παίρνει το άθροισμα όταν έχουμε ισοπαλία είναι 0,και άρα έχουμε αντίστοιχα

$$\begin{aligned} & 0 + \theta = 0 \\ & \theta = 0 \end{aligned}$$

Ξέρω ότι φαίνεται περίεργο, αλλά δεν είναι λάθος. Εφ’όσων θέλουμε οι ισοπαλίες να δίνουν u ίσο με το 0, αναγκαστικά το θ θα είναι 0. Αν έπαιρνε οποιαδήποτε άλλη τιμή, το u θα ήταν διάφορο του 0, και άρα η συνάρτηση ενεργοποίησης δε θα κατηγοριοποιούσε σωστά τον αγώνα ως ισοπαλία.

Αφού βρήκαμε και το κατώφλι, ουσιαστικά τελειώσαμε! Το μόνο που μένει είναι να επιβεβαιώσουμε ότι τα κάναμε όλα σωστά, βγάζοντας για κάθε πιθανό αποτέλεσμα το u, και τι τελικά θα απαντούσε η φ, πού δηλαδή θα κατηγοριοποιούσε το κάθε διάνυσμα εισόδου.

Γ	Φ	Αποτέλεσμα	Γ*1	Φ*(-1)	Γ*1+Φ*(-1)	u=Γ*1+Φ*(-1)+θ*1	φ(u)
0	1	Ήττα	0	-1	-1	-1	Ήττα
0	2	Ήττα	0	-2	-2	-2	Ήττα
0	3	Ήττα	0	-3	-3	-3	Ήττα
1	2	Ήττα	1	-2	-1	-1	Ήττα
1	3	Ήττα	1	-3	-2	-2	Ήττα
2	3	Ήττα	2	-3	-1	-1	Ήττα
0	0	Ισοπαλία	0	0	0	0	Ισοπαλία
1	1	Ισοπαλία	1	-1	0	0	Ισοπαλία
2	2	Ισοπαλία	2	-2	0	0	Ισοπαλία
3	3	Ισοπαλία	3	-3	0	0	Ισοπαλία
1	0	Νίκη	1	0	1	1	Νίκη
2	0	Νίκη	2	0	2	2	Νίκη
2	1	Νίκη	2	-1	1	1	Νίκη
3	0	Νίκη	3	0	3	3	Νίκη
3	1	Νίκη	3	-1	2	2	Νίκη
3	2	Νίκη	3	-2	1	1	Νίκη

Βλέπουμε ότι η φ(u) δίνει σωστά το ίδιο αποτέλεσμα με το πραγματικό. Άρα η υλοποίησή μας είναι σωστή.

Το κατώφλι στις περισσότερες περιπτώσεις δε θα μας δίνεται τόσο εύκολα. Ας πούμε, για χάρη πληρότητας, ότι η συνάρτηση ενεργοποίησης ήταν η λιγότερο προφανής

$$\varphi(u) = \begin{cases} NIKH, & u \geq 1 \\ I\Sigma O\Pi A\Lambda IA, & -1 < u < 1 \\ HTTA, & u \leq -1 \end{cases}$$

Θεωρώντας τα ίδια βάρη, θα είχαμε για τις νίκες την ανισότητα

$$\begin{aligned} 1 + \theta &\geq 1 \\ \theta &\geq 0 \end{aligned}$$

Η μέγιστη τιμή που παίρνει το άθροισμα όταν έχουμε ήττα, είναι -1, και άρα έχουμε αντίστοιχα

$$\begin{aligned} (-1) + \theta &\leq -1 \\ \theta &\leq 0 \end{aligned}$$

Καρφώθηκε το πιο είναι το κατώφλι πάλι, αλλά ας πούμε ότι δεν το είδαμε. Η μόνη τιμή που παίρνει το άθροισμα όταν έχουμε ισοπαλία είναι 0, και άρα έχουμε αντίστοιχα

$$\begin{aligned} -1 < 0 + \theta < 1 \\ -1 < \theta < 1 \end{aligned}$$

Άρα το θ πρέπει να είναι μεταξύ -1 και 1, μικρότερο ή ίσο του 0, και μεγαλύτερο ή ίσο του 0. Ποια είναι η μόνη τιμή που πληρεί όλες αυτές τις προϋποθέσεις; Το 0. Το κατώφλι, όπως θα δούμε, μπορεί να μην παίρνει μία τιμή, αλλά ένα εύρος τιμών, και οποιαδήποτε τιμή και αν διαλέξουμε από αυτό το εύρος, το αποτέλεσμα να είναι σωστό. Σε αυτό το πρόβλημα, έτυχε μόνο μία τιμή να ταιριάζει σε όλες τις προϋποθέσεις. Όπως και να'χει, οποιαδήποτε τιμή ταιριάζει, θα δίνει σωστό αποτέλεσμα, εφ'όσων έχουμε σχεδιάσει το perceptron σωστά.

