Two Households, Both Alike In Dignity

Geometry and Topology in Machine Learning

Bastian Rieck

JMM 2025 · Seattle

Machine learning

Two perspectives

Machine learning

Two perspectives

What is a neural network?

Perspective I: Neural networks as universal function approximators

Theorem (Universal function approximation)

Let $\sigma \in C(\mathbb{R}, \mathbb{R})$ *be a non-polynomial activation function. For every* $n, m \in \mathbb{N}$*, every compact subset* $K \subseteq \mathbb{R}^n$*, every function* $f \in C(K, \mathbb{R}^m)$ *and* $\epsilon > 0$*, there exist* $k \in \mathbb{N}$*,* $\mathbf{A} \in \mathbb{R}^{k \times n}$*,* $b \in \mathbb{R}^k$*, and* $\mathbf{C} \in \mathbb{R}^{m \times k}$ *such that*

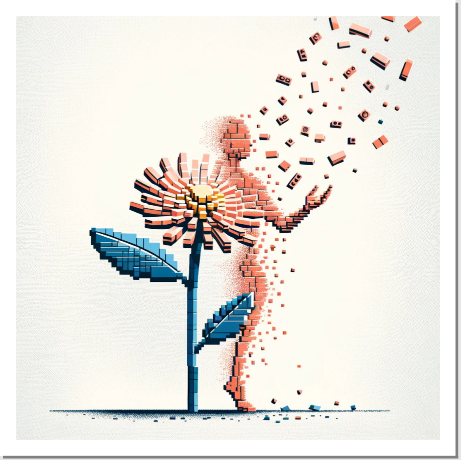
$$\sup_{x \in K} \|f(x) - g(x)\| < \epsilon,$$

where $g(x) = \mathbf{C}\sigma(\mathbf{A}x + b)$.

A. Pinkus, *Approximation theory of the MLP model in neural networks*, *Acta Numerica* 8, 1999, pp. 143–195

What is a neural network?

Perspective II: Neural networks as computational building blocks



The State of the Art in Graph Learning

The book

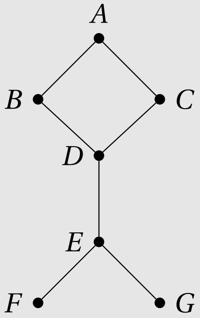
M. M. Bronstein, J. Bruna, T. Cohen and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*, 2021, arXiv: `2104.13478`

Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

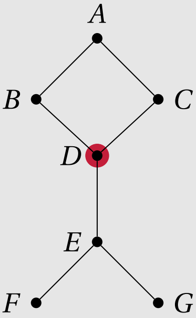


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

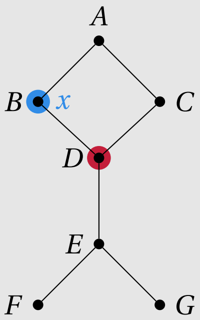


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

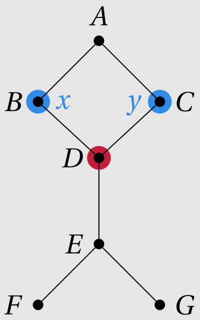


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

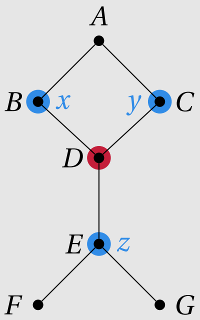


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.

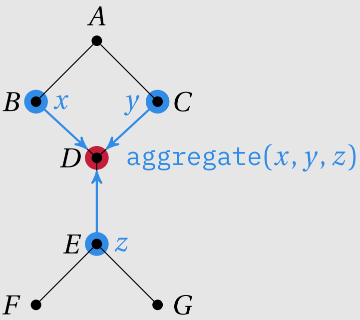


Message passing

The foundational paradigm of geometric deep learning

Idea

Neighbouring nodes in a graph may exchange *messages*, i.e. vectors $x, y, z \in \mathbb{R}^d$, which are subsequently *aggregated* to obtain a single representation.



Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

A simple message-passing neural network

$$h_i^{(t)} = \sigma\left(\mathbf{W_1^{(t)}} h_i^{(t-1)} + \mathbf{W_2^{(t)}} \sum_{j \sim i} h_j^{(t-1)}\right)$$

Message passing

Example

Setup

- $(G, \mathbf{X})$: graph with n vertices and node attributes in $\mathbb{R}^d$
- $h_i^{(t)}$: hidden attributes of vertex i at step t

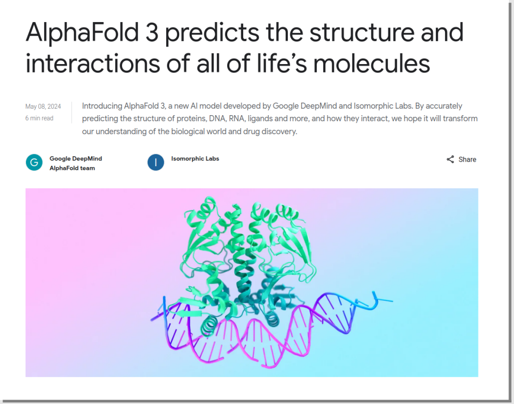
A simple message-passing neural network

$$h_i^{(t)} = \sigma\left(\mathbf{W_1^{(t)}} h_i^{(t-1)} + \mathbf{W_2^{(t)}} \sum_{j \sim i} h_j^{(t-1)}\right)$$

Generalisations

Message passing also works on topological domains like cell complexes, CW complexes, or simplicial complexes!

Not all paradigms are *required* in practice



AlphaFold 3 predicts the structure and interactions of all of life's molecules

May 08, 2024
6 min read

Introducing AlphaFold 3, a new AI model developed by Google DeepMind and Isomorphic Labs. By accurately predicting the structure of proteins, DNA, RNA, ligands and more, and how they interact, we hope it will transform our understanding of the biological world and drug discovery.

Google DeepMind AlphaFold team

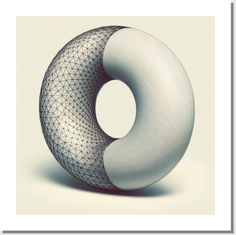
Isomorphic Labs

Share

*We find that **no invariance or equivariance** with respect to global rotations and translation of the molecule are required in the architecture [...]*

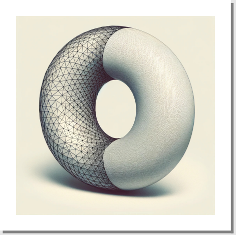
Rethinking the Foundations

Geometry and topology are *dual*



Geometry	=	fine details	+	quantitative answers
Topology	=	fundamental properties	+	qualitative answers

Geometry and topology are *dual*



Geometry	=	fine details	+	quantitative answers
Topology	=	fundamental properties	+	qualitative answers

Data has shape, shape has meaning, and meaning begets understanding.

Gunnar Carlsson, paraphrased

Example

Ollivier–Ricci curvature

Definition

Let G be a graph with its shortest-path metric d and μ_v be a probability measure on G for node $v \in V$. The *Ollivier–Ricci curvature* of a pair of nodes $i \neq j \in V$ is then defined as

$$\kappa_{\mathrm{OR}}(i, j) := 1 - \frac{\mathrm{W}_1(\mu_i, \mu_j)}{\mathrm{d}(i, j)}, \tag{1}$$

where W_1 refers to the first *Wasserstein distance* between μ_i and μ_j.

History

First introduced by Ollivier for metric (measure) spaces, this notion of curvature was quickly adopted for use in the graph setting.

Y. Ollivier, *Ricci curvature of Markov chains on metric spaces*, *Journal of Functional Analysis* 256.3, 2009, pp. 810–864

Example

Useful properties of κ_{OR}

Lower bound

It is sufficient to know the values of κ_{OR} for each edge (i, j). If $\kappa_{\mathrm{OR}}(i, j) \geq K$ for edges $(i, j) \in E$, then $\kappa_{\mathrm{OR}}(k, l) \geq K$ for *all* pairs of vertices (k, l).

