

Μη Γραμμικοί Ταξινομητές - Το δίκτυο Perceptron

ECE-TEL830 ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ

Αθανάσιος Κούτρας
Αναπληρωτής Καθηγητής

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών,
Παν. Πελοποννήσου

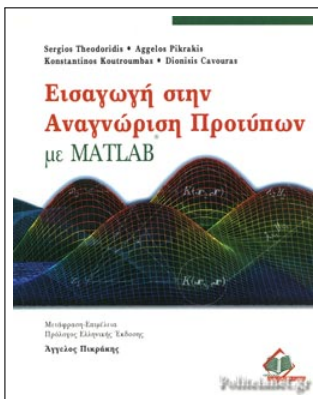
7 Μαΐου 2023

Περιγραμματα διάλεξης

- 1 Εισαγωγή
- 2 Το perceptron δύο επιπέδων
- 3 Πολυεπίπεδα perceptrons
- 4 Εκπαίδευση δικτύων perceptron
- 5 Σχολιασμός
- 6 Παραλλαγές οπισθοδιάδοσης
- 7 Η διαδικασία της αναγνώρισης
- 8 Πρακτικά ζητήματα εκπαίδευσης

Υλικό μελέτης

Theodoridis S., Piskrakis, A., Koutroumbas K., Cavouras, D., "Εισαγωγή στην αναγνώριση προτύπων με MATLAB", ΚΕΦΑΛΑΙΟ 2

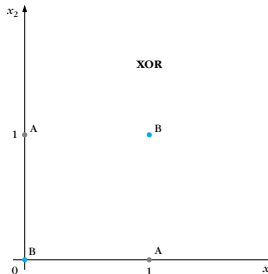


Εισαγωγή

- Στη διάλεξη αυτή θα παρουσιαστεί μια διαφορετική προσέγγιση για το σχεδιασμό πιο αποδοτικών ταξινομητών οι οποίοι μπορούν να εφαρμοστούν και να λειτουργήσουν καλύτερα σε προβλήματα μη γραμμικά διαχωρίσιμων κλάσεων από ότι οι γραμμικοί ταξινομητές που παρουσιάστηκαν σε προηγούμενες διαλέξεις.
- Συγκεκριμένα θα παρουσιάσουμε την περίπτωση των ταξινομητών τύπου perceptron οι οποίοι αποτελούν επέκταση από το βασικό perceptron που παρουσιάστηκε σε προηγούμενες διαλέξεις, και χρησιμοποιεί περισσότερα από ένα επίπεδα.

Το πρόβλημα XOR

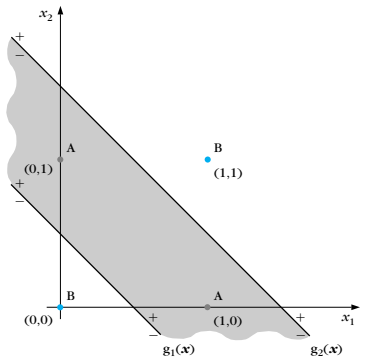
- Θεωρούμε το απλό πρόβλημα το οποίο στηρίζεται στην πολύ γνωστή boolean συνάρτηση αποκλειστικού ή (Exclusive OR - XOR).
- οι συναρτήσεις αυτού του τύπου μπορούν να θεωρηθούν ως προβλήματα ταξινόμησης.
- αυτό μπορεί να εφαρμοστεί καθώς η έξοδος τους είναι 0 ή 1 και το x ταξινομείται σε μια από τις δύο κλάσεις $A(1)$ ή $B(0)$
- η θέση των κλάσεων στον χώρο φαίνεται στο παρακάτω σχήμα.
- σε αντίθεση με τα προβλήματα AND και OR, το πρόβλημα XOR είναι μη γραμμικά διαχωρίσιμο (μπορείτε να δείτε γιατί;)



Σχήμα: Οι κλάσεις A, B για το πρόβλημα XOR

Το perceptron δύο επιπέδων

- Για να μπορέσουμε να διαχωρίσουμε τις κλάσεις A και B στο πρόβλημα XOR, μια πρώτη σκέψη θα ήταν να σχεδιάσουμε δύο ευθείες αντί για μία.
- μια τέτοια λύση παρουσιάζεται στο παρακάτω σχήμα με δύο γραμμές $g_1(x) = 0$, $g_2(x) = 0$ και τις περιοχές για τις οποίες έχουμε $g_1(x) \leq 0$ και $g_2(x) \leq 0$



