

# Αναφορά Εργασίας Υπολογιστικής Νοημοσύνης ΜΥΕ-035

Παναγιώτης Ζούμπας 4873  
Άγγελος Μπριντζής 4741  
Αδαμάντιος Δημήτριος Κάλλης 4685

## 1. Εισαγωγή

Στην παρούσα εργασία υλοποιήθηκαν δύο διαφορετικά προβλήματα της Υπολογιστικής Νοημοσύνης.

Η **Άσκηση 1** αφορά πρόβλημα επιβλεπόμενης μάθησης (ταξινόμηση) με χρήση πολυεπίπεδου νευρωνικού δικτύου (MLP), ενώ η **Άσκηση 2** αφορά πρόβλημα μη επιβλεπόμενης μάθησης (ομαδοποίηση) με χρήση του αλγορίθμου k-means.

Για κάθε άσκηση ακολουθήθηκαν πιστά τα βήματα που περιγράφονται στην εκφώνηση και πραγματοποιήθηκε πειραματική μελέτη των παραμέτρων.

## 2. Άσκηση 1 – Σύνολο Δεδομένων Ταξινόμησης (ΣΔΤ)

### Σύνοψη Κλάσεων – Άσκηση 1 (ΣΔΤ + MLP)

**Config:** Συγκεντρώνει όλες τις σταθερές παραμέτρους της Άσκησης 1.

Ορίζει τη διάσταση εισόδου  $d = 2$ , τον αριθμό κατηγοριών  $K = 4$ , το πλήθος νευρώνων των τριών κρυμμένων επιπέδων, καθώς και τις παραμέτρους εκπαίδευσης όπως το learning rate, το κριτήριο σύγκλισης (EPS) και τον μέγιστο αριθμό εποχών.

Με αυτόν τον τρόπο επιτυγχάνεται ευκολότερη ρύθμιση και πειραματισμός χωρίς αλλαγές στον κύριο κώδικα.

**SDTLoader:** Είναι υπεύθυνη για τη φόρτωση του συνόλου δεδομένων ταξινόμησης από αρχείο CSV. Διαβάζει τα σημεία  $(x_1, x_2)$  και μετατρέπει τις κατηγορίες της εκφώνησης (1–4) σε labels 0–3, κατάλληλα για χρήση στο νευρωνικό δίκτυο. Απομονώνει έτσι πλήρως τη διαδικασία εισόδου δεδομένων από την εκπαίδευση του μοντέλου.

**LabeledExample:** Αναπαριστά ένα δείγμα εκπαίδευσης ή ελέγχου.

Αποθηκεύει τις εισόδους  $(x_1, x_2)$ , την ακέραια ετικέτα της κατηγορίας, καθώς και το αντίστοιχο one-hot target vector. Χρησιμοποιείται τόσο στο forward pass όσο και στο backpropagation του δικτύου.

**Activation:** Υλοποιεί τις συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στο νευρωνικό δίκτυο. Περιλαμβάνει τις συναρτήσεις tanh, logistic και ReLU, καθώς και τις παραγώγους τους, οι οποίες είναι απαραίτητες για την εκπαίδευση με backpropagation. Η επιλογή διαφορετικής συνάρτησης στο τρίτο κρυμμένο επίπεδο επιτρέπει τη σύγκριση της επίδρασής τους στη γενικευτική ικανότητα του δικτύου.

**MLP:** Υλοποιεί το πολυεπίπεδο νευρωνικό δίκτυο της άσκησης. Περιλαμβάνει την αρχικοποίηση των βαρών και bias, τον υπολογισμό του forward pass, την εκτέλεση του backpropagation και την εφαρμογή των ενημερώσεων βαρών. Το δίκτυο αποτελείται από τρία κρυμμένα επίπεδα και ένα επίπεδο εξόδου κατάλληλο για πολυκατηγορική ταξινόμηση.

**Gradients:** Χρησιμοποιείται για την αποθήκευση των παραγώγων του σφάλματος ως προς όλα τα βάρη και τα bias του δικτύου. Επιτρέπει τη συσσώρευση gradients σε mini-batches και τη μετέπειτα μέση τιμή τους πριν την ενημέρωση των παραμέτρων. Με αυτόν τον τρόπο υποστηρίζεται η εκπαίδευση με mini-batch gradient descent.

