ΕΠ08 Αναγνώριση Προτύπων – Μηχανική Μάθηση

2η Εργασία

Τύπος εργασίας: **Ατομική**
Ημερομηνία παράδοσης: **Κυριακή 23/06/2024 23:55 (Δεν θα δοθεί παράταση)**
Τρόπος παράδοσης: **Αποκλειστικά μέσω του eclass**
Σύνολο βαθμών: 100 (20% του τελικού βαθμού του μαθήματος)

Η εργασία είναι ατομική και αποτελείται από 4 ερωτήματα. Συνιστάται ιδιαίτερα, να αφιερώσετε χρόνο ώστε να κατανοήσετε το θεμελιώδη λογισμό και τη λογική πίσω από τα ερωτήματα της εργασίας και να αποφύγετε την αναζήτηση έτοιμων λύσεων στο διαδίκτυο. Αν ωστόσο συμβουλευτείτε ή/και χρησιμοποιήσετε οποιοδήποτε υλικό ή/και κώδικα που είναι διαθέσιμα στο διαδίκτυο, πρέπει να αναφέρεται σωστά τη πηγή ή/και το σύνδεσμο στην ιστοσελίδα από όπου αντλήσατε πληροφορίες. Σε κάθε περίπτωση, **η αντιγραφή** τμήματος ή του συνόλου της εργασίας **δεν είναι αποδεκτή** κα στη περίπτωση που διαπιστωθεί αντιγραφή θα μηδενιστούν στο μάθημα όλα τα εμπλεκόμενα μέρη. Θα υπάρξει προφορική εξέταση της εργασίας.

Θα πρέπει να υποβάλετε ένα μόνο αρχείο Interactive Python NoteBook (Jupyter Notebook) μέσω του εργαλείου “Εργασίες” του eclass, ακολουθώντας την εξής σύβαση ονομασίας για το αρχείο σας: **Επώνυμο_ΑριθμόςΜητρώου.ipynb**

Τόσο ο κώδικας Python όσο και οι απαντήσεις σας στις ερωτήσεις πρέπει να είναι ενσωματωμένα στο ίδιο IPython notebook. Μπορείτε να χρησιμοποιήσετε κελιά επικεφαλίδας για να οργανώσετε περαιτέρω το έγγραφό σας.

Σημαντικό: Το IPython notebook που θα παραδώσετε θα πρέπει βεβαιωθείτε ότι ανοίγει και εκτελείται στο Google Colab.

Σε αυτή την εργασία θα ασχοληθήτε με την πρόβλεψη μουσικού είδους από σήματα μουσικής με τη χρήση νευρωνικών δικτύων. Συγκεκριμένα, ο στόχος είναι να ταξινομήσουμε 1 δευτερόλεπτο μουσικού σήματος στα εξής είδη: κλασσική μουσική, ποπ, ροκ, και μπλουζ. Για κάθε 1 δευτερόλεπτο σας παρέχονται δύο ειδών αναπαραστάσεις του ηχητικού σήματος: (i) MFCCs, και (ii) mel-spectograms.

Τα MFCCs είναι συντελεστές του φάσματος ισχύος μετασχηματισμένοι με βάση την κλίμακα mel, μία κλίμακα που είναι κοντά στον τρόπο που αντιλαμβάνεται ο άνθρωπος τα ηχητικά σήματα μέσω της ακοής. Στην δική μας περίπτωση χρησιμοποιούμε 13 συντελεστές οι οποίοι υπολογίζονται για κάθε 50 msec και επομένως για κάθε μουσικό κομμάτι του dataset προκύπτει μία ακολουθία από 20 feature vectors διάστασης 13. Για να αναπαραστήσουμε αυτή την πληροφορία μέσω ενός στατικού διανύσματος, το οποίο είναι ευκολότερο στην χρήση, υπολογίζουμε για κάθε έναν από τους 13 συντελεστές την μέση τιμή και την τυπική

του απόκλιση από την ακολουθία των 20 χρονικών στιγμών. Καταλήγουμε λοιπόν με ένα 1-D διάνυσμα 26 χαρακτηριστικών για κάθε μουσικό κομμάτι του dataset.