Μέρος 2 – The logic gate that could

Καλά όλα αυτά, αλλά για λογικές πύλες δεν είπαμε ακόμα. Τι σχέση έχει η μπάλα με τις λογικές πύλες; Στο επίπεδο των νευρωνικών δικτύων, οι ομοιότητες είναι πολλές!

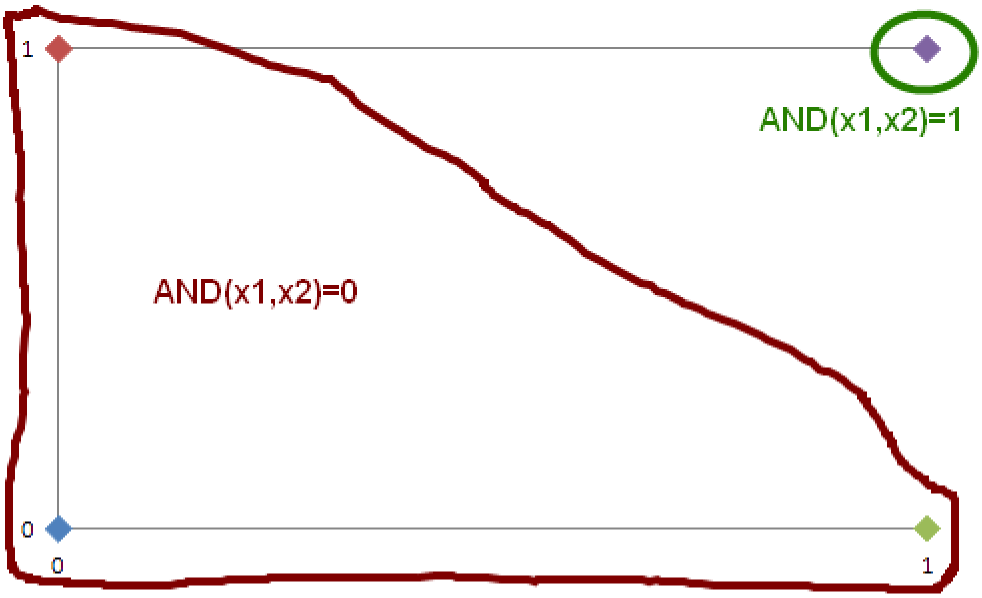
Η λογική πύλη δεν είναι τίποτα άλλο παρά ένα κύκλωμα που δέχεται μία ή περισσότερες εισόδους, και απαντάει βγάζοντας την ανάλογη τάση στην έξοδο, αφού εσωτερικά εκτελέσει την ανάλογη λογική πράξη. Ανεξαρτήτως των προδιαγραφών των λογικών πυλών και το αν βγάζουν 1KV για θετική έξοδο και -500V για αρνητική, η σύμβαση είναι να τις σκεφτόμαστε ως δυαδικά στοιχεία. Κυκλώματα που παίρνουν 0 και 1, και απαντούν με 0 και 1. Παρατηρείτε το μοτίβο;

Ό,τι είσοδο και να δώσουμε σε μια λογική πύλη, θα μας απαντήσει είτε 0, είτε 1. Άρα οι λογικές πύλες κάνουν ακριβώς αυτό που περιγράφαμε και πριν, κατηγοριοποίηση σε κλάσεις, δύο μάλιστα στην περίπτωση μας: την «κλάση 0», και την «κλάση 1».

Ας πάρουμε για παράδειγμα την πύλη AND. Η πύλη AND μας επιστρέφει 1 μόνο στην περίπτωση που όλες οι είσοδοι είναι 1. Για δύο εισόδους (για λόγους απλότητας της αναπαράστασης), έχουμε τον εξής πίνακα αλήθειας.

x1	x2	AND (x1, x2)
0	0	0
0	1	0
1	0	0
1	1	1

Αν σχεδιάσουμε αυτές τα 4 πιθανά διανύσματα εισόδου στο δισδιάστατο χώρο, προκύπτει το εξής.

