



HAL
open science

Graph Neural Networks to Advance Anticancer Drug Design

Asmaa Rassil, Hiba Chougrad, Hamid Zouaki

► **To cite this version:**

Asmaa Rassil, Hiba Chougrad, Hamid Zouaki. Graph Neural Networks to Advance Anticancer Drug Design. 16th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2020, Neos Marmaras, Greece. pp.216-226, 10.1007/978-3-030-49161-1_19. hal-04050573

HAL Id: hal-04050573

<https://inria.hal.science/hal-04050573v1>

Submitted on 29 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Graph Neural Networks to Advance Anticancer Drug Design

Asmaa Rassil¹, Hiba Chougrad², and Hamid Zouaki³

¹ Faculty of science, University Chouaib Doukkali, El Jadida, Morocco
Laboratory of Computer Science and Mathematics and their Applications.
asmaarassil@gmail.com

² National School of Applied Sciences, University Sidi Mohamed Ben Abdellah, Fez, Morocco
Laboratory of Intelligent Systems, Georesources and Renewable Energies.
chougrad.hiba@gmail.com

³ Faculty of science, University Chouaib Doukkali, El Jadida, Morocco
Laboratory of Computer Science and Mathematics and their Applications.
hamid.zouaki@yahoo.fr

Abstract. Predicting the activity of chemical compounds against cancer is a crucial task. Active chemical compounds against cancer help pharmaceutical drugs producers in the conception of anticancer medicines. Still the innate way of representing chemical compounds is by graphs, the machine learning algorithms can not handle directly the anticancer activity prediction problems. Dealing with data defined on a non-Euclidean domain gave rise to a new field of research on graphs. There has been many proposals over the years, that tried to tackle the problem of representation learning on graphs. In this work, we investigate the representation power of Node2vec for embedding learning over graphs, by comparing it to the theoretical framework Graph Isomorphism Network (GIN). We prove that GIN is a deep generalization of Node2vec. We then exert the two models Node2vec and GIN to extract regular representations from chemical compounds and make predictions about their activity against lung and ovarian cancer.

Keywords: Graph Neural Networks · Transductive learning · Inductive learning · Representation vectors · Anticancer activity prediction.

1 Introduction

Application domains of Artificial Intelligence (AI) became concerned in the last few years with treating complex problems defined on irregular data structure as graphs. The conception of pharmaceutical drugs [18, 29, 9] through the prediction of chemical compounds activities against cancer [23, 22] need more than tabular representations to be solved. The natural way of modeling molecular data for an anticancer activity prediction problem is by graphs. Representing chemical compounds in the shape of graphs keep the structure of the data unchanged, and so no information from the initial collected data will get lost. Graphs are

2 Problem Formulation

A graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is defined by the vertex set $\mathbf{V} = \{v_i : i = 1, 2, \dots, N\}$ and the edges $e_{ij} \in \mathbf{E}$ connecting pairs of nodes v_i and v_j in the graph. For each node v in the graph, we note its neighbors set by $\mathcal{N}(v) = \{u : (u, v) \in E\}$. The adjacency matrix $\mathbf{A} = (a_{ij})_{1 \leq i \leq N, 1 \leq j \leq N}$ outlines the graph structure by setting $a_{ij} = 1$ for connected nodes $e_{ij} \in E$ and $a_{ij} = 0$ if $e_{ij} \notin E$.

2.1 Graph Embedding

Graph embedding frameworks [3, 1, 6, 19, 21, 26] are unsupervised approaches for learning on graph structured data. The main purpose of network embedding is to map input graphs to a regular low dimensional space where the use of standard machine learning algorithms become feasible [14, 8, 5]. To preserve the graph topology and the structural information on graphs in the embedding space, the network embedding variants propose pairwise similarity measures upon the graph structure $S_G(u, v)$ and learn the node embeddings by preserving the defined similarity measure in the embedding space $S_E(e_u, e_v)$ through the following loss function:

$$L[S_G(u, v), S_E(e_u, e_v)]$$

where $e_v \in \mathbf{R}^d$ is the corresponding embedding vector of the node $v \in \mathbf{V}$ in the embedding space $E = \mathbf{R}^d$ and $L(\cdot)$ is a loss function.

