ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

A Tropical View of Graph Neural Networks

Davide Bacciu, Francesco Landolfi, and Danilo Numeroso

Università di Pisa - Department of Computer Science
Largo Bruno Pontecorvo, 3, 56127, Pisa - Italy

Abstract. Learning dynamic programming algorithms with Graph Neural Networks (GNNs) is a research direction which is increasingly gaining popularity. Prior work has demonstrated that in order to learn such algorithms, it is necessary to have an "alignment" between the neural architecture and the dynamics of the target algorithms, and that GNNs align, in fact, with dynamic programming. Here, we provide a different view of this alignment, studying it through the lens of tropical algebra. We show that GNNs can approximate dynamic programming algorithms up to arbitrary precision, provided that their input and output are appropriately pre- and post-processed.

1 Introduction

The *dynamic programming* (DP) paradigm [2] aims at finding a solution to a complex problem by recursively breaking it down to simpler sub-problems whose solutions are cached and reused for efficiency. A DP algorithm can be schemed as

$$\text{Answer}_i^{(k+1)} = \text{Update}(\{\text{Answer}_j^{(k)} \mid j = 1, \dots, n\}), \tag{1}$$

where $\text{Answer}_i^{(k)}$ is the solution of the i-th sub-problem at iteration k, while Update computes the Answer of a sub-problem from the Answers of the previous iteration [11, 12]. A recent line of research has focused on finding a parallelism between DP and what Graph Neural Networks [GNNs, 1] are able to compute and reason upon. Xu et al. [11] provided a first valuable characterization in this sense, by introducing the notion of *algorithmic alignment* which—simplifying—states that we can achieve a lower sample complexity if the model's architecture "aligns" with an algorithm's computation graph. The authors also provide as an example the alignment between the Update function of the Bellman-Ford algorithm [BF, 3, 6, 9] and a GNN, defined respectively as

$$\mathbf{d}_v^{(k+1)} = \min_{u \in N(v)} \mathbf{A}_{uv} + \mathbf{d}_u^{(k)}, \text{ and} \tag{BF}$$

$$\mathbf{x}_v^{(k+1)} = \bigoplus_{u \in N(v)} \text{MLP}(\mathbf{x}_u^{(k)}, \mathbf{x}_v^{(k)}), \tag{GNN}$$

where $\bigoplus$ is an aggregation function and MLP is a (learnable) function. In this example, however, an alignment was performed not only in the model's architecture, but also in the *operations* performed by the Update function, as the aggregation function of eq. (GNN) is set to $\bigoplus = \min$. This kind of "semiring" alignment can be found also in a later work of the same authors [12]. Moreover,

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

the works of Corso et al. [5] and Zhu et al. [13] demonstrate empirically that, when this kind of alignment is not possible (that is, there is no known algorithm to align to), using a linear combination of multiple semirings can improve the performance of GNNs on real-world tasks.

In the following we will show that GNNs can approximate *min*-aggregated DP algorithms up to an arbitrary precision. We will show that by drawing connections between the field of real numbers, where classical neural networks lie on, and "tropical" semirings, effectively casting GNNs under the framework of tropical algebra. Based on this, we introduce a proofing framework that demonstrates that a *sum*-aggregated GNN, equipped with appropriate *encoding* (and *decoding*) functions to (and from) its latent space can approximate DP algorithms up to an arbitrary precision. We provide practical proof examples for well-known reachability and shortest-path algorithms. We confide our work provides the needed framework to reason about algorithmic alignment of GNNs at a more abstract and general level than single algorithm alignment, while providing useful hints to guide the design of effectively "aligned" GNNs.

2 Background on tropical algebra

Tropical semiring. The *tropical semiring* [4, 8] is the set $\mathbb{T} = \mathbb{R} \cup \{\infty\}$, equipped with the following generalized addition and multiplication: for any $x, y \in \mathbb{T}$,

$$x \oplus y = \min(x, y) \qquad \text{and} \qquad x \odot y = x + y,$$

where $\mathbb{0} = \infty$ and $\mathbb{1} = 0$ are, respectively, the absorbing and identity elements. Matrix multiplication in the tropical semiring are defined analogously to the classical one, that is, given $\mathbf{A} \in \mathbb{T}^{l \times m}$ and $\mathbf{B} \in \mathbb{T}^{m \times n}$,

$$\big[\mathbf{A} \odot \mathbf{B}\big]_{ij} = \bigoplus_k \mathbf{A}_{ik} \odot \mathbf{B}_{kj} = \min_k \mathbf{A}_{ik} + \mathbf{B}_{kj}.$$

