ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ακαδημαϊκό έτος: **2020-2021**
Εξεταστική: **Σεπτέμβριος**
Μάθημα: **Αναγνώριση Προτύπων**
Εξάμηνο: **5ο**

Ονοματεπώνυμο	**ΑΜ**
Μπαντάνα Δανάη Ιωάννα	Π17081
Ρούντου Άννα Φανή	Π17113
Σαγιέντ Ιωσήφ	Π15123

«Τεκμηρίωση και Αναλυτική Περιγραφή των Αλγορίθμων στα ερωτήματα i, ii και iii»

Περιεχόμενα

Εισαγωγή

Με τον όρο «Μηχανική Μάθηση» εννοούμε τη μελέτη αλγορίθμων που μπορούν να αυτό-βελτιώνονται μέσω της αξιοποίησης εμπειρίας και δεδομένων. Οι αλγόριθμοι αυτοί έχουν σκοπό τη δημιουργία μοντέλων, που θα περιγράφουν ένα σετ δεδομένων, γνωστά ως «δεδομένα εκπαίδευσης», προκειμένου να κάνουν προβλέψεις ή να παίρνουν αποφάσεις χωρίς την συμβολή του ανθρώπου.

Το ζητούμενο στην εργασία είναι ο ορισμός τέτοιων μοντέλων με σκοπό τον καλύτερο διαχωρισμό των δεδομένων και την ομαδοποίησή τους σε κλάσεις. Θέλουμε, επομένως, να δημιουργήσουμε μοντέλα ταξινόμησης που για κάποιες τιμές/βάρη θα διαχωρίζουν με τον καλύτερο δυνατό τρόπο το σύνολο των δεδομένων μας.

Για να το επιτύχουμε αυτό, αναγκαία προϋπόθεση αποτελεί ο ορισμός των βασικών συναρτήσεων βάσει των οποίων θα κατασκευάσουμε τα μοντέλα ταξινόμησης. Ένας ταξινομητής εκφράζεται μέσω μίας συνάρτησης πρόβλεψης, που περιγράφει όσο καλύτερα γίνεται το ζητούμενο, και μίας συνάρτησης κόστους/λάθους, που μας δίνει το πόσο απέχει η τιμή πρόβλεψης του ταξινομητή από την πραγματική τιμή. Στόχος, λοιπόν, για κάθε ερώτημα είναι η δημιουργία ενός μοντέλου μηχανικής μάθησης που με τις κατάλληλες τιμές/βάρη στη συνάρτηση πρόβλεψης να ελαχιστοποιείται το κόστος/λάθος για ένα σετ δεδομένων.

1. Ερώτημα i

Να υλοποιήσετε τον Αλγόριθμο Ελάχιστου Μέσου Τετραγωνικού Σφάλματος (**Least Mean Squares**), ώστε ο εκπαιδευμένος ταξινομητής να υλοποιεί την συνάρτηση διάκρισης της μορφής $\boldsymbol{g_k(\psi_\kappa(m))}$: $\mathbb{R}^3 \rightarrow \{\boldsymbol{H}, \boldsymbol{D}, \boldsymbol{A}\}$ για κάθε στοιχηματική εταιρεία. Να αναγνωρίσετε την στοιχηματική εταιρεία τα προγνωστικά της οποίας οδηγούν σε μεγαλύτερη ακρίβεια ταξινόμησης.

Αναλυτική Περιγραφή Αλγόριθμου του «Least Mean Squares»

Για το ερώτημα αυτό, θέλουμε να κατασκευάσουμε ένα μοντέλο/ταξινομητή που θα διακρίνει το σύνολο των αποδόσεων κάθε στοιχηματικής εταιρίας σε 3 υπερεπίπεδα («Νίκη Εντός», «Ισοπαλία», «Νίκη Εκτός») με τη χρήση του «**Least Mean Squares**» αλγορίθμου. Έπειτα, θα συγκριθεί η αποδοτικότητα του ταξινομητή βάση των προβλέψεων του για κάθε στοιχηματική. Η συνάρτηση πρόβλεψης που θα χρησιμοποιηθεί για κάθε ένα υπερεπίπεδο ορίζεται ως:

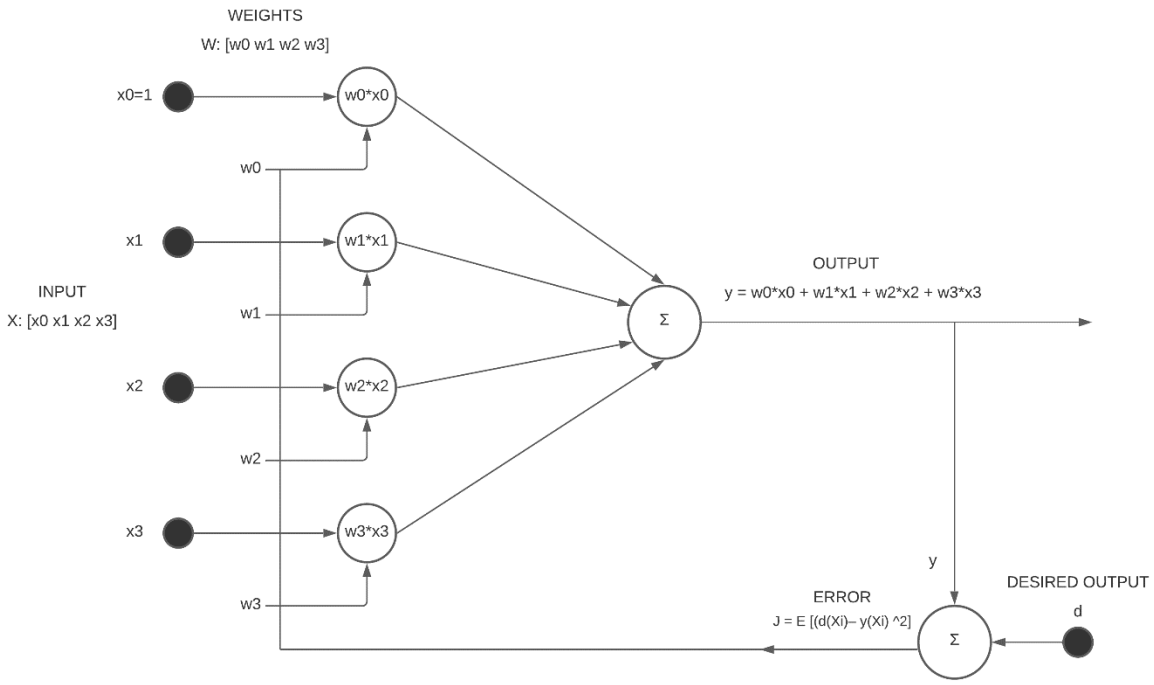
$y = w0*x0+w1*x1+w2*x2+w3*x3$, όπου

- 'w_i': τα βάρη που πρέπει να υπολογιστούν για την ελαχιστοποίηση του σφάλματος του ταξινομητή
- 'x_i': οι αποδόσεις κάθε στοιχηματικής. Το 'x0'=1 αποτελεί ένα σταθερό όρο και χρησιμοποιείται για να συμπεριλαμβάνεται πάντα η τιμή 'w0' που αποτελεί το κατώφλι του ταξινομητή. Στο 'x1' αποδίδεται η απόδοση για τη περίπτωση της «Νίκης Εντός», στο 'x2' για τη περίπτωση της «Ισοπαλίας» και στο 'x3' για τη περίπτωση της «Νίκης Εκτός».

