ΠΜΣ-07 Τεχνητή Νοημοσύνη και Εφαρμογές στο Διαδίκτυο των Πραγμάτων
Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

2η Εργασία

Έκδοση 2023-1.0

Ημερομηνία Παράδοσης: 23/5/2023

Διδάσκων: Χρήστος Δίου
Επικουρική Διδασκαλία: Βασίλης Γκολέμης

1 Εισαγωγή

Στην εργασία αυτή θα ασχοληθούμε με την ανάπτυξη μοντέλων Συνελικτικών Νευρωνικών Δικτύων (ΣΝΔ) για την κατηγοριοποίηση (classification) εικόνων. Θα χρησιμοποιήσουμε το σύνολο δεδομένων που παρουσιάστηκε στην εργασία Scene-free multi-class weather classification on single images και περιέχει εικόνες εξωτερικών τοπίων. Στόχος είναι η κατηγοριοποίηση των εικόνων σε 4 κλάσεις (labels) ανάλογα με την κατάσταση του καιρού. Συγκεκριμένα, το dataset περιέχει 1125 εικόνες με τις εξής ετικέτες: (α) cloudy: 300, (β) rain: 215, (γ) shine: 253, (δ) sunrise: 357. Οι φωτογραφίες έχουν συλλεγεί από διάφορες ελεύθερες πλατφόρμες εικόνων (π.χ. Flicker, Picasa, MojiWeather). Το dataset θα μπορούσε να χρησιμοποιήθει για την εκπαίδευση ενός μοντέλου Μηχανικής Μάθησης που θα λειτουργεί στα πλαίσια ενός IoT συστήματος ενός έξυπνου σπιτιού. Στο Figure 1 παρουσιάζουμε ενδεικτικά δύο εικόνες από κάθε κλάση. Το σύνολο δεδομένων μπορείτε να το κατεβάσετε από εδώ. Η εργασία σας θα πρέπει να υλοποιηθεί σε γλώσσα Python, χρησιμοποιώντας τις βιβλιοθήκες Numpy, Pandas, Scikit-learn και Tensorflow. Αν προτιμάτε, μπορείτε να χρησιμοποιήσετε και Pytorch αντί για Tensorflow. Επομένως, θα χρειαστεί να δημιουργήσετε ένα Python περιβάλλον με τις παραπάνω βιβλιοθήκες. Για αυτόν τον σκοπό σας παρέχουμε ένα αρχείο `requirements.txt` με όλα τα απαραίτητα πακέτα.

Στόχος είναι η εξοικείωσή σας με όλα τα βήματα που απαιτεί ένα πρόβλημα πρόβλεψης με ΣΝΔ, δηλαδή (α) φόρτωση και προετοιμασία των δεδομένων, (β) ανάπτυξη μοντέλων πρόβλεψης, (γ) αξιολόγηση των προτεινόμενων λύσεων, καθώς και η συγγραφή αναφοράς για την παρουσίαση των αποτελεσμάτων.

1.1 Παραδοτέα

Θα πρέπει να παραδώσετε ένα αρχείο `<id>.zip`, όπου `<id>` ο ΑΜ σας. Το αρχείο θα περιέχει **είτε** (α) τον κώδικα της υλοποίησής σας (ένα αρχείο .py ή ένα αρχείο .ipynb) και ένα αρχείο PDF με την αναφορά σας **είτε** (β) ένα αρχείο `.ipynb` που θα περιέχει και τον κώδικα της υλοποίησής σας και την αναφορά σας διατυπωμένη σε κατάλληλα Markdown κελιά ανάμεσα στον κώδικα.

1.2 Χρήσιμες Γενικές Συμβουλές

- Αν κολλήσετε κάπου, ρωτήστε
- Ορισμένα από τα δίκτυα χρειάζονται αρκετό χρόνο για να εκπαιδευτούν (τουλάχιστον σε CPU). Αναπτύξτε τον κώδικά σας σε μικρά σύνολα δεδομένων (π.χ. 5-10 εικόνες) ώστε να σιγουρευτείτε ότι όλα λειτουργούν σωστά προτού περάσετε στο πλήρες σύνολο δεδομένων.
- Αξιοποιήστε την υποδομή GPU και TPU των google Colab ή/και Kaggle

(i) cloudy

(ii) cloudy

(iii) shine

(iv) shine

(v) sunrise

(vi) sunrise

(vii) rain

(viii) rain

Σχήμα 1: Παραδείγματα από το σετ εικόνων που θα χρησιμοποιήσουμε

2 Φόρτωση Δεδομένων

2.1 Βήμα 1 - Δημιουργία λίστας με τις εικόνες

Για την φόρτωση των εικόνων σας δίνεται η βοηθητική συνάρτηση:

```
get_dict_with_files_per_class(<directory>)
```

στο script `utils.py`. Περάστε ως όρισμα `directory` τον φάκελο στον οποίον έχετε κάνει `unzip` τις εικόνες και η συνάρτηση θα σας επιστρέψει ένα `Python Dictionary` με τις διευθύνσεις όλων των εικόνων ανα κλάση.

