

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

### ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

#### ΜΟΝΤΕΛΟ PERCEPTRON

**Σκοπός της άσκησης:** Η εκτίμηση της επίδοσης ενός ταξινομητή τύπου Perceptron σε άγνωστα δεδομένα. Θα γίνει χρήση της μεθόδου διασταύρωσης (Cross-Validation) και τα κριτήρια επίδοσης:

1. Ακρίβεια (accuracy)
2. Ευστοχία (precision)
3. Ανάκληση (recall)
4. F-Measure
5. Ευαισθησία (Sensitivity)
6. Προσδιοριστικότητα (Specificity)

#### Βήματα υλοποίησης:

1. Χρησιμοποιήστε το σύνολο δεδομένων IRIS από το προηγούμενο εργαστήριο, καθώς και τον κώδικα από το εργαστήριο αυτό. Θυμίζουμε ότι τα πρότυπα χωρίστηκαν σε δύο κλάσεις ως εξής:
  - Κλάση 0 (στόχος  $t=0$ ): αποτελείται από τα πρότυπα των κατηγοριών "Iris-setosa" + "Iris-virginica",
  - Κλάση 1 (στόχος  $t=1$ ): αποτελείται από τα πρότυπα της κατηγορίας "Iris-versicolor".
2. Θα εφαρμοστεί η μέθοδος `train_test_split()` για  $K=9$  folds.
3. Πριν από το Cross-Validation loop ζητήστε από το χρήστη να δώσει μέγιστο πλήθος εποχών και βήμα εκπαίδευσης  $\beta$ .
4. Στο Cross-Validation loop θα πρέπει να κάνετε τα εξής:

Για κάθε *fold*

- Έχετε ήδη δημιουργήσει τους επαυξημένους πίνακες προτύπων `xtrain1` και `xtest1` προσθέτοντας σε κάθε πρότυπο τον αριθμό 1. Ομοίως έχετε δημιουργήσει τα διάνυσμα στόχων `ttrain` και `ttest`. Αποφασίστε αν οι τιμές των στόχων θα είναι 0/1, είτε -1/1 και γράψτε τον υπόλοιπο κώδικα ανάλογα με την απόφαση που πήρατε. Επίσης έχετε βρει το πλήθος των προτύπων στο train set ( $P_{train}$ ) και στο test set ( $P_{test}$ ).
- Υπολογίστε το βέλτιστο διάνυσμα βαρών  $\tilde{w}$  εκπαιδεύοντας το μοντέλο Perceptron με τα πρότυπα και τους στόχους εκπαίδευσης: `xtrain1`, `ttrain`. Θα χρησιμοποιήσετε τη function `perceptron()` την οποία θα πρέπει να γράψετε εσείς.

```
def perceptron( x, t, MAXEPOCHS, beta ):  
    # Είσοδος x : πίνακας με τα επαυξημένα πρότυπα  
    # Είσοδος t : διάνυσμα με τους πραγματικούς στόχους (0/1 ή -1/1)  
    # Είσοδοι MAXEPOCHS : μέγιστο πλήθος εποχών  
    # Είσοδοι beta : βήμα εκπαίδευσης  
    # Επιστρέφει w : τελικό επαυξημένο διάνυσμα βαρών.
```

- Ο αλγόριθμος Perceptron λειτουργεί ως εξής:

Δώσε τυχαία αρχική τιμή στο διάνυσμα βαρών

Για κάθε εποχή εκπαίδευσης

Για κάθε πρότυπο  $p$

Υπολόγισε την διέγερση του νευρώνα  $u = \tilde{\mathbf{x}}[p, :] \tilde{\mathbf{w}}$

Υπολόγισε την έξοδο του νευρώνα

- Εφόσον οι στόχοι έχουν τιμές 0/1 τότε

$$y = \begin{cases} 0, & \text{αν } u < 0 \\ 1, & \text{αν } u \geq 0 \end{cases}$$

