Image classification model based on GAT

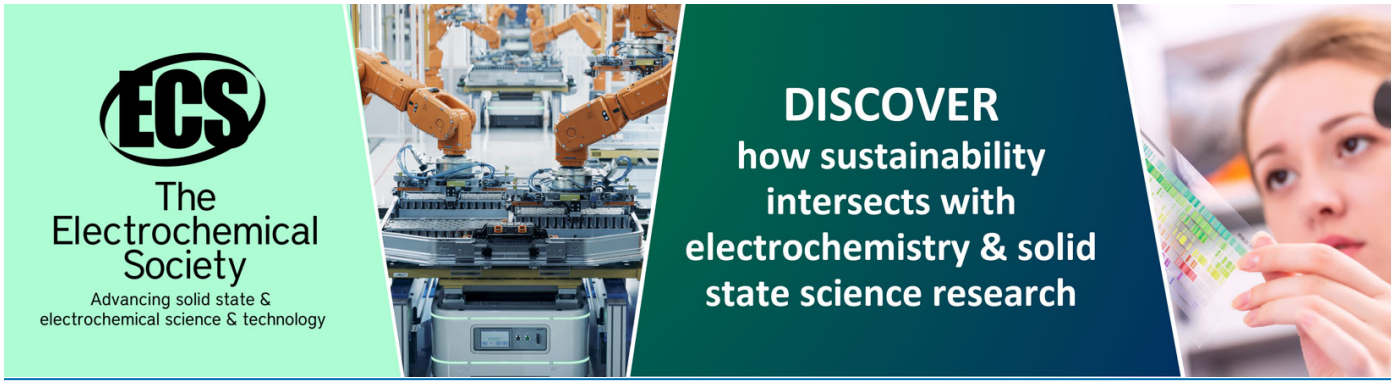

Image classification model based on GAT

Binghui Xu[1*], Sizhe Ding[2] and Yan Zhang[3]

[1] Taizhou Vocational & Technical College, Taizhou, Zhejiang, 318000, China

[2] School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei, 430072, China

[3] International college of Zhengzhou university, Zhengzhou, Henan, 450000, China

[*]Corresponding author's e-mail: 0708chengji@163.com

Abstract. In the field of image classification, graph neural network (GNN) is a kind of structured data modeling architecture with larger functions. However, there are still some problems, such as low efficiency of updating nodes, fixed network parameters and the inability to effectively model the information features of some edges in the graph. In order to solve these problems, this paper introduces attention mechanism on the basis of GNN to improve it, proposes a graph attention network (GAT), establishes a double-layer GAT model, and uses regularization method in model iterative training to achieve image classification. The model is applied to three datasets for experiments. The experimental results show that the average classification accuracy of the proposed model is high and it has good application performance.

1. Introduction

Graph is a kind of data structure. A graph can be defined as a tuple. Research shows that machine learning method has been paid more and more attention to graph structure data, because graph structure can represent many real world data, including protein composition, image analysis, scene description, software engineering, natural language processing, natural science and social interaction on social networks. As a unique non Euclidean data structure, graph research focuses on node classification, link prediction and graph classification. Compared with the vector data, the discrete and semi-structured characteristics of the graph, as well as the large difference between the connectivity and size of the graph in different instances, make it more difficult to learn the mapping between the original graph data and the learning landmark. At present, deep learning has been successfully applied to image classification, semantic segmentation, machine translation and other fields. The underlying data representation of these tasks usually has a grid like structure. By applying them to all input locations, these architectures can effectively reuse their local filters with learnable parameters. Inspired by this, scholars try to expand the neural network to deal with graph data. In the early work, cyclic neural network is used to deal with directed acyclic graph. The disadvantage of this method is that the nodes inside the graph structure data are originally disordered. When cyclic neural network is used to deal with the data of graph structure, it will artificially give a sequence to the nodes inside the graph, which makes the generalization ability of the model weak, Therefore, how to capture key structural information from graph based objects is still a challenging problem in machine learning community.