2.2 Βήμα 2 - Αποθήκευση εικόνων με σωστή ιεραρχία

Το περιβάλλον keras έχει ένα σύνολο απο βοηθητικές συναρτήσεις φόρτωσης δεδομένων, εφόσον τα έχετε αποθηκευμένα σε κατάλληλη δομή. Για τον λόγο αυτό δημιουργήστε μια συνάρτηση με όνομα `store_in_keras_structure()` η οποία θα παίρνει ως είσοδο το `Python Dictionary` του Βήματος 1 και θα δημιουργεί ένα directory με όνομα `'data/'`, ακολουθώντας με την παρακάτω δομή:

```
dataset/
---shine/
------shine_image_1.jpg
------shine_image_2.jpg
...
---rain/
------rain_image_1.jpg
------rain_image_2.jpg
...
---sunrise/
------sunrise_image_1.jpg
------sunrise_image_2.jpg
...
---cloudy/
------cloudy_image_1.jpg
------cloudy_image_2.jpg
...
```

2.3 Βήμα 3

Αξιοποίηστε την συνάρτηση `keras.image_dataset_from_directory` για την τελική φόρτωση των δεδομένων.

3 Απλό Συνελικτικό Δίκτυο

Υλοποιήστε συνάρτηση

```
cnn_simple(num_classes)
```

η οποία δέχεται ως όρισμα το πλήθος των κλάσεων του συνόλου δεδομένων και επιστρέφει ένα μοντέλο ΣΝΔ το οποίο αποτελείται από

1. Ένα επίπεδο προεπεξεργασίας που μετασχηματίζει τις τιμές της εικόνας στο $[0, 1]$ από το $[0, 255]$
2. Ένα συνελικτικό επίπεδο με 8 φίλτρα 3×3, με έξοδο ίση με την είσοδο και συνάρτηση ενεργοποίησης `ReLU`
3. Ένα επίπεδο συγκέντρωσης (Max pooling) με βήμα 2

4. Ένα συνελικτικό επίπεδο με 16 φίλτρα 3×3, με έξοδο ίση με την είσοδο και συνάρτηση ενεργοποίησης `ReLU`
5. Ένα επίπεδο συγκέντρωσης (Max pooling) με βήμα 2
6. Ένα επίπεδο μετατροπής σε 1 διάσταση (`Flatten`)
7. Ένα πλήρως συνδεδεμένο επίπεδο με 32 νευρώνες και συνάρτηση ενεργοποίησης `ReLU`
8. Ένα επίπεδο με `num_classes` εξόδους και συνάρτηση ενεργοποίησης `softmax`

3.1 Εκπαίδευση

Χρησιμοποιήστε το 60% ως σύνολο εκπαίδευσης, το 20% ως σύνολο επικύρωσης και το υπόλοιπο 20% ως σύνολο δοκιμής. Δημιουργήστε ένα μοντέλο ΣΝΔ καλώντας τη συνάρτηση `cnn_simple` και εκπαιδεύστε το χρησιμοποιώντας

- Τον αλγόριθμο βελτιστοποίησης Adam με ρυθμό εκμάθησης 10^3 , $\beta_1 = 0.9$ (ρυθμός ενημέρωσης πρώτης ροπής) $\beta_2 = 0.99$ (ρυθμός ενημέρωσης δεύτερης ροπής)
- Την κατηγορική διεντροπία ως συνάρτηση απώλειας
- Την ορθότητα (accuracy) ως μετρική αξιολόγησης
- `batch_size: 64`
- 20 εποχές μέγιστη διάρκεια εκπαίδευσης
- Πρόωρο τερματισμό της εκπαίδευσης (Early Stopping) αν δεν παρουσιαστεί μείωση της απώλειας στο σύνολο επικύρωσης για 5 συνεχείς εποχές

3.2 Αξιολόγηση

Ποια είναι η ορθότητα (accuracy) του μοντέλου σας στο σύνολο εκπαίδευσης, στο σύνολο επικύρωσης και στο σύνολο δοκιμής; Θα χρειαστεί να δημιουργήσετε μια συνάρτηση:

```
confusion_matrix(model, subset)
```

όπου `subset` παίρνει τις τιμές `'train'` , `'val'` , `'test'` και επιστρέφει τον πίνακα σύγχυσης (confusion matrix) για το αντίστοιχο σύνολο. Με βάση τον πίνακα σύγχυσης υπολογίστε (a) τις γενικές (σε όλο το dataset) τιμές accuracy, precision, recall, καθώς και (b) τις τιμές precision, recall, f1 score ανα κλάση. Σχολιάστε την επίδοση του μοντέλου σας.

