



## ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΜΑΘΗΣΗΣ

# Νευρωνικά Δίκτυα και Ευφυή Υπολογιστικά Συστήματα

## Άσκηση 3: Βαθιά Μάθηση

Στόχος σας είναι να βελτιστοποιήσετε την απόδοση μοντέλων Βαθιάς Μάθησης στο σύνολο δεδομένων CIFAR-100 χρησιμοποιώντας την βιβλιοθήκη TensorFlow 2. Σε σχέση με τις πρώτες δύο ασκήσεις του μαθήματος, το βασικό παραδοτέο μπορεί να είναι και μια γραπτή αναφορά (PDF, Word κλπ) που να συνοδεύεται από τα αρχεία .ipynb και .py σας για τον έλεγχο των αποτελεσμάτων.

Καταρχάς, τρέξτε και μελετήστε το notebook [“Εργασία 3 TensorFlow 2 - CIFAR-100.ipynb”](#). Το notebook περιλαμβάνει πολλές αναφορές στο documentation του TensorFlow 2 το οποίο πρέπει να μελετήσετε εκτενώς.

Το TF2 είναι το μόνο δεκτό ML framework για την άσκηση (δεν πρέπει να γίνει ούτε χρήση του tensorflow.compat.v1).

Στα προβλήματα Βαθιάς Μάθησης είναι απαραίτητη η επιτάχυνση με GPU. Από τις λύσεις cloud μπορείτε να δουλέψετε είτε στο Colab είτε στο Kaggle.

- [Ανοίξτε το στο Colab](#) και κάντε ένα αντίγραφο στο drive σας
- [Κάντε fork τον πυρήνα στο Kaggle](#)

Εάν διαθέτετε κάρτα γραφικών μπορείτε να δουλέψετε τοπικά κατεβάζοντας το αρχείο ipynb.

Για όλες τις υλοποιήσεις που θα κάνετε θα δουλέψετε με κάποιες υποκατηγορίες του CIFAR-100 που αντιστοιχούν με μοναδικό τρόπο στην ομάδα σας, θέτοντας στο notebook ως team\_seed τον αριθμό της ομάδας σας.

Στο notebook θα βρείτε στην ενότητα “Εισαγωγή και επισκόπηση του συνόλου δεδομένων” τον κώδικα που σας δίνει τα ονόματα των κλάσεων σας καθώς και τα σχετικά index τα οποία πρέπει να χρησιμοποιήσετε σε όποια υλοποίηση και αν κάνετε.

Ξεκινήστε από τα μοντέλα του notebook και προχωρήστε στον ορισμό νέων μοντέλων, είτε εκ του μηδενός (“from scratch”) είτε με μεταφορά μάθησης (transfer learning). Για κάθε μοντέλο συμπεριλάβετε μια σύντομη περιγραφή της αρχιτεκτονικής του και των βασικών ιδιοτήτων του. Βελτιστοποιήστε τα ακολουθώντας όσα αναφέρονται στην ενότητα “Βελτίωση της απόδοσης με πειράματα” ή/και πρόσθετες βελτιστοποιήσεις.

Περιγράψτε τη βελτιστοποίησή των μοντέλων σας, τις επιλογές και τα συμπεράσματά σας.

### Παρατηρήσεις ως προς τη βελτιστοποίηση:

- Ένα μοντέλο βελτιστοποιείται ως προς τον εαυτό του, όχι ως προς τα άλλα μοντέλα. Αν για παράδειγμα φέρετε με μεταφορά μάθησης ένα state-of-the-art μοντέλο, θα εξετάσετε πόσο μπορείτε να βελτιστοποιήσετε το ίδιο και δεν θα το συγκρίνετε με ένα πολύ απλό δίκτυο “from scratch”. Αυτό δεν σημαίνει ότι δεν μπορείτε να παρουσιάσετε συγκρίσεις μεταξύ μοντέλων σε πίνακες και γραφήματα (βλ. και επόμενη παρατήρηση).
- Η βελτιστοποίηση μπορεί να αφορά στην απόδοση ως προς τη μετρική που χρησιμοποιούμε (ορθότητα) στο σύνολο ελέγχου αλλά και σε άλλες ιδιότητες που έχουν να κάνουν με την εκπαίδευση: απαιτούμενη μνήμη, αριθμός παραμέτρων, χρόνος εκπαίδευσης, συμπεριφορά ως προς την υπερεκπαίδευση κ.ο.κ. Μπορείτε να παρουσιάσετε συγκριτικά αποτελέσματα “πριν και μετά” της βελτίωσης αυτών ιδιοτήτων ποσοτήτων ακόμα και για το ίδιο μοντέλο, ασχέτως της απόλυτης απόδοσής του ως προς την ορθότητα σε σχέση με άλλα μοντέλα.
- Προσπαθήστε να δοκιμάσετε όλες τις δυνατότητες βελτιστοποίησης που αναφέρονται στο notebook.

