ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Beyond Homophily with Graph Echo State Networks

Domenico Tortorella and Alessio Micheli

University of Pisa - Department of Computer Science
Largo B. Pontecorvo 3, 56127 Pisa - Italy

Abstract. Graph Echo State Networks (GESN) have already demonstrated their efficacy and efficiency in graph classification tasks. However, semi-supervised node classification brought out the problem of over-smoothing in end-to-end trained deep models, which causes a bias towards high homophily graphs. We evaluate for the first time GESN on node classification tasks with different degrees of homophily, analyzing also the impact of the reservoir radius. Our experiments show that reservoir models are able to achieve better or comparable accuracy with respect to fully trained deep models that implement ad hoc variations in the architectural bias, with a gain in terms of efficiency.

1 Introduction

Graphs provide a useful structure to represent relations between entities, such as paper citations or web page networks. A plethora of neural models have been proposed to solve graph-, edge-, and node-level tasks [1], most of them sharing an architecture structured in layers that perform local aggregations of node features. This architectural bias, where node features are progressively smoothed in deeper layers via local aggregation [2], is the source of most of the issues that graph neural models are facing. This bias towards locally homogeneous graphs is more apparent in node classification tasks, where graphs presenting a significant number of inter-class edges, i.e. a low *homophily* degree, present a challenge to convolutive models. Graph Echo State Network (GESN) [3] is an efficient model within the reservoir computing (RC) paradigm. In RC, input data is encoded via a randomly-initialized reservoir, while only a linear readout requires training. GESN has already been successfully applied to graph-level classification tasks [4]. In this paper, we analyze for the first time its application to node classification tasks, focusing in particular on the efficacy in tackling low-homophily graphs.

2 Node classification and homophily

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with node feature vectors $\mathbf{u}_v \in \mathbb{R}^U$ for each node $v \in \mathcal{V}$. We also denote by $\mathcal{N}_r(v)$ the set of nodes within r hops of node v, and by $\mathbf{A}$ the graph adjacency matrix. The goal of a semi-supervised node classification task is to learn a model from a subset of graph nodes with known target labels $\{(\mathbf{u}_v, y_v)\}_{v \in \mathcal{V}_{\text{train}}}$, in order to infer the node labels $y_v \in \{1, ..., C\}$ for the remaining nodes $\mathcal{V} \setminus \mathcal{V}_{\text{train}}$ using the network structure and input features $\mathbf{u}_v$. Most common graph convolutional models are structured in L layers, where each layer learns an embedding for each node based on an increasingly large receptive field. These layers can be formalized as [5]

$$\mathbf{h}_v^{(\ell)} = \textsc{combine}\left(\mathbf{h}_v^{(\ell-1)}, \textsc{aggregate}(\{\mathbf{h}_{v'}^{(\ell-1)} : v' \in \mathcal{N}_1(v)\})\right), \qquad (1)$$

where node embeddings $\mathbf{h}_v^{(\ell)} \in \mathbb{R}^H$ of layer ℓ are obtained by aggregating the previous embeddings $\mathbf{h}_{v'}^{(\ell-1)}$ of node v's 1-hop neighbors via $\textsc{aggregate}(\cdot)$, and then combined with the node's previous embeddings $\mathbf{h}_v^{(\ell-1)}$ via $\textsc{combine}(\cdot)$; for $\ell = 1$, $\mathbf{h}_v^{(0)} = \mathbf{u}_v$. The final layer L either directly predicts the one-hot encoding of target label y_v, or is followed by an MLP that serves this purpose. The whole model is trained end-to-end by typically minimizing the cross-entropy loss.

