README

CodeAppendix

Code for paper "EGAT: Edge-Featured Graph Attention Network"

DOI:https://doi.org/10.1007/978-3-030-86362-3_21

Dependencies

in requirements.txt

- python 3.7
- pytorch 1.5.0
- pytorch-lightning 0.7.3
- torch-geometric 1.4.3
- numpy 1.18.1
- scipy 1.5.1
- wandb 0.9.4 (please run `wandb off` to disable the sync to wandb server)
- torch_scatter 2.1.1
- torch_sparse 0.6.17

File Structure

- data: Directory for dataset.(This file needs to be created by yourself and placed in the `EGAT-refine-main` directory, parallel to files such as `model` and `trainer`.)
- model:
    - node.py: The node module of EGAT.
    - edge.py: The edge module of EGAT.
    - mgcn.py: The edge and node modules of MGCN, including the EGAT_MGCN (AttentionVertexModule)
    - nnconv.py: The node module of NNConv, including the EGAT_NNConv (AttentionNNConv)
    - net.py: The network structure of EGAT, for both AMLSim (AMLSimNet) and citation networks (Cora, Citeseer and PubMed) (CitationNet). The structure of CitationNet is hard coded.

    - container_abcs.py: This file contains Python’ s built-in abstract base classes for collections.
- trainer: The training process (see: pytorch-lightning) of AMLSim and citation networks.
- transforms: The transformers of dataset.
- dataset.py: Some of the preprocessing of AMLSim and all the preprocessing of citation networks.
- main.py: The entry file.
- config.yml: Hyperparameter config file.

Usage

Dataset Prepare

Please copy all datasets to the `data` directory.(available at this url)

Hyperparameters

You can control the hyperparameter in `config.yml` . where the meaning of each hyperparameter is commented .

- net: amlsim *# amlsim, citation; Indicate which network structure to use*
- lr: 0.005 *# learning rate*
- leaky_slope: 0.2 *# the leakage coefficient for the Leaky ReLU activation function*
- dropout: 0.3 *# the dropout ratio, used to reduce overfitting*
- vertex_feature_ratio: 0.5 *# the aggregation method for node features. (greater than 0.5, uses more concatenation of node features; less than 0.5, uses more averaging of node features)*
- vertex_type: mgcn *# egat, egat_merge, egat_split, mgcn, mgcn_att, nnconv, nnconv_att; The type of node modules (only work for amlsim)*
- edge_type: mlp *# mlp, egat, mgcn; The type of edge modules (only work for amlsim)*
- edge_order: after *# before, after, parallel; The order inside EGAT layer (only work for amlsim)*
- dataset: AMLSim-10K-merge-hard-batch *# AMLSim-10K-merge-hard-batch, cora, citeseer, pubmed; The dataset to use*
- l2norm: 0.0005 *# l2 normalization*
- vertex_feature: 128 *# dimension of node features*
- edge_feature: 128 *# dimension of edge features*
- layers: 6 *# layer number (only work for amlsim)*
- predict_hidden: true *# whether to use a hidden layer for prediction (only work for amlsim)*

- batchnorm_order: post *# pre, post, none; The position of batchnorm*
- update_method: residual *# residual, gru, none; The updating method of node and edge features (only work for amlsim)*
- layer_aggregation_method: concat *# last, concat; The merge layer (only work for amlsim)*
- heads: 8 *# heads of EGAT in the multi-head attention mechanism*
- symmetric: true *# whether to use symmetric strategy (whether the attention scores for edges between two nodes are the same)*
- seed: 1170 *# random seed for dataset (only work for citation networks)*

Train

Run `python main.py` to train the model. The results are reported in the terminal.

Attention

- If using CPU-only for training, you need to remove the suffix ".cuda" from some code.
- If you encounter an error: *ImportError: cannot import the name 'container_abcs' from 'torch._six'*, the solution is to click on the error link, navigate to a corresponding Python file in the torch library, and modify the import statement.

```
from torch._six import string_classes,container_abcs  # original import statement
import collections.abc as int_classes,container_abcs  # modified import statement
```