Επομένως, για κάθε υπερεπίπεδο θα πρέπει να υπολογισθεί το διάνυσμα βαρών που ελαχιστοποιεί το σφάλμα του ταξινομητή. Συγκεκριμένα στο «Ερώτημα i» θέλουμε να ελαχιστοποιήσουμε το μέσο τετραγωνικό σφάλμα (**Least Mean Squares**) ανάμεσα στην επιθυμητή και τη πραγματική έξοδο του ταξινομητή. Άρα η συνάρτηση κόστους ορίζεται ως:

$J = E [(d(x_i) – y(x_i)) ^2]$, όπου

- '$d(x_i)$': η επιθυμητή έξοδος του ταξινομητή, δηλαδή το πραγματικό αποτέλεσμα του αγώνα i.
- '$y(x_i)$': η πραγματική έξοδος του ταξινομητή, δηλαδή η πρόβλεψη του ταξινομητή για τον αγώνα i.



Επίσης έχουμε:

- ‘πίνακα Χ διαστάσεων m*(n+1)’: ο οποίος σε κάθε γραμμή περιέχει ένα διαφορετικό σετ χαρακτηριστικών ‘x_i’, ενώ σε κάθε στήλη διαφορετική τιμή για τα xi. Το x0 έχει παντα την τιμή 1 αφού χρησιμοποιείται για τον υπολογισμό του σταθερού όρου.
- ‘διάνυσμα d’: θα είναι το διάνυσμα των πραγματικών αποτελεσμάτων για κάθε εναν απο τους αγώνες. Θα χρησιμοποιήσουμε την μέθοδο «one vs all» κατα την οποία θα θέσουμε 1 στην τιμή του αποτελέσματος του οποίου ψάχνουμε και 0 στα υπόλοιπα. Δηλαδή, εάν επιθυμούμε να βρούμε τη συνάρτηση που χωρίζει τις ‘Νίκες Εντός’ από τις άλλες δύο κλάσεις, θα δώσουμε την τιμή +1 στους αγώνες οπού το αποτέλεσμα ηταν ‘Νίκη Εντός’ και 0 σε οποιαδήποτε άλλη περίπτωση.
- ‘διάνυσμα W μεγέθους n=4’: θα περιγράφει τα βάρη του εκάστοτε υπολογισμού. Κατα την αρχικοποίηση του θα έχει μηδενικές τιμές για κάθε ένα από τα τέσσερα στοιχεία του.

Όπως προαναφέρθηκε, η εκάστοτε προσαρμογή των τιμών των βαρών, ώστε να εκτιμηθεί το βέλτιστο διάνυσμα βαρών, επιτυγχάνεται με την ελαχιστοποίηση της συνάρτησης κόστους. Η συνάρτηση κόστους που καλούμαστε να ελαχιστοποιήσουμε στην προκειμένη περίπτωση είναι η αναμενόμενη τιμή του μέσου τετραγωνικού σφάλματος. Αυτό, όμως, προϋποθέτει την γνώση των υποκείμενων κατανομών που σε εμάς είναι άγνωστες, αλλιώς θα ήταν ευκολότερο να χρησιμοποιήσουμε έναν Μπεϋζιανό ταξινομητή. Εφόσον μας λείπουν οι απαραίτητες στατιστικές πληροφορίες,

θα προσεγγίσουμε το πρόβλημα αυτό υπολογιστικά με τη μεθοδολογία «Robbins-Monro». Η μεθοδολογία αυτή στηρίζεται στον επαναληπτικό υπολογισμό και ενημέρωση του επόμενου βάρους με βάση το τρέχον βάρος, την συνάρτηση κόστους καθώς και μίας τιμής 'α'. Ο τύπος που προκύπτει είναι ο εξής:

w[i+1] = w[i] + a_i*J(xi,w[i]), όπου

- 'w[i]': τρέχον βάρος της συνάρτησης
- 'a_i': τιμή που βοηθάει στη σύγκλιση του αποτελέσματος. Προέρχεται από τον υπολογισμό μίας τιμής 'α' διά του εξεταστέου πλήθους του σετ δεδομένων την τρέχουσα στιγμή i. Δηλαδή, a_i = a/i.
- 'J': η συνάρτηση κόστους $J = (d(x_i)-y(x_i)) ^ 2$