**Trainer:** υλοποιεί τη διαδικασία εκπαίδευσης του MLP. Εκτελεί την εκπαίδευση σε mini-batches μεγέθους  $L$ , ανακατεύει τα δεδομένα σε κάθε εποχή και εφαρμόζει το κριτήριο διακοπής με βάση τη σύγκλιση του σφάλματος (EPS) και τον ελάχιστο αριθμό εποχών. Αποτελεί την υλοποίηση των Βημάτων 6 και 7 της εκφώνησης.

**Evaluator:** Υπολογίζει την ακρίβεια ταξινόμησης του δικτύου στο σύνολο ελέγχου. Συγκρίνει την προβλεπόμενη κατηγορία με την πραγματική ετικέτα και επιστρέφει το ποσοστό σωστών ταξινομήσεων. Χρησιμοποιείται για την αξιολόγηση και την επιλογή του μοντέλου με τη μεγαλύτερη γενικευτική ικανότητα.

**ExperimentMain:** αποτελεί το κύριο πρόγραμμα της Άσκησης 1.

Φορτώνει τα δεδομένα ΣΔΤ, εκτελεί πειράματα για διαφορετικές τιμές του mini-batch μεγέθους και διαφορετικές συναρτήσεις ενεργοποίησης, αποθηκεύει τα αποτελέσματα σε αρχείο CSV και επιλέγει το καλύτερο μοντέλο βάσει ακρίβειας στο σύνολο ελέγχου. Για το βέλτιστο μοντέλο, αποθηκεύει επίσης για κάθε παράδειγμα του συνόλου ελέγχου αν ταξινομήθηκε σωστά ή λάθος.

**GenerateSDT:** Είναι υπεύθυνη για τη δημιουργία του Συνόλου Δεδομένων Ταξινόμησης (ΣΔΤ).

Δημιουργεί τυχαία σημεία  $(x_1, x_2)$  στο τετράγωνο  $[0,2] \times [0,2]$  και τα κατατάσσει σε μία από τέσσερις κατηγορίες σύμφωνα με τις συνθήκες που ορίζονται στην εκφώνηση.

Το παραγόμενο σύνολο δεδομένων αποθηκεύεται σε αρχείο CSV και χρησιμοποιείται ως είσοδος για την εκπαίδευση και τον έλεγχο του νευρωνικού δικτύου.

## 2. Άσκηση 1 – Σύνολο Δεδομένων Ταξινόμησης (ΣΔΤ)

### 2.1 Δημιουργία και φόρτωση δεδομένων

Σύμφωνα με την εκφώνηση, δημιουργήθηκαν 8000 τυχαία σημεία  $(x_1, x_2)$  στο τετράγωνο  $[0,2] \times [0,2]$ .

Τα δεδομένα χωρίστηκαν σε:

- 4000 παραδείγματα εκπαίδευσης
- 4000 παραδείγματα ελέγχου

Η κατάταξη κάθε σημείου σε μία από τις τέσσερις κατηγορίες  $C_1, C_2, C_3, C_4$  έγινε βάσει των 18 συνθηκών της εκφώνησης.

Η φόρτωση των δεδομένων υλοποιήθηκε στην κλάση **SDTLoader**, όπου οι κλάσεις 1–4 μετατράπηκαν εσωτερικά σε labels 0–3 για τη σωστή λειτουργία του δικτύου.

---

### 2.2 Αρχιτεκτονική Νευρωνικού Δικτύου

Υλοποιήθηκε πολυεπίπεδο νευρωνικό δίκτυο (MLP) με:

- $d = 2$  εισόδους
- $K = 4$  εξόδους (μία για κάθε κατηγορία)
- τρία κρυμμένα επίπεδα με αριθμό νευρώνων  $H_1, H_2, H_3$

Η αρχιτεκτονική και οι σταθερές της άσκησης ορίστηκαν στην κλάση **Config**.

