TOPIC 17: GNN OPTIMIZATION

Qian Wang
wang432@usc.edu

Yi Deng
dengyi@usc.edu

Muhao Guo
muhaoguo@usc.edu

1 ABSTRACT

In this report, we briefly introduce what we have done for our GNN project by now. We first describe our problem, then the project timeline, which includes the papers we have read, the lecture nodes we have reviewed and the algorithms we have implemented, is listed. Also, the algorithms we used are performed through pseudo-code and analyzed with flow chart. In the end, we sketched out our implementation and discussed the challenges we may come across.

2 PROBLEM DESCRIPTION

Graph study is trending now due to the ability of representing the real world such as social networks, bioinformatics, molecular structures, circuit elements etc. And nowadays GNNs has become a powerful and popular tool for machine learning on graphs. The purpose of this project is to implement GNNs/GCNs based on PyTorch to solve problems in social medias. Then solve another circuit-based problem in CAD by applying GNN.

3 PROJECT TIMELINE

Papers we read

- Semi-supervised Classification with Graph Convolutional Networks *by Thomas N. Kipf and Max Welling*
- Graph Convolutional Neural Networks for Web-scale Recommender Systems *by Rex Ying, Ruining He, Kaifeng Chen, etc.*
- DeepWalk: Online Learning of Social Representations *by Bryan Perozzi, Rami Al-Rfou and Steven Skiena*
- node2vec: Scalable Feature Learning for Networks *by Aditya Grover and Jure Leskovec*
- skipgram: Distributed Representations of Words and Phrases *by Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean*

Lecture Nodes we have reviewed

- Stanford CS224W: Graph Neural Networks
- Stanford CS224W: Graph Neural Networks: Hands-on Session

- Stanford CS224W: Message Passing and Node Classification
- Stanford CS224W: Machine Learning with Graphs (Video)

Expect to do

- Keep updating the Github repo we created for GNN research.
- Complete the GNN Google Colab tutorial, and apply GNN model into our project.
- Explore the data flow to our model, and try to see the components of each data batch.
- Make a specific report of our datasets, including number of nodes, edges, feature size, classe number, and show how they fed to our model.
- Visualize the embeddings of our datasets using t-SNE feature reduction, with or without the Stanford template.

4 ANALYSIS

Algorithm

- The graph convolutional operator(GCN) from the "Semi-supervised Classification with Graph Convolutional Networks" paper
  A multi-layer Graph Convolutional Network (GCN) with the following layer-wise propagation rule:
  $H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)})$

  $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $W^{(l)}$ is a layer-specific trainable weight matrix. $\sigma(\cdot)$denotes an activation function, such as the $ReLU(\cdot) = max(0, \cdot)$. $H^{(l)} \in R^{N \times D}$ is the matrix of activations in the l^{th} layer; $H^{(0)} = X$.

5 IMPLEMENTATION

The implementations are based on PyTorch.We used a GCN model to solve node classification problem.

Data

dataset	num of edges	num of nodes	size of node features	num of num_classes
cora	126842	19793	8710	70
cora_ml	16316	2995	2879	7
citeseer	10674	4230	602	6
dblp	105734	17716	1639	4
pubmed	88648	19717	500	3