Two principal branches of graph embedding are proposed in the literature [14] matrix factorization [3, 1, 6, 19] and random walk [20, 12]. Matrix factorization approaches define the proximity measure over the graph $S_G(u, v)$ either by an adjacency-based pairwise proximity measure ($S_G(u, v) = A_{uv}$) [1], or by multi-hop pairwise proximity measure captured by the k-th power of the adjacency matrix of the graph ($S_G(u, v) = A_{uv}^k$) [6] (see figure 2). The adjacency-based approaches push adjacent nodes $(u, v) \in E$ in the graph $S_G(u, v) = A_{uv} = 1$ to be equal in the embedding space $S_E(u, v) = e_v^T e_u = 1$ through the following loss function:

$$\|A_{uv} - e_v^T e_u\|_2^2 \quad (1)$$

Multi-hop pairwise proximity measures ($S(u, v) = A_{uv}^k$) [6] push adjacent nodes captured by A^k to be equal in the embedding space $S_E(u, v) = e_v^T e_u$ through the following loss function:

$$\|A_{uv}^k - e_v^T e_u\|_2^2 \quad (2)$$

Thus, nodes located in k-hop distance from each other, will have similar embedding vectors. Random walk [20, 12] approaches for graph embedding are heuristic methods for embedding learning. To learn embeddings, authors in [20, 12] optimize the probability of visiting nodes v that tend to co-occur in random walks on the graph starting from a node u

$$p(v | u) = \frac{\exp(e_v^T e_u)}{\sum_{v_i \in \mathbf{V}} \exp(e_v^T e_{v_i})} \quad (3)$$

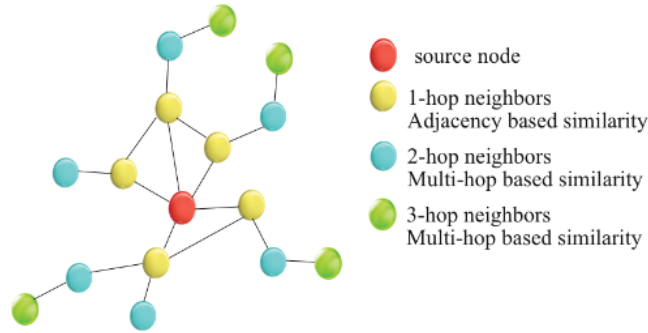


Fig. 2. For every node (i.e source node), matrix factorization approaches learn embedding vectors of source nodes based on information from local neighbors (i.e 1-hop neighbors) or distant neighbors.

Matrix factorization gives a deterministic definition of similarity upon the graph. Otherwise, random walk methods adopt a stochastic proximity measure based on randomly generated walks on the graph. Through two searching parameters p and q , Node2vec [12] controls the propagation of the generated walk. For $p > \max(q, 1)$ we reduce the probability of sampling an already visited node in the next two steps, therefore distant nodes from a source node u are pushed to have similar embeddings [12]. However, for $p < \min(q, 1)$, the walk stays local, then closest nodes to the source are pushed to have similar embeddings (figure 3).

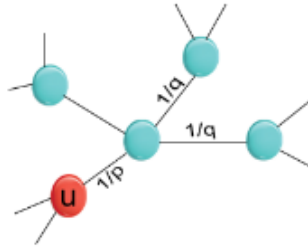


Fig. 3. The parameters p and q control the propagation of the generated walk on the graph. For small values of p , local neighbors are likely to be visited. However, when setting the parameter q to a small value, distant nodes tend to co-occur in the random walk

