



Towards More Robust GNN Training with Graph Normalization for GraphSAINT

Yuying Wang^(✉) and Qinfen Hao

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
wangyuying@ict.ac.cn

Abstract. Graph Neural Networks (GNNs) field has a dramatically development nowadays due to the strong representation ability for data in non-Euclidean space, such as graphs. However, with the larger graph datasets and the trend of more complex algorithms, the stability problem appears during model training. For example, GraphSAINT algorithm will not converge in training with a probability range from 0.1 to 0.4. In order to solve this problem, this paper proposes an improved GraphSAINT method. Firstly, a proper graph normalization strategy is introduced into the model as a neural network layer. Secondly, the structure of the model is modified based on the normalization strategy to normalize the original input data and the input data of the middle layer. Thirdly, the training process and the inference process of the model are adjusted to fit this normalization strategy. The improved GraphSAINT method successfully eliminates the instability and improves the robustness during training. Besides, it accelerates the training procedure convergence of the GraphSAINT algorithm and reduces the training time by about a quarter. Furthermore, it also achieves an improvement in the prediction accuracy. The effectiveness of the improved method is verified by using the citation dataset of Open Graph Benchmark (OGB).

Keywords: Graph Neural Networks (GNNs) · Graph normalization · Training stability · Link prediction

1 Introduction

The Graph Neural Networks (GNNs), which was firstly proposed in 2009 [1], has been developed rapidly in recent years due to the powerful processing ability for data in non-Euclidean space, for example the graph data [2–6]. Nowadays, GNNs are widely used in many areas such as social networks [7, 8], drug discovery [9] and recommendation [10, 11]. As mentioned in [12], there have been mainly four GNNs categories so far including Recurrent Graph Neural Networks (RecGNNs) [13, 14], Convolutional Graph Neural Networks (ConvGNNs) [15–23], Graph Autoencoders (GAEs) [24–27] and Spatial-temporal Graph Neural Networks (STGNNs) [28–30]. Among them, ConvGNNs realize the generalization of the convolution from grid data to graph data, whose typical model is Graph Convolution Network (GCN) [31, 32].

In order to improve the generalization performance of GCN for new nodes, the Graph Sampling based INductive learning meThod (GraphSAINT) algorithm [33] is proposed.

GraphSAINT can realize the effective training of the deep GCNs by using a special mini-batch construction way. This algorithm obtains a set of subgraphs by sampling the original training graph and then builds a GCN based on the subgraphs. Therefore, the graph sampling strategy is the main contribution of GraphSAINT. Besides, this strategy also alleviates the problem of the neighbor explosion, so that the number of neighboring nodes no longer increases exponentially with the number of layers in GraphSAINT. Moreover, compared with GraphSAGE [34], GraphSAINT also enhances the processing capability for large graphs by applying the subgraph sampling method. Therefore, although GraphSAINT uses the same inductive framework as GraphSAGE, GraphSAINT and GraphSAGE are different on sampling method: GraphSAINT samples multiple subgraphs from original dataset to construct minibatch for training; while GraphSAGE adopts the neighbor-node sampling method to generate the node embeddings.

Although GraphSAINT solves the problem of neighbor explosion and has the stronger generalization ability than GCN, the use of the graph sampling strategy makes it difficult training the network. In general, nodes which have the higher influence on each other should be selected to form a subgraph with the higher probability. This ensures that the nodes can support each other within the subgraph. However, such sampling strategy leads to different node sampling probabilities and introduces bias in the mini-batch estimator. In [33], the normalization techniques are developed to deal with this issue so that the feature learning will not give priority to the more frequently sampled nodes. As a result, GraphSAINT effectively solves the problems of instability and non-convergence faced in the training process, and obtains a good performance improvement on the classification task.

However, experiments show that the stability problem of GraphSAINT in the training process appears when GraphSAINT is applied to solve the link prediction task [35–38] in different application areas [39–43] on the citation dataset of the standard Open Graph Benchmark (OGB) [44]. Link prediction is widely used in many areas such as recommendation system [45–47], biological networks [48] and knowledge graph completion [49]. Different with node classification, the main task of link prediction is to judge whether two nodes in a network are likely to have a link. This stability problem in the training process means that the normalization techniques in [33] are insufficient to improve the training quality for the link prediction task. Thus, it is difficult to avoid falling into non-convergence during training, which appears that the value of the training loss suddenly rises and remains forever unchanged. This is a problem that has a big impact on the GNN model development.