Η παραπάνω διαδικασία εύρεσης των βέλτιστων βαρών για τη συνάρτηση πρόβλεψης θα επαναληφθεί για κάθε ένα από τα σενάρια «Νίκη Εντός», «Ισοπαλία» και «Νίκη Εκτός» για κάθε μία από τις τέσσερις στοιχηματικές εταιρίες «B365», «BW», «IW» και «LB». Έτσι, τελικά θα έχουμε 3*4 ζευγάρια βαρών. Τέλος, για κάθε ένα από τα παραπάνω ζευγάρια εφαρμόζεται η μέθοδος της 10-πλής διεπικύρωσης.

2. Ερώτημα ii

Να υλοποιήσετε τον Αλγόριθμο Ελάχιστου Τετραγωνικού Σφάλματος (**Least Squares**), ώστε ο εκπαιδευμένος ταξινομητής να υλοποιεί την συνάρτηση διάκρισης της μορφής $\boldsymbol{g_k(\psi_\kappa(m))}$: $\mathbb{R}^3 \rightarrow \{\boldsymbol{H}, \boldsymbol{D}, \boldsymbol{A}\}$ για κάθε στοιχηματική εταιρεία. Να αναγνωρίσετε την στοιχηματική εταιρεία τα προγνωστικά της οποίας οδηγούν σε μεγαλύτερη ακρίβεια ταξινόμησης.

Αναλυτική Περιγραφή Αλγόριθμου του «Least Squares»

Για το ερώτημα αυτό, θέλουμε να κατασκευάσουμε ένα μοντέλο/ταξινομητή που θα διακρίνει το σύνολο των αποδόσεων κάθε στοιχηματικής εταιρίας σε 3 υπερεπίπεδα («Νίκη Εντός», «Ισοπαλία», «Νίκη Εκτός») με τη χρήση του «**Least Squares**» αλγορίθμου. Έπειτα, θα συγκριθεί η αποδοτικότητα του ταξινομητή βάση των προβλέψεων του για κάθε στοιχηματική. Η συνάρτηση πρόβλεψης που θα χρησιμοποιηθεί για κάθε ένα υπερεπίπεδο ορίζεται ως:

