Μηχανική Μάθηση: Εργασία 2

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Πελοποννήσου

14 Μαΐου 2026
Λήξη Προθεσμίας: 4 Ιουνίου 2026

Θέμα 1 (Οπισθοδιάδοση) (0.4 μονάδες)

Όπως έχουμε δεί στο μάθημα, η οπισθοδιάδοση είναι ο κύριος αλγόριθμος που χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων. Η οπισθοδιάδοση υπολογίζει πώς θα πρέπει να αλλάξει κάθε παράμετρος (δηλ. βάρος) σε ένα νευρωνικό δίκτυο, εφαρμόζοντας τον κανόνα της αλυσίδας προς τα πίσω. Έστω ένα νευρωνικό δίκτυο το οποίο δεδομένου ενός διανύσματος εισόδου $\mathbf{x}$ το οποίο έχει ετικέτα y, υπολογίζει την απώλεια ως εξής (προς τα εμπρός πέρασμα):

$$\begin{aligned}\mathbf{z} &= \mathbf{W}^{(1)}\,\mathbf{x} + \mathbf{b}^{(1)}\\ \mathbf{h} &= \mathrm{ReLU}(\mathbf{z})\\ \hat{y} &= \mathbf{w}^{(2)\top}\mathbf{h} + b^{(2)}\\ \mathcal{L} &= \frac{1}{2}(y-\hat{y})^2\end{aligned}$$

(α) Υπολογίστε το προς τα πίσω πέρασμα του δικτύου. Δηλαδή πρέπει να υπολογίσετε την μερική παράγωγο της απώλειας $\mathcal{L}$ ως προς τις παραμέτρους του νευρωνικού δικτύου, δηλαδή τις παρακάτω ποσότητες:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{(2)}} \qquad \frac{\partial \mathcal{L}}{\partial b^{(2)}} \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}} \qquad \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(1)}}$$

Σημειώστε ότι στις διαφάνειες συμβολίζουμε τις παραπάνω μερικές παραγώγους με μια μπάρα πάνω από την ποσότητα του παρονομαστή (δηλαδή το $\partial\mathcal{L}/\partial\mathbf{w}^{(2)}$ ως $\overline{\mathbf{w}^{(2)}}$). Σημειώστε επίσης ότι η παράγωγος της συνάρτησης $\mathrm{ReLU}(\cdot)$ είναι ίση με:

$$\mathrm{ReLU}'(x) = \begin{cases}1 & \text{αν } x > 0\\ 0 & \text{αν } x \le 0\end{cases} = \mathrm{sign}\Big(\mathrm{ReLU}(x)\Big)$$

(β) Αν η είσοδος $\mathbf{x}$ που δίνεται στο νευρωνικό δίκτυο και η ετικέτα της εισόδου y ορίζονται ως εξής:

$$\mathbf{x} = \begin{bmatrix}1.2\\ 0.5\end{bmatrix} \qquad y = 0.5$$

και οι παράμετροι του νευρωνικού δικτύου είναι οι παρακάτω:

$$\mathbf{W}^{(1)} = \begin{bmatrix}0.2 & -1.5\\ -0.8 & 1.2\\ -1 & -0.4\end{bmatrix} \qquad \mathbf{b}^{(1)} = \begin{bmatrix}1.2\\ 0.5\\ -1\end{bmatrix} \quad \mathbf{w}^{(2)} = \begin{bmatrix}-1.8\\ 1.2\\ 0.8\end{bmatrix} \qquad \mathbf{b}^{(2)} = -0.4$$

να πραγματοποιηθεί το προς τα εμπρός πέρασμα και να υπολογιστεί η απώλεια $\mathcal{L}$ και έπειτα να πραγματοποιηθεί το προς τα πίσω πέρασμα και να υπολογιστούν οι τιμές των μερικών παραγώγων του ερωτήματος (α).

Θέμα 2 (Συνελικτικά Νευρωνικά Δίκτυα) (0.3 μονάδες)

Δίνονται οι πίνακες $\mathbf{I}_1$ και $\mathbf{I}_2$ οι οποίοι αποτελούν αναπαραστάσεις δυο εικόνων οι οποίες απεικονίζουν τον χαρακτήρα Τ′ (σε διαφορετική θέση σε κάθε εικόνα), καθώς και ένα φίλτρο $\mathbf{K}$ (ή πυρήνας) ενός συνελικτικού επιπέδου ενός νευρωνικού δικτύου.

