

## ΕΠ08 Αναγνώριση Προτύπων – Μηχανική Μάθηση

### 1<sup>η</sup> Εργασία: Τα βασικά της μηχανικής μάθησης

Τύπος εργασίας: **Ατομική**

Ημερομηνία παράδοσης: **Τρίτη 31/03/2022, 23:55**

Τρόπος παράδοσης: **Αποκλειστικά μέσω του eclass**

Σύνολο βαθμών: **100** (10% του τελικού βαθμού του μαθήματος)

Σε αυτή, τη πρώτη, εργασία του μαθήματος Αναγνώρισης Προτύπων – Μηχανική Μάθηση καλείστε να απαντήσετε ερωτήματα που αφορούν στο βασικό μαθηματικό υπόβαθρο των μεθοδολογιών μηχανικής μάθησης. Η εργασία είναι **ατομική** και αποτελείται από τρία ερωτήματα. Συνιστάται ιδιαίτερα, να αφιερώσετε χρόνο ώστε να κατανοήσετε τον θεμελιώδη λογισμό πίσω από τα ερωτήματα της εργασίας και να αποφύγετε την αναζήτηση έτοιμων λύσεων στο διαδίκτυο. Αν ωστόσο συμβουλευτείτε ή/και χρησιμοποιήσετε οποιοδήποτε υλικό ή/και κώδικα που είναι διαθέσιμος στο διαδίκτυο, πρέπει να αναφέρεται σωστά τη πηγή ή/και το σύνδεσμο στην ιστοσελίδα που αντλήσατε πληροφορίες. Σε κάθε περίπτωση, η αντιγραφή τμήματος ή του συνόλου της εργασίας δεν είναι αποδεκτή και στη περίπτωση που διαπιστωθεί αντιγραφή θα μηδενιστούν στο μάθημα όλα τα εμπλεκόμενα μέρη.

Θα πρέπει να υποβάλετε **ένα μόνο αρχείο Notebook IPython (Jupyter notebook) μέσω του εργαλείου εργασίες του eclass**, ακολουθώντας την εξής σύβαση ονομασίας για το αρχείο σας: *Επώνυμο\_ΑριθμόςΜητρώου.ipynb*

Τόσο ο κώδικας Python όσο και οι απαντήσεις σας στις αναλυτικές/αριθμητικές ερωτήσεις πρέπει να είναι ενσωματωμένα στο ίδιο IPython notebook. Οι μαθηματικές πράξεις μπορούν να ενσωματωθούν στο IPython notebook είτε χρησιμοποιώντας LaTeX σημειογραφία είτε ως εικόνες (π.χ. φωτογραφία χειρόγραφου). Μπορείτε να χρησιμοποιήσετε κελιά επικεφαλίδας για να οργανώσετε περαιτέρω το έγγραφό σας.

**Σημαντικό:** Το IPython notebook που θα παραδώσετε θα πρέπει βεβαιωθείτε ότι ανοίγει και να εκτελείται στο google colab.

#### [Ερώτημα 1: Πράξεις με Διανύσματα και Πίνακες] (10 βαθμοί)

Η εξοικείωση με το λογισμό διανυσμάτων και πινάκων είναι απαραίτητη για την κατανόηση και την υλοποίηση αλγορίθμων μηχανικής μάθησης. Για το σκοπό αυτό, καλείστε να απαντήσετε στα παρακάτω ερωτήματα.

Χρησιμοποιώντας το πακέτο random της NumPy να δημιουργήσετε:

- i) Δύο τυχαίους πίνακες ακέραιων αριθμών  $\mathbf{X} \in \mathbb{Z}^{3 \times 4}$  και  $\mathbf{Y} \in \mathbb{Z}^{4 \times 3}$ .
- ii) Δύο τυχαία διανύσματα ακέραιων αριθμών  $\mathbf{a} \in \mathbb{Z}^4$  και  $\mathbf{b} \in \mathbb{Z}^4$

1.1 Να υπολογίσετε το εσωτερικό γινόμενο των διανυσμάτων  $\mathbf{a}$  και  $\mathbf{b}$ , δηλαδή  $\mathbf{a}^T \mathbf{b}$

1.2 Να υπολογίσετε το γινόμενο πίνακα-διανύσματος  $\mathbf{Xa} \in \mathbb{Z}^3$

1.3 Να υπολογίσετε το γινόμενο πινάκων  $\mathbf{XY} \in \mathbb{Z}^{3 \times 3}$

1.4 Να υπολογίσετε την Ευκλείδεια νόρμα (l2-norm) του διανύσματος  $\mathbf{a}$

Στις παραπάνω ερωτήσεις καλείστε να εκτελέσετε τους υπολογισμούς αναλυτικά βήμα προς βήμα με το χέρι, παραθέτοντας όλα τα ενδιάμεσα βήματα και να επιβεβαιώσετε τα αποτελέσματα με τη χρήση της NumPy.