From the analysis of the CNN training method, we had some new discovery in training stability. Stochastic gradient descent and its variants such as momentum [50] and Adagrad [51] have been widely used to train the neural networks. The training process is complicated. Besides, as the network gets deeper, small changes to the network parameters will amplify [52]. In the process of constantly adapting to the new distribution, the distributions of layers' inputs present a problem called covariate shift [53–58], which is harmful for the neural network convergence. Ioffe and Szegedy proposed the Batch normalization (BN) mechanism to reduce the internal covariate shift and accelerate the convergence of the deep neural nets [52]. This mechanism makes use of the mean and variance to normalize the data values over each mini-batch, which allows us to set a

higher learning rate and drop the Dropout [59]. However, this effective BN mechanism has not been widely used in GNNs due to the fewer network layers [12]. Nowadays, as the graphs become larger and the tasks become more complex, the GNN models are more complicated and the difficulty of training the GNN models is also increasing rapidly, which leads to the stability problem during training. Thus, the application of the BN mechanism is of great significance to the robustness of training on large graphs.

Therefore, in order to solve the stability problem in the training process of the link prediction task, we propose an improved GraphSAINT method by adjusting the BN strategy to the special training and inference process of GraphSAINT based on the OGB in this paper. By applying the normalization strategy during training, we achieve the elimination of the instability during training successfully. Moreover, we also realize a reduction in the training time and gain an increase in accuracy under the premise of maintaining the original link prediction accuracy. The effectiveness of our method is validated by the citation dataset of the OGB.

Inspired by [60], the paper is organized as follows: Firstly, Sect. 2 is the related work about the GraphSAINT, especially its sampling strategy and the typical batch normalization techniques; Next, Sect. 3 describes our improved GraphSAINT and the training method; Then, Sect. 4 shows the comparative experiment results based on the citation dataset; At last, conclusions are given in Sect. 5.

2 Related Work

2.1 The Sampling Strategy of GraphSAINT

GCN achieves one-hop neighbors' information aggregation by using the adjacent matrix [31, 32]. However, because of the using of the adjacent matrix, when a new node is added into the graph, we must adapt the adjacent matrix and re-train the GCN model based on the new adjacent matrix of the adapted graph data to obtain all node's new embeddings. Therefore, as mentioned in Sect. 1, GCN causes a lack of generalization performance for unseen nodes. Besides, it has a high time cost. To overcome this shortcoming, GraphSAGE is proposed [34]. In this method, the neighbors of the target node are sampled and a new aggregation function is learned to aggregate neighbor nodes and generate the embedding vector of a target node, which avoids applying the adjacent matrix and reduces training costs by sampling the nodes of each layer of GNN.

Furthermore, compared with GraphSAGE, GraphSAINT makes it possible to solve the learning tasks on the large graphs by designing a sampler called SAMPLE to obtain subgraphs [33]. Besides, this subgraph sampling method can also improve the generalization performance like GraphSAGE. With the help of the sampling strategy, GraphSAINT can deal with the Neighbor Explosion problem better.

In order to preserve the connectivity features of the graph, bias in the mini-batch estimation will almost inevitably introduced by the sampler. Therefore, in [33], the self-designed normalization techniques are introduced to eliminate deviations. The key step is to estimate the sampling probability of each node, edge, and subgraph.

The sampling probability distribution $P(u)$ of the node u is

$$P(u) \propto \left\| \tilde{\mathbf{A}}_{:,u} \right\|^2 \quad (1)$$

where \mathbf{A} is the adjacency matrix and $\tilde{\mathbf{A}}$ is the normalized one, that is $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$, \mathbf{D} is the diagonal degree matrix.

