A Multi-view Image Sets Classification Based on Graph Convolutional Neural Network

Xun LIU1*, XiaoLin CHEN1, GuoQing XIA1, HuaZhen CHEN1, HeZhong YE1, and ZhanChi CHEN1

1Software Engineering Institute of Guangzhou, Guangzhou 510900, China
(e-mail: liuxun.stf@gmail.com; 1060177942@qq.com; xgq@sise.com.cn; chz@sise.com.cn; yhz@sise.com.cn; czc@sise.com.cn)
* Corresponding author. email: liuxun.stf@gmail.com

Abstract: Multi-view images represent the target object from various perspectives. We propose a new spectral graph convolutional neural network (BSGCN) for multi-view image sets, which uses spectral graph convolution technology to process multi-view image sets. With the batch normalization techniques, the BSGCN speeds up the model convergence and improves the generalization capabilities of the model. We analyze the performance of two variants of BSGCN and its model by experiments on multi-view datasets. Experimental results show that the BSGCN is 2.84% and 7.26% higher than the classic graph CNN [17] on Modelnet10 and Modelnet40 multi-view datasets, respectively, and 0.34% higher than the classical CNN [6] on Modelnet40 multi-view dataset.

1. Introduction

Convolutional neural network (CNN) has made breakthroughs in images, such as image classification [1], image retrieval [2], and image segmentation [3]. In the CNN model, the training and test images are typically single-shot images. However, multi-view images represent the information of the target object from various perspectives. Therefore, it is very valuable to make full use of multi-view information to improve the accuracy of CNN models.

In recent years, researchers have developed feature descriptors for multi-views image to improve image set classification accuracy. Multi-view descriptors are mainly divided into four categories.

1) Multi-view descriptor based on a two-dimensional planar image [4]–[6], which combines the neighborhood view features of a plurality of two-dimensional images. These various feature descriptors have the advantage of a small number of parameters and a low dimensionality.

2) Voxel-based multi-view descriptors [7]–[11], which utilize three-dimensional information of images and contain all spatial structural features of multi-view images. Those descriptors are the earliest and most studied method in the multi-view image set classification. The above two multi-view descriptors process data in Euclidean space, however, they cannot handle irregular data such as 3D point clouds.

3) Multi-view descriptors based on 3D point cloud [12], [13] they have excellent performance in multi-view image recognition. However, the input data of the 3D point cloud is high dimensional, which requires huge storage space and computing resources. The processing of high dimensional data is prone to dimensional hazards and over-fitting.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
4) Graph CNN-based multi-view descriptor [14], [15], which have the capability to process data of arbitrary structure, including grid structure and graph structure data. Compared to voxel-based and 3D point cloud-based methods, the GCNN-based descriptor would significantly reduce the input data dimension.

In this paper, we propose a novel spectral graph convolutional neural network (BSGCN) by using batch normalization technology. The BSGCN model is composed of 2-layer graph convolutional layer, 2-layer graph max-pooling layer, two batch normalization layers, a fully connected layer and a soft max layer. The model has low input data dimensions, no pre-training, and simple network layers. The proposed model accelerates the convergence of the network and improves the generalization performance of the network.

The main contributions of this paper are as follows:
1) We propose a new spectral graph convolutional neural network with batch normalization layers model, which can process the data of arbitrary structure.
2) The model does not require pre-training, and could achieve high precision on multi-view datasets.
3) The input data of the model has a low dimensionality, which significantly reduces storage space and computing resources.

2. Method

2.1. Spectral Filtering on Graph
Given the input graph signal \( x \), \( x \) is defined by the spectral filter \( g_\theta \) as

\[
y = g_\theta (L) x = g_\theta (U \Lambda U^T) x = U g_\theta (\Lambda) U^T x g_\theta (\Lambda). \tag{1}
\]

\( g_\theta (\Lambda) \) is defined as

\[
g_\theta (\Lambda) = \text{diag}(\theta). \tag{2}
\]

\( \theta \in \mathbb{R}^n \) is a vector of Fourier coefficients, which is updated by back-propagation algorithm after random initialization. Due to the poor spatial localization of (2) and the computational complexity of \( O(n) \), Defferrard et al. [16] proposes a localized filter \( g_\theta (\Lambda) \) in order to solve the above problem.

\[
g_\theta (\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k. \tag{3}
\]

\( \theta \in \mathbb{R}^K \) is a vector of polynomial coefficients, and \( K \) is the receptive field of the convolution kernel of the graph filter. Generally, \( K \) is much smaller than \( n \). Defferrard et al. [16] uses Chebyshev polynomial to approximate \( g_\theta (\Lambda) \).