The choice of functions in (1) determines the architectural bias of the model. For example, GCN [6] layers are defined as $\mathbf{h}^{(\ell)} = \mathrm{relu}(\hat{\mathbf{A}}\mathbf{h}^{(\ell-1)}\mathbf{\Theta}^{(\ell)})$, where $\hat{\mathbf{A}}$ is the normalized adjacency matrix, $\mathbf{\Theta}^{(\ell)}$ are learnable weights, and $\mathbf{h}^{(\ell)}$ is the row stack of node features for layer ℓ. It has been shown that stacking more than three or four layers of graph convolution causes a degradation in accuracy [2], since representations $\mathbf{h}_v^{(\ell)}$ converge asymptotically to a fixed point of $\hat{\mathbf{A}}$ as ℓ increases, or more generally, to a low-frequency subspace of the graph spectrum. This problem is known as *oversmoothing*. Indeed, by acting as a low-pass filter, GCNs are biased in favor of tasks whose graphs present a high degree of homophily, that is nodes in the same neighborhood mostly share the same class [7]. Formally, homophily in a graph can be quantified [7] as the intra-class edges ratio

$$\mathfrak{h}_{\mathcal{G}} = |\{(v, v') \in \mathcal{E} : y_v = y_{v'}\}| \,/\, |\mathcal{E}|\,. \qquad (2)$$

Changes in the model architectural bias have been proposed to improve classification on low homophily graphs. Some solutions individuated by [7] are:

1. separate ego and neighborhood representations in (1), by aggregating on open node neighborhoods $\mathcal{N}_r(v) \setminus \{v\}$ and combining by concatenation;
2. extend aggregation to multi-hop neighborhoods $\mathcal{N}_r(v)$, $r > 1$, e.g. as in graph convolutions with Chebyshev polynomial filters [8];
3. exploit also the representations $\mathbf{h}_v^{(\ell)}$ computed at each intermediate layer $\ell < L$ to make predictions, e.g. as in Jumping-Knowledge networks [9].

H2GCN [7] incorporates all three architectural solutions. Alternative solutions include altering the graph structure to improve the homophily degree, in order to increase the ratio of intra-class edges in node neighborhoods [10, 11].

3 Reservoir computing for graphs

Reservoir computing is a paradigm for the efficient design of recurrent neural networks. Input data is encoded by a randomly initialized reservoir, while only the task prediction layer requires training. Graph Echo State Networks (GESNs) extended the reservoir computing paradigm to graph-structured data [3], and have already demonstrated their effectiveness in graph classification tasks [4]. Node embeddings are recursively computed by the dynamical system

$$\mathbf{x}_v^{(k)} = \tanh\left(\mathbf{W}_{\text{in}}\,\mathbf{u}_v + \textstyle\sum_{v' \in \mathcal{N}_1(v)} \hat{\mathbf{W}}\,\mathbf{x}_{v'}^{(k-1)}\right), \quad \mathbf{x}_v^{(0)} = \mathbf{0}, \tag{3}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{H \times U}$ and $\hat{\mathbf{W}} \in \mathbb{R}^{H \times H}$ are the input-to-reservoir and the recurrent weights, respectively (input bias is omitted). Equation (3) is iterated over k until the system state converges to fixed point $\mathbf{x}_v^{(\infty)}$, which is used as the embedding. The existence of a fixed point is guaranteed by the Graph Embedding Stability (GES) property [4], which also guarantees independence from the system's initial state $\mathbf{x}_v^{(0)}$. A necessary condition [12] for the GES property is $\rho(\hat{\mathbf{W}}) < 1/\alpha$, where $\rho(\cdot)$ denotes the spectral radius of a matrix, i.e. its largest absolute eigenvalue, and $\alpha = \rho(\mathbf{A})$ is the graph spectral radius. This condition also provides the best estimate of the system bifurcation point, i.e. the threshold beyond which (3) becomes asymptotically unstable. Reservoir weights are randomly initialized from a uniform distribution in $[-1, 1]$, and then rescaled to the desired input scaling and reservoir spectral radius, without requiring any training. While in graph-level task node features are aggregated to provide global embeddings, for node classification tasks we directly apply a linear readout to node embeddings $\mathbf{y}_v = \mathbf{W}_{\text{out}}\,\mathbf{x}_v^{(\infty)} + \mathbf{b}_{\text{out}}$, where the weights $\mathbf{W}_{\text{out}} \in \mathbb{R}^{C \times H}, \mathbf{b}_{\text{out}} \in \mathbb{R}^{C}$ are trained by ridge regression on one-hot encodings of target classes y_v.