Το φασματογράφημα (spectrogram), που είναι ο δεύτερος τρόπος αναπαράστασης που θα χρησιμοποιήσουμε, είναι μία δισδιάστατη αναπαράσταση που δείχνει την χρονική εξέλιξη του φάσματος συχνοτήτων. Εάν στο spectrogram εφαρμόσουμε την κλίμακα mel, παίρνουμε το *mel-spectrogram* ή melgram με το οποίο και θα δουλέψουμε στην παρούσα εργασία. Υπολογίζοντας το mel-spectrogram και αντιστρέφοντας τους άξονες χρόνου και συχνότητας, προκύπτει για κάθε στοιχείο του συνόλου δεδομένων (μουσική καταγραφή) ένας πίνακας 21 (χρόνος) x 128 (συχνότητα).

Τα δεδομένα που θα χρησιμοποιήσετε βρίσκονται εδώ και είναι χωρισμένα στα σύνολα train (3200 δείγματα), validation (800 δείγματα) και test (1376 δείγματα) τα οποία θα χρησιμοποιηθούν για εκπαίδευση, εύρεση υπερπαραμέτρων και αξιολόγηση της ικανότητας γενίκευσης αντίστοιχα.

Ακολουθήστε τις οδηγίες των παρακάτω ερωτημάτων και ετοιμάστε τις απαντήσεις σας τρέχοντας τον κώδικά σας στο Google Colab. Το framework που θα πρέπει να χρησιμοποιηθεί για τον προγραμματισμό των νευρωνικών είναι **είτε το Pytorch ή το Tensorflow**.

[Ερώτημα 1: Feedforward Neural Network] (30 βαθμοί)

Βήμα 1: Φόρτωση δεδομένων (mfccs) (1 μονάδα)

Ξεκινάμε φορτώνοντας τα mfcc δεδομένα για train, validation και test μέσω των αντίστοιχων numpy αρχείων X.npy και labels.npy. Στην συνέχεια μετασχηματίζουμε τα labels από strings (classical, blues etc) σε ακέραιους αριθμούς από 0 μέχρι 3, κρατώντας την αντιστοίχισή τους με τα ονόματα των κλάσεων. Τέλος φορτώνουμε τα δεδομένα μας σε 3 Pytorch dataloaders (ένα για κάθε σύνολο δεδομένων) με batch size 16, ώστε να μπορούν να χρησιμοποιηθούν στα μοντέλα μας. Δώστε επίσης το όρισμα shuffle=True στους train και validation dataloaders.

Βήμα 2: Ορισμός Νευρωνικού Δικτύου (7 μονάδες)

Να ορίσετε μία κλάση πλήρως συνδεδεμένου νευρωνικού δικτύου (fully connected neural network) το οποίο έχει διάσταση εισόδου 26 και αποτελείται από 3 επίπεδα με αριθμούς νευρώνων 128, 32 και 4 αντίστοιχα, όπου 4 είναι ο αριθμός των κλάσεων που θα προβλεφθούν. Επισήμανση: Μην χρησιμοποιήσετε activation functions σε αυτό το στάδιο της εργασίας.

Βήμα 3: Ορισμός διαδικασίας εκπαίδευσης (7 μονάδες)

Να ορίσετε μία συνάρτηση για την εκπαίδευση του δικτύου. Συγκεκριμένα η λειτουργία της συνάρτησης δεδομένου ενός αριθμού εποχών, ενός optimizer, ενός dataloader, μιας συνάρτησης κόστους και ενός νευρωνικού δικτύου είναι η εξής: 1) υπολογίζει την έξοδο του νευρωνικού δεδομένου ενός batch από δείγματα, 2) υπολογίζει και τυπώνει το loss 3) ενημερώνει τα βάρη του δικτύου, και επιστρέφει το νευρωνικό δίκτυο, όταν ολοκληρωθεί η εκπαίδευση.

Βήμα 4: Ορισμός διαδικασίας αξιολόγησης (3 μονάδες)

Να ορίσετε αντίστοιχα μία συνάρτηση αξιολόγησης, η οποία περνάει όλα τα batches ενός dataloader από το μοντέλο υπολογίζοντας τις προβλέψεις του χωρίς να ενημερώνει τα βάρη. Μέσω των προβλέψεων η συνάρτηση να υπολογίζει και να επιστρέφει (i) το loss, (ii) το f1 macro averaged, (iii) το accuracy, και (iv) confusion matrix.