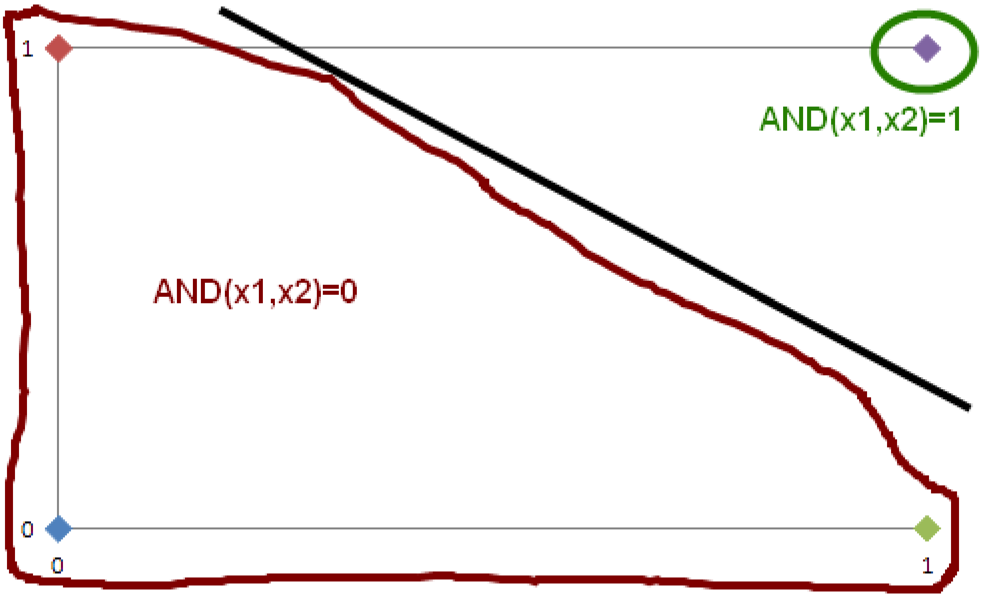


Για να θυμηθούμε τα βήματα που απαιτούνται ξανά.

1. Πίνακας εισόδων
2. Διάγραμμα στο χώρο
3. Γραμμικός διαχωρισμός κλάσεων

4. Ορισμός βαρών
    a. Υπόθεση ότι $w_i = 1$
    b. Πρέπει να είναι ίσα;
    c. Πρέπει μήπως κάποια να είναι αρνητικά;
5. Ορισμός κατωφλιού
6. Επιβεβαίωση

Το 1 και το 2 ολοκληρώθηκαν. Όπως είπαμε και στο $1^{ο}$ μέρος, για να μπορεί να υλοποιηθεί ένα πρόβλημα κατηγοριοποίησης σε κλάσεις με perceptron, πρέπει οι κλάσεις να είναι γραμμικά διαχωρίσιμες. Δύο είσοδοι σημαίνουν δισδιάστατη αναπαράσταση, και άρα κάνουν αρκετά εύκολο το να δούμε ότι πράγματι, μπορούν να χωριστούν γραμμικά τα σημεία εισόδου που δίνουν 1 στην έξοδο, και αυτά που δίνουν 0. Αν δυσκολεύεστε να αντιληφθείτε το γιατί, πάρτε ένα μολύβι και προσπαθήστε να το τοποθετήσετε μεταξύ των δύο ομάδων εισόδων, και να τις χωρίσετε. Θα δείτε ότι πράγματι αυτό γίνεται.



Αφού λοιπόν βλέπουμε ότι μπορεί να υλοποιηθεί η πύλη AND με Perceptron, καιρός να βρούμε τα βάρη. Αρχικά υποθέτουμε ότι το πρόβλημα λύνεται αν τα βάρη είναι και τα δύο 1. Δε βασίζουμε κάπου αυτή την υπόθεση, απλά είναι ο πιο εύκολος τρόπος να δούμε μια αρχική συμπεριφορά του προβλήματος στην κατηγοριοποίηση βάσει αθροίσματος, την οποία κάνουν τα perceptron.

x1	x2	AND (x1, x2)	x1*1	x2*1	x1*1+x2*1
0	0	0	0	0	0
0	1	0	0	1	1
1	0	0	1	0	1
1	1	1	1	1	2

Βλέπουμε λοιπόν ότι το πρόβλημα αυτό συμπεριφέρεται εξ'αρχής ωραία, γιατί υπάρχει ένας ξεκάθαρος διαχωρισμός μεταξύ των εισόδων που δίνουν έξοδο 1 και αυτών που δίνουν 0. Συγκεκριμένα, παρατηρούμε ότι για έξοδο 0 το άθροισμα$\sum_{i=1}^{n} x_i w_i$ είναι μικρότερο ή ίσο του 1, ενώ για έξοδο 1 το άθροισμα είναι 2.