The contractivity of (3) is a sufficient condition for the GES property [12]. However, the contractivity of graph convolution layers has also been linked to the degradation of representativeness in deep models [11]. Graph rewiring solutions to the homophily bias, such as [10], greatly increase the edges of a graph, which in turn leads to an increase of α and a decrease in contractivity. Therefore, in our experiments we will explore also values of the reservoir radius beyond the stability threshold, in this case by arbitrarily fixing the number of iterations of (3) to K. Indeed, we can interpret the K iterations of (3) as equivalent to K graph convolution layers with weights shared among layers and input skip connections. While in deep GCNs convergence to a fixed point of the graph convolution operator, due to stacking too many layers, has been linked to the oversmoothing issue [2], GESNs can in principle avoid that by selecting a reservoir radius $\rho \gg 1/\alpha$.

4 Experiments and discussion

We evaluate GESN on six node classification tasks with low homophily degree (≤ 0.3) and three tasks with high homophily degree (> 0.7). We adopt the same 10 scaffold splits 48%/32%/20% of [7], averaging results in each fold over 10 different reservoir initializations. We explore a number of units ranging from 2^4 to 2^{12}, input scaling factors from 1 to $\frac{1}{320}$, readout regularization values from 10^{-5} to 10^2, and reservoir radii $\rho\alpha \in [0.1, 9.5]$ with steps of 0.2 (up to 35 with larger steps for Squirrel and Chameleon). Embeddings are computed with at most $K = 100$ iterations of equation (3).

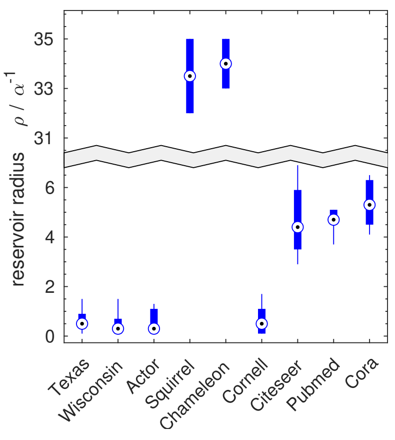

Fig. 1: Reservoir radii selected on each task.

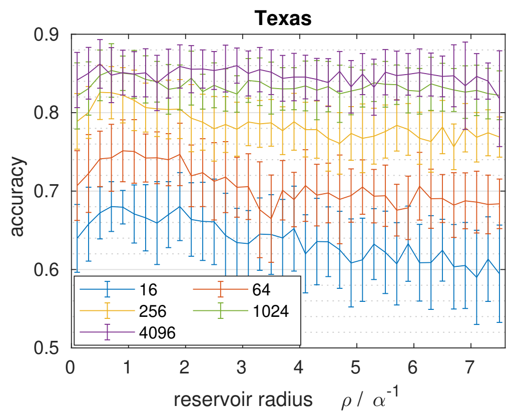

Fig. 2: Impact of reservoir radius and units on classification accuracy for Texas.