Curvature characterises graphs

If $\kappa_{\mathrm{OR}}(i, j) \geq K > 0$ for all edges $(i, j) \in E$, then for $i, j \in V$, we have

$$\mathrm{d}(i, j) \leq \frac{\mathrm{W}_1(\delta_i, \mu_i) + \mathrm{W}_1(\delta_j, \mu_j)}{\kappa_{\mathrm{OR}}(i, j)},$$

where δ_i, δ_j refer to Dirac probability measures centred at node i and j. Thus,

$$\mathrm{diam}(G) \leq \frac{\sup_i \mathrm{W}_1(\delta_i, \mu_i)}{K}.$$

Example

Ollivier–Ricci curvature, applications

Use curvature to evaluate *graph generative models*.

J. Southern*, J. Wayland*, M. Bronstein and B. Rieck, *Curvature Filtrations for Graph Generative Model Evaluation*, Advances in Neural Information Processing Systems, ed. by A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt and S. Levine, vol. 36, Curran Associates, Inc., 2023, pp. 63036–63061, arXiv: `2301.12906 [cs.LG]`

Use curvature as an efficient *graph descriptor* for graph-learning tasks.

L. O'Bray*, B. Rieck* and K. Borgwardt, *Filtration Curves for Graph Representation*, Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA: Association for Computing Machinery, 2021, pp. 1267–1275

Example

What is representation learning?

Slogan

Mapping *things* into *vectors* in $\mathbb{R}^n$ via maps that are often *learnable*.

Example

What is representation learning?

Slogan

Mapping *things* into *vectors* in $\mathbb{R}^n$ via maps that are often *learnable*.

- T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, Preprint, 2013, arXiv: `1301.3781 [cs.CL]`
- A. Grover and J. Leskovec, *`node2vec`: Scalable Feature Learning for Networks*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864
- C. Hacker, *`k-simplex2vec`: A Simplicial Extension of* `node2vec`, 'Topological Data Analysis and Beyond' Workshop at NeurIPS, 2020

Example

Representation learning using the Euler Characteristic Transform

Learn appropriate directions for calculating the ECT and solve classification tasks on point clouds, graphs, and simplicial complexes.

E. Röell and B. Rieck, *Differentiable Euler Characteristic Transforms for Shape Classification*, International Conference on Learning Representations, 2024, arXiv: `2310.07630 [cs.LG]`

Use a *local* variant of the ECT to solve node-classification tasks.

J. von Rohrscheidt and B. Rieck, *Diss-l-ECT: Dissecting Graph Data with local Euler Characteristic Transforms*, Preprint, 2024, arXiv: `2410.02622 [cs.LG]`

A simplicial perspective

Using cochains in representation learning

A simplicial perspective

Using cochains in representation learning

Suggestion

- Each simplex is represented by *evaluating* all cochains.

A simplicial perspective

Using cochains in representation learning

Suggestion

- Each simplex is represented by *evaluating* all cochains.
- Cochains are *feature maps* for the representations.

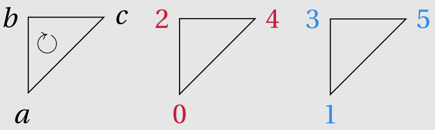
A simplicial perspective

Using cochains in representation learning

Suggestion

- Each simplex is represented by *evaluating* all cochains.
- Cochains are *feature maps* for the representations.

Example



$$\begin{array}{c} a \\ b \\ c \end{array} \begin{pmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{pmatrix}$$

Using higher-order cochains with differential forms

K. Maggs, C. Hacker and B. Rieck, *Simplicial Representation Learning with Neural k-forms*, International Conference on Learning Representations, 2024, arXiv: `2312.08515 [cs.LG]`

Kelly

Celia

Using higher-order cochains with differential forms

K. Maggs, C. Hacker and B. Rieck, *Simplicial Representation Learning with Neural k-forms*, International Conference on Learning Representations, 2024, arXiv: `2312.08515 [cs.LG]`

Kelly

Celia

Transcending message passing

- Need a simplicial complex S with node embeddings in $\mathbb{R}^n$.
- Extend node embeddings to $\phi\colon S \to \mathbb{R}^n$.
- Learn differential k-forms on the ambient space.
- Obtain representations by *integration*.