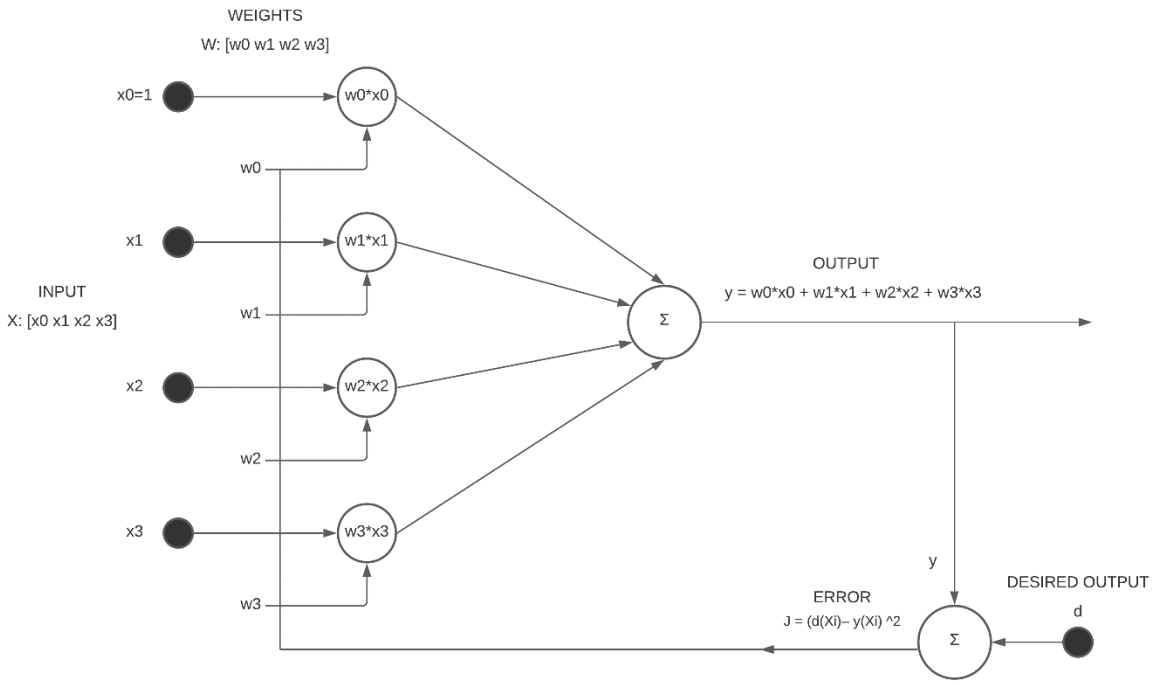
y = w0*x0+w1*x1+w2*x2+w3*x3, όπου

- 'w_i': τα βάρη που πρέπει να υπολογιστούν για την ελαχιστοποίηση του σφάλματος του ταξινομητή
- 'x_i': οι αποδόσεις κάθε στοιχηματικής. Το 'x0'=1 αποτελεί ένα σταθερό όρο και χρησιμοποιείται για να συμπεριλαμβάνεται πάντα η τιμή 'w0' που αποτελεί το κατώφλι του ταξινομητή. Στο 'x1' αποδίδεται η απόδοση για τη περίπτωση της «Νίκης Εντός», στο 'x2' για τη περίπτωση της «Ισοπαλίας» και στο 'x3' για τη περίπτωση της «Νίκης Εκτός».

Επομένως, για κάθε υπερεπίπεδο θα πρέπει να υπολογισθεί το διάνυσμα βαρών που ελαχιστοποιεί το σφάλμα του ταξινομητή. Συγκεκριμένα στο «Ερώτημα ii» θέλουμε να ελαχιστοποιήσουμε το τετραγωνικό σφάλμα (**Least Squares**) ανάμεσα στην επιθυμητή και τη πραγματική έξοδο του ταξινομητή. Άρα η συνάρτηση κόστους ορίζεται ως:

J = (d(x_i) – y(x_i)) ^2, όπου

- 'd(x_i)': η επιθυμητή έξοδος του ταξινομητή, δηλαδή το πραγματικό αποτέλεσμα του αγώνα i.
- 'y(x_i)': η πραγματική έξοδος του ταξινομητή, δηλαδή η πρόβλεψη του ταξινομητή για τον αγώνα i.



Επίσης έχουμε:

- ‘πίνακα X διαστάσεων m*(n+1)’: ο οποίος σε κάθε γραμμή περιέχει ένα διαφορετικό σετ χαρακτηριστικών ‘x_i’, ενώ σε κάθε στήλη διαφορετική τιμή για τα xi. Το x0 έχει παντα την τιμή 1 αφού χρησιμοποιείται για τον υπολογισμό του σταθερού όρου.
- ‘διάνυσμα d’: θα είναι το διάνυσμα των πραγματικών αποτελεσμάτων για κάθε εναν απο τους αγώνες. Θα χρησιμοποιήσουμε την μέθοδο «one vs all» κατα την οποία θα θέσουμε 1 στην τιμή του αποτελέσματος του οποίου ψάχνουμε και -1 στα υπόλοιπα. Δηλαδή, εάν επιθυμούμε να βρούμε τη συνάρτηση που χωρίζει τις ‘Νίκες Εντός’ από τις άλλες δύο κλάσεις, θα δώσουμε την τιμή +1 στους αγώνες οπού το αποτέλεσμα ηταν ‘Νίκη Εντός’ και -1 σε οποιαδήποτε άλλη περίπτωση.
- ‘διάνυσμα W μεγέθους n=4’: θα περιγράφει τα βάρη του εκάστοτε υπολογισμού. Κατα την αρχικοποίηση του θα έχει μηδενικές τιμές για κάθε ένα από τα τέσσερα στοιχεία του.