Βήμα 5: Εκπαίδευση δικτύου (3 μονάδες)

Εκπαιδεύστε το νευρωνικό δίκτυο στο training set χρησιμοποιώντας τα εξής:

- optimizer: stochastic gradient descent
- learning rate: 0.002
- loss function: cross-entropy loss
- αριθμός εποχών: 30

Στην συνέχεια χρησιμοποιήστε την συνάρτηση αξιολόγησης του προηγούμενου ερωτήματος για να υπολογίσετε τις επιδόσεις του εκπαιδευμένου μοντέλου στο test set. Τι επιδόσεις πετυχαίνετε;

Βήμα 6: Εκπαίδευση δικτύου με GPU (2 μονάδες)

Να επαναλάβετε το βήμα 5, αλλά αυτή την φορά να έχετε αρχικά μεταφέρει τα δεδομένα και το αρχικοποιημένο νευρωνικό σας δίκτυο στην GPU του colab. Βεβαιωθείτε ότι η εκπαίδευση τρέχει στην GPU και τυπώστε τις διαφορές στους χρόνους εκτέλεσης σε GPU και CPU. Βεβαιωθείτε ότι το colab session σας περιλαμβάνει χρήση GPU - η οποία είναι δωρεάν.

Βήμα 7: Επιλογή μοντέλου (7 μονάδες)

Κατά την διάρκεια εκπαίδευσης (30 εποχές) προκύπτουν διαφορετικά στιγμιότυπα του νευρωνικού μας, δηλαδή μοντέλα που έχουν διαφορετικά βάρη. Κατά την διαδικασία βελτιστοποίησης, δεν γνωρίζουμε ποιο στιγμιότυπο του μοντέλου μας έχει την καλύτερη δυνατότητα γενίκευσης. Για τον λόγο αυτό θα χρησιμοποιήσουμε το validation set στο τέλος

κάθε εποχής ώστε να αξιολογούμε τα στιγμιότυπα του μοντέλου. Αποθηκεύστε το μοντέλο που έχει την καλύτερη επίδοση στην μετρική f1 για το validation set και χρησιμοποιήστε το για να μετρήσετε την απόδοση στο test set. Να σχολιάσετε τα αποτελέσματα.

Σημείωση: Για τα επόμενα βήματα της εργασίας πρέπει να εργαστείτε με τον ίδιο τρόπο χρησιμοποιώντας το validation set για να βρείτε το κατάλληλο στιγμιότυπο

[Ερώτημα 2: Convolutional Neural Network] (30 βαθμοί)

Στο ερώτημα αυτό θα χρησιμοποιήσουμε τα mel-spectrograms σαν εισόδους σε συνελικτικά νευρωνικά δίκτυα (Convolution Neural Networks – CNNs) με σκοπό την ταξινόμηση τους σε μουσικά είδη.

Βήμα 1: Φόρτωση δεδομένων (spectrograms) (2 μονάδες)

Ακολουθήστε την διαδικασία του βήματος 1 στο ερωτήμα 1, αλλά αυτή τη φορά για τα melgrams. Να οπτικοποιήσετε ένα τυχαίο melgram από κάθε κλάση.

Βήμα 2: Ορισμός Νευρωνικού Δικτύου (7 μονάδες)

Να ορίσετε ένα CNN το οποίο αποτελείται από

- Ακολουθία τεσσάρων συνελικτικών επιπέδων, με kernel size 5, ώστε να επιτυγχάνεται η εξής ακολουθία καναλιών: 1, 16, 32, 64, 128
- Η έξοδος του τελευταίου συνελικτικού επιπέδου εισέρχεται σε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο 4 επιπέδων με αριθμό νευρώνων: x (διάσταση εξόδου συνελικτικού δικτύου), 1024, 256, 32, out_dim

Επισήμανση: Μην χρησιμοποιήσετε activation functions σε αυτό το στάδιο της εργασίας.

Βήμα 3: Εκπαίδευση δικτύου (7 μονάδες)

Να αλλάξετε, όπου χρειάζεται, και να εκτελέσετε, την διαδικασία εκπαίδευσης και αξιολόγησης ώστε να μπορεί να εκπαιδευτεί και το νέο νευρωνικό δίκτυο.