0	1	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$\mathbf{I}_1$

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	0	0	0	1	0	0
0	0	0	0	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$\mathbf{I}_2$

1	1	1
1	-1	-1
1	-1	-1

$\mathbf{K}$

(α) Εφαρμόστε το φίλτρο στις δυο εικόνες και κατασκευάστε τους νέους πίνακες που προκύπτουν. Έπειτα εφαρμόστε max pooling με μέγεθος παραθύρου 2×2 και κατασκευάστε τους νέους πίνακες που προκύπτουν.

(β) Καταφέρνει το φίλτρο $\mathbf{K}$ να αναγνωρίσει την ύπαρξη του χαρακτήρα Τ′ στις δυο εικόνες και γιατί;

(γ) Έστω ένα άλλο φίλτρο $\mathbf{K}'$ το οποίο φαίνεται παρακάτω.

1	1	1
1	0	0
1	0	0

$\mathbf{K}'$

Έστω επίσης ότι μας δίνεται ένα σύνολο εικόνων που απεικονίζουν χαρακτήρες κι έστω ότι οι τιμές όλων των εικονοστειχείων βρίσκονται μεταξύ 0 και 1 (0=λευκό, 1=μαύρο). Καταφέρνει το φίλτρο $\mathbf{K}'$ να αναγνωρίσει την ύπαρξη αποκλειστικά του χαρακτήρα Τ′ στις εικόνες που μας έχουν δοθεί;

Θέμα 3 (Transformer) (0.4 μονάδες)

Έστω ότι έχουμε εκπαιδεύσει ένα μοντέλο Transformer ώστε να πραγματοποιεί μετάφραση προτάσεων από Ελληνικά σε Αγγλικά. Θέλουμε να μεταφράσουμε την πρόταση «Μου αρέσει η μηχανική μάθηση». Τα διανύσματα των λέξεων της παραπάνω πρότασης (μαζί με την κωδικοποίηση θέσης) είναι τα ακόλουθα:

$$\mathbf{v}_{\text{Μου}} = \begin{bmatrix} -1.1 \\ -1.4 \\ 0.4 \end{bmatrix} \quad \mathbf{v}_{\text{αρέσει}} = \begin{bmatrix} 0.4 \\ 2.5 \\ 0.1 \end{bmatrix} \quad \mathbf{v}_{\text{η}} = \begin{bmatrix} -0.8 \\ -2.0 \\ -0.2 \end{bmatrix} \quad \mathbf{v}_{\text{μηχανική}} = \begin{bmatrix} 1.2 \\ 1.8 \\ -1.0 \end{bmatrix} \quad \mathbf{v}_{\text{μάθηση}} = \begin{bmatrix} 1.0 \\ 2.2 \\ -0.8 \end{bmatrix}$$

Όπως είπαμε στο μάθημα, στην καρδιά κάθε επιπέδου του κωδικοποιητή του Transformer βρίσκεται ένας μηχανισμός αυτοπροσοχής. Έστω ότι οι πίνακες $\mathbf{W}_K, \mathbf{W}_Q, \mathbf{W}_V$ του μηχανισμού αυτοπροσοχής του πρώτου επιπέδου του κωδικοποιητή Transformer είναι οι εξής:

$$\mathbf{W}_K = \begin{bmatrix} 1.6 & -2.4 & -1.7 & 0.2 \\ -0.6 & -0.3 & -0.5 & -0.2 \\ -0.5 & -0.1 & -0.2 & 0 \end{bmatrix} \quad \mathbf{W}_Q = \begin{bmatrix} -1.0 & 0.1 & -1.0 & 0.5 \\ -0.4 & -0.1 & 0 & -1.3 \\ 0.4 & 0.6 & -1.4 & 0.1 \end{bmatrix} \quad \mathbf{W}_V = \begin{bmatrix} 0.6 & -1.0 & 1.3 & 0 \\ -0.5 & -0.4 & 1.0 & -3.1 \\ 1.4 & -1.5 & -0.6 & 0.2 \end{bmatrix}$$

Υπολογίστε την έξοδο του μηχανισμού αυτοπροσοχής (δηλαδή τα νέα διανύσματα των πέντε λέξεων). Δείξτε όλα τα βήματα.