2.2 Graph Neural Networks

Contrary to graph embedding, graph neural networks (GNNs) [2, 13, 11, 7, 28] are deep and inductive approaches for representation learning on graphs. Through an end-to-end network, GNNs learn jointly the embeddings or representation vectors of the nodes and solve the defined problem on the graph structure. Thus, the representation vectors are learned based on the intrinsic topology of the graph and the task we aim to solve. The representation learning process involves two steps (i) aggregating neighboring nodes then (ii) combining them with the node being processed through the following recursive schema:

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}(\{h_u^{(k-1)} : u \in \mathcal{N}(v)\}) \quad (4)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}(h_v^{(k-1)}, a_v^{(k)}) \quad (5)$$

where $h_v^{(k)}$ is the learned representation vector of node v in the k th layer of the network. Variants of aggregation and combination functions give the GNN variants. Graph Isomorphism Network (GIN) [25] is a theoretical variant of GNNs which combines the aggregation and combination step in the following formula:

$$h_v^{(k)} = \text{MLP}^{(k)}[(1 + \epsilon^{(k)})h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}] \quad (6)$$

where $\epsilon^{(k)}$ is a model parameter. Graph Isomorphism Network (GIN) [25] is as powerful as the Weisfeiler-Lehman (WL) test of isomorphism [24]. Thus, the learned graph representation vectors are as powerful as WL in capturing differences of graph topologies.

3 Deep Generalization of Node2vec

Learning embedding vectors by maximizing the likelihood of visiting the neighbors generated through a combination of Breadth-first Sampling (BFS) and Depth-first Sampling (DFS) strategies bestow Node2vec [12] with an impressive representation power.

When setting the parameter p to a small value (figure 3), nodes belonging to similar communities [27] in a graph are forced to learn approximately the same embeddings. However, when setting the parameter q to a small value (figure 3), nodes representing the same structural role in different communities [15] are pushed to learn similar embeddings. Therefore, according to the given task Node2vec [12] is able to learn suitable embeddings.

In this work, we demonstrate the analogy between Node2vec [12] and the powerful theoretical framework GIN [25]. And, we prove that Node2vec is closely equivalent to the Weisfeiler-Lehman (WL) test of isomorphism [24].

We consider a shallow neural network with one hidden layer \mathbf{h} as described in figure 4. Given a node u , we maximize the probability of visiting a node $v \in \mathcal{N}(u)$ by generating a random walk starting from u . We initialize the embedding of u

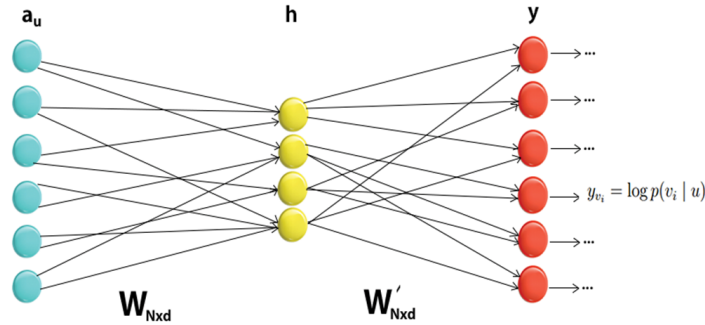


Fig. 4. Unsupervised representation Learning on graphs. starting from a source node u defined by its corresponding column in the adjacency matrix $\hat{A} = A + I_N$, we update the embedding of u by maximizing the probability of visiting the neighbors $v \in \mathcal{N}(u)$ generated by a random walk. The embedding we are looking for are defined by the parameters W and W' of the model.

by its corresponding column $a_u \in \mathbf{R}^N$ in the adjacency matrix $\hat{A} = A + I_N$, then we feed it as input to the network (figure 4). We denote the $N \times d$ matrices corresponding to the model parameters in the input-to-hidden and the hidden-to-output sections of the network by W and W' respectively. We assume that W and W' are embedding matrices of the input graph. Each row of W and W' is a d -dimensional embedding vector e_v of a node $v \in \mathbf{V}$.