**Ο ρόλος του διαφορετικού αριθμού κατηγοριών ανά ομάδα (20 μέχρι 80).** Μικρός αριθμός κατηγοριών γενικά σημαίνει ευκολότερο πρόβλημα αλλά και λιγότερα συνολικά δεδομένα και αντιθετοαντίστροφα για μεγάλο αριθμό κατηγοριών. Αυτό προφανώς έχει επίδραση στους χρόνους εκτέλεσης, στην επίδοση και στην υπερεκπαίδευση ή μη μικρών και μεγάλων δικτύων.

Μας ενδιαφέρει να δούμε τι πετυχαίνετε με 80 κατηγορίες και να μας αναφέρετε χρόνους εκπαίδευσης.

**Σημειώσεις για την επίδοση των GPUs.** Σας παραθέτουμε τις μετρήσεις που λάβαμε για τους χρόνους εκτέλεσης ανά βήμα (step) -lower is better- για το απλό συνελικτικό και το VGG16 του notebook στο Colab, στο Kaggle και σε μια φυσική GeForce RTX 2080 Ti καθώς και το AI-Score τους -higher is better- σύμφωνα με τη βιβλιοθήκη [ai-benchmark](#). Το απλό CNN έχει 128.420 εκπαιδευσίμους παράγοντες ενώ το VGG16 14.765.988.

	Colab (Tesla T4)	Kaggle (Tesla P100)	RTX 2080 Ti
simple CNN (ms/step)	15	15	7
VGG16 (ms/step)	75	53	28
AI-Score	14248	20983	28368

Παρακαλούμε αναφέρετε μας σε ποιο cloud δουλεύετε. Αν χρησιμοποιείτε φυσική GPU αναφέρετε μας το AI-Score της χρησιμοποιώντας το ai-benchmark.

**Συμπληρωματικά ερωτήματα.** Πέραν της προηγούμενης αναφοράς σας ως προς τα μοντέλα, τη βελτιστοποίησή και την απόδοσή τους, μπορείτε να αναλύστε και τα ακόλουθα θέματα αν δεν τα έχετε ήδη συμπεριλάβει, πάντα με βάση τα πειράματα που εκτελέσατε:

- Μεταφορά μάθησης vs εκπαίδευση εκ του μηδενός (“from scratch”)
- Επίδραση της επαύξησης δεδομένων (data augmentation)
- Επίδραση του πλήθους των επιπέδων που θα εκπαιδεύουν (fine-tuning) κατά τη μεταφορά μάθησης
- Επίδραση του πλήθους των δεδομένων/κλάσεων στην απόδοση του μοντέλου
- Επίδραση του ρυθμού μάθησης (learning rate)
- Επίδραση του αλγόριθμου βελτιστοποίησης (optimizer)
- Επίδραση του μεγέθους δέσμης (batch size)
- Επίδραση του μεγέθους των εικόνων (resize input)

**Παράδοση.** Μπορείτε να δουλέψετε ελεύθερα σε δικά σας notebooks, δεν χρειάζεται να επεκτείνετε απαραίτητα αυτό της εκφώνησης, αρκεί να χρησιμοποιήσετε τις προκαθορισμένες κατηγορίες εικόνων για την ομάδα σας.

Μην ξεχάσετε να συμπεριλάβετε στην αναφορά και στα notebooks σας ονοματεπώνυμο, αριθμούς μητρώου και ομάδας. Παράδοση zip στο στο eclass, ipynb, py και αν σας βολεύει σύντομη αναφορά. Όχι δεδομένα προφανώς, και τα κελιά εκτελεσμένα με ορατό output.

**Ερωτήσεις.** Στο forum του μαθήματος στο eclass, στην [περιοχή συζητήσεων της τρίτης άσκησης](#).

**FAQ.** Το γνωστό [FAQ](#) είναι εκεί και για την τρίτη άσκηση.