Τι παρατηρείτε; Μπορεί να εκπαιδευτεί το δίκτυο;

Δοκιμάστε να το εκπαιδεύσετε τόσο στην CPU όσο και στην GPU. Τι παρατηρείτε για τους χρόνους εκτέλεσης; Υπάρχει διαφορετική αναλογία σε σχέση με την εκπαίδευση του δικτύου του ερωτήματος 1; Να αιτιολογήσετε την απάντησή σας.

Βήμα 4: Pooling and padding (7 μονάδες)

Να τροποποιήσετέ το παραπάνω δίκτυο ώστε να εφαρμόζεται padding 2 στοιχείων στα συνελικτικά επίπεδα τα οποία ακολουθούνται από max pooling με kernel size 2. Να σχολιάσετε την χρησιμότητα των δύο αυτών στοιχείων. Τι επίδοση πετυχαίνετε; Τι παρατηρείτε για τον χρόνο εκπαίδευσης;

Βήμα 5: Activation functions (7 μονάδες)

Το νευρωνικό δίκτυο που έχουμε φτιάξει μέχρι στιγμής εφαρμόζει αποκλειστικά γραμμικούς μετασχηματισμούς στα δεδομένα. Για να μπορέσει το δίκτυο να "μάθει" πιο σύνθετες συσχετίσεις στα δεδομένα θα πρέπει να εισάγουμε μη-γραμμικές συναρτήσεις ενεργοποίησης (non-linear activation functions). Να εφαρμόσετε την συνάρτηση ενεργοποίησης ReLU τοποθετώντας την σε κάθε συνελικτικό επίπεδο (μετά την πράξη της συνέλιξης και πριν το pooling), και σε κάθε είσοδο των γραμμικών επιπέδων. Πως επηρεάζει την απόδοση η αλλαγή αυτή;

[Ερώτημα 3: Improving Performance] (30 βαθμοί)

Σε αυτό το ερώτημα θα προσπαθήσουμε να χρησιμοποιήσουμε τεχνικές και εργαλεία της Βαθιάς Μάθησης για να βελτιώσουμε την επίδοση του CNN.

Βήμα 1: Reproducibility (6 μονάδες)

Για να βελτιώσουμε την απόδοση του δικτύου θα πρέπει να δοκιμάσουμε διάφορες τεχνικές. Για να είμαστε σίγουροι αν μια τεχνική βελτιώνει την απόδοση θα πρέπει να εκπαιδεύουμε το δίκτυο υπό ακριβώς τις ίδιες συνθήκες. Αυτό σημαίνει πως η αρχικοποίηση των βαρών και η σειρά των δεδομένων στα batches θα πρέπει να είναι κάθε φορά ίδια, ώστε η επίδοση του δικτύου να μην εξαρτάται από κάποιο καλύτερο σημείο αρχικοποίησης ή κάποια "ευνοϊκότερη" διαχείριση των δεδομένων.

Για τον λόγο αυτό θα πρέπει να κάνετε "seed" όλες τις απαραίτητες βιβλιοθήκες και αλγορίθμους. Συμβουλευτείτε το ακόλουθο notebook και κάντε τις απαραίτητες αλλαγές στον κώδικά σας. Δοκιμάστε να τρέξετε 2 φορές την ίδια ακριβώς διαδικασία εκπαίδευσης. Θα πρέπει να πετυχαίνετε ακριβώς το ίδιο loss σε κάθε εποχή του train και τις ίδιες επιδόσεις στο test set.

Βήμα 2: Αλγόριθμοι βελτιστοποίησης (8 μονάδες)

Υπάρχουν διαφορετικοί αλγόριθμοι βελτιστοποίησης ενός νευρωνικού δικτύου. Να δοκιμάστε ένα σύνολο από optimizers που αναφέρονται εδώ, και φτιάξτε ένα πινακάκι που

στις στήλες θα περιέχει τους αλγόριθμους και στις γραμμές τις μετρικές accuracy και f1. Τι διαφορές παρατηρείτε στην επίδοση;

Βήμα 3: Batch Normalization (7 μονάδες)

Τα νευρωνικά δίκτυα λειτουργούν καλύτερα εάν τα δεδομένα έχουν συγκεκριμένες στατιστικές ιδιότητες. Για να πετύχουμε αυτές τις ιδιότητες συνήθως πραγματοποιούμε normalization στο input. Παρ' όλα αυτά καθώς εφαρμόζουμε περισσότερα επίπεδα του δικτύου και καθώς ενημερώνονται τα βάρη, ενδέχεται οι στατιστικές ιδιότητες του input κάθε layer να διαφέρουν από το ένα batch στο άλλο. Αυτό δεν βοηθά τον αλγόριθμο εκπαίδευσης αφού προσπαθεί να μάθει από δεδομένα των οποίων αλλάζει λίγο η κατανομή μεταξύ των batches. Μία λύση σε αυτό το πρόβλημα είναι να εφαρμόζουμε, σε κάθε layer, normalization σε όλα τα στοιχεία του batch.