In the forward pass,

$$W^T a_u = \sum_{v \in \mathcal{N}(u) \cup u} W_{\cdot, k}^T = e_u + \sum_{v \in \mathcal{N}(u)} e_v^T = h_u = \mathbf{h}$$

the updated embedding vector $h_u \in \mathbf{R}^d$ of the input node u is a summation of the embedding vectors e_u of u and its surrounding neighbors $\{e_v : v \in \mathcal{N}(u)\}$ given by the embedding matrix W . Node2vec adopts a sum-aggregator to aggregate neighboring nodes and utilizes the collected information to update the initial node embedding a_u . Thus, we may proclaim that node2vec is a shallow approximation of the deep and inductive Graph Isomorphism Network GIN (Eq.6).

The output of the network before applying activation is,

$$z_i = W'_{\cdot, i}^T \mathbf{h} = W'_{\cdot, i}^T h_u = h_{v_i}^T h_u \quad \forall i = 1, \dots, N$$

using a log-softmax activation function $\Phi(\cdot)$, we obtain (Eq.3) corresponding to Node2vec:

$$\Phi(h_{v_i}^T h_u) = \log p(v_i | u) = \log \frac{\exp(h_{v_i}^T h_u)}{\sum_{v \in \mathbf{V}} \exp(h_v^T h_u)} \quad (7)$$

We can conclude that, Node2vec learns embedding vectors of the graph nodes using approximately the same process of one layer GIN. Now, considering that

GIN can capture and distinguish between different graph topologies as would the WL test of isomorphism, we may assert that embeddings learned by Node2vec will also be powerful in capturing differences of graph topologies.

4 Experiments and Results

To predict anticancer activity of chemical compounds against lung and ovarian cancer, we investigate the representation power of Node2vec for embedding learning on graphs and two variants of the inductive Graph Isomorphism Network (GIN) using two benchmark datasets NCI-1 and NCI-109 for lung and ovarian cancer respectively. In the following paragraphs we provide a description of the datasets, details about the models architecture and configurations, and the obtained results achieved by each model for predicting anticancer activity.

4.1 Datasets

The NCI-1 and NCI-109 graph datasets [23, 22] are provided by the National Cancer Institute. Each NCI dataset contains thousands of chemical compounds modelled as graphs. The atoms are represented as nodes and bonds as edges in the graph representation of the chemical compounds (figure 5). Statistics about NCI-1 and NCI-109 graph datasets are reported in Table 1.

NCI-1 dataset contains two categories of chemical compounds active and non-active against the non-small cell lung cancer. Active compounds are labeled by +1 and non-active by -1. NCI-109 dataset contains two categories of chemical compounds active and non-active against ovarian cancer. Active compounds are labeled by +1 and non-active by -1.

Table 1. Datasets statistics The table summarizes the number of graphs, the average number of nodes, the average number of edges and the number of node labels in NCI-1 and NCI-109 datasets.

Datasets	Dataset Statistics			
	<i>Graphs</i>	<i>Nodes(avg.)</i>	<i>Edges(avg.)</i>	<i>Node Labels</i>
NCI-1	4110	29.8	32.3	37
NCI-109	4127	29.6	32.1	38

4.2 Implementation Details and Results

To predict the anticancer activity of chemical compounds against lung and ovarian cancer, we train the embedding model Node2vec [12] on NCI-1 and NCI-109 datasets with the following hyper-parameters: embedding dimensionality

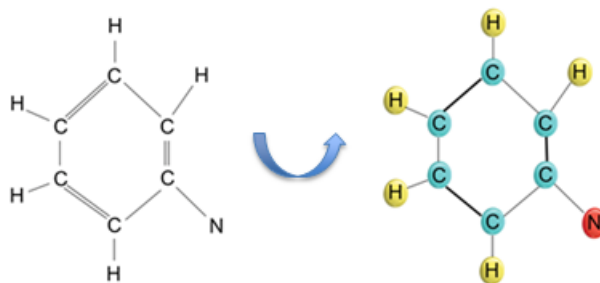


Fig. 5. Graph representation of a chemical compound The left figure represents a chemical compound and the right figure represents its corresponding graph representation where C refers to carbon atom, H to Hydrogen atom and N to Nitrogen atom.

