Title: Topological deep learning

URL: https://en.wikipedia.org/wiki/Topological_deep_learning

PageID: 76401012

Categories: Category:Deep learning, Category:Topology

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor

Isolation forest

Autoencoder

Deep learning

Feedforward neural network

Recurrent neural network LSTM GRU ESN reservoir computing

LSTM

GRU

ESN

reservoir computing

Boltzmann machine Restricted

Restricted

GAN

Diffusion model

SOM

Convolutional neural network U-Net LeNet AlexNet DeepDream

U-Net

LeNet

AlexNet

DeepDream

Neural field Neural radiance field Physics-informed neural networks

Neural radiance field

Physics-informed neural networks

Transformer Vision

Vision

Mamba

Spiking neural network

Memtransistor

Electrochemical RAM (ECRAM)

Q-learning

Policy gradient

SARSA

Temporal difference (TD)

Multi-agent Self-play

Self-play

Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

Topological deep learning (TDL) [1] [2] [3] [4] [5] [6] is a research field that extends deep learning to handle complex, non-Euclidean data structures. Traditional deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), excel in processing data on regular grids and sequences. However, scientific and real-world data often exhibit more intricate data domains encountered in scientific computations , including point clouds , meshes , time series , scalar fields graphs , or general topological spaces like simplicial complexes and CW complexes . [7] TDL addresses this by incorporating topological concepts to process data with higher-order relationships, such as interactions among multiple entities and complex hierarchies. This approach leverages structures like simplicial complexes and hypergraphs to capture global dependencies and qualitative spatial properties, offering a more nuanced representation of data. TDL also encompasses methods from computational and algebraic topology that permit studying properties of neural networks and their training process, such as their predictive performance or generalization properties. [8] [9] [10] [11] [12] [13] [14] The mathematical foundations of TDL are algebraic topology , differential topology , and geometric topology . Therefore, TDL can be generalized for data on differentiable manifolds , knots , links, tangles, curves, etc.

History and motivation

Traditional techniques from deep learning often operate under the assumption that a dataset is residing in a highly-structured space (like images , where convolutional neural networks exhibit outstanding performance over alternative methods) or a Euclidean space . The prevalence of new types of data, in particular graphs , meshes , and molecules , resulted in the development of new techniques, culminating in the field of geometric deep learning , which originally proposed a signal-processing perspective for treating such data types. [15] While originally confined to graphs, where connectivity is defined based on nodes and edges, follow-up work extended concepts to a larger variety of data types, including simplicial complexes [16] [3] and CW complexes , [8] [17] with recent work proposing a unified perspective of message-passing on general combinatorial complexes. [1]

An independent perspective on different types of data originated from topological data analysis , which proposed a new framework for describing structural information of data, i.e., their "shape," that is inherently aware of multiple scales in data, ranging from local information to global information. [18] While at first restricted to smaller datasets, subsequent work developed new descriptors that efficiently summarized topological information of datasets to make them available for traditional machine-learning techniques, such as support vector machines or random forests . Such descriptors ranged from new techniques for feature engineering over new ways of providing suitable coordinates for topological descriptors, [19] [20] [21] or the creation of more efficient dissimilarity measures . [22] [23] [24] [25]

Contemporary research in this field is largely concerned with either integrating information about the underlying data topology into existing deep-learning models or obtaining novel ways of training on topological domains.

Learning on topological spaces

Focusing on topology in the sense of point set topology , an active branch of TDL is concerned with learning on topological spaces, that is, on different topological domains.

An introduction to topological domains

One of the core concepts in topological deep learning is the domain upon which this data is defined and supported. In case of Euclidean data, such as images, this domain is a grid, upon which the pixel value of the image is supported. In a more general setting this domain might be a topological domain . Next, we introduce the most common topological domains that are encountered in a deep learning setting. These domains include, but not limited to, graphs, simplicial complexes, cell complexes, combinatorial complexes and hypergraphs.

Given a finite set S of abstract entities, a neighborhood function N {\displaystyle {\mathcal {N}}} on S is an assignment that attach to every point x {\displaystyle x} in S a subset of S or a relation. Such a function can be induced by equipping S with an auxiliary structure . Edges provide one way of defining relations among the entities of S . More specifically, edges in a graph allow one to define the notion of neighborhood using, for instance, the one hop neighborhood notion. Edges however, limited in their modeling capacity as they can only be used to model binary relations among entities of S since every edge is connected typically to two entities. In many applications, it is desirable to permit relations that incorporate more than two entities. The idea of using relations that involve more than two entities is central to topological domains. Such higher-order relations allow for a broader range of neighborhood functions to be defined on S to capture multi-way interactions among entities of S .

Next we review the main properties, advantages, and disadvantages of some commonly studied topological domains in the context of deep learning, including (abstract) simplicial complexes, regular cell complexes, hypergraphs, and combinatorial complexes.

