Κ23γ: Ανάπτυξη λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2019-20

3η Εργασία: Πρόγνωση της έντασης του ανέμου με προεκπαιδευμένο βαθύ νευρωνικό δίκτυο στη γλώσσα Python (3.7) με την χρήση του Keras API επί της πλατφόρμας μηχανικής μάθησης Tensorflow

Ιωάννης Μήτρου	Μαρία Ξανθοπούλου
AM : 1115201400108	AM : 1115201400300

Από τα ζητούμενα υλοποιήθηκαν όλα, με παραλλαγή και μικρές ελλείψεις στο τελευταίο ζητούμενο με το clustering, καθώς το πρώτο μέρος της προηγούμενης εργασίας ήταν ελλιπές για να μπορέσουμε να βγάλουμε συμπεράσματα!

Προαπαιτούμενα για την εκτέλεση των προγραμμάτων:

Keras (με backend Tensorflow), Python3.7 και pandas (καθώς και όλα τα άλλα που αναφέρονται στις διαφάνειες)

1. Αρχεία που παραδίδονται και Εκτέλεση

Μέσα στο παραδοτέο έχουμε 5 φακέλους : `Prediction`, `New_representations` `Clusters`, `Input` και έναν φάκελο με προγράμματα της 2ης εργασίας.

Το αρχείο δεδομένων <`nn_representations.csv`> βρίσκεται στο φάκελο `Input`. Το `actual.csv` και το N2 είναι ξανά σε κάθε φάκελο που τα χρειάζεται.

- Ο πρώτος φάκελος (<`Prediction`>) περιέχει τα απαραίτητα αρχεία δεδομένων για το Α ερώτημα και το αρχείο κώδικα `predict.py`.

Εκτέλεση : `python3.7 predict.py -i ../Input/nn_representations.csv`

Έξοδος: Ένα αρχείο `output.txt` που αποθηκεύεται στο φάκελο.

- Ο δεύτερος φάκελος (<`New_representations`>) περιέχει τα απαραίτητα αρχεία για το Β ερώτημα. Εδώ έχουμε 2 αρχεία που μπορούμε να εκτελέσουμε (ουσιαστικά παράγουν το ίδιο αρχείο, αλλά με 2 διαφορετικούς τρόπους). Περισσότερες λεπτομέρειες για αυτό παρακάτω.

Εκτέλεση: `python3.7 new_representations.py -i ../Input/nn_representations.csv`

β' τροπος: `python3.7 new_reps_proper.py -i ../Input/nn_representations.csv`

Έξοδος: Το αρχείο `new_representations.csv`

- Ο τρίτος φάκελος (`Clusters`) περιέχει τα απαραίτητα αρχεία για το Γ ερώτημα:

```
<python3.7 clusters.py -i ../Input/nn_representations.csv –n
        <numberofclusters> >
```

Εκτελεί εσωτερικά τον kmeans με python!!

```
<python3.7 cluster_forC.py -i
         ../new_representations/new_representations.csv>
```

(αφού έχουμε φτιάξει το new_representations.csv)

Φτιάχνει τα δεδομένα στην κατάλληλη μορφή για να τα διαβάσει η προηγούμενη εργασία!

2. <u>Αλγόριθμοι και Ροή των προγραμμάτων</u>

Α) Στο `predict.py` τρέχουμε το pretrained Νευρωνικό Δίκτυο και βρίσκουμε κάποια στατιστικά. Συγκεκριμένα, (αφού ελέγξουμε για σωστά arguments), διαβάζουμε το csv μέσω της εντολής `pd.read_csv()` της pandas χρησιμοποιώντας ως arguments <`header = None και index_col = 0`>, καθώς δεν έχουμε header και θέλουμε να χρησιμοποιήσουμε τις ημερομηνίες ως index. Έχοντας έτσι το `dataframe`, καλούμε την `predict` και βρίσκουμε μετά τα στατιστικά συγκρίνοντας την έξοδο με τα δεδεομένα του `actual.csv`. Τα στατιστικά MAE και MSE υπολογίζονται μέσω της συνάρτησης `evaluate` αφού κληθεί πρώτα η συνάρτηση `compile` και περαστούν ως παράμετροι. Επειδή, όμως, το MAPE είχε πρόβλημα με αυτόν τον τρόπο, το υπολογίζουμε ξεχωριστά διατρέχοντας τα αρχεία και βρίσκοντας το μέσο κάθε γραμμής (ώστε να μην διαιρούμε ποτέ με το 0).

