ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

A bag of nodes primer on weightless graph classification*

Raul B. Barbosa[1], Diego Carvalho[2], Priscila M. V. Lima[1], Felipe M. G. França[1]

1- Federal University of Rio de Janeiro (UFRJ) - PESC/COPPE
Cidade Universitária, Centro de Tecnologia, Bloco H, Sala 319 - 21941-972 - Brazil

2- Federal Centre for Technological Education of Rio de Janeiro (CEFET/RJ)
Av. Maracanã, 229, 20271-110 - Brazil

Abstract. This paper proposes a weightless architecture for graph classification scenarios. This architecture is a three-headed arrangement composed of graph hand-picked features, a quantization method and a final classifier. Although multiple new strategies for graph classification have been proposed in recent years, it is still necessary to settle comparable studies with respect to weightless neural networks. The proposed architecture is evaluated along with other baseline classifiers and independent strategies, showing that weightless architectures are able to compete with other well-established methods such as graph kernels.

1 Introduction

In recent years, our community has witnessed a growing interest in graph-based machine learning. Ranging from the somewhat recent graph neural networks [1] to classical kernel methods [2], graph learning is a difficult and valuable field that has intrinsic problems due to graphs' non-euclidean nature. If it is trivial to represent a graph as an adjacency matrix, any graph learning process has to come up with a graph representation that must deal with node and edge labeling invariance. This learning, especially if deep, can be costly and require amounts of time and data. Hence, simpler, faster and cheaper models are still sought-after.

Our work is focused on a fundamental branch in graph learning, the classification of entire graphs with a corresponding label. This problem has applications that have been arisen from different fields of study and industry, such as social networks, chemistry and biology [3]. Going in the opposite direction of the recent trend, we rely on the simplest and traditional approach of a bag of nodes (as defined in [4]), driven by graph statistics that are used in the form of entire distributions, later binarized, as input to the WiSARD[5] classifier.

Although WiSARD-based classifiers have demonstrated good results in other domains [6], our work pioneers in joining both graph learning and weightless classifiers. We evaluate our new architecture first in respect to its sanity, in a controlled and synthetic network science scenario and then in widely used public datasets [3], demonstrating its feasibility towards the desired task.

*This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq, FAPERJ and DIPPG - CEFET/RJ.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

2 Background

2.1 On network centralities and graph measures

Many features and measures were developed throughout these years to analyze graphs and networks[1]. Yet, here we pinpoint three simple effective measures leveraged by our work.

Degree – according to [4], the node degree is the most elementary of the graph measures. A graph G is denoted by $G = (V, E)$, where V is the set of vertices and E the set of edges, being an edge $e_{i,j}$ a pair $= (v_i, v_k) \mid v_i, v_k \in V$. The degree $d(i)$ of a node v_i is defined as the number of nodes connected to a certain node $v_i \in V$. In other words, it is $|N(i)|$ where

$$N(i) = \{v_k \in V \mid e_{i,k} \in E, i \neq k\} \tag{1}$$

A graph can be directed or undirected. If a graph is directed, $e_{i,j}$ is distinguishable from $e_{j,i}$, thus previous definition turns out to be the node's out-degree $d_{out}(i)$ of v_i, being a node's in-degree $d_{in}(i)$ analogously defined as $|N_{in}(i)|$ where

$$N_{in}(i) = \{v_k \in V \mid e_{k,i} \in E, i \neq k\} \tag{2}$$

PageRank centrality – a node's centrality describes how important is a node in a given network, another basic concept in network analysis [7]. The PageRank centrality [8] $pr(i)$ of a node v_i, of the most used node centralities, is recursively defined next, where α is the damping factor, a constant $\in (0, 1)$:

$$pr(i) = \alpha \sum_{k \in N_{in}(i)} \frac{1}{d_{out}(k)} pr(k) + (1 - \alpha) \tag{3}$$

.

The Onion-Decomposition/Spectrum – the onion decomposition [9] is a modification of the k-core decomposition, a method to define a core-periphery structure in networks, that provides comprehension of a graph in multiple scales. A k-core is a subgraph of G such that every vertex has degree at least k. The onion decomposition algorithm [9] gathers an extra information during the k-core decomposition algorithm called layer. A layer is the number of iterations until a given node is reached by the algorithm. Although straightforward, this little extra information leads to another interesting feature named the onion spectrum, which is simply the fraction of all nodes found in a given layer.

2.2 On WiSARD-based classifiers

The WiSARD [5] is a hardware-intended n-tuple classifier, being extremely lightweight. Its a pseudo-random memory-oriented model, which fits the definition of a weightless neural network, originally designed for supervised learning. Recently it got an extension for semi-supervised and unsupervised learning under the name of ClusWiSARD [10]. Basic functionality refers to the division a

[1]Here we use both terms as synonyms.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

binary input vector into n tuples, each with q bits that maps a RAM address. These addresses are further written to learn patterns and read to distinguish between learned examples for classification.

2.3 On quantization strategies

Binarization and quantization stategies deserve a chapter when dealing with WiSARD classifiers, because of its dependency on binary vector inputs. A number of works have been published and here we highlight some of the strategies evaluated in our work.