---

### 2.3 Συναρτήσεις ενεργοποίησης

Σύμφωνα με την εκφώνηση:

- στα κρυμμένα επίπεδα χρησιμοποιήθηκαν συναρτήσεις **tanh**, **logistic** ή **ReLU**
- στο επίπεδο εξόδου χρησιμοποιήθηκε κατάλληλη συνάρτηση για πολυκατηγορική ταξινόμηση

Η υλοποίηση των συναρτήσεων ενεργοποίησης και των παραγώγων τους έγινε στην κλάση **Activation**.

---

## 2.4 Εκπαίδευση του δικτύου

Η εκπαίδευση υλοποιήθηκε με **mini-batch gradient descent**, όπως ζητείται. Το μέγεθος mini-batch  $L$  επιλέχθηκε από το σύνολο:

$$L \in \left\{ \frac{N}{10}, \frac{N}{20}, \frac{N}{100}, \frac{N}{200} \right\}$$

Για κάθε mini-batch:

1. Υπολογίστηκαν τα gradients με backpropagation
2. Έγινε μέσος όρος των gradients
3. Ενημερώθηκαν τα βάρη του δικτύου

Η διαδικασία αυτή υλοποιήθηκε στις κλάσεις **Trainer**, **Gradients** και **MLP**.

---

## 2.5 Κριτήριο τερματισμού

Η εκπαίδευση σταματά όταν:

- έχει συμπληρωθεί ελάχιστος αριθμός epochs, και
- η μεταβολή του συνολικού σφάλματος εκπαίδευσης είναι μικρότερη από προκαθορισμένο όριο  $\epsilon$

Το  $\epsilon$  επιλέχθηκε πειραματικά, καθώς δεν ορίζεται ρητά στην εκφώνηση.

---

## 2.6 Αξιολόγηση και επιλογή καλύτερου μοντέλου

Μετά την εκπαίδευση, το κάθε μοντέλο αξιολογήθηκε στο σύνολο ελέγχου με μέτρο την **ακρίβεια ταξινόμησης**.

Η κλάση **Evaluator** υπολογίζει την ακρίβεια, ενώ η **ExperimentMain** συγκρίνει όλα τα πειράματα και επιλέγει το μοντέλο με τη μέγιστη γενικευτική ικανότητα.

---

## 2.7 Εκτύπωση παραδειγμάτων συνόλου ελέγχου

Για το δίκτυο με την καλύτερη γενίκευση, τα παραδείγματα του συνόλου ελέγχου αποθηκεύτηκαν σε αρχείο CSV.

Για κάθε σημείο καταγράφεται αν ταξινομείται σωστά (1) ή λανθασμένα (0), ικανοποιώντας την απαίτηση της εκφώνησης για διαφορετικό “στυλ” σωστών και λανθασμένων ταξινομήσεων.

**Σχόλιο:** Το learning rate δεν ορίστηκε ως μία σταθερή τιμή για όλα τα πειράματα, αλλά επιλέχθηκε να μεταβάλλεται ανάλογα με το μέγεθος του mini-batch  $L$ .

Η επιλογή αυτή βασίζεται στο γεγονός ότι το μέγεθος του mini-batch επηρεάζει άμεσα τη σταθερότητα και το μέγεθος των ενημερώσεων των βαρών. Για μεγάλα mini-batches, το gradient αποτελεί καλύτερη προσέγγιση του πραγματικού gradient του συνολικού σφάλματος, επιτρέποντας τη χρήση μεγαλύτερου learning rate χωρίς έντονες ταλαντώσεις. Αντίθετα, για μικρότερα mini-batches, το gradient παρουσιάζει μεγαλύτερη στοχαστικότητα, οπότε απαιτείται μικρότερο learning rate για την αποφυγή αστάθειας.

Συγκεκριμένα, χρησιμοποιήθηκε το ακόλουθο σχήμα:

- μεγαλύτερο learning rate για μεγάλα  $L$ ,
- μικρότερο learning rate για μικρότερα  $L$ .

Για την εκτέλεση του προγράμματος εκτελείτε μέσα στον φάκελο src :

```
javac -encoding UTF-8 ask1/*.java
```

```
java ask1.ExperimentMain
```