Θέμα 4 (Αναδρομικά Νευρωνικά Δίκτυα) (0.4 μονάδες)

Θα υλοποιήσετε ένα αναδρομικό νευρωνικό δίκτυο (συγκεκριμένα ένα Long Short-Term Memory (LSTM) μοντέλο) το οποίο θα παράγει κείμενο στην Ελληνική γλώσσα. Συγκεκριμένα, το μοντέλο θα δουλεύει σε επίπεδο χαρακτήρα. Θα εκπαιδευτεί παίρνοντας ως είσοδο μια ακολουθία χαρακτήρων και προβλέποντας τον επόμενο χαρακτήρα της ακολουθίας. Το σύνολο δεδομένων που θα χρησιμοποιηθεί είναι αποθηκευμένο στο αρχείο `lyrics.txt` και περιέχει στίχους από περισσότερα από 1000 Ελληνικά τραγούδια.

Σας δίνεται ένας αρχικός κώδικας για να ξεκινήσετε στο αρχείο `text_gen.py`. Θα πρέπει να υλοποιήσετε τα ακόλουθα:

- τη μέθοδο `create_training_data()` η οποία δημιουργεί τα παραδείγματα εκπαίδευσης. Κάθε παράδειγμα εκπαίδευσης είναι μια ακολουθία 30 χαρακτήρων από το κείμενο και η ετικέτα του είναι ο χαρακτήρας ο οποίος ακολουθεί τους 30 χαρακτήρες. Με άλλα λόγια, η μέθοδος θα πρέπει να επιστρέφει έναν πίνακα `X` με μέγεθος (n_samples, 30, vocab_size) που περιέχει όλες τις ακολουθίες 30 χαρακτήρων που υπάρχουν στο σύνολο δεδομένων (και οι οποίες ακολουθούνται από τουλάχιστον ένα χαρακτήρα). Κάθε χαρακτήρας της ακολουθίας αναπαρίσταται ως one-hot διάνυσμα. Θα πρέπει επίσης να επιστρέφει έναν πίνακα `y` με μέγεθος (n_samples, vocab_size) που θα δίνει πληροφορία για το ποιός είναι ο επόμενος χαρακτήρας για κάθε ακολουθία του πίνακα `X`. Και εδώ ο επόμενος χαρακτήρας αναπαρίσταται ως one-hot διάνυσμα.

- τη μέθοδο `create_model()` η οποία δημιουργεί το μοντέλο που θα προβλέπει τον επόμενο χαρακτήρα. Το μοντέλο μπορεί να υλοποιηθεί ως ένα Sequential μοντέλο στο Keras και θα πρέπει να αποτελείται από τα ακόλουθα επίπεδα:

  1. ένα επίπεδο LSTM το οποίο αποτελείται από 128 κρυφές μονάδες και μετασχηματίζει τις one-hot αναπαραστάσεις των χαρακτήρων της ακολουθίας που δίνεται ως είσοδος. Μπορείτε να χρησιμοποιήσετε την κλάση `LSTM` του Keras για να υλοποιήσετε αυτό το επίπεδο. Μιας και θέλουμε να εξάγουμε τα διανύσματα που παράγονται για όλους τους χαρακτήρες και όχι μόνο για τον τελευταίο χαρακτήρα της ακολουθίας πρέπει η παράμετρος `return_sequences` του επιπέδου να τεθεί στην τιμή True.
  2. ένα δεύτερο επίπεδο LSTM το οποίο αποτελείται από 128 κρυφές μονάδες. Αυτό το επίπεδο θέλουμε να εξάγει μόνο την αναπαράσταση του τελευταίου χαρακτήρα της ακολουθίας, την οποία θα χρησιμοποιήσουμε ώστε να προβλέψουμε τον επόμενο χαρακτήρα.
  3. ένα πλήρως συνδεδομένο επίπεδο το οποίο για κάθε ακολουθία παράγει πιθανότητες για το ποιός θα είναι ο επόμενος χαρακτήρας της πρότασης.