Όπως προαναφέρθηκε, η προσαρμογή των τιμών των βαρών κάθε φορά, ώστε να εκτιμηθεί το βέλτιστο διάνυσμα βαρών, επιτυγχάνεται με την ελαχιστοποίηση της συνάρτησης κόστους. Η συνάρτηση κόστους που καλούμαστε να ελαχιστοποιήσουμε στην προκειμένη περίπτωση είναι μία τετραγωνική συνάρτηση. Λόγω της ιδιαιτερότητας αυτής της συνάρτησης, δεν θα χρησιμοποιήσουμε τον αλγόριθμο «Gradient Descent», αλλά θα λύσουμε αναλυτικά το σύστημα των εξισώσεων για κάθε σετ δεδομένων. Έτσι, με τον αλγόριθμο «**Least Squares**» προσεγγίζουμε παλινδρομικά

τη λύση του συστήματος. Τα προσαρμοσμένα βάρη που θα έχουμε τελικά θα αποτελούν και τα βέλτιστα, καθώς η τετραγωνική συνάρτηση αποτυπώνεται γραφικά ως μία παραβολή, δηλαδή είναι μία κυρτή συνάρτηση με ένα ολικό ελάχιστο.

Ο τύπος υπολογισμού των βαρών που προκύπτει είναι ο εξής:

W = ((X.T * X)^-1) * (X.T * d), όπου

- '.T': η διαδικασία αναστροφής του πίνακα.
- 'X': το σετ δεδομένων
- 'd': οι επιθυμητές τιμές του ταξινομητή, δηλαδή τα πραγματικά αποτελέσματα των αγώνων.

Η παραπάνω διαδικασία εύρεσης των βέλτιστων βαρών για τη συνάρτηση πρόβλεψης θα επαναληφθεί για κάθε ένα από τα σενάρια «Νίκη Εντός», «Ισοπαλία» και «Νίκη Εκτός» για κάθε μία από τις τέσσερις στοιχηματικές εταιρίες «B365», «BW», «IW» και «LB». Έτσι, τελικά θα έχουμε 3*4 ζευγάρια βαρών. Τέλος, για κάθε ένα από τα παραπάνω ζευγάρια εφαρμόζεται η μέθοδος της 10-πλής διεπικύρωσης.

3. Ερώτημα iii

Να υλοποιήσετε ένα πολυστρωματικό νευρωνικό δίκτυο, ώστε ο εκπαιδευμένος ταξινομητής να υλοποιεί μια συνάρτηση διάκρισης της μορφής $\boldsymbol{g(\Phi(m))}$: $\mathbb{R}^{\mathbf{28}} \rightarrow \{\boldsymbol{H}, \boldsymbol{D}, \boldsymbol{A}\}$, όπου το $\boldsymbol{\Phi(m)} \in \mathbb{R}^{\mathbf{28}}$ αντιστοιχεί στο πλήρες διάνυσμα χαρακτηριστικών του κάθε αγώνα που δίνεται από την σχέση:

$$\boldsymbol{\Phi(m)} = [\boldsymbol{\varphi(h)}, \boldsymbol{\varphi(\alpha)}, \boldsymbol{\psi_{B365}(m)}, \boldsymbol{\psi_{BW}(m)}, \boldsymbol{\psi_{IW}(m)}, \boldsymbol{\psi_{LW}(m)}]$$

