Topological Deep Learning

Mustafa Hajij
Santa Clara University
mhajij@scu.edu

Karthikeyan Natesan ramamurthy
IBM Research
knatesa@us.ibm.com

Aldo Guzmán-Sáenz
IBM Research
aldo.guzman.saenz@ibm.com

Kyle Istvan
kyleistvan@gmail.com

Abstract

Data can naturally be modeled using topological terms. Indeed, the field of topological data analysis fundamentally relies on the idea that the shape of data carries important information that can be utilized to study and analyze the underlying data. In this article, we define a topological framework of data in the context of supervised classification using a neural network. We demonstrate that topological considerations of the available data exclude certain architectures from being trained to perform classification successfully.

1 Introduction

As machine learning practitioners are starting to realize the importance of understanding data used to train models, mathematical notions of data must be revisited from points of view that are not necessarily statistical. The purpose of this article is to introduce a topological perspective for data in the context of neural network classifiers. Using this topological machinery, we show when the classification problem is impossible for neural networks by considering the topology of the underlying data. We also show how the architecture of a neural network cannot be chosen independently from the data's topology. To demonstrate these mechanisms, we provide an example dataset and demonstrate the changes in topology that occur as it is acted upon by a neural network.

2 Previous Work

The earliest hints of which we are aware related to our work appear in a blog by C. Olah [10]. Olah performed a number of topological experiments illustrating the importance of considering the topology of the underlying data when making a neural network. In [8], the activations of a binary classification neural network were considered as point clouds upon which the layer functions of the network act. The topologies of these activations were then studied using persistent homology [2]. The authors Zeiler et. al. in [13] introduced a visualization technique that gives insight into the intermediate layers of convolutional neural networks. The authors in [12] also provide ways to visualize and interpret a given convolutional network by looking at filter activations.

3 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: $Net := f_L \circ \cdots \circ f_1$ 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 $(z_1, \ldots, z_{m_i})$ to get a vector $(\sigma(z_1), \ldots, \sigma(z_{m_i}))$.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

4 Data In a Topological Setting

The purpose of this section is define data using the language of topology.

4.1 Topological Data

We use M^n to refer 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. See Fig 1 for an illustration.

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 [3, 6].[1]

4.2 Topologically Labeled Data

Let (D,h) be topological data with $h : D \to \mathcal{D} \subset E$. Let $\mathcal{Y} = \{l_1, \ldots, l_d\}$ 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 corresponds to labeled data in the typical statistical setting for a supervised classification machine learning problem. Specifically, here we think of each set $\mathcal{D}_k := g^{-1}(l_k) \subset \mathcal{D}$ as the support of the distribution of points inside $\mathcal{D}$ which have particular label l_k. Moreover, d is the number of labels in the dataset in the statistical setting.

5 The Topological Classification Problem

With the above setting we now demonstrate how to realize the classification problem as a topological one. 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 $g : \mathcal{D}_L \subset \mathbb{R}^n \to \mathcal{Y}$ 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 d disjoint embedded k-dimensional discs $A_1, \ldots, A_d$ in $\mathbb{R}^k$ such that $f(\mathcal{D}_k) \subset A_k$, for $1 \leq k \leq d$. See Fig 1 for an illustration.*

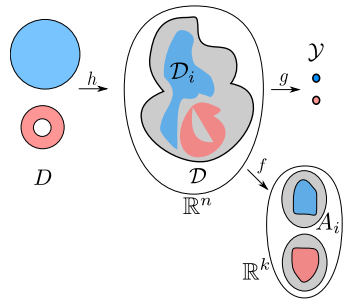

Figure 1: Illustration of Definition 1. A topological classifier *separates* the data by sending each labeled part to a different and easily separably regions of the classifier's codomain.

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 a topological framing 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.

5.1 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 5.1. *Let (D, h, g) by a topologically labeled data with $h : D \longrightarrow \mathcal{D} \subset \mathbb{R}^n$ and $g : \mathcal{D}_L \to \{l_1, l_2\}$. Then there exists a topological classifier $f : \mathbb{R}^n \to \mathbb{R}$ that separates (D,h,g).*

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

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

Proposition 5.1 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 $F : \mathbb{R}^n \longrightarrow \mathbb{R}^k$ that separates (D,h,g). This can be done by defining $F = (f_1, f_2)$ where $f_1 : \mathbb{R}^n \longrightarrow [0,1]$ is the continuous function guaranteed by Urysohn's Lemma and $f_2 : \mathbb{R}^n \longrightarrow \mathbb{R}^{k-1}$ is an arbitrary continuous function. This function F clearly separates (X,h,g). We record this fact in the following proposition.

Proposition 5.2. *Let (D,h,g) by a topologically labeled data with $h : D \to \mathcal{D} \subset \mathbb{R}^n$ and $g : \mathcal{D}_L \to \{l_1, l_2\}$. Then for any $k \geq 1$ there exists a continuous map $f : \mathbb{R}^n \to \mathbb{R}^k$ that separates (D, h, g).*

Proposition 5.2 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 5.2, asserts the existence of a topological classifier f that separates any given topologically labeled data.

