GAU-Nets: Graph Attention U-Nets for Image Classification

Haoran Zhao, Chengwei Zhang*
School of Artificial Intelligence, Hebei University of Technology, Tianjin 300132, China
zcw2000zcw@gmail.com

Abstract. Graph neural network is a research hotspot in the field of deep learning recently, and its application has become more and more extensive. A new graph neural network model is proposed, Graph Attention U-Nets (GAU-Nets for short), which has strong processing capabilities for graph structures. In GAU-Nets, the graph data structure is preliminarily processed through GCN, and then extracted the features by the graph pooling and Unpooling blocks. We innovatively added the attention mechanism to GAU-Nets to avoid forgetting the important information. We have done a lot of image classification experiments on MS-COCO and other datasets. The experimental results prove that GAU-Nets perform better than other traditional graph neural network models. Without bells and whistles, our GAU-Nets method has an accuracy of 69.1% on the MS-COCO data set and 82.1% on the VOC 2007 data set, which has surpassed all benchmark methods.

1. Introduction
In recent years, due to the expansion of graph data and the development of graph expression capabilities, researchers in the field of deep learning have shown interest in the study of graph neural networks. Different from traditional pictures and texts, graph has a unique non-European data structure. Due to its excellent performance, GNN has been widely used in the field of graph analysis [1]. Kipf et al. [2] Proposed Graph Convolutional Networks (GCN), which alleviates the over-fitting problem of the graph in the local neighborhood structure by restricting the hierarchical convolution operation. A dual graph convolutional network (DGCN) [3] is proposed to jointly Consider local and global consistency. GAT [4] add the Attention mechanism in the graph neural networks. These models have achieved good performance and solved many problems.

In this paper, a new graph neural network model is proposed, Graph Attention U-Net (GAU-Nets for short), which adds graph attention mechanisms to Graph U-Nets [5]. We apply GAU-Nets to the image classification, one of the basic computer vision problems. We choose the challenging problem to provide experimental evidence for our proposed contributions. After the preprocessing of pictures, GAU-Nets have learned the graph structure formed by pictures well with strong classification ability. Compared with other graph neural networks, our GAU-Nets achieves better performance.

2. Graph Attention U-Nets

2.1. Graph Convolutional Networks
GCN was first proposed by Kipf et al. [2], and the forward propagation method is defined in each layer as:
\[ X_{l+1} = \sigma \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X_l W_l \right), \]

Where \( X_l \) contains the feature matrix in the \( l \)-th layer, and \( \tilde{A} \) is set as the self-loops from the input adjacent matrix \( A \). At the same time, \( \tilde{A} \) is normalized by a diagonal matrix \( \tilde{D} \) in each layer. \( W_l \) is a trainable weight matrix that applies a linear transformation to feature vectors.

2.2. Graph Pooling Layer and Graph Unpooling Layer

We refer to Graph Pooling Layer (gPool) and Graph Unpooling Layer in the original Graph U-nets [5]. In the gPool Layer, we are going to generate a smaller subset by selecting a subset of nodes. We employ a trainable projection vector \( P \). For getting the features, we project all node features to one dimension with \( k \)-max pooling for nodes selected. As we use one dimension footprint of each node for the selection, the connectivity is consistent across nodes in the new graph. \( y_i \) represents the amount of node information we can save when predicting the \( P \) direction, in this case it is defined as \( y_i = x_i P / \| P \| \). In the Graph Unpooling Layer, the location information of the selected node in the corresponding Unpooling layer is retained, and we use this information to return the location of the node.

2.3. Graph Attention Mechanism

We integrate the attention mechanism into the distribution. It follows a self-attention strategy and calculates the hidden state of each node by processing its neighbors. The single graphic attention layer is defined as:

\[
\alpha_{ij} = \frac{\exp \left( \text{LeakyReLU} \left( a^T [W_h_i || W_h_j] \right) \right)}{\sum_{k \in N_i} \exp \left( \text{LeakyReLU} \left( a^T [W_h_i || W_h_k] \right) \right)}
\]

where \( N_i \) represents the neighborhoods of node \( i \), \( \alpha_{ij} \) is the attention coefficient of node \( j \) to \( i \) in the graph. The input set, \( h = \{ h_1, h_2, \ldots, h_N \} \), is node features to the layer. \( N \) is the number of nodes and \( F \) is the number of features of each node. The layer produces a new set of node features as its output as \( h' = \{ h'_1, h'_2, \ldots, h'_N \} \) [1]. \( W \) is the metric weight of the common transition and \( a \) is the weight vector of the single layer of the neural network process.