The sampling probability distribution $P_{u,v}^{(l)}$ of the edge (u, v) in the l^{th} GCN layer is

$$P_{u,v}^{(l)} \propto \frac{1}{\text{deg}(u)} + \frac{1}{\text{deg}(v)} \quad (2)$$

The sampling probability distribution $P_{u,v}$ of the subgraph is

$$P_{u,v} \propto \mathbf{B}_{u,v} + \mathbf{B}_{v,u} \quad (3)$$

where $\mathbf{B}_{u,v}$ can be interpreted as the probability of a random walk to start at u and end at v in L hops, $\mathbf{B}_{v,u}$ can be interpreted as the probability of a random walk to start at v and end at u in L hops, $\mathbf{B} = \tilde{\mathbf{A}}^L$, L means L layers which can be represented as a single layer with edge weights. Thus, the sampling probabilities of each node, edge, and subgraph are all well-estimated. Then, the subgraphs obtained by sampling will be used for GraphSAINT training.

2.2 The Typical Normalization Technique

The normalization techniques are proposed to eliminate the Internal Covariate Shift, which is caused by the change in the distributions of internal nodes of a deep network in the process of training, and offer the faster training [53, 58].

The typical normalization technique for mini-batch presented in [52] basically follows the mathematical statistics: Firstly, the mini-batch mean is calculated based on the values of each data point over a mini-batch; Next, the mini-batch variance is calculated based on the mini-batch mean; Then, each data point over a mini-batch can be normalized by subtracting the mini-batch mean and then dividing by the mini-batch variance; Finally, the scale and shift parameters are introduced and learned for each data point over a mini-batch.

3 Methodology

As mentioned in Sect. 1, the original normalization techniques of GraphSAINT, which are effective for the node classification task, are insufficient to improve the training quality for the link prediction task. Therefore, an improved GraphSAINT training algorithm is proposed.

Since the sampled subgraphs in GraphSAINT are based on the connectivity rules of the nodes, it can get the edge sampling probability with the smallest variance. In contrast, for node selection, it uses the Random node sampler. Therefore, the node feature data of each sampled subgraph do not obey the standard normal distribution. Assume the graph dataset to be processed is a whole graph $\zeta = (V, \xi)$ with N nodes $v \in V$, edges $(v_i, v_j) \in \xi$. For the node v_i in a sampled subgraph ζ_s of ζ according to SAMPLE, its feature $h_{i,s}$ has d elements. In order to normalize the distributions of the inputs to reduce the internal covariate shift, the input node feature vector can be normalized by

$$\hat{h}_{i,s} = \frac{h_{i,s} - \mu_s}{\sqrt{\sigma_s^2}}$$

$$\mu_s = \frac{1}{d} \sum_{i=1}^d h_{i,s} \sigma_s^2 = \frac{1}{d} \sum_{i=1}^d (h_{i,s} - \mu_s)^2 \quad (4)$$

where μ_s and σ_s^2 are the mean and the variance for the node v_i in the node feature dimension and are computed over the training data set.

Therefore, by means of the node-wise normalization technology in each subgraph, each node feature vector is normalized by making its mean zero and variance 1. Besides, based on the typical normalization principle, the training time can also be effectively shortened by applying the node-wise normalization technology in each subgraph.

Algorithm 1 The improved GraphSAINT training algorithm on citation dataset in OGB

Input: Training graph $\zeta = (V, \xi)$; GraphSAINT sampler;

Output: Loss and accuracy; GCN model with trained weights;

- 1: Pre-processing: Directed graph to undirected graph; Sampled subgraph ζ_s of ζ
 - .
 - 2: **for** each subgraph ζ_s **do**
 - 3: GCN construction on ζ_s and **before** each active layer RELU **do**
 - 4: Apply the normalization technology on the output of the convolutional layer
 - 5: Forward propagation to calculate loss according to MRR.
 - 6: Backward propagation to update weights.
 - 7: **end for**.
-

The whole training process of the improved GraphSAINT is illustrated in Algorithm 1. Before the training starts, we perform a pre-processing on ζ to convert the directed graph to the undirected graph and obtain the sampled subgraph ζ_s with the given SAMPLE [33]. Then an iterative training process is conducted via SGD to update model weights. Each iteration uses an independently subgraph ζ_s . Next, the original GCN is modified through applying the normalization technology on the output of the convolutional layer, which is also the original input of the RELU layer. Finally, the modified GCN on ζ_s is built to generate embeddings and the loss can then be calculated according to Mean Reciprocal Rank (MRR). In MRR, the score of the first matched result is 1, the score of the second matched result is 1/2, and the score of the n th matched result is 1/n. If there is no matching sentence, the score is 0. The final score is the sum of all scores.

