Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2021-22

3η Προγραμματιστική Εργασία

Πρόγνωση τιμών μετοχών με χρήση αναδρομικού νευρωνικού δικτύου LSTM. Ανίχνευση ανωμαλιών με χρήση νευρωνικού δικτύου αυτοκωδικοποίησης LSTM ανά μετοχή. Αναζήτηση και συσταδοποίηση μετοχών βάσει της αναπαράστασης των χρονοσειρών που προκύπτει από συνελικτική αυτοκωδικοποίηση. Σύγκριση αποτελεσμάτων αναζήτησης και συσταδοποίησης με την αρχική αναπαράσταση. Για την υλοποίηση του νευρωνικού δικτύου θα χρησιμοποιηθεί η γλώσσα Python 3 και η προγραμματιστική διεπαφή Keras επί της πλατφόρμας νευρωνικών δικτύων TensorFlow.

Η εργασία πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο την Παρασκευή 14/1/2022 στις 23.59.

Περιγραφή της εργασίας

Α) Κατασκευάστε αναδρομικό νευρωνικό δίκτυο LSTM πολλαπλών στρωμάτων. Θα πραγματοποιήσετε πειράματα εκπαίδευσης του δικτύου με διαφορετικές τιμές υπερπαραμέτρων [αριθμού στρωμάτων, μεγέθους στρωμάτων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size)], ώστε να ελαχιστοποιήσετε το σφάλμα (loss). Θα πρέπει να αποφευχθεί η υπερπροσαρμογή (overfitting) μέσω της χρήσης dropout στρωμάτων. Η μέθοδος παραμετροποιείται επίσης ως προς τον προσδιορισμό της εισόδου (τιμές υστέρησης) και της εξόδου του παράγωγου προβλήματος μηχανικής μάθησης με επίβλεψη. Τα δεδομένα του συνόλου εισόδου χωρίζονται κατάλληλα σε σύνολο εκπαίδευσης (training set) και σε σύνολο ελέγχου (test set). Βάσει των πειραμάτων, επιλέγετε τη βέλτιστη δομή για το νευρωνικό δίκτυο και τη βέλτιστη μοντελοποίηση του προβλήματος πρόγνωσης. Το πρόγραμμα οπτικοποιεί τα αποτελέσματα της πρόγνωσης μέσω γραφικής παράστασης στην οποία απεικονίζεται τόσο η εξέλιξη των πραγματικών τιμών όσο και η εξέλιξη των τιμών βάσει της πρόγνωσης.

Β) Κατασκευάστε αναδρομικό νευρωνικό δίκτυο LSTM αυτοκωδικοποίησης χρονοσειρών το οποίο θα περιλαμβάνει στρώματα κωδικοποίησης και αποκωδικοποίησης. Θα πρέπει να πραγματοποιήσετε πειράματα εκπαίδευσης του δικτύου με διαφορετικές τιμές υπερπαραμέτρων [αριθμού στρωμάτων, μεγέθους στρωμάτων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size), στρώματα dropout] ώστε να ελαχιστοποιήσετε το σφάλμα (loss) αποφεύγοντας την υπερπροσαρμογή (overfitting). Τα δεδομένα του συνόλου εισόδου πρέπει να χωριστούν κατάλληλα σε σύνολο εκπαίδευσης (training set) και σε σύνολο ελέγχου (test set). Βάσει των πειραμάτων, επιλέγετε τη βέλτιστη δομή για το νευρωνικό δίκτυο. Επιλέγετε διαφορετικά κατώφλια **μέσου απόλυτου σφάλματος** (που προκύπτει από τη διαφορά των πραγματικών τιμών των μετοχών από τις τιμές που προέκυψαν από την αυτοκωδικοποίηση) για την ανίχνευση ανωμαλιών. Οι ανωμαλίες που ανιχνεύτηκαν επισημαίνονται κατάλληλα στην γραφική παράσταση που απεικονίζει την εξέλιξη των πραγματικών τιμών των μετοχών.