Comparisons among topological domains

Each of the enumerated topological domains has its own characteristics, advantages, and limitations:

Simplicial complexes Simplest form of higher-order domains. Extensions of graph-based models. Admit hierarchical structures, making them suitable for various applications. Hodge theory can be naturally defined on simplicial complexes. Require relations to be subsets of larger relations, imposing constraints on the structure.

Simplest form of higher-order domains.

Extensions of graph-based models.

Admit hierarchical structures, making them suitable for various applications.

Hodge theory can be naturally defined on simplicial complexes.

Require relations to be subsets of larger relations, imposing constraints on the structure.

Cell Complexes Generalize simplicial complexes. Provide more flexibility in defining higher-order relations. Each cell in a cell complex is homeomorphic to an open ball, attached together via attaching maps. Boundary cells of each cell in a cell complex are also cells in the complex. Represented combinatorially via incidence matrices.

Generalize simplicial complexes.

Provide more flexibility in defining higher-order relations.

Each cell in a cell complex is homeomorphic to an open ball, attached together via attaching maps.

Boundary cells of each cell in a cell complex are also cells in the complex.

Represented combinatorially via incidence matrices.

Hypergraphs Allow arbitrary set-type relations among entities. Relations are not imposed by other relations, providing more flexibility. Do not explicitly encode the dimension of cells or relations. Useful when relations in the data do not adhere to constraints imposed by other models like simplicial and cell complexes.

Allow arbitrary set-type relations among entities.

Relations are not imposed by other relations, providing more flexibility.

Do not explicitly encode the dimension of cells or relations.

Useful when relations in the data do not adhere to constraints imposed by other models like simplicial and cell complexes.

Combinatorial Complexes [1] : Generalize and bridge the gaps between simplicial complexes, cell complexes, and hypergraphs. Allow for hierarchical structures and set-type relations. Combine features of other complexes while providing more flexibility in modeling relations. Can be represented combinatorially, similar to cell complexes.

Generalize and bridge the gaps between simplicial complexes, cell complexes, and hypergraphs.

Allow for hierarchical structures and set-type relations.

Combine features of other complexes while providing more flexibility in modeling relations.

Can be represented combinatorially, similar to cell complexes.

Hierarchical structure and set-type relations

The properties of simplicial complexes, cell complexes, and hypergraphs give rise to two main features of relations on higher-order domains, namely hierarchies of relations and set-type relations. [1]

A rank function on a higher-order domain X is an order-preserving function rk : X → Z , where rk (x) attaches a non-negative integer value to each relation x in X , preserving set inclusion in X . Cell and simplicial complexes are common examples of higher-order domains equipped with rank functions and therefore with hierarchies of relations. [1]

Relations in a higher-order domain are called set-type relations if the existence of a relation is not implied by another relation in the domain. Hypergraphs constitute examples of higher-order domains equipped with set-type relations. Given the modeling limitations of simplicial complexes, cell complexes, and hypergraphs, we develop the combinatorial complex, a higher-order domain that features both hierarchies of relations and set-type relations. [1]

The learning tasks in TDL can be broadly classified into three categories: [1]

Cell classification : Predict targets for each cell in a complex. Examples include triangular mesh segmentation, where the task is to predict the class of each face or edge in a given mesh.

Complex classification : Predict targets for an entire complex. For example, predict the class of each input mesh.

Cell prediction : Predict properties of cell-cell interactions in a complex, and in some cases, predict whether a cell exists in the complex. An example is the prediction of linkages among entities in hyperedges of a hypergraph.

In practice, to perform the aforementioned tasks, deep learning models designed for specific topological spaces must be constructed and implemented. These models, known as topological neural networks, are tailored to operate effectively within these spaces.

Topological neural networks

Central to TDL are topological neural networks (TNNs) , specialized architectures designed to operate on data structured in topological domains. [2] [1] Unlike traditional neural networks tailored for grid-like structures, TNNs are adept at handling more intricate data representations, such as graphs, simplicial complexes, and cell complexes. By harnessing the inherent topology of the data, TNNs can capture both local and global relationships, enabling nuanced analysis and interpretation.

Message passing topological neural networks

In a general topological domain, higher-order message passing involves exchanging messages among entities and cells using a set of neighborhood functions.