Let G be a graph whose adjacency matrix $\mathbf{W} \in \mathbb{T}^{n \times n}$ represents the pairwise distances between two nodes, that is

$$\mathbf{W}_{uv} = \begin{cases} d_{uv} \in \mathbb{T} & uv \in E(G), \\ \mathbb{1}\ (=0) & u = v, \\ \mathbb{0}\ (=\infty) & \text{otherwise.} \end{cases} \tag{2}$$

One can see that $\mathbf{W}^{\odot k} = \overbrace{\mathbf{W} \odot \mathbf{W} \odot \cdots \odot \mathbf{W}}^{k \text{ times}}$ computes the matrix representing the k-hop shortest paths between any two pair of nodes. Moreover, for any $k \geq n-1$, $\mathbf{W}^{\odot k}$ is the matrix representing the distances between all pairs of nodes (as $\mathbf{W}^{\odot n-1} \odot \mathbf{W} = \mathbf{W}^{\odot n-1}$) [8]. Given $\boldsymbol{\chi}_v \in \mathbb{T}^n$ the vector defined as

$$[\boldsymbol{\chi}_v]_u = \begin{cases} \mathbb{1}\ (=0) & \text{if } u = v \\ \mathbb{0}\ (=\infty) & \text{otherwise,} \end{cases} \tag{3}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

we can see that $\mathbf{W}^{\odot n-1} \odot \boldsymbol{\chi}_v \in \mathbb{T}^n$ computes the distances of all the nodes from v, which is known as the Bellman-Ford algorithm [3, 6, 9] for computing the single source shortest paths (SSSP).

Maslov quantization. Let $h > 0$ and $\mathbb{R}_+$ be the set of non-negative real numbers. We can define the maps $Q_h : \mathbb{T} \to \mathbb{R}_+$ and $D_h : \mathbb{R}_+ \to \mathbb{T}$ respectively as

$$Q_h(x) = \begin{cases} 0 & \text{if } x = \infty \\ e^{-x/h} & \text{otherwise;} \end{cases} \qquad D_h(x) = \begin{cases} \infty & \text{if } x = 0 \\ -h \ln x & \text{otherwise.} \end{cases} \tag{4}$$

These functions are typically referred to, respectively, as *Maslov quantization* and *dequantization* maps [4, 7], as an analogy to the quantization procedure in physics, and h is an analog for the Planck constant [7].

The set of non-negative real numbers $\mathbb{R}_+$ forms a semi-ring endowed with the classical addition and multiplication operators ($+$ and $\cdot$). Clearly, objects mapped in the "quantum" semi-ring can be mapped back in the tropical one, as $\forall x \in \mathbb{T}.\ x = D_h(Q_h(x))$. Moreover, there is a tight correspondence between the operators in the two semi-rings, as operations performed in the "quantum" semi-ring also approximate their generalized counterparts in the tropical one. By defining the generalized addition and multiplication

$$x \oplus_h y = D_h(Q_h(x) + Q_h(y)) \qquad \text{and} \qquad x \odot_h y = D_h(Q_h(x) \cdot Q_h(y))$$

for any two $x, y \in \mathbb{T}$, we have that

$$x \odot_h y = -h \ln(e^{-x/h} e^{-y/h}) = x + y = x \odot y,$$
$$x \oplus_h y = -h \ln(e^{-x/h} + e^{-y/h}) \xrightarrow[h \to 0]{} \min(x, y) = x \oplus y.$$

Analogously, matrix operations also approximate the generalized dot product in the tropical semi-ring: going back to the example in the previous paragraph, we have that $\lim_{h \to 0} D_h[Q_h[\mathbf{W}]^k] = \mathbf{W}^{\odot k}$ for any k ($f[\mathbf{A}]$ denotes element-wise application of f on the matrix $\mathbf{A}$). A similar result is obtained in the SSSP example, as $\lim_{h \to 0} D_h[Q_h[\mathbf{W}]^k \cdot \mathbf{e}_v] = \mathbf{W}^{\odot k} \odot \boldsymbol{\chi}_v$, where $\mathbf{e}_v \in \mathbb{R}_+^n$ is the classical standard basis vector, defined as in eq. (3) with $\mathbb{0} = 0$ and $\mathbb{1} = 1$. An example of the dynamic programming approximation in the classical algebra can be found, e.g., in [8, §1.2].