Όπως είναι λογικό, στην AND και οι δύο είσοδοι συνεισφέρουν εξ'ίσου στο να προκύψει το αποτέλεσμα, άρα τα βάρη θα είναι κατ'απόλυτη τιμή ίσα. Το πρόσημό τους θα είναι και αυτό το ίδιο, αφού πέρα από το ότι συνεισφέρουν το ίδιο, συνεισφέρουν προς την ίδια κατεύθυνση, χρειάζονται το ένα το άλλο κατά κάποιον τρόπο. Σε αντίθεση με πριν, που αν είχαμε βάλει ένα γκολ, όταν δεχόμασταν ένα το αποτέλεσμα γινόταν χειρότερο, τώρα αν έχουμε τη μία είσοδο στο 1 και γυρίσει και η άλλη από το 0 στο 1, η πύλη αλλάζει το αποτέλεσμα για να απαντήσει θετικά.

ΟΚ λοιπόν, τα βάρη είναι ίδια, και ίδιου προσήμου. Γιατί όμως να τα αφήσω στο +1 ή στο +125, και να μην τα κάνω και τα δύο -1 ή -125; Η απάντηση πάντα κρίνεται από τη συνάρτηση ενεργοποίησης. Μια πολύ γνωστή συνάρτηση ενεργοποίησης που χρησιμοποιείται συχνά, ειδικά στην κατηγοριοποίηση σε δύο κλάσεις, είναι η McCulloch-Pitts:

$$\varphi(u) = \begin{cases} +1, & u \geq 0 \\ 0, & u < 0 \end{cases}$$

Βλέπουμε λοιπόν ότι για μη αρνητικά u, το perceptron θα απαντά +1, ενώ για αρνητικά u, θα απαντάει 0. Και εδώ τίθεται το κρίσιμο ερώτημα. Οι είσοδοι που δίνουν μεγαλύτερα αθροίσματα σε σχέση με τις υπόλοιπες εισόδους, θέλουμε να δίνουν και μεγαλύτερο u σε σχέση με τις υπόλοιπες;

Ας το σκεφτούμε λίγο. Η φ θέλουμε να απαντάει 1, όταν και η AND θα απαντούσε 1. Άρα όταν το άθροισμα βγαίνει 2 επειδή και οι δύο είσοδοι είναι 1, θέλουμε η φ να απαντάει θετικά. Και πώς θα γίνει αυτό; Θα πρέπει η u να είναι μη αρνητική. Καλά ως εδώ, τι γίνεται στην αντίθετη περίπτωση; Η φ θέλουμε να απαντάει 0, προφανώς, όταν και η AND θα απαντούσε 0. Άρα όταν το άθροισμα βγαίνει μικρότερο ή ίσο του 1 επειδή τουλάχιστον μία από τις εισόδους δεν είναι 1, θέλουμε η u να είναι αρνητική, ώστε η φ να απαντάει 0.

Ποιο είναι το μοτίβο εδώ; Το βλέπετε; Μεγαλύτερο άθροισμα $\sum_{i=1}^{n} x_i$ (ναι, χωρίς τα βάρη, γι'αυτό και αρχικά τα ορίζουμε στο 1) θέλουμε να συνεπάγεται και μεγαλύτερο u. Άρα τα βάρη πρέπει να συνεισφέρουν στην αύξηση του αθροίσματος $\sum_{i=1}^{n} x_i w_i$. Συνεπώς, πρέπει να είναι θετικά!

Ίδια κατ'απόλυτη τιμή, ίδιο πρόσημο, θετικά. Ποια είναι η πιο εύκολη επιλογή; Μα το +1 προφανώς! Άρα δε χρειάζεται να πειράξουμε τα βάρη που έχουμε θέσει, αφήνουμε και τα δύο στο +1.

Το επόμενο βήμα είναι να ορίσουμε το κατώφλι. Για να μελετήσουμε και μια διαφορετική περίπτωση, ας πούμε ότι για κάποιο λόγο το βάρος κατωφλίου είναι -2. Θα μπορούσε να είναι και -1, αλλά ας το κάνουμε πιο δύσκολο, μόνο και μόνο για να δείξουμε ότι το κατώφλι δεν είναι τίποτα άλλο παρά άλλη μία (σταθερή) είσοδος με προκαθορισμένο βάρος. Θυμηθείτε ότι

$$u = \sum_{i=1}^{n} x_i w_i + \theta w_\theta.$$

Γνωρίζουμε τις εισόδους και τα βάρη, το βάρος κατωφλίου, γνωρίζουμε και τι τιμές θέλουμε να παίρνει το u, άρα μένει να βρούμε το θ, το κατώφλι το ίδιο δηλαδή.