Σχήμα: Γραμμές απόφασης για τον διαχωρισμό των κλάσεων A και B

Το perceptron δύο επιπέδων

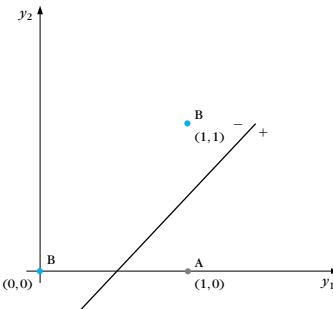
- για την επίλυση του προβλήματος, το αντιμετωπίζουμε σε δύο διαδοχικές φάσεις.
- στην πρώτη φάση, υπολογίζουμε τη θέση ενός διανύσματος χαρακτηριστικών x σε σχέση με κάθε μια από τις δύο γραμμές απόφασης
- στην δεύτερη φάση, συνδυάζουμε τα προηγούμενα αποτελέσματα και βρίσκουμε τη θέση του x σε σχέση με αμφότερες γραμμές (εντός ή εκτός της σκιασμένης περιοχής)

Α' φάση

- για την υλοποίηση των προηγούμενων γραμμών απόφασης (υπερεπίπεδα) $g_1(\cdot), g_2(\cdot)$ για την πρώτη φάση των υπολογισμών, χρησιμοποιούμε δύο perceptrons με εισόδους x_1, x_2 και τα κατάλληλα συναπτικά βάρη (όπως έχουμε δείξει σε προηγούμενη διάλεξη)
- οι έξοδοι είναι ίσες με $y_i = f(g_i(\mathbf{x}))$, $i = 1, 2$ με τη συνάρτηση ενεργοποίησης $f(\cdot)$ να είναι η βηματική συνάρτηση με επίπεδα τα 0 και 1
- αυτό που συμβαίνει είναι ο μετασχηματισμός του διανύσματος εισόδου \mathbf{x} σε ένα νέο διάνυσμα $y = [y_1 y_2]^T$.

Β' φάση

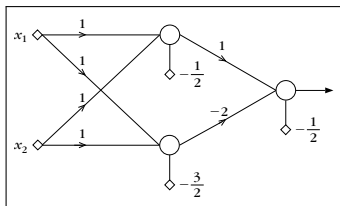
- η απόφαση στη δεύτερη φάση βασίζεται στα μετασχηματισμένα δεδομένα τα οποία θέλουμε να διαχωρίσουμε.
- αυτό είναι εύκολο όπως φαίνεται από το παρακάτω σχήμα γιατί το πρόβλημα έχει μετασχηματιστεί σε πρόβλημα γραμμικού διαχωρισμού, η λύση του οποίου είναι η τρίτη γραμμή $g(\mathbf{y})$ η οποία υλοποιείται μέσω ενός νέου νευρώνα.



Σχήμα: Γραμμή απόφασης που σχηματίζεται από τον νευρώνα του δεύτερου επιπέδου για το πρόβλημα XOR

Το perceptron δύο επιπέδων

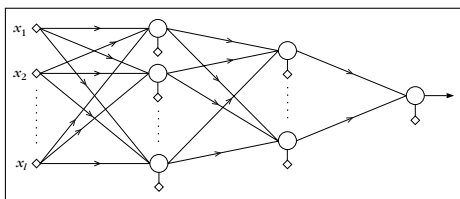
- το παρακάτω σχήμα δείχνει μια πιθανή υλοποίηση των προηγούμενων βημάτων
- η αρχιτεκτονική αυτή ονομάζεται **αρχιτεκτονική πολλών επιπέδων** ή **perceptron δύο επιπέδων** ή **εμπροσθοτροφοδοτούμενο perceptron** (feedforward network)
- οι δύο νευρώνες (κόμβοι) του πρώτου επιπέδου εκτελούν τους υπολογισμούς της πρώτης φάσης και αποτελούν το **κρυφό επίπεδο**
- ο μοναδικός νευρώνας του δεύτερου επιπέδου εκτελεί τους υπολογισμούς της τελικής φάσης και συνιστά το **επίπεδο εξόδου**
- στην αρχή έχουμε το **επίπεδο εισόδου** το οποίο περιλαμβάνει μη-επεξεργαστικούς κόμβους όπου εφαρμόζονται τα δεδομένα εισόδου (παράμετροι)



