

Project Βαθιάς Μάθησης (50% του τελικού βαθμού)

Ανάλυση Ιατρικών Εικόνων MedMNIST με CNN, Transfer Learning & Vision Transformers (PyTorch)

Σε αυτό το project θα αναλύσετε ιατρικές εικόνες από ένα dataset της συλλογής **MedMNIST** (<https://medmnist.com/>), που περιλαμβάνει εικόνες μικρής ανάλυσης από διαφορετικές ιατρικές κατηγορίες με έτοιμα train/test/validation splits.

Το project χωρίζεται σε τρία μέρη. Στο πρώτο θα σχεδιάσετε και θα εκπαιδεύσετε ένα CNN από την αρχή, πειραματιζόμενοι με τεχνικές κανονικοποίησης και regularization για να κατανοήσετε την επίδρασή τους στη μάθηση και τη γενίκευση. Στο δεύτερο θα χρησιμοποιήσετε ένα προεκπαιδευμένο CNN, αξιοποιώντας feature extraction και fine-tuning, για να δείτε τα πλεονεκτήματα του transfer learning σε μικρά datasets. Στο τρίτο θα εφαρμόσετε έναν μικρό Vision Transformer (π.χ., DeiT) με pretrained weights, για να εξετάσετε πώς η self-attention και τα patches επηρεάζουν την απόδοση σε σύγκριση με τα CNN.

Το project σας εισάγει στην κατασκευή και εκπαίδευση μοντέλων, στη χρήση προεκπαιδευμένων CNN και Transformers, και σας επιτρέπει να πειραματιστείτε με διαφορετικές τεχνικές βελτιστοποίησης, κανονικοποίησης και regularization, ενώ αναπτύσσετε κριτική ικανότητα στην επιλογή αρχιτεκτονικής και παραμέτρων.

0. Περιγραφή του Dataset

Πριν ξεκινήσετε με τα μοντέλα, επιλέξτε ένα dataset από τη συλλογή **MedMNIST**. Περιγράψτε το dataset που επιλέξατε, ώστε να κατανοήσετε τα δεδομένα με τα οποία θα δουλέψετε.

Ζητούμενα:

- Περιγράψτε τον τύπο των εικόνων (π.χ. grayscale ή RGB) και την ανάλυση (π.χ. 28×28 ή 32×32).
- Αναφέρετε τον αριθμό των κλάσεων και τη σημασία τους (π.χ. τύποι όγκων, κατηγορίες ιστών, κλπ.).
- Καταγράψτε το μέγεθος του dataset (train/test/validation splits) και αν υπάρχει ανισορροπία στις κλάσεις.
- Συμπεριλάβετε μερικά παραδείγματα εικόνων.

1. CNN από την αρχή

Σε αυτό το μέρος θα σχεδιάσετε και θα εκπαιδεύσετε ένα CNN από το μηδέν για ταξινόμηση εικόνων από το dataset της MedMNIST που επιλέξατε. Ο στόχος είναι να κατανοήσετε σε βάθος τη συμπεριφορά ενός CNN σε μικρά ιατρικά datasets και πώς τεχνικές κανονικοποίησης και regularization επηρεάζουν την εκπαίδευση, τη σταθερότητα και τη γενίκευση.

Ζητούμενα και οδηγίες:

- Σχεδιάστε ένα CNN με τουλάχιστον 3 συνελκτικά blocks, κάθε block να περιλαμβάνει Conv2d → ReLU → MaxPooling2d. Προτείνεται να ξεκινήσετε με 32, 64 και 128 φίλτρα αντίστοιχα, kernel size 3×3. (μπορείτε να πειραματιστείτε και με περισσότερα ανάλογα με την υπολογιστική δύναμη που διαθέτετε.)
- Προσθέστε ένα fully connected layer στο τέλος με αριθμό νευρώνων ίσο με τον αριθμό των κλάσεων.
- Πειραματιστείτε με **Batch Normalization** μετά από κάθε Conv2d και/ή **Layer Normalization** στο τέλος κάθε block.
- Δοκιμάστε **Dropout** στο fully connected layer με ποσοστά, 0.2, 0.5 και 0.7.
- Χρησιμοποιήστε **Weight Decay (L2 regularization)** με τιμές 1e-4, 1e-3 και 1e-2.
- Χρησιμοποιήστε **Adam optimizer**, με learning rate ενδεικτικά 1e-3, αλλά μπορείτε να πειραματιστείτε με τιμές μεταξύ 1e-4 και 5e-3.
 - Χωρίς normalization ή dropout: 1e-3
 - Με BatchNorm ή LayerNorm: 5e-4
 - Με Dropout ή Weight Decay: 1e-3 ή 1e-4 ανάλογα με το συνδυασμό τεχνικών
- Batch size: 64, Epochs: 30–50.
- Παρακολουθήστε **training και validation curves** (loss & accuracy) για κάθε συνδυασμό τεχνικών.
- Υπολογίστε **test accuracy** και δημιουργήστε **confusion matrix**.
- Συγκρίνετε τα αποτελέσματα και σχολιάστε πώς οι διαφορετικές τεχνικές επηρεάζουν την απόδοση και τη γενίκευση.

