GEOM-GCN: GEOMETRIC GRAPH CONVOLUTIONAL NETWORKS

Term Project

2020-2022

Submitted to: Dr. Ali Ahmed

Submitted by:

Aiman Younas- MSDS20087

Qazi Danish Ayub -MSDS20075

Muhammad Mubashir Ali -MSDS20085

Hira Saleem - MSDS20102

Mahnoor Imran -MSDS20088

Introduction

Message passing neural networks such as Graphical convolution network, Gated graph neural network, and graph neural network are used for learning different applications based on graphs. the general working of the MPNNS is defined as the node sends features as a message to the nodes in its neighborhood and then update its features by accumulating the features it receives as a message. Although MPNNs is already successfully implemented in different scenarios it has its own some fundamental weakness that this paper addresses. The goal of this research is to offer a new GCN framework that addresses the shortcomings of existing GCN techniques, especially the loss of structural neighbor information and the failure to preserve important dependencies amongst remote nodes.

Methodology

The study uses the concept of structural neighborhood in which it defines the neighbors of a node in a graph and latent space through embedding methods.
space mapping to create a second kind of neighborhood in latent space: comparing the distance between nodes with the parameter, where the parameter's magnitude rises from 0 to the number of neighborhoods in graph space. For each node v, the researchers used a two-stage convolution approach.

1) The nodes with the same relation with v are combined for each neighborhood type;

2) The resulting nodes are combined again into a new feature vector. This method helps you to get around the concerns mentioned above. The results reveal that the approach outperforms previous GCN solutions in the vast majority of circumstances, often by a significant margin.

Objective Functions

$$e_{\mathrm{(i,r)}}^{\mathrm{v,l+1}} = p(\{h_u^l | u \in N_i(v), \tau(z_v , z_u) = r\}) , \forall_i \in \{g, s\}, \forall_r \in R$$

$$m_v^{l+1} = q_{i \in \{g,s\}, r \in R} ((e_{\mathrm{(i,r)}}^{\mathrm{v,l+1}} , (i, r)))$$

$$h_v^{l+1} = \sigma(W_l . m_v^{l+1})$$

Optimizer

Adam optimizer

Experiments

Experminets on Optimizers:

In this section, we have changed the optimizers to produce good results. After Adam, one of the best optimizers is Stochastic gradient descent due to scalable issues on collab we have tested on one dataset that is Cora. The specification of this dataset is as follows, Nodes:2708, Edges:5429, Features:1433, and Classes:7. For constructing state-of-the-art deep learning models, an optimizer that trains as fast as Adam and as efficiently as SGD is given as Adabound in this paper [1]. We have also used this optimizer to generate the results for our Cora dataset.

Result on SGD Optimizer:
Gave very bad validation accuracy.

Result on Adabound Optimizer:
Adabound: Gave very good validatioon accuracy on the 10 splits of dataset. It gave around 75 % mean accuracy on testset.

GEOM_GCN Model:

As far as this model is concerned, we had difficulty putting it into practice. And, after reviewing certain official complaints about the code, we got to the conclusion that the offered code had some flaws that must be addressed. We attempted to resolve some of the difficulties, but there were still some that needed to be addressed.
To compensate, we read the article and ran a dry run of the code to have a better understanding of the problem and to identify concerns that can assist in the improvement of this model.

1) Smart way of determining a node's radius in Latent space:

Radius is, in fact, a critical hyper-parameter for Geom-GCN. The radius defines the proximity of that node, if the radius is small then the neighborhood in latent space becomes too small, and relevant information cannot be gathered effectively. When the radius increases too much the proximity of that node increases, too much unwanted information becomes part of the relevant information. We recommend that they should run some radius sensibility testing on each dataset to select a good radius.

2) Effective ways for selecting Node Embeddings:

We can see that some variations with neighborhoods defined by only one embedded space perform better than the original Geom-GCN. On the other side, there are a lot of versions that perform poorly. That is, the effectiveness of Geom-GCN would be determined by the embedded spaces chosen.
As a result, we believe that developing an end-to-end system that is capable of automatically determining the appropriate embedding spaces for Geom-GCN is critical for future work.

Reference

[1] Adaptive Gradient Methods with Dynamic Bound of Learning Rate -Liangchen Luo, Yuanhao Xiong, Yan Liu, Xu Sun- ICLR 2019 Conference