



## Άσκηση 4

### Ταξινόμηση βίντεο με αρχιτεκτονικές RNN & Transformer

#### Αντικείμενο - Στόχοι

- ✓ Εξοικείωση με δεδομένα τύπου χρονοσειράς
- ✓ Εξοικείωση με αρχιτεκτονικές RNN & Transformer
- ✓ Αξιολόγηση αποτελεσμάτων

Η ταξινόμηση βίντεο αποτελεί μια σημαντική εφαρμογή που αξιοποιείται σε συστήματα συστάσεων (recommendation), ασφάλειας (security), ανάλυσης συμπεριφοράς (sentiment analysis) κλπ. Πρόκειται για ένα πρόβλημα τύπου Seq2Vec και το dataset που θα χρησιμοποιήσουμε είναι μια απλουστευμένη εκδοχή του [UCF101](#), το οποίο είναι ένα dataset αναγνώρισης δράσης (action recognition) που αποτελείται από βίντεο που συλλέχθηκαν από το YouTube και εμπεριέχει συνολικά 101 κατηγορίες.

Για λόγους απλούστευσης, θα χρησιμοποιήσουμε μια απλουστευμένη εκδοχή αυτού του σετ δεδομένων που θα εμπεριέχει 5 κατηγορίες.

#### B1: Προετοιμασία δεδομένων

- Κατεβάστε τα δεδομένα από αυτό το [link](#).
- Σχεδιάστε και υλοποιήστε έναν αλγόριθμο τροφοδότησης δεδομένων βίντεο

#### B2: Προεπεξεργασία δεδομένων

- Χρησιμοποιείστε ένα προεκαπιδευμένο CNN στο ImageNet για να παράξετε χαρακτηριστικά για κάθε frame των βίντεο. Θεωρείστε αυτό το CNN ως μη-εκπαιδευμένο για τη συνέχεια της άσκησης
- Προβείτε στους κατάλληλους μετασχηματισμούς στα δεδομένα πριν εξάγετε τα χαρακτηριστικά

#### B3: Σχεδιασμός, υλοποίηση και εκπαίδευση αρχιτεκτονικής RNN

- Χρησιμοποιείστε αρχιτεκτονική τουλάχιστον 2 στρώσεων τύπου LSTM ή GRU
- Εκπαιδεύστε το μοντέλο σας στο πρόβλημα της ταξινόμησης

#### B4: Σχεδιασμός, υλοποίηση και εκπαίδευση αρχιτεκτονικής Transformer

- Χρησιμοποιείστε αρχιτεκτονική Transformer
  - Χρησιμοποιήστε μόνο το encoder μέρος του Transformer
  - Εφαρμόστε κατάλληλο positional encoding και υπολογίστε μάσκα για την περιοχή της χρονοσειράς που δεν περιέχει δεδομένα (λόγω zero-padding στα πρώτα χρονικά βήματα κατά τη δημιουργία του batch)
  - Εφαρμόστε 1D Max-pooling στην έξοδο του encoder κατά τη “χρονική” διάσταση ή χρησιμοποιήστε απευθείας κάποιο classification token
- Εκπαιδεύστε το μοντέλο σας στο πρόβλημα της ταξινόμησης

#### B5: Αξιολόγηση

- Αξιολογήστε τα αποτελέσματα των μοντέλων σας στο σετ ελέγχου χρησιμοποιώντας τις ακόλουθες μετρικές
  - Accuracy
  - Precision
  - Recall
  - F1-score
  - Confusion matrix
- Υπολογίστε τις μετρικές συνολικά και ανά κατηγορία όπου αυτό έχει νόημα
- Σχολιάστε τα αποτελέσματα

## Ζητούμενα

εκπονήστε τεχνική έκθεση περιγράφοντας τις διαδικασίες που ακολουθήσατε, απαντώντας και στα παρακάτω ερωτήματα

1. (50%) Εκτελέστε τα παραπάνω βήματα. Βασιστείτε στο [πρότυπο notebook](#) που υπάρχει στο github του μαθήματος

2. (50%) Δώστε απαντήσεις στα παρακάτω ερωτήματα:

1. Υποθέτουμε ότι χρησιμοποιήσατε μία bi-directional αρχιτεκτονική RNN. Θα μπορούσατε να χρησιμοποιήσετε ένα τέτοιο μοντέλο αν θα έπρεπε να κάνετε online προβλέψεις σε real-time συνεχείς ροές δεδομένων βίντεο; Αν ναι, πως; Αν όχι, γιατί;
2. Υποθέτουμε πως σκοπεύετε να χρησιμοποιήσετε το μοντέλο σας για μια εφαρμογή ασφαλείας (π.χ. εντοπισμός παράξενων συμπεριφορών σε ένα mall). Ποια θεωρείτε ότι είναι η πιο σημαντική μετρική για την αξιολόγηση του μοντέλου σας;
3. Εκπαιδεύετε το RNN μοντέλο σας και η εκπαίδευση αποκλίνει. Έπειτα από μελέτη παρατηρείτε ότι η τιμή της παραγώγου λαμβάνει πολύ υψηλές τιμές για μερικές παραμέτρους του δικτύου. Ποιο πρόβλημα πιστεύετε ότι υπάρχει; Τι θα μπορούσατε να κάνετε ώστε να το αντιμετωπίσετε; (Προτείνετε τουλάχιστον 2 λύσεις)

### Σημειώσεις

- Μαζί με την τεχνική έκθεση (pdf) να παραδώσετε και τον πηγαίο κώδικα που αναπτύξατε.
- Συστήνεται η χρήση της Pytorch και της Pytorch Lightning. Επιτρέπεται επίσης Keras/Tensorflow, JAX ή CNTK. Αν επιθυμείτε να εκπονήσετε την εργασία σε κάποια άλλη βιβλιοθήκη επικοινωνήστε πρώτα με τους διδάσκοντες του μαθήματος. Σε περίπτωση που δεν επιλέξετε την Pytorch, καλείστε να “μεταφέρετε” στην βιβλιοθήκη της επιλογής σας και το τμήμα της άσκησης που είναι ήδη υλοποιημένο στο πρότυπο notebook.

### Χρήσιμα links

- <https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html?highlight=nn%20embedding#torch.nn.Embedding>
- <https://pytorch.org/docs/stable/generated/torch.nn.Transformer.html#torch.nn.Transformer>
- <https://pytorch.org/docs/stable/generated/torch.nn.TransformerEncoder.html#torch.nn.TransformerEncoder>
- [https://pytorch.org/docs/stable/generated/torch.nn.Module.html?highlight=register%20buffer#torch.nn.Module.register\\_buffer](https://pytorch.org/docs/stable/generated/torch.nn.Module.html?highlight=register%20buffer#torch.nn.Module.register_buffer)
- [https://pytorch-lightning.readthedocs.io/en/latest/accelerators/accelerator\\_prepare.html](https://pytorch-lightning.readthedocs.io/en/latest/accelerators/accelerator_prepare.html)

