Graph-based deep learning for graphics classification

Pau Riba, Anjan Dutta, Josep Lladós and Alicia Fornés
Computer Vision Center, Computer Science Department,
Universitat Autònoma de Barcelona, Barcelona, Spain
Email: {priba, adutta, josep, afornes}@cvc.uab.es

***Abstract*—Graph-based representations are a common way to deal with graphics recognition problems. However, previous works were mainly focused on developing learning-free techniques. The success of *deep learning* frameworks have proved that learning is a powerful tool to solve many problems, however it is not straightforward to extend these methodologies to non euclidean data such as graphs. On the other hand, graphs are a good representational structure for graphical entities. In this work, we present some deep learning techniques that have been proposed in the literature for graph-based representations and we show how they can be used in graphics recognition problems.**

I. Introduction

Since the field of graphics recognition deals with 2-dimensional data (symbols, line drawings, maps etc.), graph-based representation has received a lot of attention within the research community [14]. This is mainly because of its triviality in the representation paradigm. In graphic recognition, there are many works that use graph based representation for symbol spotting [11], [3], symbol recognition [4], [10], map registration [16] etc. All these works use to incorporate a graph-based representation for the underlying data and, depending on the problem requirements, they propose an approximate graph matching or classification approach (graph kernel, graph indexing, graph hashing etc) to reach the final goal.

The success of convolutional neural networks (CNNs) [7] in different fields as computer vision and natural language processing, has increased the interest of extending these frameworks to other domains such as non-Euclidean structures, as graphs and manifolds. The previously mentioned extensions are called *geometric deep learning*[1] and has recently increased its popularity [6], [9]. Following this trend, in this article, we show some experimental results carried out with the recently proposed framework involving graph neural network on graphical symbol datasets.

The rest of the paper is organised as follows: Section II introduces the concept of message neural passing given an input graph. Section III presents some preliminar results and discussion on graphics problems. Finally Section IV draws the conclusion and introduces the future work.

II. Deep graph models for graphics recognition

In this work, the *message passing neural network* (MPNN) framework presented by Gilmer *et al.* [6] has been used to classify graph representations of graphics symbols[2]. This particular framework is able to reformulate previously proposed neural network models on graphs and design a new one. Graphs representing graphical symbols are very challenging to test the state of the art methods because of existance of noise and distortion. Despite these simple datasets, there are complex problems such as graph-based keyword spotting [13] where graphs are much bigger and complex, and learning based techniques can be very important there.

The input of the proposed network is an attributed graph with node and edge labels. As an output, the proposed network can either learn individual node embeddings or a global information that summarize the whole graph. The latter can be done through the use of a readout function. The forward pass of the network is defined at two stages, first *message passing phases* and finally an optional *readout phase*.

The message passing phase is computed T times propagating information along the graph. Firstly, a message function M_t computes the information sent from vertices $w \in \mathcal{N}(v)$ to vertex v through the edge e_{vw} at iteration t. All the processed messages are summed to obtain the global message m_v^{t+1} received by v. Intuitively, the message passing phase considers the neighbouring nodes to update the node attributes in an iterative manner. Furthermore, the neighbourhood of the nodes depend on the number iterations through which the message passing procedure takes place.

$$m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw}), \tag{1}$$

where h_w^t and h_v^t are the hidden states of nodes v and w at iteration t. Then, the hidden state of node v is updated with an update function U_t depending on the message m_v^{t+1}.

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{2}$$

After all the message passing iterations are computed, the nodes contain local information that can be used for different tasks, such as classification, at vertex level. However, a global descriptor of the graph can be obtained using a readout

[1] http://geometricdeeplearning.com/

[2] Third party implementation: https://github.com/priba/nmp_qc

function R that should be invariant to permutations of the set of node states.

$$\hat{y} = R(\{h_v^T \mid v \in G\}) \tag{3}$$

Here M_t, U_t and R are learnable and differentiable functions that can be defined depending on the problems to be solved. Previously proposed methodologies, such as, [8], [2] can be reformulated within this framework.

In this paper, we are focused on the functions proposed by Li *et al.* [8]. This model assumes discrete edge types, and the message function is formulated as $M(h_v, h_w, e_{vw}) = A_{e_{vw}} h_w$, where $A_{e_{vw}}$ is a learned matrix for each possible edge label. The update function is $U = \mathrm{GRU}(h_v, m_v)$, where GRU is the Gated Recurrent Unit [1]. Finally, the readout function is defined as $R = \sum_{v \in V} \sigma(i(h_v^{(T)}, h_v^0)) \odot (j(h_v^T))$, where i and j are neural networks. Gilmer *et al.* [6] modified the message function as $M(h_v, h_w, e_{vw}) = A(e_{vw}) h_w$ in order to allow continuous edge attributes, where $A(e_{vw})$ is a neural network which maps the edge vector to a matrix. Furthermore, they added the set2set model [15] as an improved readout function which often provides better performance.