  Συνεπώς, η είσοδος στο μοντέλο είναι ένα σύνολο από ακολουθίες 30 χαρακτήρων κι έχει μέγεθος (batch_size, 30, vocab_size). Η έξοδος του μοντέλου πρέπει να είναι πιθανότητες για το ποιός θα είναι ο επόμενος χαρακτήρας κάθε ακολουθίας που δόθηκε ως είσοδος, δηλαδή να έχει μέγεθος (batch_size, vocab_size).

- τη μέθοδο `generate_text()` η οποία ξεκινάει από τους 30 παρακάτω χαρακτήρες «οταν τα ματια σου κοιταω θελω », και παράγει 200 επιπλέον δίνοντας κάθε φορά ως είσοδο στο μοντέλο τους 30 τελευταίους χαρακτήρες μέχρι εκείνο το σημείο. Η μέθοδος θα πρέπει να διαλέγει τυχαία τον επόμενο χαρακτήρα από το σύνολο των χαρακτήρων με πιθανότητα ίση με αυτή που επιστρέφει το μοντέλο ως πρόβλεψη.

Εκπαιδεύστε το μοντέλο για 50 εποχές με μέγεθος μικροδέσμης (batch size) ίσο με 128. Μιας και προβλέπουμε πιθανότητες για vocab_size διαφορετικές κλάσεις (δηλαδή τους διαφορετικούς διαθέσιμους χαρακτήρες), η συνάρτηση απώλειας θα πρέπει να είναι η διασταυρωμένη εντροπία για πολλαπλές κλάσεις. Χρησιμοποιήστε τον αλγόριθμο Adam ως μέθοδο βελτιστοποίησης με ρυθμό μάθησης (learning rate) ίσο με 0.001. Εμφανίσετε το κείμενο που παρήγαγε το μοντέλο. Βγάζει νόημα το κείμενο αυτό;

⚠ Προτείνεται να εκτελέσετε τον κώδικα στην πλατφόρμα Google Colab όπου το πακέτο Keras είναι ήδη προ-εγκατεστημένο και να εκπαιδεύσετε το μοντέλο στη GPU που παρέχει η πλατφόρμα, μιας και εκπαίδευση σε CPU για 50 εποχές ενδέχεται να διαρκέσει πολλές ώρες.

Θέμα 5 (Data Challenge) (0.5 μονάδες)

Στα πλαίσια της εργασίας αυτής θα πρέπει να λάβετε μέρος σε ένα Data Challenge το οποίο φιλοξενείται στην πλατφόρμα Kaggle, μια πλατφόρμα όπου εταιρείες και ερευνητές δημοσιεύουν τα δεδομένα τους σε μορφή διαγωνισμών και άνθρωποι από όλο τον κόσμο ανταγωνίζονται για να παράγουν τα καλύτερα μοντέλα. Το Data Challenge στο οποίο θα λάβετε μέρος είναι ιδιωτικό, δηλαδή πρόσβαση έχουν μόνο οι φοιτητές του μαθήματος και βρίσκεται στον ακόλουθο σύνδεσμο: `https://www.kaggle.com/competitions/ml-uop-2026`. Για να μπορέσετε να συμμετάσχετε θα πρέπει να χρησιμοποιήσετε τον ακόλουθο σύνδεσμο: `https://www.kaggle.com/t/20933ba73189446e9d8339fad0f4b1e3`. Όσοι έχετε δημιουργήσει ομάδες δυο ατόμων, αφού συνδεθείτε θα πρέπει να κάνετε merge τους λαγαριασμούς σας σε μια ομάδα.

