Topological Deep Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Topological deep learning is a formalism that is aimed at introducing topological language to deep learning for the purpose of utilizing the minimal mathematical structures to formalize problems that arise in a generic deep learning problem. In this article, we define and study the classification problem in machine learning in a topological setting. Using this topological framework, we show when the classification problem is possible or not possible in the context of neural networks. Finally, we demonstrate how our topological setting immediately illuminates aspects of this problem that are not as readily apparent using traditional tools.

1. Introduction

Recent years have witnessed increased interest in the role topology plays in machine learning and data science [5]. Topology is a natural tool that allows the formulation of many longstanding problems in these fields. For instance, *persistent homology* [10] has been overwhelmingly successful at finding solutions to a vast array of complex data problems [1, 2, 3, 6, 7, 9, 12, 16, 17, 18, 19, 20, 23, 25, 26].

On the other hand, the role that topology plays in deep learning is still mostly restricted to techniques that attempt to enhance machine learning models [14, 4, 28]. However, we believe that topology can and will play a central role in deep learning and AI in general. Our purpose of this article is to introduce *topological deep learning*, a formalism that is aimed at introducing topological language to deep learning for the purpose of utilizing the minimal mathematical structures to formalize problems that arise in a generic deep learning problem.

To this end we define and study the classification problem in a topological setting. Using this topological machinery, we show when the classification problem is possible or not possible in the context of neural networks. Finally, we show how the architecture of a neural network cannot be chosen independently from the topology of the underlying data. To demonstrate these results, we provide an example dataset and show how it is acted upon by a neural net from this topological perspective.

2. Background

A *neural network*, or simply a *network*, is a function $Net : \mathbb{R}^{d_{in}} \longrightarrow \mathbb{R}^{d_{out}}$ defined by a composition of the form:

$$Net := f_L \circ \cdots \circ f_1 \tag{1}$$

where the functions f_i, $1 \leq i \leq L$ are called the *layer functions*. A layer function $f_i : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}^{m_i}$ is typically a continuous, piece-wise smooth function of the following form: $f_i(x) = \sigma(W_i(x) + b_i)$ where W_i is an $m_i \times n_i$ matrix, b_i is a vector in $\mathbb{R}^{m_i}$, and $\sigma : \mathbb{R} \longrightarrow \mathbb{R}$ is an appropriately chosen nonlinear function that is applied coordinate-wise on an input vector $(z_1, \cdots, z_{m_i})$ to get a vector $(\sigma(z_1), \cdots, \sigma(z_{m_i}))$.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

3. Data In a Topological Setting

In the present article we clearly distinguished between data and the functions that operate on it.This distinction is important because data as a separate mathematical object have complex properties that intertwine non-trivially with the functions, that also have unique properties, that operate on the data. The purpose of this section is define the notion of data using topological notions.

3.1. Topological Data

Denote by M^n to a manifold M of dimension n. Let $D = M_1^{i_1} \cup M_2^{i_2} \cdots \cup M_k^{i_k}$ be a disjoint union of k compact manifolds. Let $h : D \longrightarrow E$ be a continuous function on D. We refer to the pair (D,h) as *topological data* and refer to E as the *the ambient space* of the topological data, or simply the ambient space of the data.

A few remarks here must be made about the above definition. First note that the definition above is consistent with the statistical version. The space E, usually some Euclidean space, represents the ambient space of a probability distribution μ from which we sample the data. The support of μ is $\mathcal{D} := h(D)$. The assumption that the data lives on a manifold-like structure is justified in the literature [11, 21]. [1]

3.2. Topologically Labeled Data

Let (D,h) be topological data with $h : D \to \mathcal{D} \subset E$. Let $\mathcal{Y} = \{l_1, \cdots, l_d\}$ be a finite set. A *topological labeling* on $\mathcal{D}$ is a closed subset $\mathcal{D}_L \subset \mathcal{D}$ along with a surjective continuous function $g : \mathcal{D}_L \to \mathcal{Y}$ where $\mathcal{Y}$ is given the discrete topology. The triplet (D,h,g) will be called *topologically labeled data*.

Topologically labeled data is a topological object that corresponds to labeled data in the typical statistical setting for a supervised classification machine learning problem.