Όπως και πριν, εξετάζουμε το u για κάθε κλάση. Όταν το u πρέπει να είναι μεγαλύτερο από κάτι, εξετάζουμε την ελάχιστη τιμή του αθροίσματος $\sum_{i=1}^{n} x_i w_i$ των εισόδων που εντάσσονται σε αυτή την κλάση. Όταν το u πρέπει να είναι μικρότερο από κάτι, εξετάζουμε τη μέγιστη τιμή.

Για την κλάση 0 λοιπόν, της οποίας μέγιστη τιμή αθροίσματος είναι το 1, θέλουμε να ισχύει

$$1 + \theta * (-2) < 0$$

$$\theta * (-2) < -1$$
$$\theta * 2 > 1$$
$$\theta > \frac{1}{2}$$

Για την κλάση 1, της οποίας ελάχιστη τιμή αθροίσματος είναι το 2, θέλουμε να ισχύει

$$2 + \theta * (-2) \geq 0$$
$$\theta * (-2) \geq -2$$
$$\theta * 2 \leq 2$$
$$\theta \leq 1$$

Συνεπώς

$\frac{1}{2} < \theta \leq 1.$

Πράγματι, όποιο και να είναι το θ, αν είναι εντός αυτού του εύρους, το αποτέλεσμα θα είναι σωστό. Επιβεβαιώνουμε με την εύκολη επιλογή 1, αλλά και με μια λίγο πιο περίεργη επιλογή, το 0,8. Χάριν πληρότητας, θα δοκιμάσουμε ως κατώφλι και το 0, που είναι εκτός του επιτρεπτού εύρους, για να δείξουμε ότι η υλοποίηση θα βγει λάθος.

x1	x2	AND (x1, x2)	x1*1	x2*1	x1*1+x2*1	u=x1*1+x2*1+1*(-2)	φ(u)	u=x1*1+x2*1+0.8*(-2)	φ(u)	u=x1*1+x2*1+0*(-2)	φ(u)
0	0	0	0	0	0	-2	0	-1.6	0	0	1
0	1	0	0	1	1	-1	0	-0.6	0	1	1
1	0	0	1	0	1	-1	0	-0.6	0	1	1
1	1	1	1	1	2	0	1	0.4	1	2	1

Πράγματι, τα πρώτα δύο κατώφλια, 1 και 0.8, έδωσαν φ(u) που ταιριάζει στην AND για την ίδια είσοδο. Αντιθέτως, το κατώφλι 0 το οποίο είναι και εκτός του επιτρεπτού εύρους, επιστρέφει κάτι διαφορετικό, κάτι λάθος.

Μέρος 3 – The gate that didn't care

Πριν περάσουμε σε ένα παράδειγμα πύλης που δεν υλοποιείται με perceptron, θέλω να δούμε ένα παράδειγμα μιας πύλης που ουσιαστικά δεν έχει λόγω ύπαρξης, μόνο και μόνο για να δείξω μια συγκεκριμένη σκέψη στην επιλογή των βαρών. Η πύλη αυτή υλοποιεί την εξής συνάρτηση:

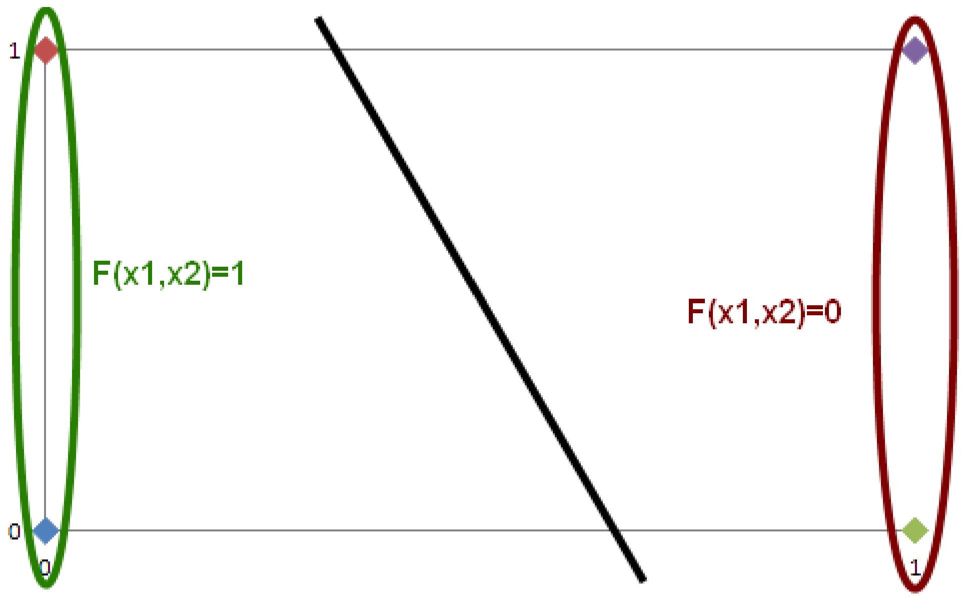
$$f(x_1, x_2) = \begin{cases} 1, & x_1 = 0 \\ 0, & \alpha\lambda\lambda o\acute{\upsilon} \end{cases}$$