Theorem 5.3. *Let (D, h, g) be topologically labeled data with $h : D \to \mathcal{D} \subset R^n$ and $g : \mathcal{D}_L \to \mathcal{Y}$. Then there exists a continuous map $f : \mathbb{R}^n \to \mathbb{R}^k$ that separates (D, h, g) for any integer $k \geq 1$.*

6 Neural Networks as Topological Classifiers

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

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

Denote by $Int(A)$ denotes the interior of a set A. The *softmax function* on d vertices $softmax : \mathbb{R}^d \longrightarrow Int(\Delta_{d-1}) \subset \mathbb{R}^d$, is defined by the composition $S \circ Exp$ where $Exp : \mathbb{R}^d \to (\mathbb{R}^+)^d$ is defined by : $Exp(x_1, \ldots, x_d) = (\exp(x_1), \ldots, \exp(x_d))$, and $S : \mathbb{R}^d \to \Delta_{d-1}$ is defined by: $S(x_1, \ldots, x_d) = (x_1 / \sum_{i=1}^d x_i, \ldots, x_d / \sum_{i=1}^d x_i)$.

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

For an input $x \in \mathcal{D}$ the point $Net(x)$ is an element of Δ_{d-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. This immediately yields the following theorem.

Theorem 6.1. *Let (D, h, g) by a topologically labeled data with, $h : D \to \mathcal{D} \subset \mathbb{R}^n$ and $g : \mathcal{D}_L \subset \mathbb{R}^n \to \{l_1, \ldots l_d\}$. A softmax classification neural network $Net : \mathbb{R}^n \to Int(\Delta_{d-1})$ separates (D, h, g) if and only if $Net(\mathcal{D}_{i+1}) \subset Int(VC(v_i))$ for $0 \leq i \leq d-1$.*

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 5.3 with the universality of neural networks [1, 4, 7].[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 [1, 7] for more details.

7 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 7.1 shows 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. **This shows that the neural networks architecture must be in general a function of the underlying training dataset.**

Theorem 7.1. *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 $Net(\mathcal{D}_1)$, $Net(\mathcal{D}_2)$.

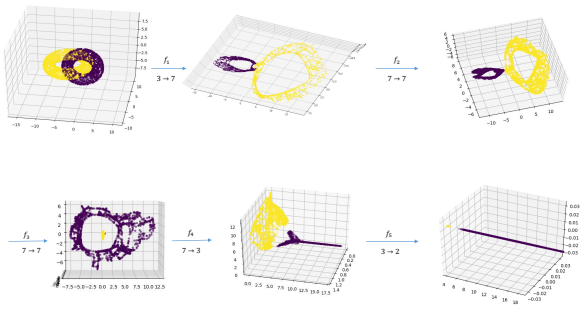

Figure 2: Unlinking the data using a neural net.

Note that in Theorem 7.1, the statement is independent of the depth of the neural network. This is also related to the work [5] which shows that skinny neural networks are not universal approximators. This is also related to the work in [9] where is was shown that a network has to be wide enough in order to successfully classify the input data. In fact, Theorem 7.1 can be extended to the case when $k = n$. Consider the dataset given in left upper corner of Fig 2. We argue that this data cannot be unlinked if every layer of a neural networks that acts on this data is chosen to go from $\mathbb{R}^3$ to $\mathbb{R}^3$. To see this first observe that any linear matrix W in a given layer, if it is not a projection, can be realized as an isotopy and possibly a reflection. Both of these continuous functions preserve link type (i.e. maintains the linking topological property). Furthermore the RELU non-linearity acts as a deformation retract which also does not change the link type. Hence the only way to untangle this dataset is to go to higher dimensions. Similar knotting and linking phenomenon can be constructed in higher dimensional spaces. This shows that a neural networks architecture must not be chosen independently without considering the nature of data.

On the other hand, the same dataset can be unlinked easily if we send the dataset to a higher dimension space. Consider for instance the network $Net : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$ defined by the composition $Net = f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1$ where $f_1 : \mathbb{R}^3 \longrightarrow \mathbb{R}^7$, $f_2 : \mathbb{R}^7 \longrightarrow \mathbb{R}^7$, $f_3 : \mathbb{R}^7 \longrightarrow \mathbb{R}^7$, $f_4 : \mathbb{R}^7 \longrightarrow \mathbb{R}^3$ and $f_5 : \mathbb{R}^3 \longrightarrow \mathbb{R}^2$. The activations are described in Figure 2.

Note that this network unlinks the input two tori provided we send this data to a higher dimension and allow enough steps (in terms of number of layers) for this unlinking to occur. Moreover, to visualize the higher dimensional activations in Figure 2 we project them to $\mathbb{R}^3$ using Isomap [11].

8 Acknowledgement

Mustafa Hajij was supported in part by the National Science Foundation (NSF, DMS-2134231).

References

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

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

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

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

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

[6] 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.

[7] 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.

[8] G. Naitzat, A. Zhitnikov, and L.-H. Lim. Topology of deep neural networks. *arXiv preprint arXiv:2004.06093*, 2020.

[9] 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.

[10] C. Olah. Neural networks, manifolds, and topology. *Blog post*, 2014.

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

[12] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[13] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.