2. GNN overview

CNN is one of the research motives of GNN. CNN can extract multi-scale local spatial features in images and combine them into highly expressive feature vectors, which makes breakthroughs in all machine learning fields, thus opening the era of deep learning. However, CNN can only operate euride data such as images (two-dimensional grid), texts (one-dimensional sequence) and other rules As for the non eurydian data like the figure, the effect of CNN processing is not good. Another motive of GNN research is graph embedding. The main purpose of graph embedding is to map the nodes in a graph into a low-dimensional vector based on the characteristics of the graph, and to retain some topological properties of the graph, so as to measure the similarity between nodes and make it more convenient to use. In the field of graph analysis, traditional machine learning methods usually rely on the characteristics of human design, but the flexibility of graph structure and the high cost of human make this method limited. However, the existing deep learning methods have serious shortcomings. First of all, the nodes in the encoder do not share any parameters, which means that the number of parameters in the model will increase linearly with the number of nodes, which will lead to low efficiency of the algorithm. Secondly, the direct embedding method lacks generalization ability, and can't deal with dynamic graphs or generalize to new graphs.

GNN came into being and is used to gather information in graph structure data based on CNN and graph embedding method, . GNN processes the data represented in the graph domain by extending the existing neural network, where each node is defined by its own characteristics and its related nodes. Experimental results show that GNN is a powerful structured data modeling architecture, but there are still some limitations. First of all, it is inefficient to update the hidden state vector of the node iteratively according to the fixed point. If the assumption of fixed point is relaxed, a multi-layer GNN can be designed to get the stable vector representation of the node and its neighborhood. Secondly, GNN uses the same parameters in the process of iterative calculation, while most commonly used neural networks use different parameters in different layers. This parameter is used for hierarchical feature extraction method. In addition, the update of hidden state vector of nodes in the graph is a step-by-step process, and the variation of RNN such as GRU and LSTM can promote this process. Thirdly, GNN can not effectively model the information features of some edges. Finally, if we focus on the representation of nodes rather than graphs, fixed points are not suitable, because the distribution values represented on fixed points will be very smooth, and there will be less information for distinguishing each node. Therefore, it is still necessary to introduce new methods to improve GNN.

3. GAT principle

Attention mechanism has been successfully applied to many tasks based on sequence. In this section, we will summarize the theoretical derivation of GAT and its advantages in application. GAT has only one single graph attention layer, and any graph attention network is constructed by superposing this layer. The coefficient of node pair (i, j) in attention mechanism is calculated by the following formula in this layer:

$$\alpha_{i,j} = \frac{e^{\left\{\text{LeakyReLu}[a^T (Wh_i \| Wh_j)]\right\}}}{\sum_{k \in N_i} e^{\left\{\text{LeakyReLu}[a^T (Wh_i \| Wh_j)]\right\}}} \quad (1)$$

Where || is a concatenated symbol, indicating that two vectors are spliced. $\alpha_{i,j}$ is the attention coefficient of node j for node i, N_i is the neighbor node set of node i in the graph. The input set of node features is expressed as $h = \{h_1, h_2, \cdots, h_N\}, h_i \in R^F$, N represents the total number of nodes, F represents the total number of features of each node. This layer generates a new set of node features $h' = \{h_1', h_2', \cdots, h_N'\}, h_i' \in R^F$ as output. $W \in R^{F \times F}$ is the weight matrix of sharing linear transformation among nodes, and $a \in R^{2F}$ is the weight vector of single-layer feedforward neural network. Formula (1) normalizes the weight vector by softmax activation function, and uses LeakyRelu function to

produce nonlinear characteristics. After obtaining the normalized attention coefficient $\alpha_{i,j}$, use $\alpha_{i,j}$ to calculate the linear combination of corresponding node features as the final output eigenvector $h^{'}$ of each node. The calculation formula is as follows:

$$h_i^{'} = \sigma(\sum_{j \in N_i} \alpha_{ij} W h_j) \tag{2}$$

Where $\sigma(\cdot)$ is the activation function, similar to the previous LeakyRelu function, in order to increase nonlinearity.