4. The Topological Classification Problem

With the above setting we now demonstrate how to realize the classification problem as a topological problem. In what follows we set $\mathcal{D}_k$ to denote $g^{-1}(l_k)$ for $l_k \in \mathcal{Y}$.

Definition 1. *Let (D,h,g) be topologically labeled data with, $h : D \to \mathcal{D} \subset \mathbb{R}^n$ and $g : \mathcal{D}_L \to \mathcal{Y}$ where $|\mathcal{Y}| = d$. A topological classifier on (D,h,g) is a continuous function $f : \mathbb{R}^n \to \mathbb{R}^k$. We say that f separates the topologically labeled data (D,h,g) if we can find k disjoint embedded k-dimensional discs $A_1, \cdots, A_k$ in $\mathbb{R}^k$ such that $f(\mathcal{D}_k) \subset A_k$.*

In general, a topologically labeled data can be knotted, linked and entangled together in a non-trivial manner by the embedding h, and the existence of a function f that separates this data is not immediate. The preceding description is an topological rewording of the classification problem typically given in a statistical setting. Indeed, a successful classifier tries to *separate* the labeled data by mapping the raw input data into another space where this data can be separated easily according to the given class.

The function f is the learning function that we try to compute, in practice. The first question one could ask in this context is one of existence: given topologically labeled data (D,h,g) when can we find a function f that separates this data? We answer this question next.

4.1. Topological Classifiers and Separability of Topologically Labeled Data

We start with the binary classification problem, namely when $|\mathcal{Y}| = 2$. We have the following proposition:

Proposition 4.1. *Let (D,h,g) by a topologically labeled data with $h : D \longrightarrow \mathcal{D} \subset \mathbb{R}^{d_{in}}$ and $g : \mathcal{D}_L \to \{l_1, l_2\}$. Then there exists a topological classifier $f : \mathbb{R}^{d_{in}} \to \mathbb{R}$ that separates (D,h,g).*

Proof. The label function $g : \mathcal{D}_L \longrightarrow \{l_1, l_2\}$ induces a partition on $\mathcal{D}_L$ into two disjoint closed sets $\mathcal{D}_1 := g^{-1}(l_1)$ and $\mathcal{D}_2 := g^{-1}(l_2)$. By Urysohn's lemma there exists a function $f^* : \mathcal{D}_L \longrightarrow [0,1]$

[1] While we make this assumption here, it not strictly necessary anywhere in our proofs.

such that $f^*(\mathcal{D}_1) = 0$ and $f^*(\mathcal{D}_2) = 1$. Since $\mathcal{D}_L$ is closed in $\mathbb{R}^{d_{in}}$ then by Tietze extension theorem there exists an extension of f^* to a continuous function $\hat{f} : \mathbb{R}^{in} \to \mathbb{R}$ such that $f^*(\mathcal{D}_L) = f(\mathcal{D}_L)$. In particular, $f(\mathcal{D}_1) = 0$ and $f(\mathcal{D}_2) = 1$. Hence the function f separates (D,h,g).

Proposition 4.1 can be easily generalized to obtain functions that separate (D,h,g) in any Euclidean space $\mathbb{R}^k$. Namely, for any $k \geq 1$ there exists a continuous map $F : \mathbb{R}^{d_{in}} \longrightarrow \mathbb{R}^k$ that separates (D,h,g). This can be done by defining $F = (f_1, f_2)$ where $f_1 : \mathbb{R}^{d_{in}} \longrightarrow [0,1]$ is the continuous function guaranteed by Urysohn's Lemma and $f_2 : \mathbb{R}^{d_{in}} \longrightarrow \mathbb{R}^{k-1}$ is an arbitrary continuous function. This function F clearly separates (X,h,g). We record this fact in the following proposition.

Proposition 4.2. *Let (D,h,g) by a topologically labeled data with $h : D \to \mathcal{D} \subset \mathbb{R}^{d_{in}}$ and $g : \mathcal{D}_L \to \{l_1, l_2\}$. Then for any $k \geq 1$ there exists a continuous map $f : \mathbb{R}^{d_{in}} \to \mathbb{R}^k$ that separates (D,h,g).*

Proposition 4.2 can be generalized to the case when the set $\mathcal{Y}$ has an arbitrary finite size. This can be done by because Urysohn's Lemma remains valid when we start with n disjoint sets instead of 2. The following theorem, which generalizes 4.2, asserts the existence of a topological classifier f that separates any given topologically labeled data.