Β) <u>α' Τρόπος</u>: Το βασικό σε αυτόν τον τρόπο είναι η κλήση της συνάρτησης του Keras `Model()`, όπου φτιάχνει ένα υπομοντέλο. Έτσι, παίρνουμε έτοιμο το output από το πρώτο hidden layer και μετά κολλάμε τα timestamps και αυτό θα είναι το αρχείο εξόδου.

<u>β' Τρόπος</u>: Αυτός ο τρόπος (που ζητήθηκε στο e-class) φτιάχνει ένα δεύτερο νευρωνικό δίκτυο με input και ένα επιπλέον στρώμα. Παίρνουμε τα βάρη του pretrained NN με την `get_weights` και με την `set_weights` τα βάζουμε στο καινούριο που θα φτιάξουμε, το οποίο μετά απλά το τρέχουμε. Επιτυγχάνει το ίδιο αποτέλεσμα με τον α' Τρόπο και ολόιδια αρχεία (κάτι που εξετάστηκε με την diff στο terminal).

Γ) Στον φάκελο για το Γ ερώτημα, υπάρχει καταρχάς το πρόγραμμα <cluster_forC.py>, το οποίο "φτιάχνει" το new_representations.csv σε μορφή κατάλληλη για να διαβαστεί από το πρόγραμμα της προηγούμενης εργασίας που είχαμε παραδώσει. Το πρόβλημα με αυτό, είναι, πως δεν ήταν ολοκληρωμένο το πρώτο σκέλος της $2^{ης}$ εργασίας, καθώς μας έλειπε το silhouette (αντ' αυτού, στη $2^{η}$ εργασία χρησιμοποιήσαμε ένα objective function που μας έδειχνε την ποιότητα της συσταδοποίησης, το οποίο όμως στη παρούσα περίπτωση δεν είναι κατάλληλο) και, επιπλέον, είναι χρονοβόρο στην εκτέλεση. Δεδομένων αυτών και του ότι δεν είναι σωστό να βγάλουμε συμπεράσματα βασισμένοι σε μία ημιτελή εργασία, μαζί με αυτό το πρόγραμμα, αναπτύχθηκαν (και υπάρχουν στο φάκελο) προγραμμα (clusters.py) clustering σε python. Η βασική συνάρτηση στο πρόγραμμα για το cluster είναι η kmeans και χρησιμοποιήθηκε με μετρική 'cityblock' (εναλλακτική διατύπωση της Manhattan). Για σύγκριση, έγιναν αντίστοιχα πειράματα clustering σε MATLAB, πάνω στα ίδια δεδομένα, με

τον ίδιο αλγόριθμο (kmeans) και την ίδια μετρική (cityblock). Τα συμπεράσματα αυτής της έρευνας συνοψίζονται παρακάτω.

Επειδή ο τελικός στόχος είναι η συσταδοποίηση των διανυσμάτων ανέμου σε 4 ή 12 clusters για χρονική συσχέτιση με εποχές και μήνες, υπολογίστηκαν και τα μέσα διανύσματα ανά ημέρα (day averages), στα οποία επίσης εφαρμόστηκε συσταδοποίηση. Τα αποτελέσματα που πήραμε από τα day averages είναι παρεμφερή με τα πρώτα, όμως τώρα υπολογίζονται πολύ ταχύτερα (στο 1/20 του χρόνου!)

3. Αποτελέσματα και Διερεύνηση

Παρακάτω παρουσιάζονται πίνακες αποτελεσμάτων και γραφήματα από την εκτέλεση των προγραμμάτων. Εξάγονται επίσης κάποια συμπεράσματα.