Definition: Higher-Order Message Passing on a General Topological Domain

Let X {\displaystyle {\mathcal {X}}} be a topological domain. We define a set of neighborhood functions N = { N 1 , … , N n } {\displaystyle {\mathcal {N}}=\{{\mathcal {N}}_{1},\ldots ,{\mathcal {N}}_{n}\}} on X {\displaystyle {\mathcal {X}}} . Consider a cell x {\displaystyle x} and let y ∈ N k (x) {\displaystyle y\in {\mathcal {N}}_{k}(x)} for some N k ∈ N {\displaystyle {\mathcal {N}}_{k}\in {\mathcal {N}}} . A message m x , y {\displaystyle m_{x,y}} between cells x {\displaystyle x} and y {\displaystyle y} is a computation dependent on these two cells or the data supported on them. Denote N (x) {\displaystyle {\mathcal {N}}(x)} as the multi-set { { N 1 (x) , … , N n (x) } } {\displaystyle \{\!\!\{{\mathcal {N}}_{1}(x),\ldots ,{\mathcal {N}}_{n}(x)\}\!\!\}} , and let h x (l) {\displaystyle \mathbf {h} _{x}^{(l)}} represent some data supported on cell x {\displaystyle x} at layer l {\displaystyle l} . Higher-order message passing on X {\displaystyle {\mathcal {X}}} , [1] [8] induced by N {\displaystyle {\mathcal {N}}} , is defined by the following four update rules:

m x , y = α N k (h x (l) , h y (l)) {\displaystyle m_{x,y}=\alpha _{{\mathcal {N}}_{k}}(\mathbf {h} _{x}^{(l)},\mathbf {h} _{y}^{(l)})}

m x k = ■ y ∈ N k (x) m x , y {\displaystyle m_{x}^{k}=\bigoplus _{y\in {\mathcal {N}}_{k}(x)}m_{x,y}} , where ■ {\displaystyle \bigoplus } is the intra-neighborhood aggregation function.

m x = ■ N k ∈ N m x k {\displaystyle m_{x}=\bigotimes _{{\mathcal {N}}_{k}\in {\mathcal {N}}}m_{x}^{k}} , where ■ {\displaystyle \bigotimes } is the inter-neighborhood aggregation function.

h x (l + 1) = β (h x (l) , m x) {\displaystyle \mathbf {h} _{x}^{(l+1)}=\beta (\mathbf {h} _{x}^{(l)},m_{x})} , where α N k , β {\displaystyle \alpha _{{\mathcal {N}}_{k}},\beta } are differentiable functions.

Some remarks on Definition above are as follows.

First, Equation 1 describes how messages are computed between cells x {\displaystyle x} and y {\displaystyle y} . The message m x , y {\displaystyle m_{x,y}} is influenced by both the data h x (l) {\displaystyle \mathbf {h} _{x}^{(l)}} and h y (l) {\displaystyle \mathbf {h} _{y}^{(l)}} associated with cells x {\displaystyle x} and y {\displaystyle y} , respectively. Additionally, it incorporates characteristics specific to the cells themselves, such as orientation in the case of cell complexes. This allows for a richer representation of spatial relationships compared to traditional graph-based message passing frameworks.

Second, Equation 2 defines how messages from neighboring cells are aggregated within each neighborhood. The function ■ {\displaystyle \bigoplus } aggregates these messages, allowing information to be exchanged effectively between adjacent cells within the same neighborhood.

Third, Equation 3 outlines the process of combining messages from different neighborhoods. The function ■ {\displaystyle \bigotimes } aggregates messages across various neighborhoods, facilitating communication between cells that may not be directly connected but share common neighborhood relationships.

Fourth, Equation 4 specifies how the aggregated messages influence the state of a cell in the next layer. Here, the function β {\displaystyle \beta } updates the state of cell x {\displaystyle x} based on its current state h x (l) {\displaystyle \mathbf {h} _{x}^{(l)}} and the aggregated message m x {\displaystyle m_{x}} obtained from neighboring cells.

Non-message passing topological neural networks

While the majority of TNNs follow the message passing paradigm from graph learning , several models have been suggested that do not follow this approach. For instance, Maggs et al. [26] leverage geometric information from embedded simplicial complexes, i.e., simplicial complexes with high-dimensional features attached to their vertices.This offers interpretability and geometric consistency without relying on message passing. Furthermore, in [27] a contrastive loss-based method was suggested to learn the simplicial representation.

Learning on topological descriptors

Motivated by the modular nature of deep neural networks , initial work in TDL drew inspiration from topological data analysis , and aimed to make the resulting descriptors amenable to integration into deep-learning models. This led to work defining new layers for deep neural networks. Pioneering work by Hofer et al., [28] for instance, introduced a layer that permitted topological descriptors like persistence diagrams or persistence barcodes to be integrated into a deep neural network. This was achieved by means of end-to-end-trainable projection functions, permitting topological features to be used to solve shape classification tasks, for instance. Follow-up work expanded more on the theoretical properties of such descriptors and integrated them into the field of representation learning . [29] Other such topological layers include layers based on extended persistent homology descriptors, [30] persistence landscapes, [31] or coordinate functions. [32] In parallel, persistent homology also found applications in graph-learning tasks. Noteworthy examples include new algorithms for learning task-specific filtration functions for graph classification or node classification tasks. [33] [34] [35]

Applications

TDL is rapidly finding new applications across different domains, including data compression, [36] enhancing the expressivity and predictive performance of graph neural networks , [16] [17] [33] action recognition, [37] and trajectory prediction. [38]

References