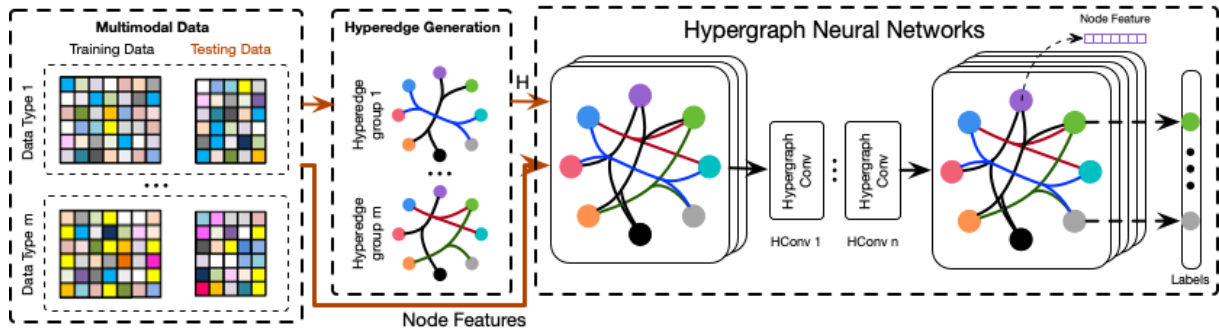

News

We have release a deep learning toolbox named DHG for graph neural networks and hypergraph neural networks. You can find many intresting things in it. Many correlation structures like simple graph, directed graph, bipartite graph, and simple hypergraph are all supported in the toolbox, as well as their visualization. More details refer to DHG!

Hypergraph Neural Networks

Created by Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong, Ji, Yue Gao from Xiamen University and Tsinghua University.



Introduction

This work will appear in AAAI 2019. We proposed a novel framework(HGNN) for data representation learning, which could take multi-modal data and exhibit superior performance gain compared with single modal or graph-based multi-modal methods. You can also check our paper for a deeper introduction.

HGNN could encode high-order data correlation in a hypergraph structure. Confronting the challenges of learning representation for complex data in real practice, we propose to incorporate such data structure in a hypergraph, which is more flexible on data modeling, especially when dealing with complex data. In this method, a hyperedge convolution operation is designed to handle the data correlation during representation learning. In this way, traditional hypergraph learning procedure can be conducted using hyperedge convolution operations efficiently. HGNN is able to learn the hidden layer representation considering the high-order data structure, which is a general framework considering the complex data correlations.

In this repository, we release code and data for train a Hypergrpah Nerual Networks for node classification on ModelNet40 dataset and NTU2012 dataset. The visual objects' feature is extracted by MVCNN(Su et al.) and GVCNN(Feng et al.).

Citation

if you find our work useful in your research, please consider citing:

```
1 @article{feng2018hypergraph,
2   title={Hypergraph Neural Networks},
3   author={Feng, Yifan and You, Haoxuan and Zhang, Zizhao and Ji,
4     Rongrong and Gao, Yue},
5   journal={AAAI 2019},
6   year={2018}
7 }
```

Installation

Install Pytorch 0.4.0. You also need to install yaml. The code has been tested with Python 3.6, Pytorch 0.4.0 and CUDA 9.0 on Ubuntu 16.04.

Usage

Firstly, you should download the feature files of modelnet40 and ntu2012 datasets. Then, configure the “data_root” and “result_root” path in config/config.yaml.

Download datasets for training/evaluation (should be placed under “data_root”) - ModelNet40_mvcnn_gvcnn_feature - NTU2012_mvcnn_gvcnn_feature

To train and evaluate HGNN for node classification:

```
1 python train.py
```

You can select the feature that contribute to construct hypergraph incidence matrix by changing the status of parameters “use_mvcnn_feature_for_structure” and “use_gvcnn_feature_for_structure” in config.yaml file. Similarly, changing the status of parameter “use_gvcnn_feature” and “use_mvcnn_feature” can control the feature HGNN feed, and both true will concatenate the mvcnn feature and gvcnn feature as the node feature in HGNN.

```
1 # config/config.yaml
2 use_mvcnn_feature_for_structure: True
3 use_gvcnn_feature_for_structure: True
4 use_mvcnn_feature: False
5 use_gvcnn_feature: True
```

To change the experimental dataset (ModelNet40 or NTU2012)

```
1 # config/config.yaml
```

```
2 #Model
3 on_dataset: &o_d ModelNet40
4 #on_dataset: &o_d NTU2012
```

License

Our code is released under MIT License (see LICENSE file for details).