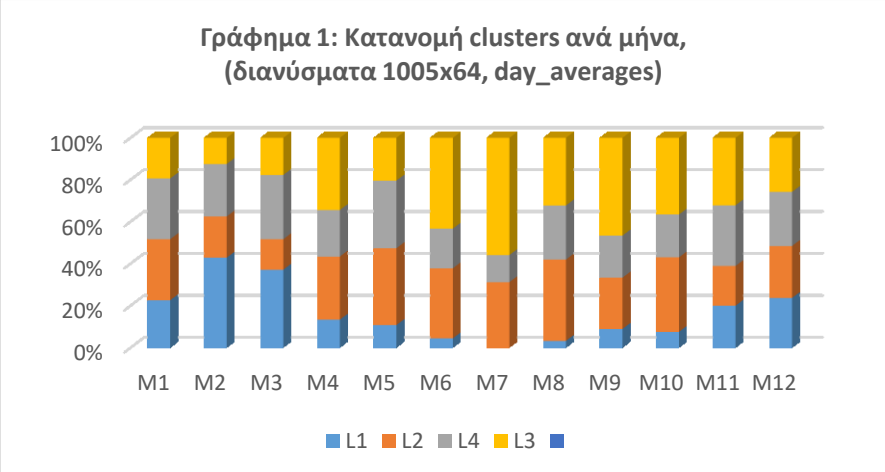
Πίνακας 1: Μέσο silhouette συστάδων (scikit.cluster kmeans, Manhattan dist.)

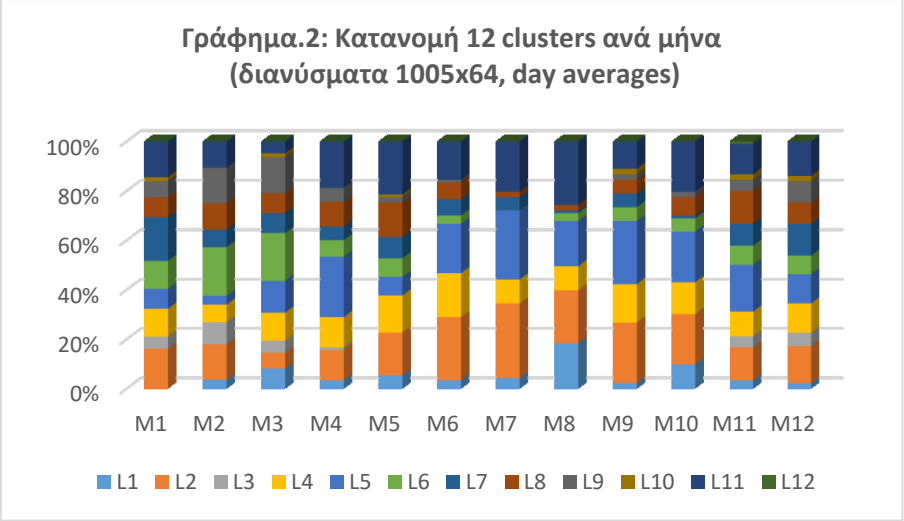
# clusters	Διανύσματα εισόδου (dim 128)	Διανύσματα ενδιάμεσου layer (dim 64)	Διανύσματα ενδιάμεσου layer (dim 64), με day averages
4	0.2629	0.4415	0.4485
12	0.2007	0.2496	0.2532

Πίνακας 2: Μέσο silhouette συστάδων (MATLAB kmeans, Manhattan dist.)

# clusters	Διανύσματα εισόδου (dim 128)	Διανύσματα ενδιάμεσου layer (dim 64)	Διανύσματα ενδιάμεσου layer (dim 64), με day averages
4	0.3293	0.5564	0.5923
12	0.2648	0.3396	0.3633

Τα παρακάτω γραφήματα δίνουν την κατανομή των clusters (υπολογισμένων σε διανύσματα day averages) ανά μήνα.





Συμπεράσματα

1. Παρατηρούμε ότι γίνεται πολύ καλύτερη συσταδοποίηση στα διανύσματα του ενδιάμεσου layer (διάστασης 64) σε σχέση με τα διανύσματα εισόδου (στο N2), διάστασης 128.
2. Η συσταδοποίηση είναι καλύτερη για μικρότερο αριθμό clusters (4, έναντι του 12).
3. Η συσταδοποίηση είναι ελαφρώς καλύτερη στα διανύσματα day averages (τους ημερήσιους μέσους όρους δηλαδή)
4. **Από τα γραφήματα μπορούμε να κάνουμε και μια χρονική συσχέτιση των clusters με μήνες και εποχές**. Αυτή η συσχέτιση είναι πιο καθαρή στην περίπτωση της συσταδοποίησης σε 4 clusters (Γράφημα 1): **οι 'μπλε' και 'γκρι' άνεμοι είναι κυρίως χειμωνιάτικοι (ειδικά οι 'μπλέ'), ενώ οι 'πορτοκαλί' και οι 'κίτρινοι' κυρίως καλοκαιρινοί (ειδικά οι 'κίτρινοι').**