Theorem 4.3. *Let (D,h,g) be topologically labeled data with $h : D \to \mathcal{D} \subset R^{d_{in}}$ and $g : \mathcal{D}_L \to \mathcal{Y}$. Then there exists a continuous map $f : \mathbb{R}^{d_{in}} \to \mathbb{R}^k$ that separates (D,h,g) for any integer $k \geq 1$.*

5. Neural Networks as Topological Classifiers

Let (D,h,g) by a topologically labeled data with, $h : D \to \mathcal{D} \subset R^{d_{in}}$ and $g : \mathcal{D}_L \to \mathcal{Y} = \{l_1, \cdots l_n\}$. Can we find a neural network defined on $R^{d_{in}}$ that separates the data (D,h,g) ? We start by framing the softmax classification networks using topological terminologies.

Typical, classification neural networks have a special layer function at the end where one uses the *softmax activation function* [2]. Denote by Δ_n the n^{th} simplex as the convex hull of the vertices $\{v_0,\cdots,v_n\}$ where $v_i = (0,...,1,...,0) \in \mathbb{R}^{n+1}$ with the lone 1 in the $(i+1)^{th}$ coordinate.

The *softmax function* on n vertices $softmax : R^n \longrightarrow Int(\Delta_{n-1}) \subset R^n$, is defined by the composition $S \circ Exp$ where $Exp : \mathbb{R}^n \to (\mathbb{R}^+)^n$ is defined by : $Exp(x_1,\cdots,x_n) = (\exp(x_1), \cdots, \exp(x_n))$, and $S : \mathbb{R}^n \to \Delta_{n-1}$ is defined by :$S(x_1,\cdots,x_n) = (x_1/\sum_{i=1}^n x_i, \cdots, x_n/\sum_{i=1}^n x_i)$.

A network Net is said to be a *softmax classification neural network* with n labels if the final layer of Net is softmax function with n vertices. Usually n is the number of labels in the classification problem. Each vertex v_i in Δ_{n-1} corresponds to precisely one label $l_{i+1} \in \mathcal{Y}$ for $0 \leq i \leq n-1$.

For an input $x \in \mathcal{D}$ the point $Net(x)$ is an element of Δ_{n-1}. By definition, the point x is assigned to the label l_{i+1} by the neural network if and only if $Net(x) \in Int(VC(v_i))$ where $VC(C)$ denotes the Voronoi cell of the set C and $Int(A)$ denotes the interior of a set A. This immediately yields the following theorem.

Theorem 5.1. *Let (D,h,g) by a topologically labeled data with, $h : D \to \mathcal{D} \subset R^{d_{in}}$ and $g : \mathcal{D}_L \subset \mathbb{R}^{d_{in}} \to \{l_1, \cdots l_n\}$. A softmax classification neural network $Net : \mathbb{R}^{d_{in}} \to Int(\Delta_{n-1})$ separates (D,h,g) if and only if $Net(\mathcal{D}_{i+1}) \subset Int(VC(v_i))$ for $0 \leq i \leq n-1$.*

Finally, to answer the question about the ability of a neural network to separate a topologically labeled data, we combine the result we obtained from Theorem 4.3 with the universality of neural networks [8, 13, 22] [3]. The universality of neural networks essentially states that for any continuous function f we can find a network that approximates it to an arbitrary precision[4]. Hence we conclude that any topologically labeled data can effectively be separated by a neural network.

[2] There are other types of classification neural networks but this is beyond the scope of our discussion here

[3] The universal approximation theorem is available in many flavors : one may fix the depth of the network and vary the width or the other way around.

[4] The closeness between functions is with respect to an appropriate functional norm. See [8, 22] for more details.

6. Shape of Data and Neural Networks

We end our discussion by briefly showing how the shape of input data is essential when deciding on the architecture of the neural network. Theorem 6.1 that if we are not careful about the choice of the first layer function of a network then we can always find a topologically labeled data that cannot be separated by this network.