3 Approximation capabilities of GNNs

In most graph representation learning scenarios, a exemplar graph G is either provided as an unweighted adjacency matrix ($\mathbf{A} \in \{0, 1\}^{n \times n}$, with $\mathbf{A}_{uv} = 1$ iff $uv \in E(G)$), or as a weighted one ($\mathbf{W} \in \mathbb{R}_+^{n \times n}$, $\mathbf{W}_{uv} > 0$ iff $uv \in E(G)$), with weights representing either distances or similarities. In all cases, we assume there is no self loop ($vv \notin E(G)$ for any $v \in V(G)$). In the following we will show how the Graph Isomorphism Network [GIN 10], defined as

$$\mathrm{GIN}_\epsilon(\mathbf{A}, \mathbf{X}) = \mathrm{MLP}\big((1 + \epsilon)\mathbf{X} + \mathbf{A}\mathbf{X}\big), \tag{5}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

approximates the SSSP problem using a simple encoder-decoder architecture that can be synthesized as $G_{\text{out}} = \text{DEC}(\text{GIN}_\epsilon^k(\text{ENC}(G_{\text{in}})))$, for some given encoder (ENC) and decoder (DEC), where GIN^k represent the k compositions of the same network, that is

$$\text{GIN}_\epsilon^k(\mathbf{A}, \mathbf{X}) = \begin{cases} \text{GIN}_\epsilon(\mathbf{A}, \mathbf{X}) & \text{if } k = 1 \\ \text{GIN}_\epsilon(\mathbf{A}, \text{GIN}_\epsilon^{k-1}(\mathbf{A}, \mathbf{X})) & \text{otherwise.} \end{cases}$$

The following propositions will only assess the fact that there exists a configuration of the parameters of ENC, GIN, and DEC that will allow the model to approximate the distances in a given task, but they will not address the problem of *how* that configuration is obtained (i.e., if a given learning or optimization algorithm will provide such configuration).

Reachability approximation. Assume we are given an unweighted graph $G = (\mathbf{A}, \mathbf{e}_v)$ in the form of an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ and a standard basis vector $\mathbf{e}_v$ with 1 on the v entry for a given $v \in V(G)$ and 0 in the other ones. Assume also $\text{ENC}(\mathbf{X}) = \mathbf{X}$, and $\text{DEC}(\mathbf{X}) = \text{MLP}(D_1[\mathbf{X}])$, with D_1 defined as in eq. (4). Then GIN_0 (and, consequentially, also GIN_ϵ) can identify the nodes that are reachable within k-hops from a given node $v \in V(G)$ (or, equivalently, in the k-hop neighborhood v, denoted as $N^k(v)$ or $N_+^k(v)$ if inclusive of v itself).

Proposition 1. *Given $G = (\mathbf{A}, \mathbf{e}_v)$, a positive constant $c > 0$, and $\mathbf{d} = \text{MLP}(D_1[\text{GIN}_0^k(\mathbf{A}, \mathbf{e}_v)])$, with GIN_0 and MLP having any number of layers with ReLU activation function, there exist a configuration of their parameters such that $|\mathbf{d}_u - \delta_u(N_+^k(v))| \leq c$ for any $u \in V(G)$, where δ_u is the Dirac delta.*

Proof. Fix $w_\theta = 1$ and $b_\theta = 0$ for every layer of GIN/MLP, if not stated otherwise. Since ReLU acts as an identity function for non-negative elements, we have that $\text{GIN}_0^k(\mathbf{A}, \mathbf{e}_v) = (\mathbf{A} + \mathbf{I})^k \mathbf{e}_v$. Let $\mathbf{W} \in \mathbb{T}^{n \times n}$ a matrix defined as in eq. (2), with $\mathbf{d}_{uw} = 0$ for all $uw \in E(G)$. For any $h > 0$ we have that

$$Q_h[\mathbf{W}] = \mathbf{A} + \mathbf{I} \qquad \text{and} \qquad Q_h[\boldsymbol{\chi}_v] = \mathbf{e}_v.$$