III. Experimental results and discussions

We considered the GREC and LETTERS dataset [12] for the experimental validation of the neural network models mentioned above. GREC is a set of graphs coming from architectural symbols, whereas LETTERS is a graph representation of 15 capital English letters.

Table III shows a comparative results between the classical graph-based techniques against some MPNN models. For MPNN models, the table shows the mean and standard deviation of 10 runs. These are some preliminary results, and more experiments will be done in the near future. Training the MPNN model on the GREC dataset (286 graphs) with 500 epochs needs 878.57 seconds, which gives an idea about time complexity of this type of network. From the experimental results, it is quite clear that the neural network models similar to MPNN are quite capable of extracting information from the graph data. However, we observed that training such models needs a huge amount of data which is not available in the community at present. This scarcity of data often leads to the problem of overfitting, which results in worse classification results.

	GREC	Letters (low)	Letters (med)	Letters (high)
GED [5]	95.5	99.3	94.4	89.1
Embedding [5]	99.2	99.8	94.9	92.9
MPNN	89.5 (±2.80)	91.3 (±1.97)	81.2 (±2.38)	64.24 (±3.48)
MPNN (without set2set)	92.98 (±2.07)	94.8 (±0.68)	86.1 (±1.81)	75.7 (±1.95)

Moreover, using set2set approach as a readout layer leads to a model that is harder to train. Hence, with the current amount of data, it is unfeasible to train a good model. Some ideas to overcome this problem is to reduce the number of parameters of the network and add data augmentation.

IV. Conclusions

Geometric deep learning for graphics recognition is proved to learn important features that are capable for classifying the underlying data. However, deep learning frameworks require a huge amount of data to be able to generalise. In graphics domain, there is no such large dataset in order to exploit all the potential of these methods. Hence, our future work will be focused on the study of neural network frameworks for graph-based graphics or symbol recognition problem. In order to solve the overfitting problem, we will explore some research lines such as reducing the model parameters proposing new message, update and readout functions and data augmentation.

Acknowledgment

This work has been partially supported by the European Union's research and innovation program under the Marie Skłodowska-Curie grant agreement No. 665919, Spanish project TIN2015-70924-C2-2-R, a FPU fellowship FPU15/06264 from the Spanish Ministerio de Educación, Cultura y Deporte, the Ramon y Cajal Fellowship RYC-2014-16831, and the CERCA Program/Generalitat de Catalunya. The Titan X Pascal and the Titan Xp used for this research was donated by the NVIDIA Corporation.

References

[1] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *SSST*, 2014.

[2] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016, pp. 3844–3852.

[3] A. Dutta, J. Lladós, and U. Pal, "Symbol spotting in line drawings through graph paths hashing," in *ICDAR*, 2011, pp. 982–986.

[4] A. Dutta, J. Lladós, and U. Pal, "Bag-of-graphpaths for symbol recognition and spotting in line drawings," in *GREC*, 2011.

[5] J. Gibert, E. Valveny, and H. Bunke, "Graph embedding in vector spaces by node attribute statistics," *PR*, vol. 45, no. 9, pp. 3072 – 3083, 2012.

[6] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *CoRR*, vol. arXiv:1511.05493, 2015.

[9] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," *CoRR*, vol. arXiv:1611.08402, 2016.

[10] N. Nayef and T. M. Breuel, "A branch and bound algorithm for graphical symbol recognition in document images," in *DAS*, 2010, pp. 543–546.

[11] R. Qureshi, J.-Y. Ramel, D. Barret, and H. Cardot, "Spotting symbols in line drawing images using graph representations," in *GREC*, 2008, pp. 91–103.

[12] K. Riesen and H. Bunke, "Iam graph database repository for graph based pattern recognition and machine learning," *S+SSPR*, pp. 287–297, 2008.

[13] M. Stauffer, A. Fischer, and K. Riesen, "A novel graph database for handwritten word images," in *S+SSPR*. Springer, 2016, pp. 553–563.

[14] K. Tombre and B. Lamiroy, "Graphics recognition-from re-engineering to retrieval," in *ICDAR*, 2003, pp. 148–155.

[15] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *ICLR*, 2016.

[16] S. Wachenfeld, K. Broelemann, X. Jiang, and A. Krüger, "Graph-based registration of partial images of city maps using geometric hashing," in *GbRPR*, 2009, pp. 92–101.