In order to stabilize the learning process of the network, GAT extends the attention mechanism by using multiple attention, that is, *K* independent attention mechanisms are used to calculate the hidden state vector, and then the eigenvectors are concatenated to get the following calculation formula

$$h_i^{'} = \Big\|_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij}^{k} W^k h_j) \tag{3}$$

In formula (3), || represents the serial symbol, α_{ij}^{k} represents the normalized attention coefficient calculated by the *k*-th attention mechanism, and W^k represents the weight matrix corresponding to the input linear transformation in the *k*-th attention mechanism. It should be noted that in such a setting, each node corresponding to the final output $h_i^{'}$ will contain $K \cdot F^{'}$ eigenvectors.. Therefore, if we want to use multi attention in the last layer of GAT, that is, the output layer, it will not be appropriate to connect the eigenvectors in series. Therefore, we use the average method, and the specific calculation formula is as follows:

$$h_i^{'} = \sigma(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} \alpha_{ij}^{k} W^k h_j) \tag{4}$$

The final output $h_i^{'}$ of the average method will contain $F^{'}$ fused feature vectors for each node, which reduces the computation and can effectively aggregate the features of nodes, and finally classify the nodes in the graph.

4. Experimental analysis

4.1Experimental result

For the comparison and evaluation of GAT model based on various good baseline methods and previous methods, there are three benchmark data sets, and there is only one large graph with many nodes in all three data sets, as shown in Table 1.

Table 1. Experimental data set

Attribute	Cora	Citeseer	Pubmed
Node number	2708	3327	19717
Edge number	5429	4732	44338
Node characteristics	1433	3703	500
Category	7	6	3
Training node	140	120	60
Verification node	500	500	500
Test node	100	1000	1000

For the three datasets in Table 1, we applied the two-layer GAT model in the experiment, and the super parameters of its architecture were optimized on the Cora datasets, and then reused by Citeser. The first layer consists of *K*=8 multi head attention. Each attention head computes $F^{'} = 8$ features (64 features in total), and then uses exponential linear elements to provide nonlinear features. The second layer is used for classification: the specific structure is that one can calculate C features (where C is the number of categories in Table 1), and then use the softmax activation function to classify. In order to solve the problem of small amount of data in training set, a lot of regularization methods are

used in GAT model. In training, L2 regularization is used and the parameter $\lambda = 0.0005$ is set. In addition, dropout that setting p=0.6 is applied to the input layer of the two-layer GAT model, and to the normalized attention coefficient, which means that in each iterative training, each node may be selected at random sampling of its neighbor nodes. It is also found in the experiment that the gat model needs to be fine tuned when training PubMed's training set: the parameter λ of L2 regularization is enhanced to 0.001.

The experimental results of comparative evaluation are shown in Table 2, and the evaluation standard is the average classification accuracy after 100 runs on the data set. In order to evaluate the benefits of attention mechanism fairly, the experiment further evaluated a GCN model, which calculated 64 hidden features, and tried to use the ReLu and ELU activation functions. Finally, GCN-64 reported the better average classification results after 100 runs on the dataset. The experimental results show that GCN-64 uses the ReLu activation function better.

Table 2. Average classification accuracy of the experiment method

Method	Cora	Citeseer	Pubmed
MLP	53.2%	47.4%	72.6%
ManiReg	58.7%	62.1%	71.7%
SemiEmb	59.4%	58.4%	72.7%
LP	68.8%	46.4%	64.2%
DeepWalk	67.7%	44.1%	66.9%
ICA	74.8%	68.1%	74.5%
Planetoid	75.9%	67.2%	77.7%
Chebyshev	80.5%	68.4%	75.1%
GCN	80.9%	71.3%	79.3%
MoNet	81.3%	—	79.2%
GCN-64	81.1%	72.9%	79.5%
GAT	84.2%	73.6%	79.5%

It can be seen from Table 2 that the experimental results successfully show that the GAT method has achieved the best results on all three datasets, which is in line with the expectation before the experiment. The "—" in Table 2 indicates that Monet cannot get results on Citeser datasets. The experiment can also improve the effect of GCN methods on Cora and Citeseer datasets, which shows that assigning different weights to nodes with the same neighbors may improve the effect of the model.