Θα μου πείτε «κάτσε λίγο, γιατί έχει δύο εισόδους αν δεν ασχολείται καθόλου με την τιμή της x_2;», και θα έχετε δίκιο ότι δεν έχει πολύ νόημα. Αλλά μην αφήσετε να σας απασχολήσει αυτό. Το όλο θέμα είναι στη σκέψη που θα χρειαστούμε στα πλαίσια αυτού του παραδείγματος.

Ο πίνακας αλήθειας αυτής της πύλης/συνάρτησης είναι:

x1	x2	F (x1, x2)
0	0	1
0	1	1
1	0	0
1	1	0

Το διάγραμμα στο δισδιάστατο χώρο που δείχνει πώς χωρίζονται οι καταστάσεις:



Βλέπουμε ότι οι δύο κλάσεις είναι γραμμικά διαχωρίσιμες, άρα μπορεί να υλοποιηθεί η πύλη αυτή με perceptron.

Θεωρώντας αρχικά τα βάρη ίσα με +1 ως σωστά, έχουμε:

x1	x2	F (x1, x2)	x1*1	x2*1	x1*1+x2*1
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	1	0	1	1	2

Αμέσως προκύπτει ένα πρόβλημα. Παρ'ό,τι η F είναι 1 για είσοδο (0,1) και 0 για είσοδο (1,0), το άθροισμα $\sum_{i=1}^{n} x_i w_i$ είναι το ίδιο και για τις 2. Προφανώς, αυτά τα βάρη δεν είναι κατάλληλα για τη ζητούμενη κατηγοριοποίηση σε κλάσεις.

Ας σκεφτούμε λίγο λοιπόν τι κάνει αυτή η πύλη, και ποιος είναι ο ρόλος των βαρών. Είπαμε ότι ουσιαστικά τα βάρη ορίζουν τη συνεισφορά κάθε μεταβλητής στο αποτέλεσμα. Η συγκεκριμένη πύλη επηρεάζεται όμως μόνο από τη x_1! Η x_2 δεν έχει απολύτως καμία συνεισφορά εδώ! Ή, ας πούμε ότι η x_2 έχει μικρότερη συνεισφορά από τη x_1, με την έννοια ότι το «καθόλου συνεισφορά» είναι λιγότερο από το «κάποια συνεισφορά».

Συνεπώς, το βάρος της x_2 θα πρέπει να είναι σίγουρα μικρότερο της x_1, και το πιο σωστό είναι αφού η x_2 έχει μηδενική συνεισφορά στο αποτέλεσμα, να έχει και μηδενικό βάρος.

x1	x2	F (x1, x2)	x1*1	x2*0	x1*1+x2*0
0	0	1	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	1	0	1	0	1

Νομίζω ότι είναι ξεκάθαρος πλέον ο διαχωρισμός του αθροίσματος μεταξύ των εισόδων που δίνουν 1 στην έξοδο, και αυτών που δίνουν 0. Το 0 δεν έχει πρόσημο, άρα δε μας ενδιαφέρει αν το πρόσημο των δύο βαρών θα είναι κοινό ή όχι. Αυτό που μας ενδιαφέρει είναι τι πρόσημο θα έχει το βάρος του x_1. Μπορεί κατ'απόλυτη τιμή να είναι σωστές οι επιλογές μας, αφού υπάρχει διαφοροποίηση, όμως το x_1 στην πραγματικότητα έχει «θετική» ή «αρνητική» συνεισφορά;

Ας το σκεφτούμε λίγο. Ανεξαρτήτως του x_2, όταν το x_1 είναι 0, η πύλη επιστρέφει 1. Όταν το x_1 γίνει 1, η πύλη επιστρέφει 0. Βλέπουμε λοιπόν ότι όταν το άθροισμα $\sum_{i=1}^{n} x_i w_i$ είναι μεγαλύτερο, ουσιαστικά το u θα πρέπει να είναι μικρότερο απ'ό,τι στις άλλες εισόδους, γιατί χρησιμοποιώντας τη McCulloch-Pitts, θέλουμε όπου η πύλη επιστρέφει 0, το u να είναι αρνητικό. Βλέπουμε λοιπόν ότι τα βάρη που έχουμε διαλέξει δεν κάνουν για την περίπτωση αυτής της πύλης. Το x_1 έχει «αρνητική συνεισφορά», γιατί καθώς αυτό μεγαλώνει, η έξοδος της πύλης μικραίνει. Και είσοδοι με αρνητική συνεισφορά, όπως μάθαμε στο 1ο μέρος, πρέπει να έχουν αρνητικό βάρος.