\[
g_\theta (\Lambda) \approx \sum_{k=0}^{K-1} \theta_k T_k(\Lambda). \tag{4}
\]

\( \theta \in \mathbb{R}^K \) is a vector of Chebyshev polynomial coefficients, \( T_k(\Lambda) \) represents a Chebyshev polynomial of \( K \) order, \( \Lambda = \frac{L}{\lambda_{\text{max}}} - I_n \), \( \lambda_{\text{max}} \) is the maximum eigenvalue of \( L \). Chebyshev polynomial is defined as

\[
T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x). \tag{5}
\]

where \( T_0(x) = 1, T_1(x) = x \). It is available from (1) (4) (5) formula.

\[
g_\theta \ast x \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}). \tag{6}
\]

where \( \tilde{L} = \frac{L}{\lambda_{\text{max}}} - I_n \), the complexity of the equation is \( O(|\varepsilon|) \).

| Table 1 | Architecture structure for BSGCN model |
| --- | --- |
| | Architecture |
| | SGC32 – BN – ReLU – GP4 – SGC64 – BN – ReLU – GP4 – FC1280 – softmax |
2.2. BSGCN Model

Our fast localized spectral graph convolutional neural networks with batch normalization layers (BSGCN) model is composed of 2-layer graph convolutional layer, 2-layer graph max-pooling layer, two batch normalization layers, a fully connected layer and a softmax layer. The model framework is shown in Table 1.

SGCn represents the spectral graph convolutional layer with n filters. BN denotes batch normalization layer, which speeds up the convergence and improves the generalization performance of the network to improve accuracy. ReLU is a nonlinear activation function that makes the network nonlinear and can represent any complex function from input to output. GPn represents the max-pooling layer with a pooling size of n. FCn represents the fully connected layer with n neurons, and the final layer is a softmax layer for multi-classification.

2.3. BSGCN Training And Testing Methods

Our training and testing methods for BSGCN are mainly derived from the idea of Kanezaki et al. [17]. In the training phase, the training sample contains images of each target image \( x_i \) \( i = 1 \ldots N \), its label \( y \in \{1, 2, \ldots, N\} \). Let \( M \) be the number of the pre-defined viewpoints and \( N \) denote the number of target object categories. We introduce an incorrect viewpoint, the label of which corresponds to the class of the non-target image. For a given label, we define a latent variable \( v_i \in \{1, 2, \ldots, M\} \) to optimize equation (7).

\[
\max_{\text{BSGCN}, \{v_i\}^M_{i=1}} \sum_{i=1}^{M} \left( \log p_{v_i,y}^{(i)} + \sum_{j \neq v_i} \log p_{j,N+1}^{(i)} \right)
\]

\[
= \max_{\text{BSGCN}, \{v_i\}^M_{i=1}} \sum_{i=1}^{M} \left( \log p_{v_i,y}^{(i)} + \sum_{j=1}^{M} \log p_{j,N+1}^{(i)} - \log p_{v_i,N+1}^{(i)} \right)
\]

\[
= \max_{\text{BSGCN}, \{v_i\}^M_{i=1}} \prod_{i=1}^{M} \frac{p_{v_i,y}^{(i)}}{p_{v_i,N+1}^{(i)}}
\]

where \( p_{v_i,y}^{(i)} \) corresponds to the probability that the predicted label is the same as the real label \( v_i \) for \( v_i \)-th viewpoint from all predefined viewpoints. \( p_{j,N+1}^{(i)} \) indicates the probability that the predicted class from the incorrect viewpoint belongs to the \( N+1 \) class. If the latent variable \( v_i \) is fixed, the BSGCN network parameters can be updated by back-propagation to maximize equation (7).

In the testing phase, except that the input image is the image from \( M' (1 \leq M' \leq M) \) viewpoints, the rest of the optimization process is the same as the training phase.

3. Experiment

3.1. Datasets

The multi-view data set in the experiment is the Modelnet10 and Modelnet40 [18]. The Modelnet10 dataset has a total of 10 categories, which includes 4899 target objects. The Modelnet40 dataset has a total of 40 categories, which includes 12, 311 target objects. The predefined viewpoints selected in this paper is \( M = 20 \), that is, each target object is an image from 20 shooting angles, and the angle of viewpoints \( M \) is the same as RotationNet [17].