2. Transfer Learning με CNN

Στο δεύτερο μέρος θα χρησιμοποιήσετε ένα προεκπαιδευμένο CNN (π.χ. VGG16, ResNet18 ή MobileNetV2 (torchvision.models)). Στόχος είναι να κατανοήσετε πότε αξίζει η εκπαίδευση μόνο του classifier head (feature extraction) και πότε το fine-tuning.

Ζητούμενα και οδηγίες:

- Αντικαταστήστε τον classifier head για να ταιριάζει με τον αριθμό κλάσεων του dataset.
- Εκπαιδεύστε αρχικά **μόνο τον classifier head** με Adam optimizer, learning rate ~1e-3, batch size ~64, για 15–25 epochs.
- Δοκιμάστε **fine-tuning των τελευταίων δύο blocks** με learning rate ~1e-4, batch size 32, για 10–15 epochs.
- Παρακολουθήστε και συγκρίνετε **training και validation curves** για feature extraction vs fine-tuning.
- Υπολογίστε **test accuracy και confusion matrix**.
- Συζητήστε διαφορές σε απόδοση, ταχύτητα σύγκλισης και γενίκευση σε σχέση με το CNN από την αρχή.

3. Μικρός Vision Transformer (DeiT)

Σε αυτό το μέρος θα χρησιμοποιήσετε έναν μικρό Vision Transformer (π.χ. DeiT-tiny μέσω της βιβλιοθήκης `timm`) με pretrained weights. Σκοπός είναι να δείτε πώς αποδίδουν οι Transformers σε μικρά datasets και να κατανοήσετε τα βασικά στοιχεία της αρχιτεκτονικής τους.

Ζητούμενα και οδηγίες:

- Κάντε **resize των εικόνων σε 224×224** και normalization σύμφωνα με το pretrained μοντέλο.
- Προσαρμόστε τον **classifier head** για τον αριθμό των κλάσεων.
- Εκπαιδεύστε κυρίως τον classifier head (**feature extraction**) με Adam optimizer, learning rate $\sim 1e-3$, batch size 32, για 10–15 epochs.
- Προαιρετικά: κάντε **fine-tuning των τελευταίων layers** με learning rate $\sim 1e-4$, batch size 16, για 5–10 epochs.
- Παρακολουθήστε **training και validation curves**, υπολογίστε test accuracy και δημιουργήστε confusion matrix.
- Συγκρίνετε τα αποτελέσματα με CNN από την αρχή και Transfer Learning CNN.
- Περιγράψτε σύντομα την αρχιτεκτονική (patches, positional embeddings, self-attention) και σχολιάστε πλεονεκτήματα και περιορισμούς σε μικρά datasets.

4. Οδηγίες Υλοποίησης και Παράδοσης

- Όλα τα μοντέλα πρέπει να υλοποιηθούν σε PyTorch, με οργανωμένο και σχολιασμένο κώδικα.
- Χρησιμοποιήστε seeds για αναπαραγωγικότητα.
- Αναφέρετε όλους τους **hyperparameters**, optimizer και learning rates που χρησιμοποιήσατε.
- Δημιουργήστε **training/validation curves** και συγκριτική αξιολόγηση όλων των μοντέλων.
- Η αναφορά PDF πρέπει να περιλαμβάνει περιγραφή dataset, ανάλυση μοντέλων και τεχνικών, πειραματικά αποτελέσματα, σύγκριση μοντέλων, συμπεράσματα και προτάσεις βελτίωσης.
- Παραδοτέα: πλήρης κώδικας (zip) και αναφορά PDF. Deadline: 14/1/2026.