Έστω λοιπόν ότι το βάρος w_1 είναι -1.

x1	x2	F (x1, x2)	x1*(-1)	x2*0	x1*(-1)+x2*0
0	0	1	0	0	0
0	1	1	0	0	0
1	0	0	-1	0	-1
1	1	0	-1	0	-1

Βλέπουμε ότι πλέον το άθροισμα είναι όπως το θέλουμε. Μικρότερες τιμές εκεί που η πύλη είναι 0, μεγαλύτερες εκεί που η πύλη είναι 1. Μπορούμε δηλαδή να συνεχίσουμε, βρίσκοντας το κατώφλι. Έστω ότι το βάρος του είναι +1, άρα έχουμε συνοπτικά:

$$\varphi(u) = \begin{cases} +1, & u \geq 0 \\ 0, & u < 0 \end{cases}$$

$$0 + \theta \geq 0$$
$$\theta \geq 0$$

$$-1 + \theta < 0$$
$$\theta < 1$$

άρα

$$0 \leq \theta < 1$$

Πράγματι, αρκεί το θ να είναι εντός αυτού του εύρους, και το perceptron θα δίνει μέσω της συνάρτησης ενεργοποίησης την ίδια απάντηση που θα έδινε και η πύλη.

x1	x2	F (x1, x2)	x1*(-1)	x2*0	x1*(-1)+x2*0	=x1*(-1)+x2*0+0.99*	ϕ(u)	u=x1*(-1)+x2*0.5+1*1	ϕ(u)	u=x1*(-1)+x2*0+1*1	ϕ(u)
0	0	1	0	0	0	0.99	1	0.5	1	1	1
0	1	1	0	0	0	0.99	1	0.5	1	1	1
1	0	0	-1	0	-1	-0.01	0	-0.5	0	0	1
1	1	0	-1	0	-1	-0.01	0	-0.5	0	0	1

Βλέπουμε ότι για κατώφλια εντός του εύρος (0,99 και 0,5), το αποτέλεσμα είναι σωστό, ενώ εκτός εύρους (1), το αποτέλεσμα είναι διαφορετικό από το ζητούμενο.

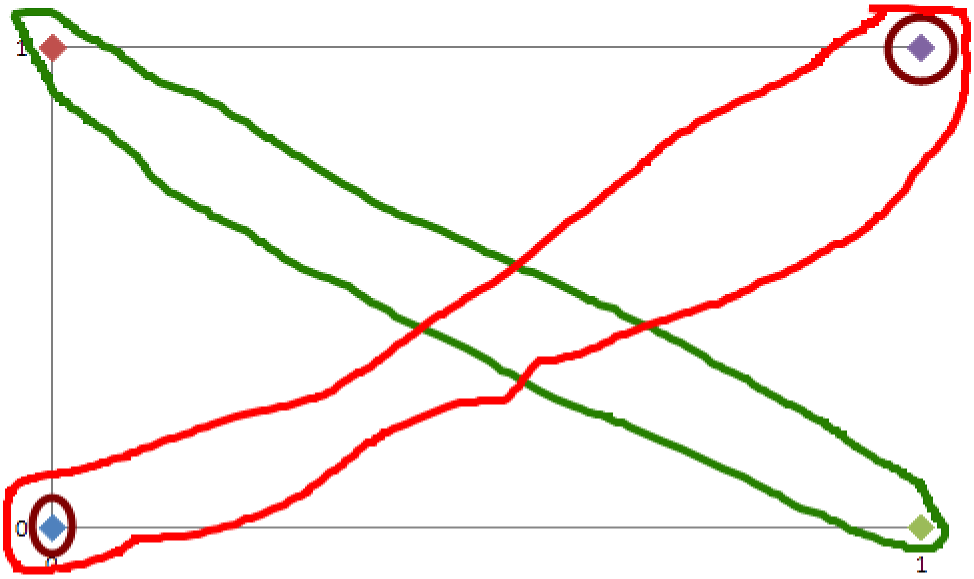
Μέρος 4 – The gate that couldn't

Πάμε να δούμε λοιπόν και ένα χαρακτηριστικό παράδειγμα λογικής δυαδικής πύλης (και άρα προβλήματος κατηγοριοποίησης σε δύο κλάσεις) που δε γίνεται να υλοποιηθεί με perceptron, επειδή δεν είναι γραμμικά διαχωρίσιμο. Ένα τέτοιο παράδειγμα είναι η πύλη XOR.