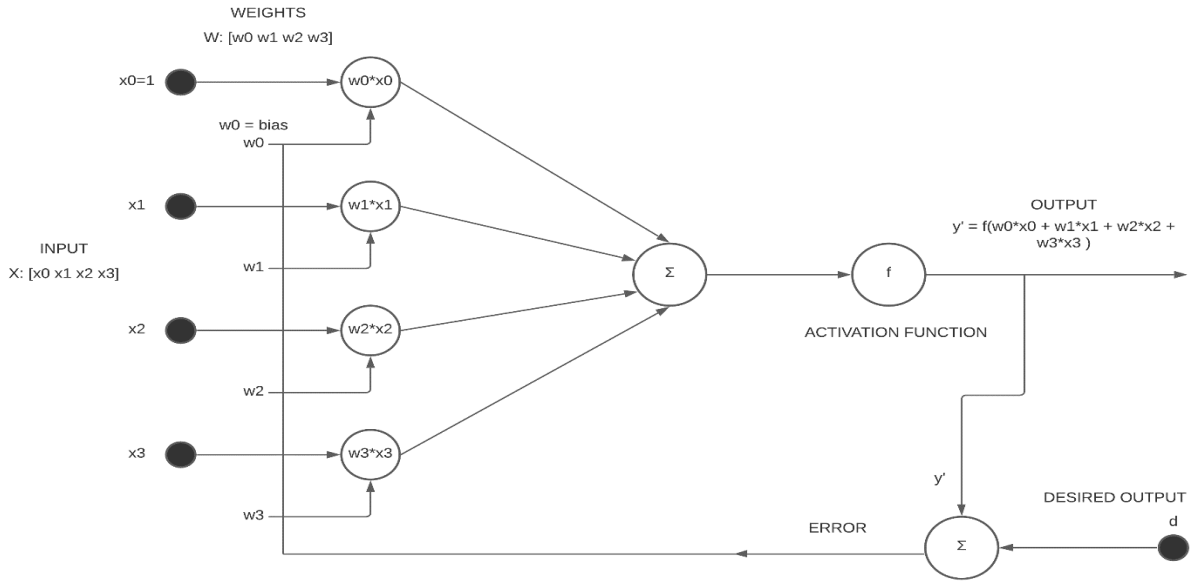
Αναλυτική Περιγραφή ενός Πολυστρωματικού Νευρωνικού Δικτύου

Για το ερώτημα αυτό, θέλουμε να κατασκευάσουμε ένα πολυστρωματικό νευρωνικό δίκτυο που θα διακρίνει το σύνολο των πλήρη διανυσμάτων χαρακτηριστικών του κάθε αγώνα. Για την καλύτερη κατανόηση της διαδικασίας της κατασκευής ενός πολυστρωματικού δικτύου, χρήσιμη είναι η σύντομη αναφορά στα δομικά και λειτουργικά χαρακτηριστικά του.

Ο 'Perceptron' είναι ένας ταξινομητής που παίρνει ως είσοδο ένα διάνυσμα 'X' με πραγματικές τιμές και το απεικονίζει σε μία τιμή εξόδου. Κάθε διάνυσμα εισόδου πολλαπλασιάζεται με ένα διάνυσμα βαρών 'W', και έπειτα προστίθεται το bias term, εξάγοντας έτσι ένα βεβαρημένο άθροισμα για κάθε είσοδο 'X_i'. Στη συνέχεια, το άθροισμα αυτό μεταφέρεται σε μία μη γραμμική συνάρτηση ενεργοποίησης («Activation Function»), υπεύθυνη για τον περιορισμό αυτής της τιμής σε ένα ορισμένο εύρος, δημιουργώντας την τελική τιμή εξόδου 'y' στον νευρώνα. Η συνάρτηση ενεργοποίησης που θα χρησιμοποιήσουμε για το ερώτημα iii θα είναι η σιγμοειδείς που θα προσαρμόζει το άθροισμα στο εύρος τιμών (0,1). Βάσει του αποτελέσματος y', τελικά ο νευρώνας ταξινομεί το εκάστοτε σετ χαρακτηριστικών 'X_i' σε μία κλάση.

Εκπαιδεύουμε ένα νευρώνα ('Perceptron') προσαρμόζοντας τα βάρη του μετά τον υπολογισμό κάθε εξόδου του 'y''. Για αυτό απαιτείται ο ορισμός μίας συνάρτηση κόστους που θα αποτιμά την απόσταση μεταξύ της εξόδου του ταξινομητή και της πραγματικής τιμής.