3.2. Experimental parameter

On Modelnet40 dataset, we construct a 16–NN graph with the same input size and use the fast Graculus [21] algorithm to calculate the coarsening of a given graph, the coarsening level is 4. The initial learning rate is 0.1 with a decay rate of 0.1 after every 100 train steps, and the learning rate is optimized by SGD.
with the momentum of 0.9. We set dropout rate to 0.5, regularization weight to 0.000005, and batch size to 40. We choose the spectral graph convolution filter kernel size of 9. On Modelnet10 dataset, we set the batch size to 200 in order to significantly reduce the training time, the remaining parameter settings remain unchanged. In all our experiments, we first reshape the image to 28×28, then normalize it to 0-1. All experiments are trained for 200 epochs.

3.3. Results
In order to fully reflect the advantages of BN, we introduce two variants of BSGCN as follows:

BSGCN-a: We only remove the BN layer after the first spectral graph convolutional layer, and the remaining layers and parameters remain unchanged. The corresponding model is SGC32−ReLU−GP4−SGC64−BN−ReLU−GP4−FC1280−softmax.

BSGCN-b: We remove all the BN layers after all spectral graph convolutional layers, and the remaining layers and parameters remain unchanged. The corresponding model is SGC32−ReLU−GP4−SGC64−ReLU−GP4−FC1280−softmax.

The comparison results between our BSGCN and the BSGCN-a and BSGCN-b are shown in Table 2 and Table 3. The results in Table 2 show that on Modelnet10 dataset, our BSGCN needs 62 epochs to reach the highest accuracy of 92.84%, which is 0.33% higher than BSGCN-a to reach the best accuracy of 92.51%. Our BSGCN is also higher than BSGCN-b to reach the best classification result of 92.51%. Although BSGCN-a and BSGCN-b models have the best same classification accuracy of 92.51%, BSGCN-a takes only 33 epochs to achieve the highest precision, and the number of epochs BSGCN-b needs to achieve the same accuracy is almost 5 times than BSGCN-a.

| Method     | Epochs to Max Accuracy | ModelNet10 Classification (Max Accuracy) |
|------------|------------------------|----------------------------------------|
| BSGCN(ours)| 62                     | 92.84%                                 |
| BSGCN-a(ours) | 33                  | 92.51%                                 |
| BSGCN-b(ours) | 150                | 92.51%                                 |

The results in Table 3 show that on Modelnet40 dataset, our BSGCN needs 27 steps to reach the accuracy of no less than 89.38%. However, the BCGCN-a and BSGCN-b need 43 and 152 epochs to reach the similar classification result, respectively. Moreover, BSGCN-a needs 159 epochs to reach the highest accuracy of 90.32%, while our BSGCN needs about 3 times fewer epochs to reach the same classification result. Compared to BSGCN-a, the highest accuracy of BSGCN-b was only 89.38%. The highest accuracy of our BSGCN is 90.44%, which is higher than the highest classification accuracy of BSGCN-a and BSGCN-b.

The comparison results of Table 2 and Table 3 show that BSGCN model designed by batch normalization technology can better and faster learn the characteristics of data and improve the accuracy of the model, and verify the validity of our BSGCN.

The comparison results of our BSGCN method and other methods are shown in Table 4. On Modelnet10 dataset, our BSGCN method achieves an accuracy of 92.84%, which is 1.34% higher than 91.5% of based on 2D planar image [5]. Recently, voxel-based [10] uses three-dimensional information
of multi-view images, which reaches very high classification result of 92.32%. However, Our BSGCN is still more competitive over [10]. Our BSGCN is 15.24% higher than 77.6% of based on 3D point cloud [12]. Compared to based on graph CNN [14], [15], our BSGCN is higher 18.54% than [14] of 74.3% and 2.84% higher than [15] of 90.0%.

| Method                  | Modelnet10 Classification (Accuracy) | Modelnet40 Classification (Accuracy) |
|-------------------------|-------------------------------------|-------------------------------------|
| MVCNN [4]               | -                                   | 90.1%                               |
| Zanuttigh and Minto [5] | 91.5%                               | 87.8%                               |
| Soltani et al. [6]      | -                                   | 82.1%                               |
| 3D-GAN [7]              | 91.0%                               | 83.3%                               |
| Xu and Todorovic [8]    | 88.00%                              | 81.26%                              |
| Arvind et al. [9]       | -                                   | 86.50%                              |
| binVoxNetPlus [10]      | 92.32%                              | 85.47%                              |
| VSL [11]                | 91.0%                               | 84.5%                               |
| PointNet [12]           | 77.6%                               | -                                   |
| PointNet [13]           | -                                   | 89.2%                               |
| Dominguez et al. [14]   | 74.3%                               | -                                   |
| ECC [15]                | 90.0%                               | 83.2%                               |
| BSGCN(ours)             | 92.84%                              | 90.44%                              |