Εισάγετε λοιπόν στην αρχιτεκτονική σας BatchNorm2d layers πριν από κάθε συνάρτηση ενεργοποίησης σε όλα τα συνελικτικά επιπέδα.

Βήμα 4: Regularization (9 μονάδες)

Δοκιμάστε να αμβλύνετε τη διαφορά του train loss απο το validation loss δοκιμάζοντας διαφορετικές τιμές: (i) weight_decay στον optimizer και (ii) dropout στα linear layers. Να αυξήσετε τον αριθμό των εποχών από 30 σε 60 και να δοκιμάστε τα (i) και (ii) μαζί και ξεχωριστά. Τι επίδοση πετυχαίνετε στο test set;

[Ερώτημα 4: Testing] (10 βαθμοί)

Με την αξιολόγηση του ταξινομητή στο test set προσπαθούμε να αποκτήσουμε μια εικόνα της ικανότητας γενίκευσής του σε δεδομένα που δεν έχει χρησιμοποιήσει κατά την εκπαίδευση. Για να δούμε πόσο κοντά είναι αυτή η εικόνα στην πραγματικότητα, θα παράξουμε προβλέψεις για δεδομένα από τον πραγματικό κόσμο (youtube videos).

Βήμα 1: Inference (3 μονάδες)

Εδώ θα χρειαστεί να φτιαχτεί μία συνάρτηση που θα παίρνει ως είσοδο ένα σύνολο δεδομένων (dataloader με shuffle=False) και ένα εκπαιδευμένο CNN και θα επιστρέφει μία λίστα με τις προβλέψεις του μοντέλου.

Βήμα 2: Κατέβασμα μουσικής από το youtube (2 μονάδες)

Το αρχείο youtube_to_melgram.ipynb που σας έχει δοθεί μαζί με τα δεδομένα περιέχει συναρτήσεις οι οποίες, δεδομένου ενός youtube url, κατεβάζουν το ηχητικό αρχείο και

υπολογίζουν μία ακολουθία από mel spectrograms (ένα για κάθε δευτερόλεπτο). Η συνάρτηση youtube_to_melgram αποθηκεύει στο αρχείο melgrams.npy την ακολουθία melgram ενός δεδομένου url. Χρησιμοποιήστε την με τουλάχιστον 1 url απο κάθε μουσικό είδος που περιέχεται στο σύνολο δεδομένων μας. Είστε ελεύθεροι να χρησιμοποιήσετε όποιο url θέλετε.

Σας δίνονται ενδεικτικά:

- κλασσική μουσική:https://www.youtube.com/watch?v=9E6b3swbnWg
- ποπ: https://www.youtube.com/watch?v=EDwb9jOVRtU
- ροκ: https://www.youtube.com/watch?v=OMaycNcPsHI
- μπλουζ: https://www.youtube.com/watch?v=l45f28PzfCI

Βήμα 3: Προβλέψεις (5 μονάδες)

Για κάθε ένα από τα μουσικά είδη χρησιμοποιήστε την συνάρτηση του βήματος 1 ώστε να παράξετε προβλέψεις. Τυπώστε ένα διάγραμμα όπου στον κατακόρυφο άξονα θα βρίσκονται οι μουσικές κλάσεις και στον οριζόντιο τα timestamps (που αντιστοιχούν σε δευτερόλεπτα). Είναι σε αντιστοιχία οι προβλέψεις του δικτύου στο χρόνο με το μουσικό είδος που ανήκει το ηχητικό περιεχόμενο των videos; Εποπτικά σχολιάστε πόσο κοντά είναι η απόδοση του ταξινομητή σας στα youtube videos (σε προβλέψεις τους ενός δευτερολέπτου) σε σχέση με την απόδοση στο test set.