Αναγνώριση Προτύπων

Εργαστηριακή Άσκηση 4

Ο ΑΛΓΟΡΙΘΜΟΣ PERCPTRON

Δασούλας Ιωάννης – 1053711

**1. ∆ιάβασµα προτύπων από ιατρικές εργαστηριακές εξετάσεις**

Αρχικά διαβάζονται οι τιμές του αρχείου liverdisorder.data μέσω της συνάρτησης [x,c] = ReadLiver(total), όπου total ο αριθμός των δειγμάτων (345). Το αρχείο περιέχει 6 μετρήσεις και μία κατηγορία, 1 ή 2. Αν η κατηγορία είναι ένα, ο εξεταζόμενος θεωρείται υγιής, ενώ αν είναι 2 θεωρείται ασθενής. Με τη συνάρτηση αποθηκεύονται στον πίνακα x οι 6 μετρήσεις σε έναν πίνακα 6x345 και στον c όλες οι κατηγορίες σε ένα διάνυσμα 1x345.

Έπειτα, με μία επανάληψη, χωρίζονται σε 2 πίνακες οι μετρήσεις των υγιών εξεταζόμενων (πίνακα H) και των ασθενών (πίνακας U).

clc;

clear all;

total = 345;

[x,c] = ReadLiver(total);

H\_index = 1; %H -> Healty

U\_index = 1; %U -> Unhealty

for i=1:total

if(c(i) == 1)

H(:,H\_index) = x(:,i);

H\_index = H\_index + 1;

elseif (c(i) == 2)

U(:,U\_index) = x(:,i);

U\_index = U\_index + 1;

end

end

**2. Υπολογίστε τους συντελεστές των γραµµικών συναρτήσεων απόφασης µε τον αλγόριθµο Perceptron**

Για την τύπωση των συντελεστών χρησιμοποιήθηκε η συνάρτηση Perceptor. Αρχικά, έπρεπε να γίνουν κάποιες αλλαγές διότι δεν λειτουργούσε εκ των προτέρων. Πρώτον, δεν υπάρχει στις διαθέσιμες συναρτήσεις, η συνάρτηση rows, οπότε οι ο αριθμός των σειρών των πινάκων x1, x2 υπολογίστηκε με τον παρακάτω τρόπο.

[x1\_rows, x1\_col] = size(x1);

[x2\_rows, x2\_col] = size(x2);

Έπειτα, έπρεπε να αντικατασταθεί η συνάρτηση printf() και αυτό έγινε με την συνάρτηση disp(sprint()), ενώ σβήστηκε και η εντολή fflush(stdout) που μάλλον αφορούσε υλοποίηση σε octave.

disp(sprintf( '%8d', sum(Rc) )) ;

Τέλος, προστέθηκε το διάνυσμα των συντελεστών Weights στην δήλωση της συνάρτησης για να εμφανιστούν τα βάρη. Η συνάρτηση πλέον όπως διαμορφώθηκε:

function [Rc,Rep,Weights] = Perceptron(x1,x2,Lr,MaxRep)

%#

%# [Rc,Rep,Weights] = Perceptron(x,c,Lr)

%#

%# Input

%# x1: Pattern Vectors for the first class

%# x2: Pattern Vectors for the second class

%# Lr: Learning rate

%# MaxRep: Maximum repetitions

%# Output

%# Rc: Correct classification rate using the C-method

%# Rep: Pattern vectors on each class

%#

NumOfP1 = columns(x1) ;

NumOfP2 = columns(x2) ;

[x1\_rows, x1\_col] = size(x1);

[x2\_rows, x2\_col] = size(x2);

Weights = 2 \* rand(x1\_rows+1,1) - 1 ;

Rep = [NumOfP1,NumOfP2] ;

TotPat = sum(Rep) ;

Rc = zeros(2,1) ;

if ( x1\_rows ~= x2\_rows )

printf( 'Error in vectors x1, x2\n' ) ;

end

%#

%# C-Error

%#

for j = 1:MaxRep

if ( rand() > 0.5 )

k = floor(rand() \* NumOfP1 + 1) ;

Pat = [x1(:,k);1] ;

Score = Weights' \* Pat ;

if ( Score < 0 )

Weights = Weights + Lr \* Pat ;

end

else

k = floor(rand() \* NumOfP2 + 1) ;

Pat = [x2(:,k);1] ;

Score = Weights' \* Pat ;

if ( Score >= 0 )

Weights = Weights - Lr \* Pat ;

end

end

Rc = zeros(2,1) ;

for i=1:NumOfP1

if ( Weights' \* [x1(:,i);1] >= 0 )

Rc(1) = Rc(1) + 1 ;

end

end

for i=1:NumOfP2

if ( Weights' \* [x2(:,i);1] < 0 )

Rc(2) = Rc(2) + 1 ;

end

end

disp(sprintf( '%8d', sum(Rc) )) ;

%drawnow('update');

if ( sum(Rc) == TotPat )

break ;

end

end

disp(sprintf( '\n' )) ;

Ο κώδικας για την εμφάνιση των συντελεστών:

[Rc,Rep,Weights] = Perceptron(H,U,1,1000);

Weights

Συντελεστές για αρχικό συντελεστή εκπαίδευσης = 1 και μέγιστο αριθμό επαναλήψεων = 1000:

-91.3350

58.5557

282.9391

-474.4684

-211.1923

56.5365

-3.4393

**3. Υπολογίστε το σφάλµα του συστήµατος ταξινόµησης**

Το ποσοστό επιτυχούς ταξινόμησης ισούται με το άθροισμα των εξετάσεων που ταξινομήθηκαν σωστά σε κάθε κατηγορία προς το πλήθος τους. Άρα, το σφάλμα υπολογίζεται έτσι:

error = 1 - ((Rc(1) + Rc(2))/total)

Για αρχικό συντελεστή εκπαίδευσης = 1 και μέγιστο αριθμό επαναλήψεων = 1000, το σφάλμα είναι περίπου 50%, αν και τα αποτελέσματα διαφέρουν λόγω του ότι ο αλγόριθμος είναι στοχαστικός.