4 Συνελικτικό δίκτυο μεγάλου βάθους

Στόχος αυτού του βήματος είναι να εξετάσετε την επίδραση του βάθους ενός συνελικτικού νευρωνικού δικτύου στην επίδοση. Για τον σκοπό αυτό θα δημιουργήσετε ένα ΣΝΔ ίδιας αρχιτεκτονικής με το προηγούμενο βήμα, με περισσότερα όμως κρυφά επίπεδα. Συγκεκριμένα, το ΣΝΔ θα αποτελείται από:

1. Ένα επίπεδο προεπεξεργασίας που μετασχηματίζει τις τιμές της εικόνας στο $[0, 1]$ από το $[0, 255]$
2. Τρία διαδοχικά συνελικτικά επίπεδο με 32 φίλτρα 3×3, με έξοδο ίση με την είσοδο και συνάρτηση ενεργοποίησης `ReLU`
3. Ένα επίπεδο συγκέντρωσης (Max Pooling) με βήμα 4
4. Τρία διαδοχικά συνελικτικά επίπεδα με 64 φίλτρα 3×3, με έξοδο ίση με την είσοδο και συνάρτηση ενεργοποίησης `ReLU`
5. Ένα επίπεδο συγκέντρωσης (Max Pooling) με βήμα 2

6. Τρία διαδοχικά συνελικτικά επίπεδα με 128 φίλτρα 3×3, με έξοδο ίση με την είσοδο και συνάρτηση ενεργοποίησης `ReLU`
7. Ένα επίπεδο συγκέντρωσης (Max Pooling) με βήμα 2
8. Ένα επίπεδο μετατροπής σε 1 διάσταση (`Flatten`)
9. Ένα πλήρως συνδεδεμένο επίπεδο με 128 νευρώνες και συνάρτηση ενεργοποίησης `ReLU`
10. Ένα επίπεδο με `num_classes` εξόδους και συνάρτηση ενεργοποίησης `softmax`

Εκπαιδεύστε και αξιολογήστε το μοντέλο όπως και στο προηγούμενο βήμα. Εντοπίστε πιθανές διαφορές στην επίδοση και σχολιάστε σε τι θεωρείτε ότι οφείλονται.

5 Προεκπαιδευμένο Νευρωνικό Δίκτυο

Σε πολλές περιπτώσεις όπου το σύνολο δεδομένων δεν είναι ιδιαίτερα μεγάλο (σχολιάστε: θα χαρακτηρίζατε το σετ δεδομένων μας με 1125 εικόνες μεγάλο;), είναι εξαιρετικά χρήσιμο να αξιοποιούμε ένα προεκπαιδευμένο ΣΝΔ και να το προσαρμόζουμε (finetuning) στα δικά μας δεδομένα. Η τεχνική αυτή είναι γνωστή ως transfer learning. Η βιβλιοθήκη Keras προσφέρει ένα σύνολο από γνωστά προεκπαιδευμένα δίκτυα, τα οπόια μπορείτε να βρείτε εδώ. Επιλέξτε όποιο θέλετε (μπορείτε να δοκιμάσετε και παραπάνω από ενα) και προσαρμόστε στο το στα δεδομένα μας. Κατά την προσαρμογή (finetuning), μπορείτε είτε να 'σταματήσετε' (freeze) την εκπαίδευση όλων βαρών πέραν του τελευταίου layer και να προσθέστε ένα δικό σας τελικό layer, ή να εκπαιδεύσετε όλα τα βάρη του δικτύου (συχνά είναι προτιμότερο αυτό να γίνεται με κάποιο μικρό learning rate). Εκπαιδεύστε και αξιολογήστε το μοντέλο όπως και στα προηγούμενα βήματα (δοκιμάστε διαφορετικές εναλλακτικές). Εντοπίστε πιθανές διαφορές στην επίδοση και σχολιάστε σε τι θεωρείτε ότι οφείλονται.

Ως παράδειγμα, για τη χρησιμοποίηση του προεκπαιδευμένου δικτύου Inception μπορείτε να χρησιμοποιήσετε την

```
keras.applications.inception_v3.InceptionV3(weights='imagenet', include_top=False)
```

6 Δοκιμή δικών σας εικόνων

Στο βήμα αυτό θα εξετάσουμε αν τα ΣΝΔ των προηγούμενων βημάτων έχουν την αναμενόμενη επίδοση σε εικόνες λίγο διαφορετικές από αυτές που εκπαιδεύτηκαν. Το παραπάνω ερώτημα, αν δηλαδή η επίδοση ενός δικτύου γενικεύεται (generalization) και σε εικόνες διαφορετικής προέλευσης είναι πολύ καίριο για την υιοθέτηση αυτών των μοντέλων σε πραγματικές εφαρμογές. Για τον λόγο αυτό σας ζητάμε να βγάλετε με το κινητό σας (ή οποια συσκευή επιθυμείτε) 10 φωτογραφίες και να τις δοκιμάσετε στα παραπάνω μοντέλα. Προσπαθήστε οι φωτογραφίες να μην είναι όλες την ίδια μέρα και υπό τις ίδιες καιρικές συνθήκες. Σχολιάστε τα αποτελέσματα. Θα εμπιστευόσασταν τα μοντέλα που εκπαιδεύσατε για μια εφαρμογή αυτόματης πρόβλεψης του καιρού;