Perceptron



Το «**Multi-layer Perceptron (MLP)»** χαρακτηρίζεται ως το τεχνητό νευρωνικό δίκτυο που περιέχει πολλαπλά στρώματα από πολλούς ταξινομητές 'Perceptron' (νευρώνες). Αποτελεί έναν 'supervised learning' αλγόριθμο, ο οποίος προσαρμόζει τα βάρη μέσω μίας διαδικασίας που ονομάζεται «Backpropagation». Διαθέτει ένα στρώμα εισόδου, ένα στρώμα εξόδου και μεταξύ αυτών μπορεί να υπάρχουν ένα ή περισσότερα επίπεδα που ονομάζονται κρυμμένα επίπεδα.

- 'Input Layer': Έχει τόσους νευρώνες όσα είναι τα χαρακτηριστικά (features) της εισόδου 'X_i', δηλαδή {xi|x1,x2,…,xm}.
- 'Hidder Layers': Τα στρώματα αυτά δεν μπορούμε να τα δούμε. Κάθε νευρώνας σε κάθε κρυμμένο επίπεδο μετατρέπει τις τιμές από το προηγούμενο στρώμα σε μία σταθμισμένη άθροιση και στη συνέχεια το αποτέλεσμα τροποποιείται από μία συνάρτηση ενεργοποίησης(Activation Function 'f').
- 'Output Layer': Λαμβάνει τις τιμές από το τελευταίο κρυφό στρώμα και τις μετατρέπει σε τιμές εξόδου.

Ένα διάνυσμα εισόδου 'X_i' περνά μέσα από το πρώτο στρώμα, του οποίου οι τιμές εξόδου συνδέονται με τις τιμές εισόδου του επόμενου στρώματος και ούτω καθεξής, μέχρι το δίκτυο να δώσει, ως αποτέλεσμα, τις εξόδους του τελευταίου στρώματος. Σε κάθε επίπεδο κάθε είσοδος πολλαπλασιάζεται με το αντίστοιχο βάρος που ενώνει αυτό και τον επόμενο νευρώνα. Ο επόμενος νευρώνας τότε προσθέτει τις τιμές από κάθε νευρώνα του πρηγούμενου στρώματος καθώς και το bias term που του αντιστοιχεί. Τότε, το τελικό αποτέλεσμα περνάει μέσα από την συνάρτηση ενεργοποίησης όπου και

αποτελεί την είσοδο για τον υπολογισμό των αθροισμάτων του επόμενου στρώματος. Η διαδικασία αυτη προχωράει μέχρι να υπολογιστούν οι νευρώνες και του τελευταίου επιπέδου.

Υπεύθυνος για τη μάθηση και την εκτίμηση των βέλτιστων βαρών του «πολυστρωματικού Perceptron-MLP» είναι ένας αλγόριθμος που ονομάζεται «Backpropagation».

Η ιδέα του αλγορίθμου «Backpropagation» βασίζεται στον υπολογισμό του σφάλματος και στον εκ νέου υπολογισμό της συστοιχίας των βαρών 'W' του τελευταίου στρώματος νευρώνων ώστε να προχωρήσει με αυτόν τον τρόπο προς τα προηγούμενα στρώματα, από πίσω προς τα εμπρός. Δηλαδή, ενημερώνει όλα τα βάρη 'W' σε κάθε στρώμα, από το τελευταίο μέχρι να φτάσει στο επίπεδο εισόδου του δικτύου. Με άλλα λόγια, υπολογίζουμε το σφάλμα μεταξύ του τι προέβλεψε το δίκτυο και της πραγματικής τιμής και τελικά υπολογίζουμε εκ νέου όλες τις τιμές βαρών, από το τελευταίο στρώμα στο πρώτο, με σκοπό πάντα να μειώσουμε το σφάλμα του νευρωνικού δικτύου.

Πολυστρωματικό Νευρωνικό Δίκτυο