Για μέγιστο αριθμό επαναλήψεων = 10.000, το σφάλμα μειώνεται στο 40% περίπου, ενώ για μέγιστο αριθμό επαναλήψεων = 100.000 το σφάλμα μειώνεται στο 30% περίπου.

**4. Μελετήστε την επίδραση που έχει ο συντελεστής εκπαίδευσης στον αλγόριθµο Perceptron**

Για την μέτρηση της αξιοπιστίας, δημιουργήθηκε ένας πίνακας που για δείχνει το ποσοστό αξιοπιστίας για κάθε συντελεστή από 0.05 εώς 1 με βήμα 0.05 και στο τέλος γίνεται το αντίστοιχο plot. Στο διάνυσμα position αποθηκεύονται όλες οι τιμές των συντελεστών και στο correct\_table το ποσοστό αξιοπιστίας για τον συντελεστή αυτό και δεδομένο μέγιστο αριθμό επαναλήψεων.

max\_rep = 10000;

index = 1;

for i=0.05:0.05:1

position(index) = i;

[Rc,Rep,Weights] = Perceptron(H,U,i,max\_rep);

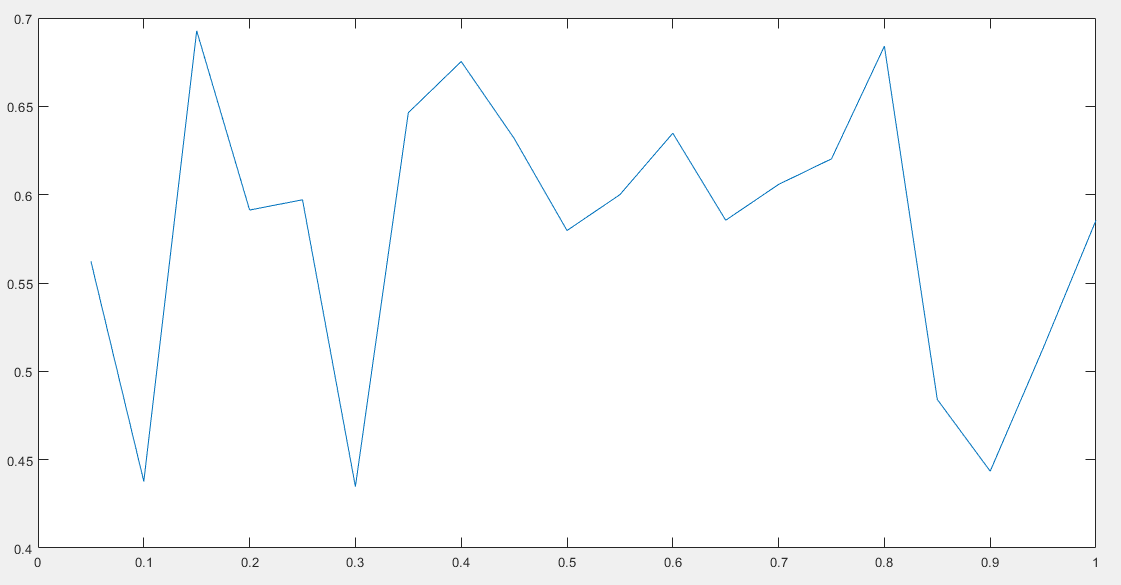
correct\_table(index) =((Rc(1) + Rc(2))/total);

index = index + 1;

end

plot(position, correct\_table)

Για μέγιστο αριθμό επαναλήψεων = 1.000:



Για μέγιστο αριθμό επαναλήψεων = 10.000:

Εικόνα που περιέχει βάρκα, νερό, άτομα, πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

O αριθμός των επαναλήψεων συναρτήσει του συντελεστή εκπαίδευσης δεν μπορούσε να υπολογιστεί καθώς ήταν πάντα ίσος με τον μέγιστο αριθμό επαναλήψεων. Αυτό σημαίνει ότι για αυτά τα δεδομένα δεν συνέκλινε ο αλγόριθμος καθώς το άθροισμα των σωστά ταξινομημένων παραδειγμάτων κυμαινόταν πάντα κοντά στο 200, ενώ για να τερματίσει ο αλγόριθμος θα έπρεπε να φτάσει το 345 ακόμα και για πολύ μικρούς συντελεστές διότι τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα.

**5. Τι παρατηρείτε από τα αποτελέσµατα των µετρήσεων**

Αρχικά, όπως επισημάνθηκε και προηγουμένως, η αξιοπιστία αυξάνεται όλο και περισσότερο με την αύξηση του μέγιστου αριθμού επαναλήψεων, όπως και αναμενόταν. Επιπλέον, θα έπρεπε με την αύξηση του συντελεστή εκπαίδευσης, να αυξάνεται ξανά η αξιοπιστία, κάτι που όπως φαίνεται από τα γραφήματα δεν συμβαίνει. Αυτό προκύπτει λόγω του ότι τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα, όπως εξηγήθηκε και προηγουμένως. Ακόμα και για πολύ μικρούς συντελεστές, δεν συνέκλινε ο αλγόριθμος και υπήρχε πάντοτε ένα ποσοστό σφάλματος, άρα με αρκετά μεγάλη βεβαιότητα προκύπτει το συμπέρασμα ότι δεν είναι γραμμικώς διαχωρίσιμα.