Figure 1: Datasets of GNNstack

Structure of GNNstack Model

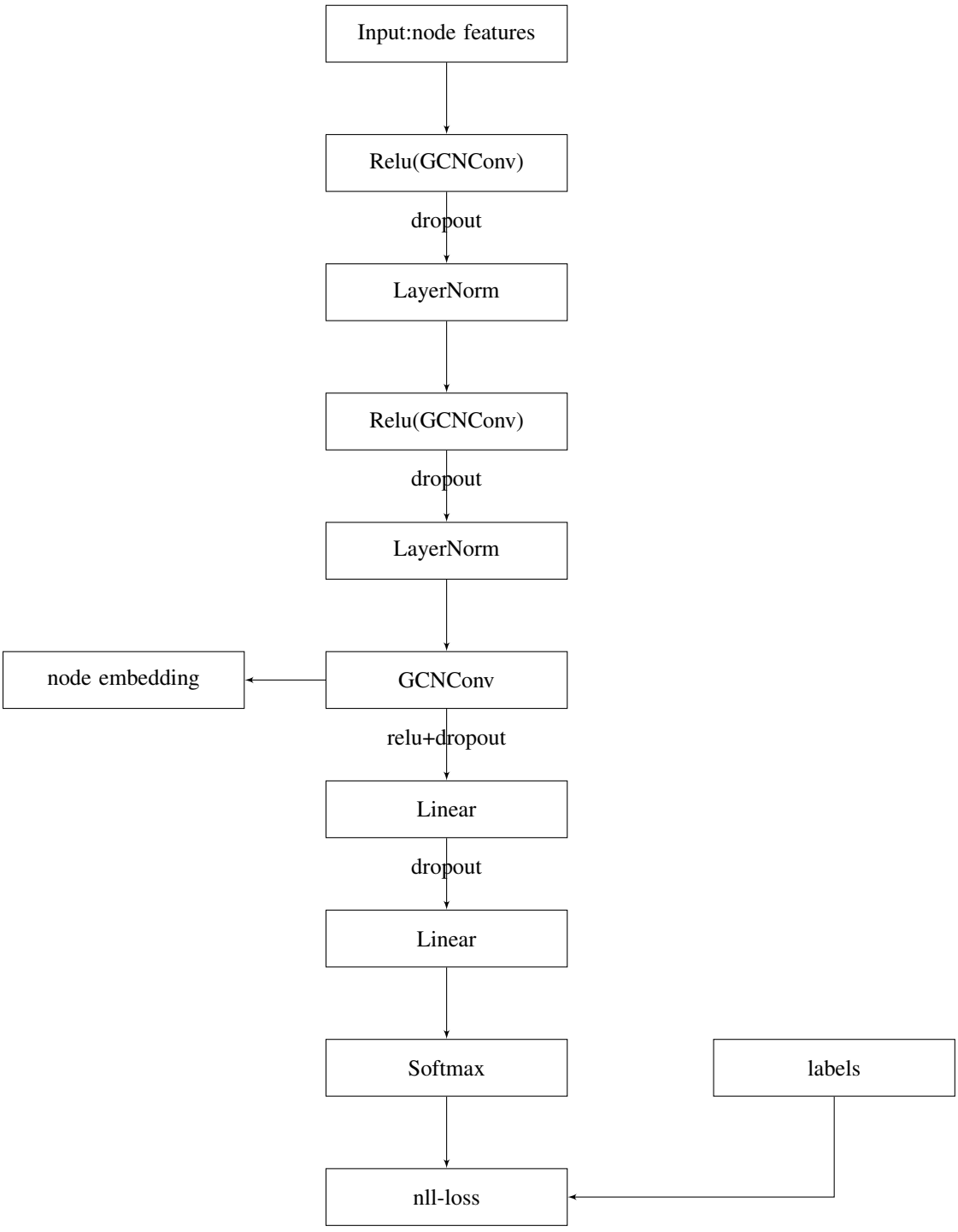

Figure 2: GNNstack model structure

-

Results:

dataset	cora	cora-ml	citeseer	dblp	pubmed
accuracy	0.8480	0.9790	0.9960	0.9434	0.9328

6 TOOL (SOFTWARE) CHALLENGES

- We mainly used PyTorch .
- Besides PyTorch, we will use Python and we work on Github, Linux and Overleaf.
- We create a virtual environment of Python.

7 PROPOSAL

Introduction
Most of the current deep learning processing data is limited to images and text, but it does not include the network.This is because the network is the connection information between nodes and it is difficult to convert this kind of information to standard data. How to solve this problem? If we can convert the network to a geometric space, in other words, assignment to each node a coordinates in a geometric space, then we can use the general machine learning methods to solve this kind of problem. So we use DeepWalk or Node2vec algorithms to give every node in the network a vector representation. And for further specific task like node classification, we can also use GCN algorihtm to train the model.

prior work summary

- We have learned how to use Pytorch.
- We have read the lecture notes from Stanford University about Machine Learning with Graphs.
- Implemented node2vec and deepwalk in pytorch.
- We have get some extra data sets (like BolgCatalog and Homo Sapiens etc.) and implement the node2vec data sets.
- Implemented GCN in pytorch and trained GNNstack model for node classification.

the key novelties of your work

- We have implement the whole GNN algorithm on some data sets
- Use data sets that are closely related to life.
- Perform a new machine learning method on the network.