Submission and Formatting Instructions for International Conference on Machine Learning (ICML 2022)

Anonymous Authors[1]

Abstract

1. Introduction

2. Related works

3. Proposed method

In this section the proposed method will be explained. Our new method have differences in architecture compare to Graph-Unets that are mentioned bellow:

3.1. Pooling layer

~~The main novelty of this paper is changes proposed in this layer.~~ ~~We propose two different pooling layers.~~ In previous work(Graph U-nets) the main idea for this layer was finding top k important nodes by their features and that method doesn't consider the structure of graph. Our proposed layer consider both features of each nodes and structure of nodes in graph to choose top k important nodes. To move towards achieving this goal we use two approaches, first using centralities of node and second generalization spectral sampling method. Now we will describe both of these methods:

3.1.1. Centrality based

For choosing most important nodes of graph just from its structure, one easy way is calculating each node centrality and choose top k of them, but in different graphs and different tasks needs different centralities to be useful and efficient. So for achieving this goal and using different centralities we propose this method. ~~In this method~~ we consider the importance of each node with linear composition of centralities. To explain it more accurate first we augment some important and easy calculating centralities to form a matrix that we call it "centrality matrix" then this matrix multiply with learnable vector in neural network. This means the neural network depending to the task and graph learns vector that its each components show how much that centrality is important. Finally the result of multiplying centrality matrix and this vector will be the vector of importance of each node according to graph structure. we add these structure scores to feature scores comes from Graph-Unets method and choose top k nodes from these scores.

Algorithm 1 Centrality Based

```
Input: Graph G = (V, E), size m
Initialize S = 0_{|V|×m}
for i = 1 to |V| do
    for j = 1 to m do
        S_ij = j-th centrality for i-th node
    end for
end for
```

3.1.2. General spectral sampling

According to paper ... they proposed a method for sampling nodes with spectral decomposition. ~~to summarize their method~~, they find smoothest signal and choose node with most energy for a sample of whole graph. The key part of our generalization is come from the fact that least eigenvalues are very near to each other so if we can use m-least of them maybe it leads to get a better result. At first we explain how the sampling method works, after spectral decomposition of Laplacian matrix and finding least eigenvalues and corresponding eigenvector of p-th power of the Laplacian matrix, we should replace each components of eigenvectors to absolute of it and then the index of biggest value will be the index of node we should sample from graph. To generalize the method of sampling we can calculate m-least eigenvalues and eigenvectors and after finding absolute values of each components of eigenvectors we can augment them to form a matrix that we call it "spectral matrix" and similarly to cen-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Algorithm 2 Generalization of Sampling

Input: Adjacency Matrix A, size m, power p
$L = NormalizedLaplacian(A^p)$
$U\Sigma V = SVD(L)$
$idx = argsort(\lambda)$
$idx = least(idx, m)$
$S = V[:, idx]$
$S = absolute(S)$

Algorithm 3 Pooling

Input: Features' Matrix X^l, Adjacency Matrix A^l, Structure Matrix S^l, Vector p, Vector q, Number k
$y_{features} = X^l p / \|p\|$
$y_{structure} = S^l q / \|q\|$
$y = y_{features} + y_{structures}$
$idx = rank(y, k)$
$\tilde{y} = sigmoid(y[idx])$
$\tilde{X}^{l+1} = X^l[idx, :]$
$X^{l+1} = \tilde{X}^{l+1} \odot (\tilde{y} 1^T)$
$A^{l+1} = A^l[idx, idx]$

trality based method we learn a vector. the way of how to choosing top k nodes is the same as before. In algorithm 2 λ is eigenvalues vector and absolute function returns matrix of absolute values of each components.

3.2. Graph convolution layer

It's well-known that GIN(Graph Isomorphism Network) is more powerful than GraphSage and GCN(Graph Convolutional Network) so it would be reasonable to replace GCN with GIN.

3.3. Unpooling layer

The main difference in this layer is we initialize features of each new nodes with weighted average of its neighbors' features.

3.4. Loss function

The loss function of these method is added with regularization term to increase accuracy in some datasets. As a result of using this regularization term is finding more smooth decision boundary. the proposed regularization term defines bellow:

$$R = \frac{1}{L} \sum_{i=1}^{L} \frac{\|W^i\|_F}{\#W^i components}, W^i \in \mathbb{R}^{m^i \times n^i} \quad (1)$$

$$R = \frac{1}{L} \sum_{i=1}^{L} \frac{\|W^i\|_F}{m^i \times n^i} \quad (2)$$

which W^i is transform matrix of i-th layer in dense network for classification.
In order to make clear why we choose this regularization format, we should tell that the common way for regularization would be $R = \frac{1}{L} \sum_{i=1}^{L} \|W^i\|_F$ but the layers dimensions are not similar to each other. to normalize these norms of each layer we divide each of them by number of their components. In other word we choose different regularization coefficient for different layers.

Algorithm 4 Unpooling

$X^{l+1} = distribute(0_{N \times C}, X^l, idx)$
$I = V \backslash idx$
$\forall\, i \in I : X_i^{l+1} = \frac{\sum_{j \in V} A_{ij} X_i^l}{\sum_{j \in V} A_{ij}}$

4. Experiments

5. Conclusion

Acknowledgements

Do not include acknowledgements in the initial version of the paper submitted for blind review.

If a paper is accepted, the final camera-ready version can (and probably should) include acknowledgements. In this case, please place such acknowledgements in an unnumbered section at the end of the paper. Typically, this will include thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

A. You can have an appendix here.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one, even using the one-column format.