As mentioned in Sect. 2.1, GraphSAINT uses the subgraphs obtained by the subgraph sampling method for training, while it uses the whole graph data to calculate the output result during inference. Therefore, the normalization operation can be added independently during inference.

4 Experiments

In this section, in order to verify the effectiveness of the improved GraphSAINT algorithm, we choose the Link prediction task based on the citation dataset of OGB (ogbl-citation). The ogbl-citation dataset is a directed graph and can be viewed as a ‘subgraph’ of the citation network called MAG [61]. In this dataset, each node represents a paper, whose title and abstract are encoded into a 128-dimensional word2vec features, and each directed edge indicates the citation relationship between two papers.

The link prediction task means that we need to predict missing citations based on the existing citations on the graph. Two of each source paper’s references are randomly dropped and the model is required to achieve the ranking of the missing two references in front of other 1000 references that are also randomly sampled from all the papers and not referenced by the source papers. According to this, MRR is chosen as the evaluation metric [44]. Besides, we use the two dropped edges of all source papers respectively for validation and testing. Naturally, the training set contains the rest of the edges.

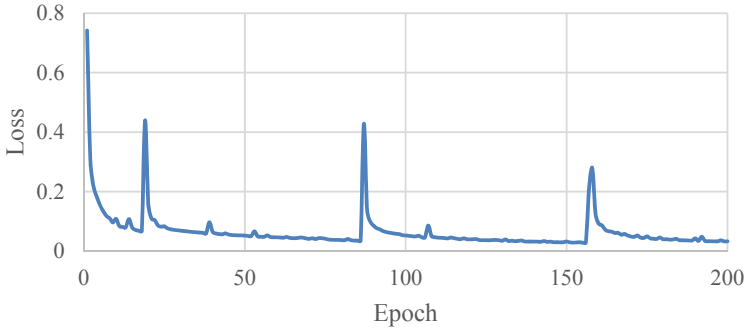
Table 1. Results for GraphSAINT on citation dataset.

GraphSAINT (Citation)		MRR		
		Training	Validation	Test
Traditional (official)		0.8626 ± 0.0046	0.7933 ± 0.0046	0.7943 ± 0.0043
Traditional	Convergence	0.8690	0.8031	0.8048
	Non-convergence	0.0010	0.0010	0.0010
Improved		0.9001 ± 0.0014	0.8335 ± 0.0020	0.8344 ± 0.0023

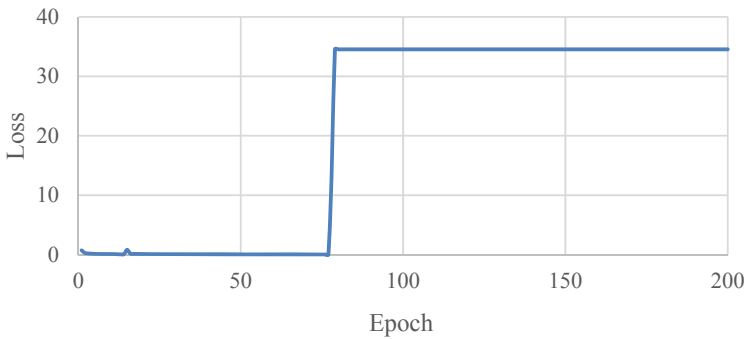
The official results of the traditional GraphSAINT on the citation dataset are given in Table 1: the MRR value of the training set is 0.8626 ± 0.0046 , the MRR value of the validation set is 0.7933 ± 0.0046 and the MRR value of the test set is 0.7943 ± 0.0043 . However, in the training of our recurrence experiment of the traditional GraphSAINT, we found that GraphSAINT algorithm will not converge in training with a probability range from 0.1 to 0.4. For a RUN where loss converges as shown in Fig. 1(a), the MRR results are consistent with the official results as shown in Table 1, i.e., the training result can reach about 0.8690, the validation result can reach about 0.8031 and the test result can also reach about 0.8048. But for a RUN where loss does not converge, the MRR results are also shown in Table 1 and the loss curve trained under different epochs is basically as shown in Fig. 1(b).

As shown in Fig. 1, we can see that one RUN contains 200 epochs. Besides, in Fig. 1(b), after the 78th epoch, the loss is suddenly and sharply increased to 34.5388 and remains unchanged. Therefore, some measures need to be taken to solve the problem of non-convergence in the training process.