Theorem 6.1. *Let Net be neural network of the form : $Net = Net_1 \circ f_1$ with $f_1 : \mathbb{R}^n \longrightarrow \mathbb{R}^k$ such that $f_1(x) = \sigma(W(x) + b)$ and $k < n$ and $Net_1 : \mathbb{R}^k \longrightarrow \mathbb{R}^d$ is an arbitrary net. Then there exists a topologically labeled data (D,h,g) with $h : D \to \mathcal{D} \subset \mathbb{R}^n$ and $g : \mathcal{D}_L \subset \mathcal{D} \to \mathbb{R}^d$ that is not separable by Net.*

Proof. Let $D = \mathcal{D} = \{x \in \mathbb{R}^n, ||x|| \leq 2\}$. Let $\mathcal{D}_L = \mathcal{D}_1 \cup \mathcal{D}_2$ where $\mathcal{D}_1 = \{x \in \mathbb{R}^n, ||x|| \leq 0{,}9\}$ and $\mathcal{D}_2 = \{x \in \mathbb{R}^n, 1 \leq ||x|| \leq 2\}$. Choose $g : \mathcal{D}_L \longrightarrow \{l_1,l_2\}$ such that $g(\mathcal{D}_1) = l_1$ and $g(\mathcal{D}_2) = l_2$. Let f_1 be a function as defined in the Theorem. The matrix $W : \mathbb{R}^n \longrightarrow \mathbb{R}^k$ where $k < n$ has a nontrivial kernel. Hence, there is a non-trivial vector $v \in \mathbb{R}^n$ such that $W(v) = 0$. Choose a point $p_1 \in \mathcal{D}_1$ and a point $p_2 \in \mathcal{D}_2$ on the line that passes through the origin and has the direction of v. We obtain $W(p_1) = W(p_2) = 0$. In other words, $f_1(p_1) = f_1(p_2)$. Hence $Net(p_1) = Net(p_2)$ and hence $Net(\mathcal{D}_1) \cap Net(\mathcal{D}_2) \neq \varnothing$ and so we cannot find two embedded disks that separate the sets $Net(\mathcal{D}_1)$, $Net(\mathcal{D}_2)$.

Note that in Theorem 6.1 the statement is independent of the depth of the neural network. This is also related to the work [15] which shows that skinny neural networks are not universal approximators. This is also related to the work in [24] where is was shown that a network has to be wide enough in order to successfully classify the input data.

To demonstrate the role that the topology of data may play in regard to the architecture of a neural network we end our discussion by considering the following example. Let Net be a neural network given by the composition $Net = f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1$. For $1 \leq i \leq 5$ maps are given by $f_i := Relu(W_i(x) + b_i)$ such that $W_1 : \mathbb{R}^2 \to \mathbb{R}^5$, $W_2 : \mathbb{R}^5 \to \mathbb{R}^5$, $W_3 : \mathbb{R}^5 \to \mathbb{R}^2$ and $W_j : \mathbb{R}^2 \to \mathbb{R}^2$ for $4 \leq j \leq 5$. Finally, the function, $f_5 = softmax(W_6(x) + b_6)$ where $W_6 : \mathbb{R}^5 \to \mathbb{R}^2$.

We train this network on the annulus dataset given in the top left Figure in 1. In Figure 1 we also trace the activations as demonstrated in Figure 1. In the Figure we visualize the activations in higher dimension by projecting them using Isomap [27] to $\mathbb{R}^3$. Our choice of this algorithm as a dimensionality reduction algorithm is driven by the fact that the dataset we work with here is essentially a manifold; as such, projecting the space to a lower dimension with the Isomap algorithm should preserve most of the topological and geometric structure of the this space.