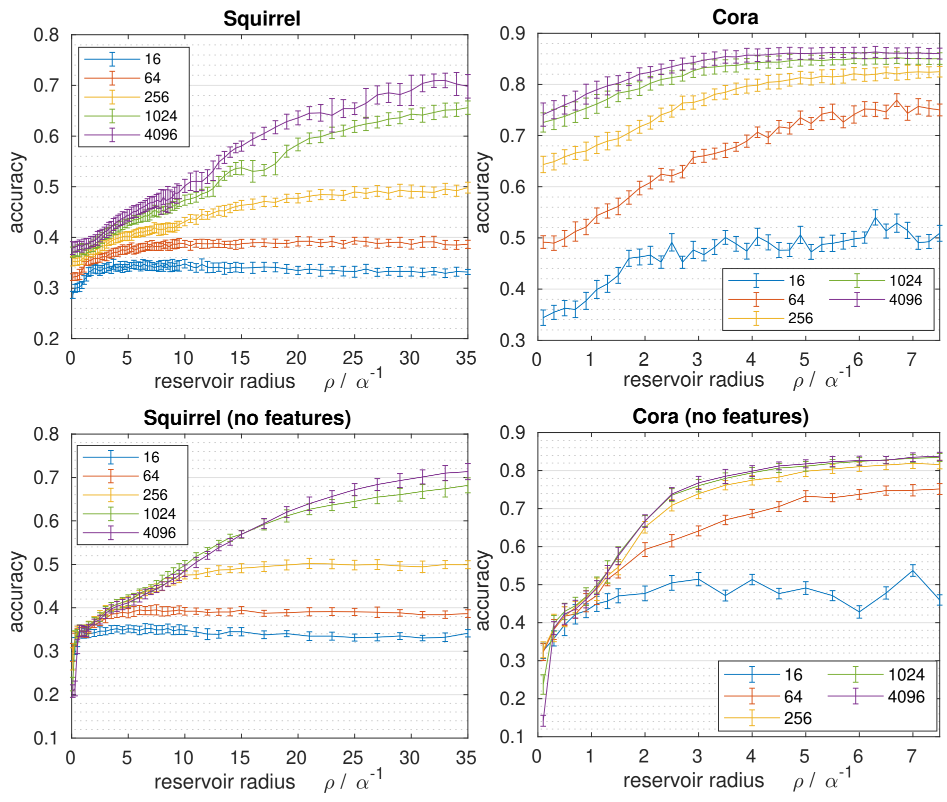

Fig. 3: Impact of reservoir radius and units on classification accuracy for Squirrel ($\mathfrak{h}_{\mathcal{G}} = 0.22$) and Cora ($\mathfrak{h}_{\mathcal{G}} = 0.81$), with and without input features.

	Texas	Wisconsin	Actor	Squirrel	Chameleon	Cornell	Citeseer	Pubmed	Cora
Homo.	0.11	0.21	0.22	0.22	0.23	0.30	0.74	0.80	0.81
Nodes	183	251	7,600	5,201	2,277	183	3,327	19,717	2,708
Edges	295	466	26,752	198,493	31,421	280	9,104	88,648	10,556
Radius	2.56	2.88	9.99	138.60	61.90	2.68	13.74	23.24	14.39
Featur.	1,703	1,703	932	2,089	2,089	1,703	3,703	500	1,433
Classes	5	5	5	5	5	5	6	3	7
GCN	$59.5_{\pm 5.3}$	$59.8_{\pm 7.0}$	$30.3_{\pm 0.8}$	$36.9_{\pm 1.3}$	$59.8_{\pm 2.6}$	$57.0_{\pm 4.7}$	$76.7_{\pm 1.6}$	$87.4_{\pm 0.7}$	$87.3_{\pm 1.3}$
+JK	$66.5_{\pm 6.6}$	$74.3_{\pm 6.4}$	$34.2_{\pm 0.9}$	$40.5_{\pm 1.6}$	$63.4_{\pm 2.0}$	$64.6_{\pm 8.7}$	$74.5_{\pm 1.8}$	$88.4_{\pm 0.5}$	$85.8_{\pm 0.9}$
+Cheby	$77.3_{\pm 4.1}$	$79.4_{\pm 4.5}$	$34.1_{\pm 1.1}$	$43.9_{\pm 1.6}$	$55.2_{\pm 2.8}$	$74.3_{\pm 7.5}$	$75.8_{\pm 1.5}$	$88.7_{\pm 0.6}$	$86.8_{\pm 1.0}$
H2GCN	$84.9_{\pm 6.8}$	$86.7_{\pm 4.7}$	$35.9_{\pm 1.0}$	$36.4_{\pm 1.9}$	$57.1_{\pm 1.6}$	$82.2_{\pm 4.8}$	$77.1_{\pm 1.6}$	$89.4_{\pm 0.3}$	$86.9_{\pm 1.4}$
MLP	$81.9_{\pm 4.8}$	$85.3_{\pm 3.6}$	$35.8_{\pm 1.0}$	$29.7_{\pm 1.8}$	$46.4_{\pm 2.5}$	$81.1_{\pm 6.4}$	$72.4_{\pm 2.2}$	$86.7_{\pm 0.4}$	$74.8_{\pm 2.2}$
GESN	$84.3_{\pm 4.4}$	$83.3_{\pm 3.8}$	$34.5_{\pm 0.8}$	$71.2_{\pm 1.5}$	$76.2_{\pm 1.2}$	$81.1_{\pm 6.0}$	$74.5_{\pm 2.1}$	$89.2_{\pm 0.3}$	$86.0_{\pm 1.0}$

Table 1: Node classification accuracy on low and high homophily graphs (average and standard deviation; results of fully trained models reported from [7]; results within one standard deviation of the best accuracy are highlighted).