- Εφόσον οι στόχοι έχουν τιμές -1/1 τότε

$$y = \begin{cases} -1, & \text{αν } u < 0 \\ 1, & \text{αν } u \geq 0 \end{cases}$$

Διόρθωσε το διάνυσμα βαρών:

- Αν  $t[p] = y$  τότε μην κάνεις τίποτε
- Αν  $t[p] \neq y$  τότε  $\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} + \beta (t[p] - y) \tilde{\mathbf{x}}[p, :]$

end

Τερμάτισε εφόσον

- δεν έγινε καμία διόρθωση των βαρών για κανένα πρότυπο
- είτε φτάσαμε στο μέγιστο αριθμό εποχών

end

Μετά το τέλος του αλγορίθμου τυπώστε στην οθόνη το πλήθος των εποχών που εκτελέστηκαν.

- Υπολογίστε την διέγερση του νευρώνα για όλα τα πρότυπα του test set:

$$u_{test}[i] = \tilde{\mathbf{x}}_{test}[i, :] \tilde{\mathbf{w}}$$

- Υπολογίστε την εκτίμηση που κάνει ο ταξινομητής για την κλάση στην οποία ανήκουν τα πρότυπα του test set:

- Εφόσον οι στόχοι έχουν τιμές 0/1 τότε

$$predict_{test}[i] = \begin{cases} 0, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- Εφόσον οι στόχοι έχουν τιμές -1/1 τότε

$$predict_{test}[i] = \begin{cases} -1, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- Καλέστε τη συνάρτηση `evaluate()` από το προηγούμενο εργαστήριο όσες φορές χρειάζεται έτσι ώστε για το συγκεκριμένο fold να υπολογίσετε το Accuracy, Precision, Recall, F-measure, Sensitivity και Specificity.
- Χρησιμοποιώντας κατάλληλο subplot σε grid 3x3 στο figure(1) τυπώστε το εξής γράφημα:
  - δείξτε με μπλε τελείες τους πραγματικούς στόχους  $t_{test}[i]$  για όλα τα πρότυπα του test set
  - δείξτε με κόκκινους κύκλους τους εκτιμώμενους στόχους  $predict_{test}[i]$  για όλα τα πρότυπα του test set

5. Μετά το τέλος του loop υπολογίστε και τυπώστε στην οθόνη τα εξής:
  1. τη μέση τιμή του Accuracy για όλα τα folds
  2. τη μέση τιμή του Precision για όλα τα folds
  3. τη μέση τιμή του Recall για όλα τα folds
  4. τη μέση τιμή του F-Measure για όλα τα folds
  5. τη μέση τιμή του Sensitivity για όλα τα folds
  6. τη μέση τιμή του Specificity για όλα τα folds
6. Τρέξτε πολλές φορές το πρόγραμμα χρησιμοποιώντας διαφορετικές τιμές MAXEPOCHS, beta και παρατηρήστε αν εμφανίζονται διαφορές στην επίδοση του αλγορίθμου.

**Σχόλια:**

- **Η εντολή break.** Η πρόωρη έξοδος από ένα loop στην Python, όπως και στις περισσότερες γλώσσες, γίνεται με την εντολή break.
- **Άνοιγμα figure για plot.** Μπορούμε να ανοίξουμε ένα συγκεκριμένο figure, πχ το figure 1, με την εντολή

```
import matplotlib.pyplot as plt
```

```
plt.figure(1)
```

Κατόπιν όλες οι εντολές plot, κλπ θα εκτελεστούν σε αυτό το figure. Αν αποφασίσουμε ότι θέλουμε τώρα να τυπώσουμε σε άλλο figure, πχ το figure 2, απλά καλούμε την εντολή

```
plt.figure(2)
```

και τώρα όλες οι εντολές plot θα εμφανίζονται στο figure 2. Μπορούμε ανά πάσα στιγμή να επιστρέψουμε στο figure 1 ξανακαλώντας

```
plt.figure(1)
```