4.2Discussion

When using GAT to realize image classification, we can generate GAT model using sparse matrix operation, which can reduce the storage complexity linearly according to the number of nodes and edges, and can allow the GAT model to run in a larger graph structure data set. However, the tensor processing framework used by GAT only supports sparse matrix multiplication of tensor with rank 2, which limits the ability of multi batch processing of the model, especially when there are multiple graphs in the data set, the effect of GAT model is not so ideal, and solving this limitation is still an important direction of future work. According to the regularity of graph structure, GPU may have no performance advantage compared with CPU because of sparse matrix operation. It should also be noted that the upper limit of the receptive field in GAT model is determined by the depth of the network, which is similar to GCN and its similar models. However, technologies like skip connections can be easily applied to the appropriate extension of network depth. Finally, because the computation of attention mechanism is parallel on all nodes of the graph, and it is parallelized in a distributed way, it may produce a lot of redundant computation, because the two adjacent nodes are neighbors to each other, when calculating the hidden state vectors of the two nodes respectively, the neighbors of the two nodes may be highly overlapped. From the above description and analysis, it can be seen that the goal of GAT is to extract the characteristics of nodes to classify the nodes in the graph, and it does not

contain the information of the edges between nodes, so it ignores part of the structural information of the graph. In the future, we need to further study how to combine the information of nodes and edges to classify the graph structure data.

5. Conclusion

In view of the shortcomings of the existing image classification algorithm GNN, this paper introduces attention mechanism and proposes an image classification model based on GAT. In this paper, the improvement motivation of GNN is described in detail, and the principle of graph attention network based on GNN is given. By setting the hidden state vector of the model learning node, the effect of classifying the nodes in the graph is finally achieved. This method is compared with the traditional GNN. Compared with the method, it not only reduces the computation, but also enhances the generalization ability of node classification. However, the disadvantage is that it ignores the structure information of the graph and cannot process the data set with multiple graphs in multiple batches. In the future, this method will be improved to solve this problem.

Acknowledgments

This work was supported by Major and Special Project of Taizhou Vocational & Technical College.

References

[1] Zhao, Z.Q., Jiao, L.C., Zhao, J.Q., et al. (2016) Discriminant Deep Belief Network for High-Resolution SAR Image Classification. Pattern Recognition, 61:686-701.
[2] Tang, P., Wang, X.G., Feng, B., et al. (2017) Learning Multi-Instance Deep Discriminative Patterns for Image Classification. IEEE transactions on image processing, 26(7):3385-3396.
[3] Yan, Z., Zhang, H., Wang, B., et al. (2016) Automatic Photo Adjustment Using Deep Neural Networks. ACM transactions on graphics, 35(2):1-15.
[4] Baskararaja, G., Manickavasagam, M. (2012) Subgraph Matching Using Graph Neural Network. Journal of Intelligent Learning Systems & Applications, 4(4):274-278.
[5] Liu, Z.N., Liu, W.Y., Chen, P.Y., et al. (2019) hpGAT: High-Order Proximity Informed Graph Attention Network. IEEE Access, 7: 123002-123012.
[6] Zhang, C.H., Yu, J., Liu, Y., et al. (2019) Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting. IEEE Access, 7: 166246-166256.
[7] Wang, R., Li, B.C., Hu, S.W., et al. (2020) Knowledge Graph Embedding via Graph Attenuated Attention Networks. IEEE Access, 8: 5212-5224.