On Modelnet40 dataset, our BSGCN method is superior to based on 2D planar images [4]–[6], voxel-based [7]–[11], based on 3D point cloud [13] and based on graph CNN [15]. Our BSGCN achieves 90.44% accuracy rate, which is higher than 83.2% of ECC [15] by 7.24%. Although the accuracy of our BSGCN is slightly more than the MVCNN [4] method of inputting $224 \times 224 \times 3$ of Imagenet1000 pre-train, the BSGCN does not need pre-train. Moreover, there are 4 layers network and the input data dimension are low($28 \times 28 \times 3$).

4. Conclusions

We propose a new graph convolutional neural network called BSGCN that can process arbitrary structure data. Our BSGCN effectively improves accuracy and accelerates the convergence speed through batch normalization technology. We analyze the results of BSGCN and two variants of BSGCN by experiments, which the experiments prove that the BSGCN designed by batch normalization technology could learn better feature of the data. The classification accuracy on Modelnet10 and Modelnet40 datasets is much higher than the graph CNN-based approach, which is higher than the other methods. In the future, we will consider using $1 \times 1$’s graph convolution and global graph pooling layers instead of the fully connected layer to greatly reduce the number of parameters.

Acknowledgment

This research was funded by the Characteristic Innovation Projects of Ordinary Universities in Guangdong Province, China (2020KTSCX212,2019KTSCX245), the Projects of South China Institute of Software Engineering of Guangzhou University (ky202012).

References

[1] A. G. Howard et al., “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.
[2] J. Yang et al., “Dynamic Match Kernel with Deep Convolutional Features for Image Retrieval,” IEEE Trans Image Processing, vol. 27, no. 11, pp. 5288-5302, Jun. 2018.
[3] L. C. Chen et al., “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834-848, 2018.
[4] H. Su et al., (2015). Multi-view convolutional neural networks for 3d shape recognition. Presented at ICCV. Proceedings of the IEEE international conference on computer vision.

[5] P. Zanuttigh et al., (2017). Deep learning for 3d shape classification from multiple depth maps. Presented at ICIP. Proceedings of IEEE International Conference on Image Processing.

[6] J. Wu et al., (2017). Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. Presented at CVPR. The IEEE conference on computer vision and pattern recognition.

[7] A. A. Soltani et al., (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Presented at ICONIP. Advances in Neural Information Processing Systems.

[8] X. Xu et al., “Beam search for learning a deep convolutional neural network of 3d shapes,” arXiv preprint arXiv:1612.04774, 2016.

[9] V. Arvind et al., “Wide and deep volumetric residual networks for volumetric image classification,” arXiv preprint arXiv:1710.01217, 2017.

[10] C. Ma et al., “Binary Volumetric Convolutional Neural Networks for 3-D Object Recognition,” IEEE Trans. Instrum. and Meas., no. 99, pp. 1-11, Jul. 2018.

[11] S. Liu et al., (2018). Learning a Hierarchical Latent Variable Model of 3D Shapes. Presented at 3DV. 2018 International Conference on 3D Vision.

[12] G. A. Garcia et al., (2016). Pointnet: A 3d convolutional neural network for real-time object class recognition. Presented at IJCNN. 2016 International Joint Conference on Neural Networks.

[13] R. Q. Charles et al., (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. Presented at CVPR. 2017 IEEE Conference on Computer Vision and Pattern Recognition.

[14] M. Dominguez et al., (2017). Towards 3D convolutional neural networks with meshes. Presented at ICIP. 2017 IEEE International Conference on Image Processing.

[15] M. Simonovsky et al., (2017). Dynamic edgeconditioned filters in convolutional neural networks on graphs. Presented at CVPR. 2017 IEEE Conference on Computer Vision and Pattern Recognition.

[16] M. Defferrard et al., (2016). Convolutional neural networks on graphs with fast localized spectral filtering. Presented at NIPS. Advances in neural information processing systems.

[17] A. Kanezaki et al., (2018). RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. Presented at CVPR. Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition.

[18] Z. Wu et al., (2015). 3D ShapeNets: A deep representation for volumetric shape modeling. Presented at CVPR. The IEEE Conference on Computer Vision and Pattern Recognition.

[19] I. S. Dhillon et al., “Weighted graph cuts without eigenvectors a multilevel approach,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 11, pp. 1944-1957, Sep. 2007.