Τα δεδομένα εκπαίδευσης και ελέγχου μπορείτε να τα κατεβάσετε από την πλατφόρμα Kaggle αφού συνδεθείτε. Σας δίνονται τα αρχεία `train.txt`, `test.txt` και `y_train.txt`. Τα δυο πρώτα αρχεία περιέχουν τα παραδείγματα εκπαίδευσης και ελέγχου, αντίστοιχα. Κάθε γραμμή των αρχείων αυτών αντιστοιχεί σε ένα άτομο και περιλαμβάνει τα εξής 12 χαρακτηριστικά τα οποία χωρίζονται μεταξύ τους με χαρακτήρα κόμμα:
(1) ηλικία
(2) εργατική τάξη
(3) εκπαίδευση
(4) οικογενειακή κατάσταση
(5) επάγγελμα
(6) σχέση
(7) φυλή
(8) φύλο
(9) κέρδος κεφαλαίου
(10) απώλεια κεφαλαίου
(11) ώρες εργασίας ανά εβδομάδα
(12) χώρα προέλευσης

Για παράδειγμα, οι γραμμές που αντιστοιχούν στα 5 πρώτα άτομα του συνόλου εκπαίδευσης έχουν την παρακάτω μορφή:

```
38, Private, HS-grad, Divorced, Adm-clerical, Unmarried, White, Female,0,0,20, United-States
45, State-gov, HS-grad, Married-civ-spouse, Exec-managerial, Wife, White, Female,0,0,40, United-States
29, Private, Bachelors, Married-civ-spouse, Exec-managerial, Husband, Black, Male,0,0,55, United-States
51, Private, Doctorate, Never-married, Prof-specialty, Not-in-family, White, Male,0,0,40, United-States
46, Private, Masters, Widowed, Prof-specialty, Not-in-family, White, Female,0,0,44, United-States
```

Για κάθε άτομο, σκοπός είναι να προβλέψουμε αν το εισόδημά του είναι χαμηλό ή υψηλό. Συνεπώς, πρόκειται για ένα πρόβλημα δυαδικής κατηγοριοποίησης. Οι κλάσεις για τα δεδομένα εκπαίδευσης (δηλαδή αν κάθε άτομο από το σύνολο εκπαίδευσης έχει χαμηλό ή υψηλό εισόδημα) βρίσκονται στο αρχείο `y_train.txt`. Αφού κάνετε προβλέψεις για τα δεδομένα ελέγχου, θα πρέπει να δημιουργήσετε ένα αρχείο με τις προβλέψεις και να το ανεβάσετε στην πλατφόρμα Kaggle η οποία θα το αξιολογήσει και θα σας εμφανίσει την ακρίβεια (accuracy) των προβλέψεών σας. Σας δίνεται το αρχείο `baseline.py` το οποίο περιέχει έναν αρχικό κώδικα για να ξεκινήσετε καθώς και τον κώδικα ο οποίος δημιουργεί το αρχείο με τις προβλέψεις που πρέπει να ανεβάσετε στο Kaggle. Ο συγκεκριμένος κώδικας μετατρέπει τις κατηγορικές μεταβλητές (δηλαδή τις μεταβλητές που οι τιμές τους προέρχονται από ένα σύνολο κατηγοριών όπως η χώρα προέλευσης) σε αριθμητικές αντιστοιχώντας κάθε τιμή της μεταβλητής σε έναν ακέραιο. Έπειτα, τα δεδομένα εκπαίδευσης και ελέγχου δίνονται ως είσοδος σε ένα ταξινομητή k-πλησιέστερων γειτόνων ο οποίος προβλέπει την κλάση στην οποία ανήκει κάθε παράδειγμα του συνόλου ελέγχου.

Σημειώστε ότι υπάρχουν κάποιες τιμές που λείπουν στα χαρακτηριστικά εργατική τάξη, επάγγελμα και χώρα προέλευσης. Οι τιμές αυτές συμβολίζονται με ένα ερωτηματικό (?). Ο κώδικας στο αρχείο `baseline.py` θεωρεί τις τιμές που λείπουν σαν μια ξεχωριστή τιμή. Θα μπορούσατε πιθανώς να τις χειριστείτε με καλύτερο τρόπο. Σημειώστε επίσης ότι η κατάλληλη κωδικοποίηση των χαρακτηριστικών ενδεχομένως να σας προσφέρει σημαντική αύξηση στην απόδοση, μεγαλύτερη από ότι η χρήση πιο ισχυρών αλγορίθμων μάθησης. Συνεπώς, θα πρέπει να αφιερώσετε κάποιο χρόνο και σε αυτό το κομμάτι.