Σχήμα: Perceptron δύο επιπέδων που λύνει το πρόβλημα XOR

Γενίκευση πολυεπιπέδων δικτύων perceptron

- το perceptron δύο επιπέδων δεν μπορεί να διαχωρίσει κλάσεις που προκύπτουν από οποιαδήποτε ένωση πολυεδρικών περιοχών γιατί ο νευρώνας εξόδου μπορεί να υλοποιήσει μόνο ένα υπερεπίπεδο.
- την ίδια δυσκολία αντιμετώπιζε το απλό δίκτυο perceptron στην επίλυση του προβλήματος XOR και ξεπεράστηκε με την κατασκευή δύο γραμμών αντί για μία.
- Με παρόμοιο τρόπο μπορούμε να χρησιμοποιήσουμε ένα επιπλέον κρυφό επίπεδο δημιουργώντας μια αρχιτεκτονική **perceptron τριών επιπέδων**.
- μπορεί ναδειχτεί ότι μια τέτοια αρχιτεκτονική μπορεί να διαχωρίσει κλάσεις που προκύπτουν από οποιαδήποτε ένωση πολυεδρικών περιοχών.



Σχήμα: Η αρχιτεκτονική ενός perceptron πολλών επιπέδων με δύο κρυφά επίπεδα και έναν νευρώνα εξόδου

- οι προηγούμενες προσεγγίσεις προϋποθέτουν ότι ξέρουμε τις περιοχές που βρίσκονται τα δεδομένα και μπορούμε να υπολογίσουμε αναλυτικά τις εξισώσεις των αντίστοιχων υπερεπιπέδων
- στην πραγματικότητα όμως αυτό δεν ισχύει καθώς το μόνο που ξέρουμε είναι κάποια δεδομένα (παραδείγματα) εκπαίδευσης και τις αντίστοιχες ετικέτες κλάσης τους.
- όπως και στην περίπτωση του perceptron, θα πρέπει να καταφύγουμε σε **αλγόριθμους μάθησης** που θα εκτιμήσουν (μάθουν) τα συναπτικά βάρη από τα διαθέσιμα διανύσματα εκπαίδευσης

Ο αλγόριθμος οπισθοδιάδοσης

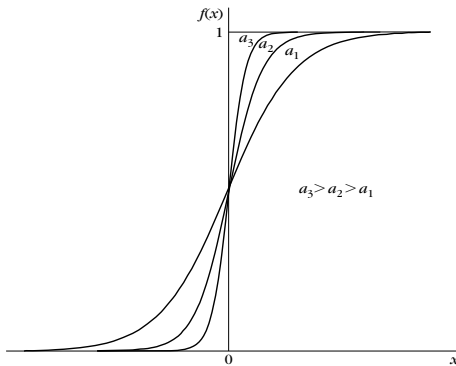
- για τον σχεδιασμό ενός perceptron πολλών επιπέδων, καθορίζουμε εκ των προτέρων την αρχιτεκτονική και υπολογίζουμε τις συναπτικές παραμέτρους ώστε να ελαχιστοποιήσουμε μια κατάλληλη συνάρτηση κόστους της εξόδου.
- ένα βασικό πρόβλημα που αντιμετωπίζει η τεχνική αυτή είναι η ασυνέχεια της βηματικής συνάρτησης (activation function) που καθιστά απαγορευτική την παραγωγή της συνάρτησης κόστους ως προς τις άγνωστες παραμέτρους (συναπτικά βάρη) με σκοπό την ελαχιστοποίηση της.
- για να ξεπεράσουμε το πρόβλημα αντί να χρησιμοποιήσουμε ως συνάρτηση ενεργοποίησης την βηματική, χρησιμοποιούμε προσεγγίσεις της από την οικογένεια των **σιγμοειδών συναρτήσεων**.
- ένας τυπικός αντιπρόσωπος είναι η **λογιστική συνάρτηση** (logistic function)

$$f(x) = \frac{1}{1 + \exp(-\alpha x)}$$

όπου α είναι παράμετρος κλίσης

Παραδείγματα σιγμοειδών συναρτήσεων

- το παρακάτω σχήμα δείχνει διαφορετικές εκδόσεις της σιγμοειδούς συνάρτησης για διάφορες τιμές της παραμέτρου α μαζί με την βηματική συνάρτηση για σύγκριση.