Plot Binarization – the initial strategy used in [11], consists of plotting $\mathbb{R}^2$ points in a canvas and using the plot image as a binary input vector.

Histogram Binarization – a thermometer encoding of the distribution histogram, where each histogram bin is a thermometer encoding of its min-max normalized value. This encoding scheme was previously used with good results in [6, 12] for example.

KernelCanvas [12] Binarization – an $\mathbb{R}^n$ division strategy that splits the space in k regions, being k defined as the number of kernels. Kernels are random points that when in $\mathbb{R}^2$ produce a space partition similar to a Voronoi diagram.

KD-Tree Binarization – an $\mathbb{R}^2$ division framework that splits the space in k regions using a KDTree, being k the number of leaves in the tree. Nodes get divided according to pre-defined priority policy. More details can be found in [11].

3 Problem and Model

We formulate our graph classification problem as learning a function $f \sim l(G)$, where $l : G \to \Sigma$, being G a graph and Σ a label set in a given dataset. Our architecture breaks f into a three function composition: $f = C \circ Q \circ M$. M is the feature/measure extraction step that, given a graph, returns an array of $\mathbb{R}^2$ extracted feature points. Q is the binarization/quantization step, that given a list of arbitrary points in $\mathbb{R}^2$, outputs a fixed-size binary vector. C is the classifier step that given a fixed-size binary vector, outputs a label as result.

The mental picture of step M is a plot of measures describing G. For all measures but the onion decomposition, the whole distribution was used under the form of an empirical complementary cumulative distribution function. The onion spectrum was used for the onion decomposition. In our work, albeit some datasets and graphs might present node and edge attributes, we restrict our study to using the graph structure as input.

4 Experiments and Results

Several datasets were used in this work, including a synthetic dataset we built named RGG, inspired by [13]. This dataset was created using graphs generated by five random graph generators available in the NetworkX[14] Python library:

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

Erdos-Renyi (GNP), Random power law tree, Connected Watts-Strogatz small-world, Holme-Kim and Barabási-Albert. Each generator serves as the class label for the dataset. The other datasets were drawn from the TUDatasets collection [3], using their cleaned versions as distributed in the PyTorch Geometric library [15]. A summary of the datasets is presented in Table 1.

dataset	# graphs	# classes	avg. $\|N\|$	avg. $\|E\|$
RGG	20480	5	139.68	813.43
MUTAG	135	2	18.85	20.83
ENZYMES	595	6	32.48	62.16
IMDB-M	321	3	22.35	124.72
COLLAB	4064	3	76.94	2333.64
NCI1	3785	2	29.87	32.37

Table 1: Dataset summary.

The performance of every combination for the C, Q and M functions listed on Table 2 was assessed, resulting in a total of 48 models (including both WiSARD-based classifiers and RandomForest baselines). Additionally, we evaluated the performance of a complete independent baseline, using the WL-Graph Kernel [16] as implemented in [17] and an SVM classifier (WL-SVM). Every combination was tested as an hypothesis in a ten-fold cross validation procedure on a Xeon E5-2630 v3 CPU with 32GB RAM.

Measures (M)	Quantization(Q)	Classifiers (C)
node degree (IN/OUT)	Plot (PB)	WiSARD (WIS)
page rank centrality (PR)	Histogram (HG)	ClusWiSARD (CWIS)
onion decomposition (OD)	KernelCanvas (KC)	RandomForest (RF)
	KDTree (KDT)	

Table 2: All hypothesis sets for measure extraction (M), quantization (Q) and classifier (C) steps.

The parameter selection methodology was exactly the same as in [18]. Grid search is used for best validation parameters and then a test phase. We also put a size constraint on our binary vectors to 2048 bits, except for the RGG dataset test where only 1024 bits were used.

As depicted by Figure 1, the results of a F_1-score of 0.99 on the RGG dataset assess the feasibility of our framework, reaching a comparable performance to the SVM baseline approach.On some datasets, such as MUTAG, it outperforms the baseline model, reaching a top 0.83 F_1-score. Concerning all tested hypotheses excluding the kernel baseline, there is no clear winner. With respect to measures, both page rank and onion decomposition ranked better in a tie scenario. As for quantization strategies, there was a slight dominance of the KDTree and Plot approaches, yet other strategies were also successful in some experiments. In

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

general, WiSARD classifiers had a comparable performance to RandomForest alternatives.

