Article Title

Frederik Valdemar Schrøder, Jens Petur Tróndarson, Mathias Møller Lybech

Aalborg University

fschra16@student.aau.dk jtrand16@student.aau.dk mlybec16@student.aau.dk

October 15, 2020

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

I. Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit [2].

II. Related work

III. Basic Theory

i. Graph Convolutional Network

Graph Convolutional Network (GCN) is a neural network architecture that operates on graphs. Given a graph $G = (V, E)$ a GCN takes the input of a adjacency matrix A with size of NxN that represents graph G and a feature matrix NxF^0, where N is the total amount of nodes and F^0 is the total amount of input features for each node.

A hidden layer in a GCN can be defined as $H^i = f(H^{i-1}, A)$ where i indicates the layer and H^0 is the previously mentioned NxF^0 feature matrix and f is a propagation function [1]. There are many different types of propagation functions. A simple example could be $f(H^i, A) = \sigma(AH^iW^i) = H^{i+1}$ where W^i is the weight matrix at layer i and σ is a non-linear activation function [1]. The intuition behind this propagation function is that the future representation of each node is calculated based on its neighbors nodes. Each time i is increased, the nodes will reach one further edge away from the original node in the graph. An issue with this propagation function could be that the value of each node now is a sum of each of its neighbors, and therefore loses its own value. This could be solved by replacing A with $\hat{A} = A + I$ where I is the degree matrix. Doing this the node considers itself a neighbor.

IV. Method

V. Discussion

VI. Conclusion

References

[1] Tobias Skovgaard Jepsen. *How to do Deep Learning on Graphs with Graph Convolutional Networks*. 2018. URL: `https://towardsdatascience.com/how-to-do-deep-learning-on-graphs-`

with-graph-convolutional-networks-7d2250723780 (visited on 10/14/2020).

[2] Rex Ying et al. "Graph Convolutional Neural Networks for Web-Scale Recommender Systems". In: KDD '18. 2018.