Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Τεχνολογίες Γραφημάτων
Εργασία

Διδάσκων: Δημήτρης Μιχαήλ

2024-2025

Σκοπός της εργασίας αυτής είναι η εξοικείωση σας με αλγορίθμους υπολογισμού ενσωματώσεων κόμβων σε γραφήματα. Ως παράδειγμα θα χρησιμοποιήσουμε τον αλγόριθμο DeepWalk [2], ο οποίος χρησιμοποιεί τυχαίους περιπάτους (random walks) για να προσεγγίσει την πιθανότητα $P(u|v)$ να δούμε έναν κόμβο u ξεκινώντας από έναν κόμβο v. Μέσω της διαδικασίας των τυχαίων περιπάτων παράγει ένα σύνολο δεδομένων που στην συνέχεια χρησιμοποεί για να προπονήσει ένα μοντέλο Skip-Gram. Στην έκδοση που θα υλοποιήσουμε στην εργασία αυτή θα χρησιμοποιήσουμε αντί για το Skip-Gram μοντέλο, το CBOW μοντέλο.

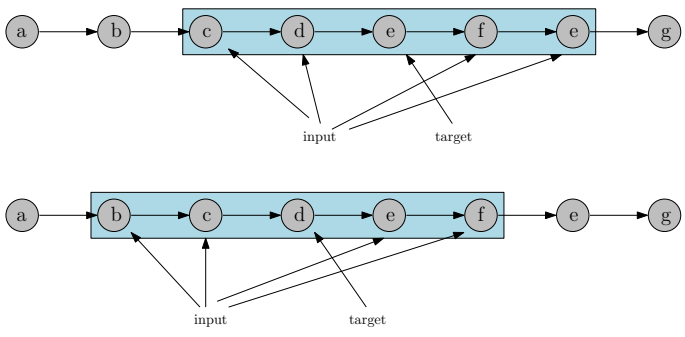
1 Τυχαία Μονοπάτια

Στο πρώτο μέρος της εργασίας καλείστε να μελετήσετε την δημοσιεύση που περιγράφει την τεχνική DeepWalk και να χρησιμοποιήσετε την βιβλιοθήκη NetworkX για να δημιουργήσετε μικρούς τυχαίους περιπάτους. Γράψτε ένα πρόγραμμα σε NetworkX που να διαβάζει ένα μη κατευθυνόμενο γράφημα και να παράγει μικρά τυχαία μονοπάτια. Φροντίστε να δέχεται το πρόγραμμα σας (π.χ από την γραμμή εντολών) τις κατάλληλες παραμέτρους. Από κάθε κόμβο του γραφήματος, εκκινήστε γ τυχαίους περιπάτους σταθερού μήκους t. Για ευκολία και για συμβατότητα με το επόμενο κομμάτι της άσκησης, χρησιμοποιήστε μια αναπαράσταση των κόμβων με ακέραιες τιμές.

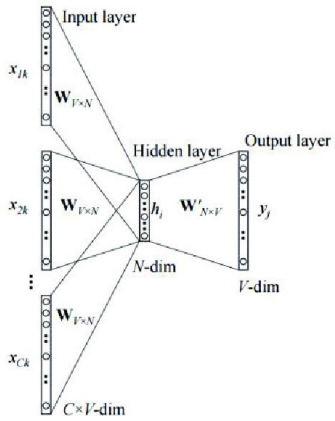
2 Υλοποίηση CBOW μοντέλου (Continous bag of words)

Στο δεύτερο μέρος της εργασίας καλείστε να υλοποιήσετε το CBOW [1] μοντέλο με την χρήση της βιβλιοθήκης pytorch και να εκπαιδεύσετε το μοντέλο με την χρήση των μονοπατιών που παράξατε στο πρώτο μέρος.

Στο μοντέλο CBOW χρησιμοποιούμε μία παράμετρο w που περιγράφει το μέγεθος του παραθύρου που κοιτάμε. Αυτό το παράθυρο κινείται κατά μήκος ενός τυχαίου μονοπατιού που έχουμε παράξει στο πρώτο μέρος της άσκησης. Για κάθε θέση του παραθύρου αυτού παράγουμε ένα δείγμα προπόνησης. Το δείγμα προπόνησης περιέχει ως είσοδο τους πρώτους $\lfloor \frac{w-1}{2} \rfloor$ και τους τελευταίους $\lfloor \frac{w-1}{2} \rfloor$ κόμβους ενώ ως label περιέχει τον ενδιάμεσο κόμβο.



Ένα παράδειγμα του CBOW μοντέλου φαίνεται παρακάτω. Ως είσοδο δέχεται τους κόμβους (1-hot vectors) και μέσω ενός embedding lookup υπολογίζει την κρυφή αναπαράσταση. Στην συνέχεια υπολογίζουμε τον μέσο όρο ή άθροισμα των κρυφών αναπαραστάσεων και μετά μέσω μίας softmax (ή hierarchical softmax σε περίπτωση μεγάλης εισόδου) βγάζουμε την έξοδο.



Το μοντέλο είναι ρηχό και οι ενσωματώσεις περιέχονται απευθείας στον πίνακα παραμέτρων W. Η εξαγωγή των ενσωματώσεων γίνεται με τον πολλαπλασιασμό με το 1-hot vector (embedding lookup). Για την υλοποίηση του μοντέλου με το Pytorch μελετήστε τις κλάσεις

- https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html
- https://pytorch.org/docs/stable/generated/torch.nn.Linear.html
- https://pytorch.org/docs/stable/generated/torch.nn.Softmax.html

3 Χρήση της Μεθόδου

Χρησιμοποιήστε το γράφημα Zachary Karate Club από τη βιβλιοθήκη NetworkX ως παράδειγμα. Υπολογίστε τις ενσωματώσεις και τρέξτε την μέθοδο t-SNE για να τα οπτικοποιήσετε. Τι παρατηρείτε; Δώστε παραδείγματα εφαρμογών όπου η μέθοδος αυτή μπορεί να χρησιμοποιηθεί. Παρουσιάστε την δουλειά σας με λεπτομέρεια στην αναφορά. Τρέξτε την υλοποίηση σας και δείξτε τα αποτελέσματα.

4 Παραδοτέα

Η άσκηση έχει ένα παραδοτέο που αποτελείται από 3 μέρη:

1. Ο πηγαίος κώδικας ο οποίος θα πρέπει να είναι γραμμένος σε python και να εκτελείται πολύ εύκολα. Μπορείτε να χρησιμοποιήσετε αν θέλετε και jupiter notebook. Δώστε και ένα requirements.txt με τις βιβλιοθήκες που χρειάζονται.
2. Μαζί με τον πηγαίο κώδικα θα πρέπει να υπάρχει και ένα αρχείο README.md το οποίο να περιγράφει αναλυτικά πως τρέχει.
3. Τέλος θα πρέπει να υπάρχει και ένα αρχείο report.pdf το οποίο να περιγράφει αναλυτικά την δουλειά σας, να εξηγεί τον κώδικα σας, και να περιέχει παραδείγματα εκτέλεσης του κώδικα σας.

Προσοχή η βαθμολόγηση δεν γίνεται μόνο με βάση την λειτουργικότητα αλλά και με βάση την ποιότητα του κώδικα. Επιπρόσθετα σημαντικό ρόλο παίζει η αναλυτική αναφορά.

References

[1] Tomas Mikolov. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 3781, 2013.

[2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.