Σχήμα: Η λογιστική συνάρτηση. Όσο μεγαλύτερη η τιμή της παραμέτρου κλίσης α , τόσο καλύτερη είναι η προσέγγιση της συνάρτησης μοναδιαίου βήματος

Παραδείγματα σιγμοειδών συναρτήσεων

- μια άλλη παραλλαγή της λογιστικής είναι η **αντισυμμετρική** σε σχέση με την αρχική: $f(-x) = -f(x)$
- αυτή ορίζεται ως:

$$f(x) = \frac{2}{1 + \exp(-\alpha x)} - 1$$

- κυμαίνεται μεταξύ 1 και -1 και ανήκει στην οικογένεια των συναρτήσεων υπερβολικής εφαπτομένης:

$$f(x) = c \frac{1 - \exp(-\alpha x)}{1 + \exp(-\alpha x)} = c \tanh\left(\frac{\alpha x}{2}\right)$$

- όλες αυτές οι παραπάνω συναρτήσεις καλούνται **συναρτήσεις συμπίεσης** καθώς η έξοδος τους περιορίζεται σε συγκεκριμένο εύρος τιμών.

Ο αλγόριθμος οπισθοδιάδοσης

- **Αρχικοποίηση:** Αρχικοποιούμε όλα τα βάρη με μικρές τυχαίες τιμές χρησιμοποιώντας γεννήτρια ψευδοτυχαίων αριθμών
- **Υπολογισμοί προς τα εμπρός:** για κάθε ένα από τα διανύσματα χαρακτηριστικών υπολογίζουμε τους εμπρόσθιους υπολογισμούς και την συνάρτηση κόστους για την τρέχουσα εκτίμηση των βαρών
- **Υπολογισμοί προς τα πίσω:** υπολογίζουμε τις παραγώγους σε κάθε κόμβο και επίπεδο του δικτύου
- **Ενημέρωση των βαρών:** Για όλους τους κόμβους, υπολογίζεται επαναεκτίμηση των βαρών

$$\mathbf{w}_j^r(new) = \mathbf{w}_j^r(old) + \Delta \mathbf{w}_j^r$$

$$\Delta \mathbf{w}_j^r = -\mu \sum_{i=1}^N \delta_j^r(i) \mathbf{y}^{r-1}(i)$$

- η συνάρτηση κόστους που χρησιμοποιείται στις πιο πάνω σχέσεις επαναπροσδιορισμού των βαρών, είναι η **συνάρτηση ελαχίστων τετραγώνων**.

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

Σχολιασμός

- για τον τερματισμό των επαναλήψεων γίνεται έλεγχος της συνάρτησης κόστους. Όταν γίνει μικρότερη από ένα ορισμένο κατώφλι, τότε τερματίζονται οι επαναλήψεις.
- ένα δεύτερο κριτήριο τερματισμού της διαδικασίας εκπαίδευσης είναι όταν το διάνυσμα των μερικών παραγώγων της συνάρτησης μεταφοράς ως προς τα βάρη γίνει μικρό.
- η ταχύτητα σύγκλισης του αλγορίθμου εξαρτάται από την τιμή της **σταθεράς μάθησης μ** . Πρέπει να είναι μικρή για να έχουμε σύγκλιση, όχι πολύ μικρή γιατί θα έχουμε πολύ αργή σύγκλιση.
- το πρόβλημα ελαχιστοποίησης είναι **μη γραμμικό**. Αν έχουμε τοπικά ελάχιστα στην επιφάνεια κόστους, ο αλγόριθμος έχει την πιθανότητα να παγιδευτεί σε κάποιο από αυτά και να μην καταλήξει στο ολικό ελάχιστο. Σε αυτή την περίπτωση θα πρέπει να γίνει εκ νέου αρχικοποίηση των παραμέτρων και εκτέλεση του αλγορίθμου.