If we set $w_{\theta_1} = h$ in the first layer of MLP, we have that

$$h \cdot D_1[Q_h[\mathbf{W}]^k \cdot Q_h[\boldsymbol{\chi}_v]] = D_h[Q_h[\mathbf{W}]^k \cdot Q_h[\boldsymbol{\chi}_v]] \xrightarrow[h \to 0]{} \mathbf{W}^{\odot k} \odot \boldsymbol{\chi}_v = \mathbf{d}_\circ,$$

where $[\mathbf{d}_\circ]_u = 0$ if $u \in N_+^k(v)$ and $[\mathbf{d}_\circ]_u = \infty$ otherwise. By setting, instead, $w_{\theta_1} = -h$ and $b_{\theta_1} = 1$ in the first layer of MLP, we obtain

$$\lim_{h \to 0} \mathbf{d} = \lim_{h \to 0} \sigma[1 - h \cdot D_1[(\mathbf{A} + \mathbf{I})^k \mathbf{e}_v]] = \max[0, 1 - \mathbf{d}_\circ] = \delta(N_+^k(v)).$$

For a $h > 0$ small enough we have that

- for $u \in N_+^k(v)$, $|\mathbf{d}_u - \delta_u(N_+^k(v))| < c$ by definition of limit, and
- for $u \notin N_+^k(v)$, $[\mathbf{d}_\circ]_u \gg 1$ and hence $\mathbf{d}_u = \delta_u(N_+^k(v)) = 0$,

thus reaching the conclusion. □

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Unweighted shortest path. Under the same assumptions, we can show instead that GIN_ϵ is able to approximate the unweighted k-hop shortest paths to a given source node (that is, the Bellman-Ford algorithm) up to a given precision.

Proposition 2. *Given $G = (\mathbf{A}, \mathbf{e}_v)$, a positive constant $c > 0$, and $\mathbf{d} = \mathrm{MLP}(D_1[\mathrm{GIN}^k_\epsilon(\mathbf{A}, \mathbf{e}_v)])$, with GIN_ϵ and MLP having any number of layers with* ReLU *activation function, there exist a configuration of their parameters such that $|\mathbf{d}_u - \mathrm{dist}(u, v)| \leq c$ for any $u \in N^k_+(v)$.*

Proof. Fix $w_\theta = 1$ and $b_\theta = 0$ for every layer of GIN/MLP, if not stated otherwise. Let $\mathbf{W} \in \mathbb{T}^{n \times n}$ a matrix defined as in eq. (2), with $\mathbf{d}_{uw} = 1$ for all $uw \in E(G)$. For any $h > 0$ we have that

$$Q_h[\mathbf{W}] = e^{-1/h} \cdot \mathbf{A} + \mathbf{I} \qquad \text{and} \qquad Q_h[\boldsymbol{\chi}_v] = \mathbf{e}_v.$$

By fixing $\epsilon = e^{1/h} - 1$, and $w_{\theta_1} = e^{-1/h}$ in the first layer of GIN, we have that

$$\mathrm{GIN}^k_\epsilon(\mathbf{A}, \mathbf{e}_v) = e^{-1/h} \cdot ((1 + \epsilon) \cdot \mathbf{I} + \mathbf{A})^k \mathbf{e}_v = (\mathbf{I} + e^{-1/h} \cdot \mathbf{A})^k \mathbf{e}_v,$$

and, by setting also $w_{\theta_1} = h$ in the first layer of MLP, we have that

$$\mathbf{d} = h \cdot D_1[Q_h[\mathbf{W}]^k \cdot Q_h[\boldsymbol{\chi}_v]] = D_h[Q_h[\mathbf{W}]^k \cdot Q_h[\boldsymbol{\chi}_v]] \xrightarrow[h \to 0]{} \mathbf{W}^{\odot k} \odot \boldsymbol{\chi}_v = \mathbf{d}_\circ,$$

where

$$[\mathbf{d}_\circ]_u = \begin{cases} \mathrm{dist}(u, v) & \text{if } u \in N^k_+(v) \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

By definition of limit (for a $h > 0$ small enough) we reach the conclusion. □

Weighted shortest path. If instead we are given a weighted graph $G = (\mathbf{W}, \mathbf{e}_v)$ in the form of a distance matrix $\mathbf{W} \in \mathbb{T}^{n \times n}$, defined as in eq. (2) (but with diagonal elements[1] set to ∞), then GIN_0 (and GIN_ϵ) can approximate a weighted shortest path if also proper encoding of the distance is provided. Specifically, by setting $\mathrm{ENC}(\mathbf{W}) = Q_1[\mathrm{MLP}[\mathbf{W}]]$, with Q_1 defined as in eq. (4) (notice that the MLP is applied element-wise).

