Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

3η Προγραμματιστική Εργασία

Εισαγωγή

Στην εργασία αυτή υλοποιήθηκε ο κώδικας για το Α και το Β ερώτημα. Έπειτα πειραματιστήκαμε με τις υπερπαραμέτρους και συγκρίναμε τα αποτελέσματα τα οποία προέκυψαν. Η μελέτη αυτή βρίσκεται στο αρχείο Μελέτη στο οποίο περιγράφουμε ακριβώς σε ποιες συγκρίσεις έχουμε μεταβεί. Λόγω περιορισμένου χρόνου, τα μοντέλα τα έχουμε εκπαιδεύσει μόνο ανά χρονοσειρά, μιας και θεωρήσαμε πιο σημαντικό να ασχοληθούμε με τον πειραματισμό και του Β ερωτήματος.

Στοιχεία Φοιτητών

Η εργασία αυτή υλοποιήθηκε από τους:

Θεόδωρος Κωνσταντόπουλος – 201600266

Ταράτσας Ιωάννης – 201700160

Github: https://github.com/GiannisTrt/projectALAP/tree/Project3/Project3

Αρχεία

Τα αρχεία τα οποία παραδίδονται είναι το forecast.py το detect.py και το research.pdf. Μέσα στο πρώτο αρχείο(forecast.py), υπάρχει ο κώδικας του Α, στο δεύτερο αρχείο(detect.py), υπάρχει ο κώδικας του Β, και στο αρχείο research.pdf η έρευνα που έχουμε πραγματοποιήσει. Παρακάτω θα αναφερθούν οι συναρτήσεις που έχουν χρησιμοποιηθεί σε κάθε αρχείο κώδικα:

Στο αρχείο *forecast.py:*

Def create_model: Η συνάρτηση αυτή δημιουργεί το μοντέλο LSTM πολλαπλών στρωμάτων και το επιστρέφει στην main

Def parse_args: Η συνάρτηση αυτή διαβάζει τα ορίσματα από το command line και τα επιστρέφει στην main

Def plot_results: Είναι μια συνάρτηση η οποία εκτυπώνει σε γραφική παράσταση τα αποτελέσματα που θέλουμε.

Def main: Η συνάρτηση main αφού διαβάζει τα ορίσματα από το command line, διαβάζει το αρχείο, και το χωρίζει σε train_set(80%) και test_set(20%). Στην συνέχεια , ομαλοποιούμε τα δεδομένα πριν την τοποθέτηση του μοντέλου με την χρήση του Min-Max Scaler για την ενίσχυση της απόδοσης. Αφού αναδιαμορφώνουμε τα δεδομένα, φτιάχνουμε το μοντέλο. Σε αυτό το σημείο έχουμε πειραματιστεί και με διάφορες τιμές στρωμάτων, αλλά και νευρώνων. Επίσης, έχουμε πειραματιστεί και ανάμεσα στους δύο optimizers Adam και

SGD(Θα αναλυθούν τα αποτελέσματα στο research.pdf). Σημαντικό θα ήταν να αναφέρουμε πως κάνουμε χρήση ενός validation_set=0.3, και πως σταματάμε το μοντέλο αφού παρατηρήσουμε ότι μετά από κάποιες επαναλήψεις σταματά να εκπαιδεύεται, με EarlyStopping, αν δεν υπάρχει βελτίωση του validation_loss . Τέλος, προετοιμάζουμε και τα δεδομένα δοκιμής, έτσι ώστε να έχουν την ίδια μορφή με τα δεδομένα με τα οποία εκπαιδεύτηκε το μοντέλο, κάνουμε τις προβλέψεις και απεικονίζουμε τα αποτελέσματα.

Στο αρχείο *detect.py:*

Αρχικά στο αρχείο detect.py αρχικά χωρίζουμε τα δεδομένα σε train και σε test. Αφού κάνουμε κανονικοποίηση με την χρήση του StandarScaler, για να ενισχύσουμε την απόδοση, δημιουργούμε ακολουθίες των 30 data. Έπειτα κατασκευάζουμε το μοντέλο LSTM Autoencoder και εκπαιδεύουμε το μοντέλο. Στην συνέχεια, πρέπει να προσδιορίσουμε και τις ανωμαλίες, και αυτό το υλοποιούμε με MAE. Βρίσκουμε την μέγιστη τιμή απώλειας, την αποθηκεύουμε και πλέον αν η απώλεια για ένα σημείο στα δεδομένα που μαντεύουμε στο test_set, ξεπερνάει αυτήν την τιμή, τότε επισημάνουμε το σημείο αυτό ως ανωμαλία.

Στο αρχείο *research.pdf:*

Περιγράφονται οι μετρήσεις και τα αποτελέσματα που έχουμε δει

Οδηγίες για την χρήση

Τα προγράμματα εκτελούνται ακριβώς με τον ίδιο τρόπο που περιγράφεται στην εκφώνηση. Η μόνη διαφορά είναι πως στο detect.py δεν βάζουμε με το χέρι το meo αλλά αφήνουμε το πρόγραμμα να το υπολογίσει. Συγκεκριμένα:

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

Το detect.py: python detect.py –d <dataset> -n number of time series selected>