As shown in Figure 1, we use the GCN layer to convert the input vector into a low-dimensional representation. Two encoder modules are stacked together, and each module contains a gPool layer and a GCN layer. In the decoder area, we have designed two decoder modules. Each module contains a gUnpool layer, an Attention gate and a GCN layer. The encoder module uses skip links to connect low-level spatial features from the encoder module to the same size volume. The output vector of the node in the last layer is the network embedding, which can be used for node classification.

![Fig.1 Construction of Graph Attention U-Nets](image)
3. Experiments and Results

3.1. Datasets Introduction
In this experiment, we used two datasets, namely MS-COCO [7] and VOC 2007 [8]. Microsoft COCO is a dataset used in image classification tasks. Its training set size reaches 82081 images. The validation set contains 40,504 pictures. These pictures are divided into 80 categories. PASCAL Visual Object Classes Challenge (VOC 2007) is also very well-known, including 9,963 images, divided into 20 categories.

3.2. Data preprocessing
Since datasets such as COCO are picture-type, we need to divide the picture into superpixel pictures through SLIC [6], and then rebuild the picture data structure as Fig. 2. We establish the superpixel graph $G$ of each image through the connection. We subtly attach two spatially $i = 1$ via the graph edges $\{E_{ij}\}$ to represent a superpixel and each graph edge $E_{ij}$. Each graph node $V_i$ is surrounded by superpixel nodes. Each graph node $V_i$ have the input features denoted as $f_i \in \mathbb{R}^d$, where the feature dimension is $d$. We computed the feature $f_i$ by averaging the features of all the pixels which is belonging to the same superpixel node $V_i$.

3.3. Classification result
We pre-trained our model on ImageNet, using SGD as our optimizer, and set the momentum to 0.9. In addition, we also used the learning rate decay method to optimize the model, with the learning-rate set as 0.01, in this case, the learning rate of our model decays by a factor of 10 after every 40 rounds of training. We used Pytorch to implement our model.

We use many evaluation criteria to evaluate the performance of the model, such as average per-class accuracy (CP), $F1$ (CF1), recall rate (CR), average overall accuracy (OP), $F1$ (OF1) and recall rate (OR). In order to more clearly see the advantages of our model, we also used the evaluation criteria of Top-3 labels and Top-5 labels.

| Methods                | MS-COCO | VOC 2007 |
|------------------------|---------|----------|
| GCN[2]                 | 63.0    | 72.0     |
| GAT[4]                 | 63.4    | 71.4     |
| Graph U-Nets[5]        | 67.4    | 79.4     |
| GAU-Nets(ours)         | 69.1    | 82.1     |

Fig. 2 Geometrical dimensions of elements
The visualization of our results shows that our GAU-nets model is better than the others in a series of indicators as Fig.3. In particular, our model performs better than others in dealing with difficult-to-recognize pictures, indicating that our model has a stronger ability to find details.

4. Conclusion

In this paper, we proposed a novel graph neural network, GAU-Nets, for image classification. Our method improves the precision of graph neural networks in learning graph data from image. Our proposed method is universal and modular, so it can be easily applied to image classification problems. We have done a lot of image classification experiments on MS-COCO and other datasets. The experimental results prove that our GAU-Nets method has an accuracy of 69.1% on the MS-COCO data set and 82.1% on the VOC 2007 data set, which has surpassed all benchmark methods and GAU-Nets has an average increase of 1.7% in MS-COCO and 2.7% in VOC 2007 datasets.

References

[1] Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2018). Graph Neural Networks: A Review of Methods and Applications. 1–22.
[2] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” ICLR 2017, 2017.
[3] J. Atwood and D. Towsley, “Diffusion-convolutional neural nets,” in NIPS 2016, 2016, pp. 1993–2001.
[4] C. Zhuang and Q. Ma, “Dual graph convolutional networks for graph-based semi-supervised classification,” in WWW2018, 2018.
[5] Gao, H., & Ji, S. (2019). Graph U-nets. 36th International Conference on Machine Learning, ICML 2019, 2019-June, 3651–3660.
[6] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., S"usstrunk, S.: Slic super-pixels. Technical report (2010)

[7] TsungYi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence ´Zitnick. Microsoft COCO: Common objects in context. In ECCV, pages 740 -755, 2014. 2, 5

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. IJCV, 88(2):303–338, 2010. 2, 5