Proposition 3. *Given $G = (\mathbf{W}, \mathbf{e}_v)$, a positive constant $c > 0$, and $\mathbf{d} = \mathrm{MLP}_\mathrm{D}[D_1[\mathrm{GIN}^k_\epsilon(Q_1[\mathrm{MLP}_\mathrm{E}[\mathbf{W}]], \mathbf{e}_v)]]$, with GIN_ϵ, MLP_E, and MLP_D having any number of layers with* ReLU *activation function, there exist a configuration of their parameters such that $|\mathbf{d}_u - \mathrm{dist}(u, v)| \leq c$ for any $u \in N^k_+(v)$.*

Proof. Fix $w_\theta = 1$ and $b_\theta = 0$ for every layer of GIN/MLP, apart in the first layer of MLP_E, where we set $w_{\theta_1} = 1/h$, and in the first layer of MLP_D, where we set instead $w_{\theta_1} = h$. As in the proof of proposition 1, we have that $\mathrm{GIN}^k_0(\mathbf{A}, \mathbf{e}_v) =$

[1] This assumption is given only for consistency with the previous settings. If we set the diagonal elements of $\mathbf{W}$ to 0, we have to set GIN_{-1} instead.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

$(\mathbf{A}+\mathbf{I})^k\mathbf{e}_v$. Noticing that $Q_1(x/h) = Q_h(x)$ and $h \cdot D_1(x) = D_h(x)$, we obtain that

$$\mathbf{d} = h \cdot D_1[Q_1[\tfrac{1}{h}\mathbf{W}]^k \cdot \mathbf{e}_v] = D_h[Q_h[\mathbf{W}]^k \cdot Q_h[\boldsymbol{\chi}_v]] \xrightarrow[h\to 0]{} \mathbf{W}^{\odot k} \odot \boldsymbol{\chi}_v = \mathbf{d}_\circ,$$

where $\mathbf{d}_\circ$ is defined as in eq. (6). By definition of limit (for a $h > 0$ small enough) we reach the conclusion. □

4 Conclusion

In this preliminary work, we have drawn connections between tropical algebra and Graph Neural Networks (GNNs), especially in the context of approximating Dynamic Programming (DP) algorithms. We proved that a *sum*-aggregated GNN can, through the use of Maslov quantization/dequantization maps, effectively approximate reachability and shortest path algorithms on graphs. By viewing GNNs under the framework of tropical semirings, we hope to unleash further research in this direction, exploring the alignment between GNNs and DP even further, and enabling to reason at a more abstract and general level than that of aligning GNNs with single specific algorithms.

References

[1] D. Bacciu, F. Errica, A. Micheli, and M. Podda. "A gentle introduction to deep learning for graphs". *Neural Networks* 129 (2020).

[2] R. Bellman. "Dynamic Programming". *Science* 153.3731 (1966).

[3] R. Bellman. "On a routing problem". *Quarterly of Applied Mathematics* 16.1 (1958).

[4] E. Brugalle, I. Itenberg, G. Mikhalkin, and K. Shaw. "Brief introduction to tropical geometry" (2000).

[5] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković. "Principal Neighbourhood Aggregation for Graph Nets". *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

[6] L. R. Ford Jr. *Network flow theory*. Tech. rep. Rand Corp Santa Monica Ca, 1956.

[7] G. L. Litvinov and V. P. Maslov. "The correspondence principle for idempotent calculus and some computer applications". *Idempotency*. 1998.

[8] D. Maclagan and B. Sturmfels. "Introduction to tropical geometry". *Graduate Studies in Mathematics* 161 (2009).

[9] A. Shimbel. "Structure in communication nets". *Proceedings of the symposium on information networks*. 1954.

[10] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. "How Powerful are Graph Neural Networks?" *International Conference on Learning Representations*. 2019.

[11] K. Xu, J. Li, M. Zhang, S. S. Du, K.-i. Kawarabayashi, and S. Jegelka. "What Can Neural Networks Reason About?" 2019.

[12] K. Xu, M. Zhang, J. Li, S. S. Du, K.-I. Kawarabayashi, and S. Jegelka. "How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks". 2020.

[13] Z. Zhu, Z. Zhang, L.-P. Xhonneux, and J. Tang. "Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction". *Advances in Neural Information Processing Systems*. Vol. 34. 2021.