After applying the improved GraphSAINT, our loss curve during training is shown by the solid line in Fig. 2. We can see that the solid loss curve is convergent. Besides, compared with the dotted line in Fig. 2, which is the loss curve during training of the traditional GraphSAINT as shown in Fig. 1(a), the improved GraphSAINT has a more stable convergence during training and a faster convergence rate. Moreover, all the three MRR values have an improvement as shown in Table 1: the training results can reach 0.9001 ± 0.0014 , the validation results can reach 0.8335 ± 0.0020 and the test results can also reach 0.8344 ± 0.0023 . Thus, the effectiveness of our improved GraphSAINT is verified.



(a)



(b)

Fig. 1. The loss curve during training of the traditional GraphSAINT (a) with convergence (b) without convergence

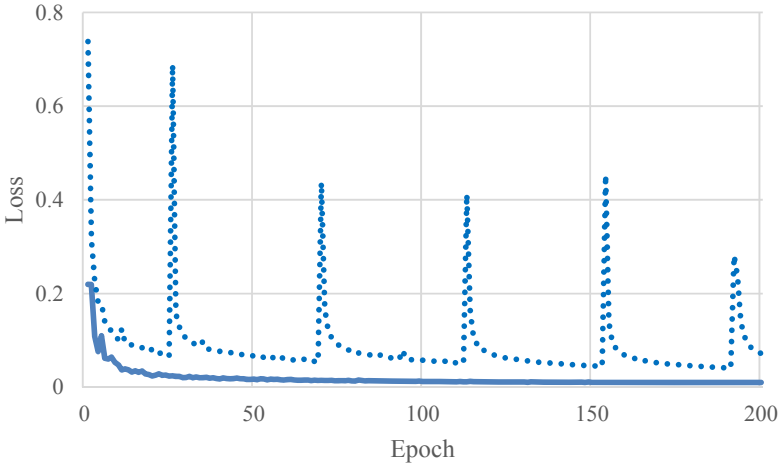


Fig. 2. The loss curve during training of the improved GraphSAINT (the solid line) and the traditional GraphSAINT with convergence in Fig. 1(a) (the dotted line).

5 Conclusions

The stability problem during training of graph neural network is crucial. In this paper, we focus on this stability problem and propose an improved GraphSAINT by applying the normalization strategy. The proposed method not only improves the robustness of the training process of the GraphSAINT, but also accelerates the convergence of the model. In the future, more attention will be paid to the distributed training methods for large graph datasets.

Acknowledgement. This research was funded by the Fifth Innovative Project of State Key Laboratory of Computer Architecture (ICT, CAS) under Grant No. CARCH 5403.

References

1. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Netw.* **20**(1), 61–80 (2009)
2. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.* **30**(3), 83–98 (2013)
3. Zang, C., Cui, P., Faloutsos, C.: Beyond Sigmoids: the NetTide model for social network growth, and its applications. In: *Proceedings of the International Conference on Association for Computing Machinery’s Special Interest Group on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, pp. 2015–2024 (2016)
4. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1), 40–51 (2007)