Accuracy results are reported in Table 1, while Fig. 1 shows the reservoir radii selected in the 10 splits. We can observe three different behaviors, exemplified in Fig. 2 and 3 (top). The number of reservoir units plays a significant role, offering best results when it is closer to the number of input features. For Texas, Wisconsin, Actor, and Cornell, the performances of GESN are closer to the accuracies of MLP, which uses only node features $\mathbf{u}_v$, and H2GCN, with reservoir radii $\rho\alpha < 1$: in this case, the graph connectivity appears to be of no use. While Squirrel and Chameleon present a low homophily degree, graph convolution models fare better than MLP: in this case graph connectivity needs to be taken into account. On these two tasks, GESN improves upon the best model accuracy by 27.3% and 12.8%, respectively, with reservoir radii selected in the range 33–35. Finally, on high homophily tasks (Citeseer, Pubmed, Cora) GESN performs generally in line with graph convolution models, which in turn do better than MLP; reservoir radii are selected in the range 4–6.

We observe how the best accuracy results are for reservoir radii well above the stability threshold, which are required when the graph connectivity needs to be leveraged in classifying nodes. To support our conclusion, in Fig. 3 (bottom) we report the accuracy on Squirrel and Cora where input features have been removed. We observe that for stable embeddings ($\rho\alpha < 1$), accuracy significantly drops below the level reached by having input features, while it reaches almost the same levels of accuracy for the values of $\rho\alpha$ selected with features, which are well beyond the region where GESN stability is guaranteed.

Finally, we underline the efficiency of GESN. Only the linear readout's $C(H+1)$ parameters require training, against the additional $O(H^2 L)$ parameters of models that need to be trained end-to-end through many gradient descent epochs (for further time comparisons, see [4]). The time required to compute node embeddings and train the readout for a model of 4096 units takes from 0.87 to 1.58 seconds on a GPU Nvidia Tesla V100, depending on graph size.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

5 Conclusion

For the first time, we have applied Graph Echo State Networks to the task of node classification. Experiments on nine graphs with different degrees of homophily have shown a classification accuracy generally in line with most fully trained models, with extraordinary improvements over two low homophily tasks. Furthermore, contrary to the theory and experiments that demonstrated the crucial role of system stability in applying GESNs to graph-level tasks, our experiments have shown that node embeddings computed in regions well beyond the theoretical stability threshold are better suited to represent the graph structure. Future work will analyze more in-depth the embedding space structure, the role of reservoir radius in conditioning the filtering properties of GESN, and the impact of reservoir spectrum.

References

[1] D. Bacciu, F. Errica, A. Micheli, and M. Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 2020.

[2] Q. Li, Z. Han, and X. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 3538–3545, 2018.

[3] C. Gallicchio and A. Micheli. Graph echo state networks. In *The 2010 International Joint Conference on Neural Networks*, pages 3967–3974, 2010.

[4] C. Gallicchio and A. Micheli. Fast and deep graph neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[5] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations*, 2019.

[6] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017.

[7] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, volume 33, pages 7793–7804, 2020.

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, volume 29, pages 3844–3852, 2016.

[9] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5453–5462, 2018.

[10] J. Gasteiger, S. Weißenberger, and S. Günnemann. Diffusion improves graph learning. In *Advances in Neural Information Processing Systems*, volume 32, pages 13298–13310, 2019.

[11] J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *10th International Conference on Learning Representations*, 2022.

[12] D. Tortorella, C. Gallicchio, and A. Micheli. Spectral bounds for graph echo state network stability. In *The 2022 International Joint Conference on Neural Networks*, 2022.