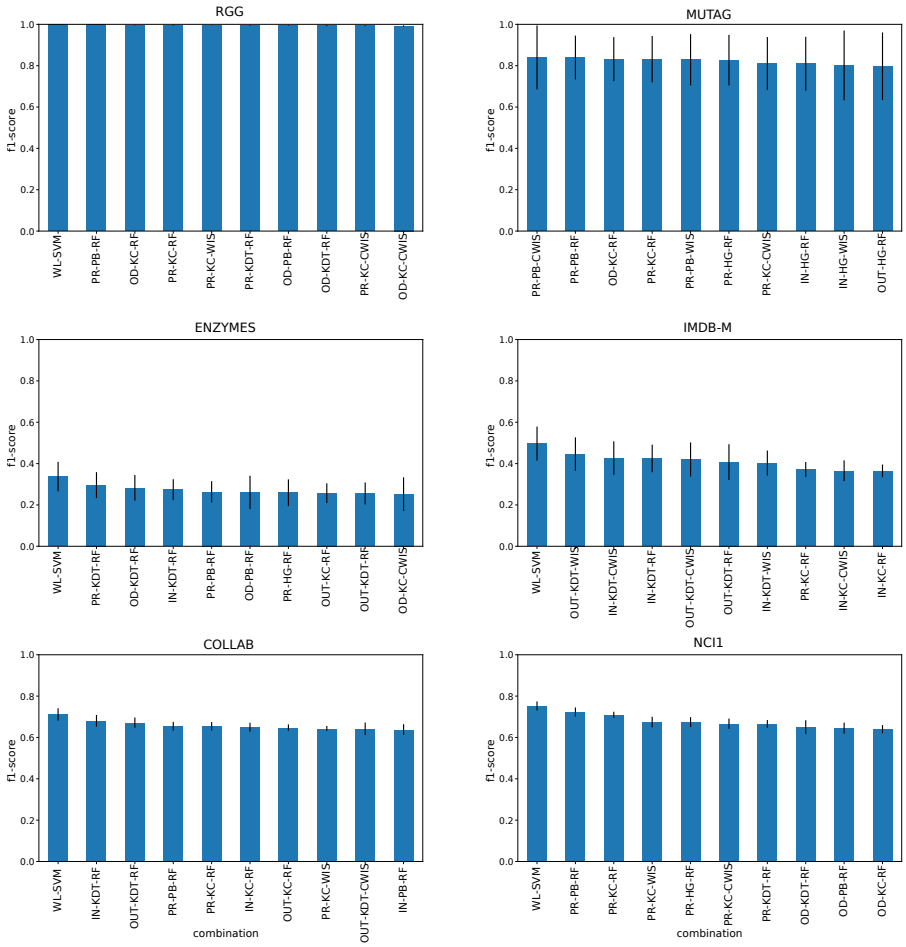

Fig. 1: Results summary: top 10 combinations for each dataset. Proposed framework performance is comparable to kernel baseline in every scenario.

5 Conclusion

In this paper we have presented a novel architecture for graph classification using weightless neural networks. Although based on naive bag of nodes features, experimental results point that it has comparable performance with other shallow learning and yet more complex alternatives, such as graph kernels. Future work aims towards the incorporation of node and edges attributes to the framework and also the inclusion of other graph representation techniques as part of the

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

learning process.

.

References

[1] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Sig. Proc. Magazine*, 34(4):18–42, 2017.

[2] N. M. Kriege, F. D. Johansson, and C. Morris. A survey on graph kernels. *Applied Network Science*, 5(1):6, Jan 2020.

[3] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Repr. Learning and Beyond (GRL+ 2020)*, 2020.

[4] W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

[5] I. Aleksander, M. De Gregorio, F.M.G. França, P.M.V. Lima, and H. Morton. A brief introduction to weightless neural systems. In *ESANN 2009*, pages 299–305, 2009.

[6] P. Xavier, M De Gregorio, F. M. G. França, and P. M. V. Lima. Detection of elementary particles with the wisard n-tuple classifier. In *ESANN 2020*, pages 643–648, 2020.

[7] M. Newman. *Networks*. Oxford university press, 2018.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[9] L. Hébert-Dufresne, J. A. Grochow, and A. Allard. Multi-scale structure and topological anomaly detection via a new network statistic: The onion decomposition. *Scientific reports*, 6(1):1–9, 2016.

[10] D. O. Cardoso, F. M. G. França, and J. Gama. WCDS: A Two-Phase Weightless Neural System for Data Stream Clustering. *New Generation Computing*, 35:391–416, 2017.

[11] R. Barbosa, D. Carvalho, D. Cardoso, and F. M. G. França. Weightless neuro-symbolic GPS trajectory classification. *Neurocomputing*, 298:100–108, 2018.

[12] D. F. P. de Souza, F. M. G. Franca, and P. M. V. Lima. Spatio-temporal Pattern Classification with KernelCanvas and WiSARD. In *BRACIS 2014*, pages 228–233, 2014.

[13] P. Avelar, H. Lemos, M. Prates, and L. Lamb. Multitask learning on graph neural networks: Learning multiple graph centrality measures with a unified network. In *International Conference on Artificial Neural Networks*, pages 701–715. Springer, 2019.

[14] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proc. of the 7th Python in Science Conference*, pages 11–15, 2008.

[15] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[16] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12:2539–2561, 2011.

[17] G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, and M. Vazirgiannis. Grakel: A graph kernel library in python. *J. Mach. Learn. Res.*, 21(54):1–5, 2020.

[18] F. Errica, M. Podda, D. Bacciu, and A. Micheli. A fair comparison of graph neural networks for graph classification. In *ICLR 2020*, 2020.