$\in \{7, 12\}$, random walk length $\in \{3, 4\}$, size of neighbors set (i.e the context size) $\in \{5, 7\}$ and for each node we generate 10 random walks.

To train the deep Graph Isomorphism Network (GIN) [25] on NCI-1 and NCI-109 datasets, we use two variants of the model using (1) multilayer perceptron and (2) single-layer perceptron as combination functions of neighboring node representations. To build the anticancer activity prediction model, we use five GIN layers each followed by batch normalization [16]. We learn the model parameters by optimizing the cross-entropy loss between the predicted labels and the true labels using the Adam optimizer [17]. We tune the following hyper-parameters: batch size $\in \{32, 42\}$, hidden units $\in \{16, 32\}$, learning rate $\in \{0.01, 0.001\}$ and dropout coefficient $\in \{0, 0.2\}$. All implementations were carried out using PyTorch Geometric [10] and the Cuda enabled GPU 12GB NVIDIA Tesla K80.

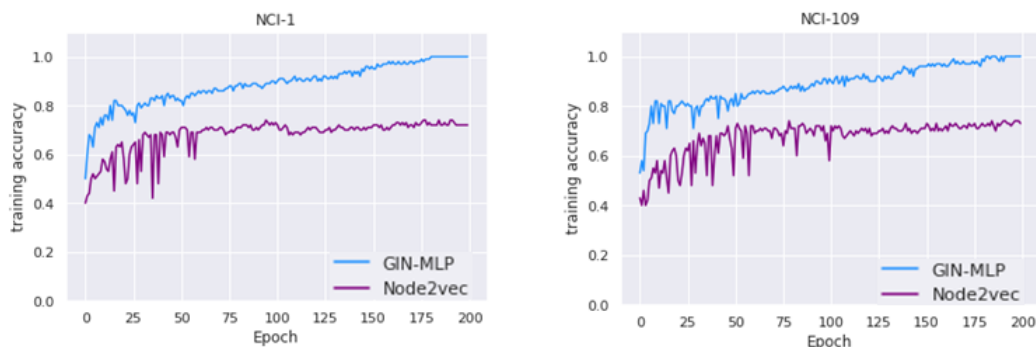


Fig. 6. Training set performance of GIN-MLP compared to Node2vec on NCI-1 and NCI-109 datasets over 100 epochs

Table 2. Binary-classification results(%) The table reports the average and standard deviations of accuracies on binary-graph classification

Datasets	Representation Learning Models		
	<i>Node2vec</i> [12]	<i>GIN-1-Layer</i> [25]	<i>GIN-MLP</i> [25]
NCI-1	54.9 \pm 1.2	83.0 \pm 3.5	83.4 \pm 2.5
NCI-109	51.4 \pm 2.3	86.8 \pm 6.0	85.0 \pm 6.7

Results reported in table 2 and figure 6 show that GIN models perfectly fit the data and report approximately the same results for predicting the anticancer activity of chemical compounds on both NCI-1 (GIN-1-layer gives an average accuracy of $83.0 \pm 3.5\%$ and GIN-MLP gives $83.4 \pm 2.5\%$) and NCI-109 (GIN-1-layer gives an average accuracy of $86.8 \pm 6.0\%$ and GIN-MLP gives $85.0 \pm 6.7\%$ accuracy) datasets, while Node2vec has a low and unstable performance on the two datasets (average accuracy of $54.9 \pm 1.2\%$ on NCI-1 for lung cancer and $51.4 \pm 2.3\%$ on NCI-109 for ovarian cancer). Those results are maybe due to the fact that GIN models are deep while Node2vec is a shallow model.