Ο προσαρμοστικός BP αλγόριθμος

- Ο αλγόριθμος αυτός προσαρμόζει την τιμή του ρυθμού εκπαίδευσης μ σε κάθε επανάληψη με βάση την τρέχουσα τιμή της συνάρτησης κόστους $J(t)$ αλλά και την προηγούμενη τιμή της $J(t - 1)$.
- Η λειτουργία της στηρίζεται στο παρακάτω:
 - αν $\frac{J(t)}{J(t - 1)} < 1$ τότε ο ρυθμός μ αυξάνεται κατά έναν παράγοντα r_i (τυπική τιμή $r_i = 1.05$)
 - αν $\frac{J(t)}{J(t - 1)} > c$ (τυπική τιμή του $c = 1.04$, τότε ο ρυθμός μ μειώνεται κατά έναν παράγοντα r_d (τυπική τιμή $r_d = 0.7$)

Η διαδικασία της αναγνώρισης

- μετά το τέλος της εκπαίδευσης, οι συνάψεις και τα κατώφλια παγιώνονται και το δίκτυο είναι έτοιμο να επιτελέσει ταξινόμηση που είναι πιο εύκολη διαδικασία.
- η ταξινόμηση γίνεται ως εξής: το άγνωστο διάνυσμα χαρακτηριστικών παρουσιάζεται στην είσοδο και ταξινομείται στην κλάση που υποδεικνύεται από την έξοδο του δικτύου.
- οι υπολογισμοί που εκτελούνται από τους νευρώνες είναι του τύπου πολλαπλασιασμός / πρόσθεση ακολουθούμενοι από μια μη γραμμικότητα.
- οι υπολογισμοί μπορούν να γίνουν εύκολα σε hardware (VLSI) αλλά και ακόμα και παράλληλα.

Πρακτικά ζητήματα εκπαίδευσης των δικτύων

Επιλογή του μεγέθους του δικτύου

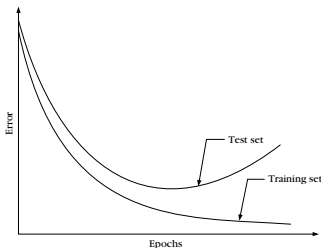
- δεν μπορούμε να επιλέξουμε μεγάλο δίκτυο καθώς θα αντιμετωπίσουμε πρόβλημα υπολογιστικής πολυπλοκότητας αλλά και της υπερ-προσαρμογής.
- θέλουμε να είναι αρκετά μεγάλο για να μάθει ποια είναι τα παρόμοια χαρακτηριστικά, αλλά και αρκετά μικρό ώστε να μην μάθει τις υποκείμενες διαφορές ανάμεσα στα δεδομένα της ίδιας κατηγορίας.

Το πρόβλημα της υπερ-εκπαίδευσης

- συνήθως διαιρούμε το σύνολο των παραδειγμάτων μας σε σύνολο εκπαίδευσης και επαλήθευσης ή δοκιμής και παρατηρούμε την μεταβολή του σφάλματος ως συνάρτηση των εποχών για κάθε σύνολο ξεχωριστά

Πρακτικά ζητήματα εκπαίδευσης δικτύων

- στο παρακάτω σχήμα παρουσιάζεται το πρόβλημα της υπερεκπαίδευσης.
- ενώ το σφάλμα εκπαίδευσης μειώνεται, από ένα σημείο και μετά το σφάλμα επαλήθευσης αυξάνεται.
- αυτό δείχνει ότι το δίκτυο έχει μάθει (προσαρμοστεί) να αναγνωρίζει τα παραδείγματα του συνόλου εκπαίδευσης και δεν μπορεί να γενικεύσει στα άγνωστα (παραδείγματα συνόλου επαλήθευσης).
- θα μπορούσαμε να σταματήσουμε την διαδικασία εκπαίδευσης στο σημείο που αρχίζουν και αποκλίνουν οι δύο καμπύλες αλλά αυτό προϋποθέτει την ύπαρξη ενός μεγάλου αριθμού συνόλων δεδομένων (κάτι που δεν έχουμε στην πράξη).



Σχήμα: Τάση του σφάλματος εξόδου ως προς τον αριθμό των εποχών που δείχνει την υπερεκπαίδευση του συνόλου εκπαίδευσης



Signal & Image Processing, Pattern Recognition Group (SIPPRE)

www.sippre-group.com