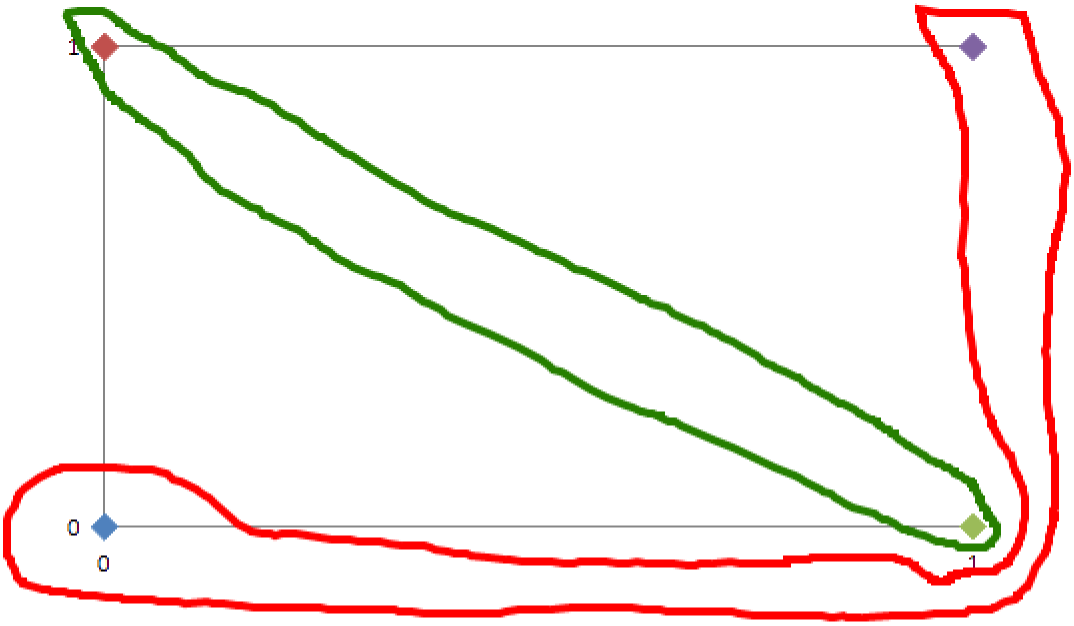
Πίνακας αλήθειας:

x1	x2	XOR (x1, x2)
0	0	0
0	1	1
1	0	1
1	1	0

Το διάγραμμά του στο δισδιάστατο χώρο:



Βλέπουμε λοιπόν ένα πρόβλημα εξ'αρχής. Αν κυκλώσουμε ομαδικά τα σημεία που προκύπτουν από διανύσματα εισόδου που δίνουν έξοδο 1, και αυτά που δίνουν έξοδο 0, οι δύο ομαδοποιήσεις «συγκρούονται», και δεν υπάρχει τρόπος να διαχωριστούν με μια γραμμή. Και όχι, η παρακάτω «εξυπνάδα» δεν αλλάζει κάτι:



Όσο και να προσπαθήσετε, δε θα μπορέσετε να τραβήξετε μία, και μόνο μία, ευθεία που να χωρίζει αυτές τις δύο κλάσεις. Απαιτούνται δύο ευθείες, αλλά τότε πλέον μιλάμε για πρόβλημα κατηγοριοποίησης σε 3 κλάσεις, που δεν είναι όμως το ζητούμενο.

Συνεπώς, η πύλη XOR δε μπορεί να υλοποιηθεί χρησιμοποιώντας perceptron.

Μέρος 5 – Τελειώσαμε επιτέλους;

Ναι, αυτό ήταν όλο κι όλο! Ελπίζω να σας βοήθησα! Για όποιες απορίες σχετικά με perceptrons και την επίλυση προβλημάτων κατηγοριοποίησης σε κλάσεις μπορείτε να απευθυνθείτε στους διδάσκοντες των Ευρετικών και της Υπολογιστικής Νοημοσύνης 1, αλλά και να μου στείλετε mail στο leftos@gmail.com. Στο ίδιο mail περιμένω και τις παρατηρήσεις και τα σχόλιά σας σχετικά με αυτόν τον οδηγό, αλλά και επισημάνσεις για όποιο λάθος μπορεί να βρείτε.

Καλή επιτυχία!