Γ) Κατασκευάστε συνελικτικό νευρωνικό δίκτυο αυτοκωδικοποίησης χρονοσειρών το οποίο θα περιλαμβάνει στρώματα συμπίεσης και αποσυμπίεσης (“bottleneck”). Θα πραγματοποιήσετε πειράματα εκπαίδευσης του δικτύου με διαφορετικές τιμές υπερπαραμέτρων [αριθμού συνελικτικών στρωμάτων,

μεγέθους συνελικτικών φίλτρων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size), διάστασης συμπίεσης (latent dimension, default=3)] ώστε να ελαχιστοποιήσετε το σφάλμα (loss) αποφεύγοντας την υπερπροσαρμογή (overfitting). Τα δεδομένα του συνόλου εισόδου πρέπει να χωριστούν κατάλληλα σε σύνολο εκπαίδευσης (training set) και σε σύνολο ελέγχου (test set). Βάσει των πειραμάτων, επιλέγετε τη βέλτιστη δομή για το νευρωνικό δίκτυο, και το διάνυσμα συμπίεσης (latent vector) χρησιμοποιείται για τη μείωση της πολυπλοκότητας των χρονοσειρών. Παράμετρος της μείωσης της πολυπλοκότητας είναι το «παράθυρο» των χρονικών στιγμών που δίνεται ως είσοδος στο νευρωνικό δίκτυο (default = 10). Χρησιμοποιήστε το εκπαιδευμένο νευρωνικό δίκτυο για την παραγωγή των νέων αναπαραστάσεων του συνόλου μετοχών που παρέχεται και αποθηκεύστε σε tab-separated αρχείο.

Δ) Χρησιμοποιήστε το παραδοτέο της 2^{ης} εργασίας και παράγετε όλα τα αποτελέσματα για τις χρονοσειρές σύμφωνα με την νέα αναπαράσταση. Συγκρίνετε με τα αποτελέσματα πριν τη μείωση της πολυπλοκότητας.

Τα πειράματα και τα αποτελέσματα των ερωτημάτων Α έως Δ περιγράφονται και σχολιάζονται αναλυτικά στην αναφορά που παραδίδεται.

ΕΚΤΕΛΕΣΗ

Το αρχείο που δίνεται στην είσοδο έχει την ακόλουθη μορφή:

```
item_id1    X11   X12   ...   X1d
.     .     .     ...   .
item_idN    XN1   XN2   ...   XNd
```

Α)

Το αρχείο εισόδου δίνεται μέσω παραμέτρου στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:

```
$python forecast.py –d <dataset> -n <number of time series selected>
```

Το πρόγραμμα `forecast.py` χρησιμοποιεί το μοντέλο που έχει επιλεχθεί και εκπαιδευτεί βάσει των πειραμάτων για την επιλογή των βέλτιστων (υπερ)παραμέτρων. Το μοντέλο εκπαιδεύεται τόσο ανά χρονοσειρά όσο και ανά σύνολο χρονοσειρών και παράγονται οι αντίστοιχες γραφικές παραστάσεις. Τα αποτελέσματα των αντίστοιχων προγνώσεων συγκρίνονται και σχολιάζονται.

Β)

Το αρχείο εισόδου δίνεται μέσω παραμέτρου στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:

```
$python detect.py –d <dataset> -n <number of time series selected> –mae <error value as double>
```

Το πρόγραμμα `detect.py` χρησιμοποιεί το μοντέλο που έχει επιλεχθεί και εκπαιδευτεί βάσει των

πειραμάτων για την επιλογή των βέλτιστων (υπερ)παραμέτρων. Το μοντέλο εκπαιδεύεται βάσει του συνόλου των χρονοσειρών. Παράγονται οι γραφικές παραστάσεις στις οποίες επισημαίνονται οι ανωμαλίες.

Γ)

Το αρχείο δίνεται μέσω παραμέτρου στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:

```
$python reduce.py –d <dataset> -q <queryset> -od <output_dataset_file> -oq
<output_query_file>
```

Το πρόγραμμα `reduce.py` χρησιμοποιεί το μοντέλο που έχει επιλεχθεί και εκπαιδευτεί βάσει των πειραμάτων για την επιλογή των βέλτιστων (υπερ)παραμέτρων. Το μοντέλο εκπαιδεύεται βάσει του συνόλου των χρονοσειρών.

Τα αρχεία εξόδου έχουν την ίδια μορφή με τα αρχεία εισόδου.

Δ) Όπως και στη $2^{η}$ εργασία. Χρησιμοποιούνται τα αρχεία εισόδου και αναζήτησης που παρήχθησαν στο ερώτημα Γ.

Επιπρόσθετες απαιτήσεις

1. Αρχείο (ή ενότητα στο Readme) που να σχολιάζει τα αποτελέσματα.
2. Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη αρχείου Readme το οποίο περιλαμβάνει κατ' ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα και περιγραφή τους, γ) οδηγίες χρήσης του προγράμματος και δ) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.
3. Η υλοποίηση του προγράμματος θα πρέπει να γίνει με τη χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας (Git ή SVN) [ομάδες 2 ατόμων].