5. Cho, K., et al.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp. 1724–1734 (2014)
6. Brockschmidt, M., Chen, Y., Cook, B., Kohli, P., Tarlow, D.: Learning to decipher the heap for program verification. In: Proceedings of the International Conference on Machine Learning Workshop on Constructive Machine Learning, Lille, France, pp. 1–5 (2015)
7. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. In: Proceedings of the 12th International Conference on Information and Knowledge Management, Louisiana, USA, New Orleans, pp. 556–559 (2003)
8. Chen, J., Ma, T., Xiao, C.: FastGCN: fast learning with graph convolutional networks via importance sampling. In: Proceedings of the 6th International Conference on Learning Representations, Vancouver, Canada, pp. 1–15 (2018)
9. Lim, J., Ryu, S., Park, K., Choe, Y.J., Ham, J., Kim, W.Y.: Predicting drug–target interaction using a novel graph neural network with 3D structure-embedded graph representation. *J. Chem. Inf. Model.* **59**(9), 3981–3988 (2019)
10. Fan, W., et al.: Graph neural networks for social recommendation. In: Proceedings of the International World Wide Web Conferences, San Francisco, United States, pp. 417–426 (2019)
11. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, USA, Honolulu, pp. 346–353 (2019)
12. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(1), 4–24 (2021)
13. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. In: Proceedings of the International Conference on Learning Representations, Caribe Hilton, San Juan, Puerto Rico, pp. 1–20 (2016)
14. Dai, H., Kozareva, Z., Dai, B., Smola, A., Song, L.: Learning steady-states of iterative algorithms over graphs. In: Proceedings of the International Conference on Machine Learning, Vienna, Austria, pp. 1114–1122 (2018)
15. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: Proceedings of the International Conference on Learning Representations, Banff, Canada, pp. 1–14 (2014)
16. Levie, R., Monti, F., Bresson, X., Bronstein, M.: CayleyNets: graph convolutional neural networks with complex rational spectral filters. *IEEE Trans. Signal Process.* **67**(1), 97–109 (2017)
17. Micheli, A.: Neural network for graphs: a contextual constructive approach. *IEEE Trans. Neural Netw.* **20**(3), 498–511 (2009)
18. Gilmer, J., Schoenholz, S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the International Conference on Machine Learning, Sydney, Australia, pp. 1263–1272 (2017)
19. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.: Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii, pp. 5115–5124 (2017)
20. Zhang, J., Shi, X., Xie, J., Ma, H., King, I., Yeung, D.Y.: GaAN: gated attention networks for learning on large and spatiotemporal graphs. In: The Conference on Uncertainty in Artificial Intelligence, Monterey, California, USA, pp. 1–11 (2018)
21. Chen, J., Zhu, J., Song, L.: Stochastic training of graph convolutional networks with variance reduction. In: Proceedings of the International Conference on Machine Learning, Vienna, Austria, pp. 941–949 (2018)

22. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: Proceedings of the Conference on Neural Information Processing Systems, Montréal, Canada, pp. 4801–4811 (2018)
23. Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S., Hsieh, C.J.: Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the International Conference on Association for Computing Machinery’s Knowledge Discovery and Data Mining, Anchorage, AK, USA, pp. 257–266 (2019)
24. Yu, W., et al.: Learning deep network representations with adversarially regularized autoencoders. In: Proceedings of the Association for Computing Machinery’s Association for the National Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 2663–2671 (2018)
25. Cao, N., Kipf, T.: MolGAN: an implicit generative model for small molecular graphs. In: Proceedings of the International Conference on Machine Learning workshop on Theoretical Foundations and Applications of Deep Generative Models, Vienna, Austria, pp. 1114–1122 (2018)
26. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. In: Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Stockholm, pp. 2609–2615 (2018)
27. Ma, T., Chen, J., Xiao, C.: Constrained generation of semantically valid graphs via regularizing variational autoencoders. In: Proceedings of the Conference on Neural Information Processing Systems, Montréal, Canada, pp. 7110–7121 (2018)
28. Seo, Y., Defferrard, M., Vandergheynst, P., Bresson, X.: Structured sequence modeling with graph convolutional recurrent networks. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) ICONIP 2018. LNCS, vol. 11301, pp. 362–373. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04167-0_33
29. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Stockholm, pp. 3634–3640 (2018)
30. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the Association for Computing Machinery’s Association for the National Conference on Artificial Intelligence, Honolulu, Hawaii, USA, pp. 922–929 (2019)
31. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations, Toulon, France, pp. 1–14 (2017)
32. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the Conference on Neural Information Processing Systems, Barcelona, Spain, pp. 3844–3852 (2016)
33. Zeng, H., Zhou, H., Srivastava, A., Kannan, R., Prasanna, V.: GraphSAINT: graph sampling based inductive learning method. In: Proceedings of the 8th International Conference on Learning Representations, pp. 1–19. Formerly Addis Ababa Ethiopia: Virtual Conference (2020)
34. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the Conference on Neural Information Processing Systems, California, USA, Long Beach, pp. 1024–1034 (2017)
35. Lü, L., Zhou, T.: Link prediction in complex networks: a survey. *Physica A Stat. Mech. Appl.* **390**(6), 1150–1170 (2011)
36. Zhang, M., Chen, Y.: Link prediction based on graph neural networks. In: Proceedings of the Conference on Neural Information Processing Systems, Montréal, Canada, pp. 5165–5175 (2018)