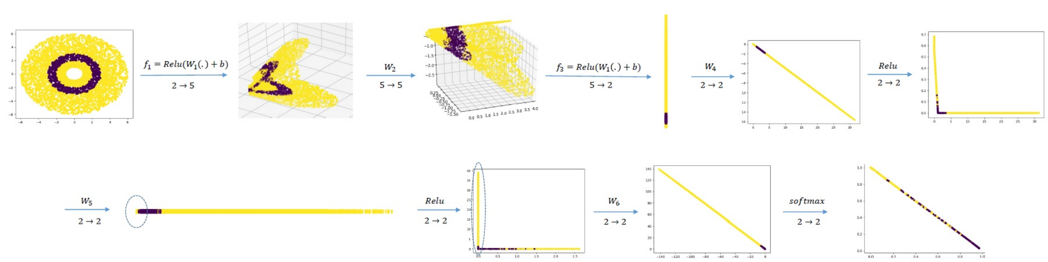

Figura 1: The topological operations performed by a network on data sampled from the annulus and colored by two lables.

Inspecting the activations in Figure 1 we make the following observation:

1. A neural network can collapse the topological space either using the nonlinear $Relu$ or by utilizing the linear part of a given layer function. This is the case with the map $f_3 : \mathbb{R}^5 \longrightarrow \mathbb{R}^2$. While the linear component is a projection onto $\mathbb{R}^2$, the network çhose"to project the space into $1-$ manifold since the second dimension is not needed for the final classification.
2. Note that the yellow components are separated by the purple one, and in order to map both of these parts to the same part of the space, the net has to glue these two parts together. Indeed, the neural network quotients parts of the space as it sees it necessary. This is visible in W_5, which acts as a projection, and again W_6.

Referencias

[1] M. Attene, S. Biasotti, and M. Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, 19(2):127–138, 2003.

[2] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *Proceedings of the 8th Conference on Visualization'97*, pages 167–ff. IEEE Computer Society Press, 1997.

[3] R. L. Boyell and H. Ruston. Hybrid techniques for real-time radar simulation. In *Proceedings of the November 12-14, 1963, fall joint computer conference*, pages 445–458. ACM, 1963.

[4] R. Brüel-Gabrielsson, B. J. Nelson, A. Dwaraknath, P. Skraba, L. J. Guibas, and G. Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2019.

[5] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[6] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization*, pages 497–504, 2004.

[7] C. Curto. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society*, 54(1):63–78, 2017.

[8] G. Cybenko. Approximations by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:183–192, 1989.

[9] Y. Dabaghian, F. Mémoli, L. Frank, and G. Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLoS Computational Biology*, 8(8):e1002581, 2012.

[10] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[11] C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[12] C. Giusti, R. Ghrist, and D. S. Bassett. Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data. *Journal of computational neuroscience*, 41:1, 2016.

[13] B. Hanin and M. Sellke. Approximating continuous functions by relu nets of minimal width. *arXiv preprint arXiv:1710.11278*, 2017.

[14] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1634–1644, 2017.

[15] J. Johnson. Deep, skinny neural networks are not universal approximators. *arXiv preprint arXiv:1810.00393*, 2018.

[16] I. S. Kweon and T. Kanade. Extracting topographic terrain features from elevation maps. *CVGIP: image understanding*, 59(2):171–182, 1994.

[17] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee. Computing the shape of brain networks using graph filtration and gromov-hausdorff metric. *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 302–309, 2011.

[18] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee. Discriminative persistent homology of brain networks. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 841–844, 2011.

[19] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, 2012.

[20] H. Lee, H. Kang, M. K. Chung, B.-N. Kim, and D. S. Lee. Weighted functional brain network modeling via network filtration. *NIPS Workshop on Algebraic Topology and Machine Learning*, 2012.

[21] N. Lei, D. An, Y. Guo, K. Su, S. Liu, Z. Luo, S.-T. Yau, and X. Gu. A geometric understanding of deep learning. *Engineering*, 2020.

[22] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239, 2017.

[23] P. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3:1236, 2013.

[24] Q. Nguyen, M. C. Mukkamala, and M. Hein. Neural networks should be wide enough to learn disconnected decision regions. *arXiv preprint arXiv:1803.00094*, 2018.

[25] M. Nicolau, A. J. Levine, and G. Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

[26] P. Rosen, B. Wang, A. Seth, B. Mills, A. Ginsburg, J. Kamenetzky, J. Kern, and C. R. Johnson. Using contour trees in the analysis and visualization of radio astronomy data cubes. *arXiv preprint arXiv:1704.04561*, 2017.

[27] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[28] F. Wang, H. Liu, D. Samaras, and C. Chen. Topogan: A topology-aware generative adversarial network.