Η ημερομηνία λήξης του διαγωνισμού είναι η **03/06/2026** και ώρα **23:59**. Για το συγκεκριμένο ερώτημα, θα πρέπει να υποβάλετε ένα σύντομο κείμενο που περιγράφει τι ακριβώς κάνατε καθώς και τον κώδικα που παρήγαγε την καλύτερη λύση σας.

⚠ Υπάρχει όριο στον αριθμό των υποβολών που μπορείτε να κάνετε κάθε μέρα στην πλατφόρμα (το πολύ 5 υποβολές ανά μέρα).

⚠ Τα δεδομένα ελέγχου στο Kaggle έχουν χωριστεί σε δυο υποσύνολα (public και private). Μέχρι τη λήξη του διαγωνισμού, οι ακρίβειες που εμφανίζονται στο leaderboard είναι αυτές στο public κομμάτι των δεδομένων ελέγχου. Μετά τη λήξη του διαγωνισμού, θα εμφανίζονται αυτές στο private κομμάτι των δεδομένων.

⚠ Ο βαθμός σας για αυτό το ερώτημα θα εξαρτηθεί από την ακρίβεια που θα επιτύχετε και τη θέση σας στο private leaderboard του διαγωνισμού.

⚠ Πρέπει να χρησιμοποιήσετε μόνο αλγορίθμους μηχανικής μάθησης που έχουμε καλύψει ή θα καλύψουμε

στο μάθημα. Αν χρησιμοποιήσετε κάποιον αλγόριθμο που δεν έχουμε καλύψει, ο βαθμός σας για αυτό το ερώτημα θα είναι **0**.

⚠ Αν έχετε ερωτήσεις σχετικά με την πλατφόρμα Kaggle, δημοσιεύστε τις ερωτήσεις αυτές στην περιοχή συζητήσεων στο e-class.

Διευκρινίσεις:

- Μπορείτε να δουλέψετε είτε μόνοι σας ή σε ομάδες 2 ατόμων (οι ομάδες πρέπει να είναι ίδιες με αυτές της πρώτης εργασίας).
  ⚠ Εργασίες με 3 ή περισσότερα άτομα δεν θα βαθμολογούνται.
- Μην ξεχάσετε να προσθέστε τα ονόματα και τους αριθμούς μητρώου των μελών της ομάδας σας στην κορυφή του jupyter notebook ή της αναφοράς που θα παραδώσετε.
- Για να βαθμολογηθούν τα θέματα που ζητούν την ανάπτυξη κώδικα, πρέπει ο κώδικας να εκτελείται και να παρουσιάζει αποτέλεσμα.
- Για τα θέματα που ζητούν την υλοποίηση αλγορίθμων, μπορείτε να μεταφέρετε τον κώδικα στο Google Colab και να παραδώσετε το/τα jupyter notebooks που θα προκύψουν. Τυχόν ερωτήσεις που υπάρχουν στα συγκεκριμένα θέματα θα πρέπει να απαντηθούν είτε στο jupyter notebook ή στην αναφορά. Αν δεν απαντηθούν οι ερωτήσεις, δεν θα λάβετε όλες τις μονάδες του θέματος.
- Η χρήση μεγάλων γλωσσικών μοντέλων ή chatbots για την απάντηση οποιουδήποτε θέματος δεν συνίσταται. Αν χρησιμοποιηθεί κάποιο τέτοιο εργαλείο, να αναφερθεί πως ακριβώς έγινε η χρήση του.
- Τα αρχεία σας θα πρέπει να υποβληθούν αποκλειστικά στην πλατφόρμα e-class, στην ενότητα Εργασίες, από ένα μόνο μέλος της ομάδας σας.
- Δημιουργήστε ένα μοναδικό αρχείο .zip (⚠ όχι .rar) που περιέχει όλα τα αρχεία σας (π.χ., αναφορά, κώδικα) εκτός των δεδομένων και υποβάλετε το συγκεκριμένο αρχείο στην πλατφόρμα e-class. Σε περίπτωση που χρησιμοποιείτε το Google Colab, μπορείτε να κατεβάσετε τον κώδικα ως ένα αρχείο .ipynb από το Google Colab πατώντας File → Download → Download .ipynb.