37. Lichtenwalter, R.N., Chawla, N.V.: Vertex collocation profiles: subgraph counting for link analysis and prediction. In: Proceedings of the 21st International Conference on World Wide Web, Lyon, France, pp. 1019–1028 (2012)
38. Li, R.H., Jeffrey, X.Y., Liu, J.: Link prediction: the power of maximal entropy random walk. In: Proceedings of the 20th International Conference on Association for Computing Machinery's Information and Knowledge Management, Glasgow, UK, pp. 1147–1156 (2011)
39. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the International Conference on Association for Computing Machinery's Knowledge Discovery and Data Mining, London, United Kingdom, pp. 974–983 (2018)
40. Chen, H., Liang, G., Zhang, X.L., Giles, C.L.: Discovering missing links in networks using vertex similarity measures. In: Proceedings of the Twenty-Seventh Conference on Annual Association for Computing Machinery's Applied Computing, Trento, Italy, pp. 138–143 (2012)
41. Monti, F., Bronstein, M., Bresson, X.: Geometric matrix completion with recurrent multi-graph neural networks. In Proceedings of the Conference on Neural Information Processing Systems, California, USA, Long Beach, pp. 3697–3707 (2017)
42. Ahn, M.W., Jung, W.S.: Accuracy test for link prediction in terms of similarity index: the case of WS and BA models. *Physica A Stat. Mech. Appl.* **429**(1), 3992–3997 (2015)
43. Hoffman, M., Steinley, D., Brusco, M.J.: A note on using the adjusted Rand index for link prediction in networks. *Soc. Netw.* **42**, 72–79 (2015)
44. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. In: Proceedings of the 34th Conference on Neural Information Processing Systems, Vancouver, Canada, pp. 1–34 (2020)
45. Aiello, L.M., Barrat, A., Schifanella, R., Cattuto, C., Markines, B., Menczer, F.: Friendship prediction and homophily in social media. *Assoc. Comput. Machinery's Trans. Web* **6**(2), 1–33 (2012)
46. Tang, J., Wu, S., Sun, J.M., Su, H.: Cross-domain collaboration recommendation. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, Beijing, China, pp. 1285–1293 (2012)
47. Akcora, C.G., Carminati, B., Ferrari, E.: Network and profile based measures for user similarities on social networks. In: Proceedings of the IEEE International Conference on Information Reuse & Integration, Las Vegas, NV, USA, pp. 292–298 (2011)
48. Turki, T., Wei, Z.: A link prediction approach to cancer drug sensitivity prediction. *BMC Syst. Biol.* **11**(5), 13–26 (2017)
49. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. *Proc. IEEE* **104**(1), 11–33 (2016)
50. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: Proceedings of the 30th International Conference on Machine Learning, Georgia, USA, Atlanta, pp. 1139–1147 (2013)
51. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12**(61), 2121–2159 (2011)
52. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, pp. 448–456 (2015)
53. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Stat. Plan. Infer.* **90**(2), 227–244 (2000)
54. Wiesler, S., Ney, H.: A convergence analysis of log-linear training. In: Proceedings of the Conference on Neural Information Processing Systems, Granada, Spain, pp. 657–665 (2011)

55. Wiesler, S., Richard, A., Schlüter, R., Ney, H.: Mean-normalized stochastic gradient for large-scale deep learning. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Florence, Italy, pp. 180–184 (2014)
56. Nair, V.H., Geoffrey, E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 30th International Conference on Machine Learning, Haifa, Israel, pp. 807–814 (2010)
57. Bengio, Y., Glorot, X.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, pp. 249–256 (2010)
58. Raiko, T., Valpola, H., LeCun, Y.: Deep learning made easier by linear transformations in perceptrons. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, pp. 924–932 (2012)
59. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
60. Misra, S.: A Step by Step Guide for Choosing Project Topics and Writing Research Papers in ICT Related Disciplines. In: Misra, S., Muhammad-Bello, B. (eds.) ICTA 2020. CCIS, vol. 1350. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69143-1_55
61. Wang, K., Shen, Z., Huang, C., Wu, C., Dong, Y., Kanakia, A.: Microsoft academic graph: when experts are not enough. *Quant. Sci. Stud.* **1**(1), 396–413 (2020)