In a nutshell

$$\underbrace{\Omega^k(\mathbb{R}^n) \xrightarrow{\int} C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})}_{\text{de Rham map}} \underbrace{\xrightarrow{\phi^*} C^k_{\text{simp}}(S;\mathbb{R})}_{\text{Restriction map}}$$

Dramatis personæ

- $\Omega^k(\mathbb{R}^n)$: k-forms on $\mathbb{R}^n$
- $C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})$: singular cochains
- $C^k_{\text{simp}}(S;\mathbb{R})$: simplicial cochains
- $\phi^*\gamma(\sigma) = \gamma(\phi|_\sigma)$ for a singular cochain $\gamma \in C^k_{\text{sing}}(\mathbb{R}^n;\mathbb{R})$.

Slogan

We replace the evaluation of feature cochains by the *integration* of differential feature forms.

How to learn differential k-forms?

Decomposition

Every k-form $\omega \in \Omega^k(\mathbb{R}^n)$ decomposes as

$$\omega = \sum_I f_I dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_k}.$$

Neural k-forms

Given a neural network $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k}}$, its *neural k-form* is

$$\omega(\sigma) = \sum_I \sigma_I dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_k}.$$

Neural k-forms = MLPs

(Multi-layer perceptrons (MLPs) constitute the simplest, best-studied neural-network architecture.)

Higher-order cochains in practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.

Higher-order cochains in practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.

Higher-order cochains in practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma) \right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.

Higher-order cochains in practice

- Assume a data set of embedded simplicial complexes $\{S_\alpha, \phi_\alpha \colon S_\alpha \to \mathbb{R}^n\}$.
- Use MLP to learn $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{\binom{n}{k} \times \ell}$.
- Integrate (S_α, ϕ_α), building the *integration matrix* $X(\alpha, \sigma) := \left[\int_{\phi_\alpha} \omega(\sigma) \right] \in \mathbb{R}^{|S_\alpha| \times \ell}$.
- Reduce $X(\alpha, \sigma)$ via $\operatorname{diag}(X(\alpha, \sigma)^T X(\alpha, \sigma))$, for instance.

Higher-order cochains

Empirical results

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
NkF (ours)	9K	83.50 ± 0.55	86.41 ± 3.64	76.70 ± 2.17

Higher-order cochains

Empirical results

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
NkF (ours)	9K	83.50 ± 0.55	86.41 ± 3.64	76.70 ± 2.17

- We can prove a *universal approximation theorem* for neural k-forms.

Higher-order cochains

Empirical results

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
NkF (ours)	9K	83.50 ± 0.55	86.41 ± 3.64	76.70 ± 2.17

- We can prove a *universal approximation theorem* for neural k-forms.
- Our integration matrices can be easily integrated into *any* architecture.

Higher-order cochains

Empirical results

	Parameters	BACE	BBBP	HIV
EGNN	1M	74.62 ± 2.58	82.67 ± 0.54	68.25 ± 6.74
GAT	135K	69.52 ± 17.52	76.51 ± 3.36	56.38 ± 4.41
GCN	133K	66.79 ± 1.56	73.77 ± 3.30	68.70 ± 1.67
GIN	282K	42.91 ± 18.56	61.66 ± 19.47	55.28 ± 17.49
NkF (ours)	9K	83.50 ± 0.55	86.41 ± 3.64	76.70 ± 2.17

- We can prove a *universal approximation theorem* for neural k-forms.
- Our integration matrices can be easily integrated into *any* architecture.
- 'Learning more with less:' Smaller models but *better* results.

Conclusion

- Machine learning needs more mathematicians.

Conclusion

- Machine learning needs more mathematicians.
- The use of geometrical-topological concepts provides a novel perspective for many machine-learning applications…

Conclusion

- Machine learning needs more mathematicians.
- The use of geometrical-topological concepts provides a novel perspective for many machine-learning applications…
- …but the use of geometrical and topological features should be *justified* and carefully assessed.

Conclusion

- Machine learning needs more mathematicians.
- The use of geometrical-topological concepts provides a novel perspective for many machine-learning applications…
- …but the use of geometrical and topological features should be *justified* and carefully assessed.
- The best time to join is *now*!

Conclusion

- Machine learning needs more mathematicians.
- The use of geometrical-topological concepts provides a novel perspective for many machine-learning applications…
- …but the use of geometrical and topological features should be *justified* and carefully assessed.
- The best time to join is *now*!

Our research