**Σημαντικό:** Για να είναι συνεπή τα τυχαία αποτελέσματα, να ορίστε το seed στη γεννήτρια τυχαίων αριθμών ως τα τελευταία 4 ψηφία του αριθμού μητρώου σας

### [Ερώτημα 2: Υπολογισμός παραγώγων] (20 βαθμοί)

Ένα πολύ ευρύ σύνολο αλγορίθμων μηχανικής μάθησης απαιτούν την επίλυση κατάλληλων προβλημάτων βελτιστοποίησης στα οποία ο υπολογισμός παραγώγων συναρτήσεων πολλών μεταβλητών είναι κεντρικής σημασίας. Για την εξοικείωσή σας με τον υπολογισμό παραγώγων και την αναλυτική επίλυση προβλημάτων βελτιστοποίησης καλείστε να απαντήσετε στα παρακάτω ερωτήματα. Αυτό το ερώτημα αποτελεί καλή ευκαιρία για να ξεκινήσετε να χρησιμοποιείται το The Matrix Cookbook στη πράξη.

The Matrix Cookbook είναι διαθέσιμο στο σύνδεσμο:

<https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>

2.1 Έστω  $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}$ , μια συνάρτηση  $d$  μεταβλητών, με  $\mathbf{A}$  συμμετρικό πίνακα (δηλ.  $\mathbf{A} = \mathbf{A}^T$ ) διάστασης  $d \times d$ , και  $\mathbf{b} \in \mathbb{R}^d$ . Να υπολογίσετε αναλυτικά βήμα προς βήμα με το χέρι, την παράγωγο της  $f(\mathbf{x})$  ως προς το διάνυσμα  $\mathbf{x} \in \mathbb{R}^d$  παραθέτοντας όλα τα ενδιάμεσα βήματα.

Υπενθυμίζεται ότι η παράγωγος μιας συνάρτησης  $f: \mathbb{R}^d \rightarrow \mathbb{R}$ ,  $\nabla f(\mathbf{x})$  ως προς το διάνυσμα  $\mathbf{x} \in \mathbb{R}^d$  είναι:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix} \in \mathbb{R}^d$$

2.2 Έστω  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{X}$  πίνακες πραγματικών αριθμών διάστασης  $d \times d$  με τάξη (rank) ίση με  $d$ . Να βρείτε, αναλυτικά και σε κλειστή μορφή (closed form solution) το ολικό ελάχιστο του παρακάτω προβλήματος βελτιστοποίησης:

$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{X}\mathbf{B}\|_F^2$$

Να εξηγήσετε με λεπτομέρεια όλη τη διαδικασία σκέψης που θα ακολουθήσετε και να παραθέσετε όλα τα ενδιάμεσα βήματα των πράξεων.

### [Ερώτημα 3: Gradient descent] (70 βαθμοί)

Ο αλγόριθμος gradient descent είναι η πιο διαδεδομένη μέθοδος βελτιστοποίησης στη μηχανική μάθηση. Σε αυτό το ερώτημα καλείστε να υλοποιήσετε σε Python τον αλγόριθμο gradient descent για την ελαχιστοποίηση συναρτήσεων δύο μεταβλητών.

Θεωρήστε τις παρακάτω συναρτήσεις:

$$f_1([x_1, x_2]^T) = f_1(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 3)^2$$

$$f_2([x_1, x_2]^T) = f_2(x_1, x_2) = (1 - (x_2 - 3))^2 + 20 \left( (x_1 + 3) - (x_2 - 3)^2 \right)^2$$

Να υλοποιήσετε σε Python και να εκτελέσετε τον αλγόριθμο gradient descent για κάθε συνάρτηση.

Να αρχικοποιήσετε τις μεταβλητές με μηδέν. Πρώτα, τρέξτε με ένα σταθερό ποσοστό εκμάθησης (learning rate) ίσο με 0.5. Να εκτελέσετε T=10 επαναλήψεις και αναφέρετε την τιμή της συνάρτησης στο τέλος κάθε βήματος (επανάληψης). Χρησιμοποιώντας τη matplotlib να σχεδιάσετε γραφήματα (ένα για κάθε συνάρτηση) που να απεικονίζουν την τιμή της συνάρτησης (κάθετος άξονας) ως συνάρτηση των επαναλήψεων (οριζόντιος άξονας).

Στη συνέχεια, να επαναλάβετε τη διαδικασία για διαφορετικές τιμές του learning rate και του πλήθους επαναλήψεων (T) και αναφέρετε τα αποτελέσματά σας με τη μορφή γραφημάτων όπως στο παραπάνω σκέλος του ερωτήματος. Πόσες επαναλήψεις χρειάζεται ο αλγόριθμος για να συγκλίνει σε στάσιμο σημείο για διαφορετικές τιμές του learning rate για κάθε συνάρτηση; Τι συμβαίνει για μεγάλες τιμές του learning rate; Τι συμβαίνει για μικρές τιμές του learning rate; Τι συμβαίνει εάν χρησιμοποιήσετε διαφορετικές τιμές αρχικοποίησης;