Subsequently, we validate the representation power of Node2vec by comparing it theoretically to the Graph Isomorphism Network (GIN) [25] which is proved to be as powerful as the Weisfeiler-Lehman(WL) test of isomorphism [24] in capturing differences on graph structures and topologies. We use Node2vec and two variants of the GIN model to auspiciously predict the activity of chemical compounds against lung and ovarian cancer over NCI-1 and NCI-109 respectively. Chemical compounds predicted to be active against lung and ovarian cancer can then be decisive in the conception of pharmaceutical anticancer drugs.

5 Conclusion

In this work, we investigate the representation learning power of Graph Neural Networks on graph data structure. We explore two powerful models that operate on graphs (1) Node2vec and (2) Graph Isomorphism Network and derive that the latter is a deep and inductive extension of Node2vec. Thus, based on the aforementioned results we are able to apprehend the representation power of Node2vec.

Predicting effectively the activity of chemical compounds against cancer helps pharmaceutical drugs producers in the fabrication of anticancer medicines. We exploit Node2vec and GIN representation power over graph data to predict the activity of chemical compounds against lung and ovarian cancer and achieve remarkable results using GIN variants.

As a future work, we would like to define a new specially-designed deep learning framework for representation learning on chemical compounds modelled as graphs. We also plan to go beyond neighbor aggregation and adopt attention mechanisms to learn the connective strength between nodes.

References

1. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed large-scale natural graph factorization. In: Proceedings of the 22nd international conference on World Wide Web. pp. 37–48 (2013)
2. Atwood, J., Towsley, D.: Diffusion-convolutional neural networks. In: Advances in neural information processing systems. pp. 1993–2001 (2016)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in neural information processing systems. pp. 585–591 (2002)
4. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**(4), 18–42 (2017)
5. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering* **30**(9), 1616–1637 (2018)
6. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM international on conference on information and knowledge management. pp. 891–900 (2015)
7. Chen, J., Ma, T., Xiao, C.: Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247* (2018)
8. Cui, P., Wang, X., Pei, J., Zhu, W.: A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* **31**(5), 833–852 (2018)
9. De Cao, N., Kipf, T.: Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973* (2018)
10. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
11. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1263–1272. *JMLR. org* (2017)
12. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
13. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)
14. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017)
15. Henderson, K., Gallagher, B., Eliassi-Rad, T., Tong, H., Basu, S., Akoglu, L., Koutra, D., Faloutsos, C., Li, L.: Rolx: structural role extraction & mining in large graphs. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1231–1239 (2012)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
18. Li, Y., Vinyals, O., Dyer, C., Pascanu, R., Battaglia, P.: Learning deep generative models of graphs. *arXiv preprint arXiv:1803.03324* (2018)
19. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1105–1114 (2016)

20. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
21. Shen, X., Pan, S., Liu, W., Ong, Y.S., Sun, Q.S.: Discrete network embedding. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 3549–3555 (2018)
22. Shervashidze, N., Schweitzer, P., Leeuwen, E.J.v., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* **12**(Sep), 2539–2561 (2011)
23. Wale, N., Watson, I.A., Karypis, G.: Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems* **14**(3), 347–375 (2008)
24. Weisfeiler, B., Leman, A.: A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Tekhnicheskaya Informatsia*, 9 (1968)
25. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? *International Conference on Learning Representations (ICLR)* (2019)
26. Yang, H., Pan, S., Zhang, P., Chen, L., Lian, D., Zhang, C.: Binarized attributed network embedding. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 1476–1481. IEEE (2018)
27. Yang, J., Leskovec, J.: Overlapping communities explain core-periphery organization of networks. *Proceedings of the IEEE* **102**(12), 1892–1902 (2014)
28. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. In: *Advances in neural information processing systems*. pp. 4800–4810 (2018)
29. You, J., Liu, B., Ying, Z., Pande, V., Leskovec, J.: Graph convolutional policy network for goal-directed molecular graph generation. In: *Advances in neural information processing systems*. pp. 6410–6421 (2018)