NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Topological Expressiveness of Neural Networks

Topology of Learning

António Leitão

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Information Management School
Universidade Nova de Lisboa

Topological Expressiveness Of Neural Networks

António Leitão

Dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science in Advanced Analytics.

February, 2021

This work was done under the supervision of:

Giovanni Petri

Senior Research Scientist

ISI Foundation, Turin, Italy

giovanni.petri@isi.it

Flávio Pinheiro

Assistant Professor

NOVA IMS, Lisbon, Portugal

fpinheiro@novaims.unl.pt

Peer-reviewed publications resulting from this work:

Giovanni Petri and **António Leitão**. "On the Topological Expressive Power of Neural Networks" In: *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond.* 2020 https://openreview.net/forum?id=I44kJPuvqPD

António Leitão and Giovanni Petri, "On the Topological Complexity of Decision Boundaries", *50th Scientific Meeting of the Italian Statistical Society*, 2020, pp. 588-594

Acknowledgements

First and foremost to my family, in particular my brother Zé and my wonderful girlfriend Giulia. All have enthusiastically supported me even though their collective insight about my academic life is a vague memory of my high-school graduation. Except you dad, of course, because tuitions are a thing.

To Giovanni Petri in particular, for teaching me basically everything that is in here. For patiently putting up with my strange ideas: the many that failed and the some that almost didn't. I believe that this work is as much his as it is mine. I thank not just him but also his whole Topological Data Analysis group at ISI Foundation in Turin: Jacob, André, Alan, Chiara, Paolo and Rossano. Who were very welcoming during my period there, however brief.

Lastly I would like to thank Flávio Pinheiro for his supportive and insightfull comments as well as Nuno Alpalhão and Daniel Alcântara for their useless ones.

All illustrations where done by the author.

Abstract

Given a neural network, how many different problems can it solve? This important and open question in deep learning is usually referred to as the problem of the expressive power of a neural network. Previous research has tackled this issue through statistical and geometrical methods. This work proposes a new method based on a topological perspective.

Topology is the field of mathematics aimed at describing spaces and functions through robust characterizing features. Topological Data Analysis is the young field developed to extract topological insight from data.

We first show that topological features of the decision boundary describe the closest notion of the intrinsic complexity of a classification problem. These topological features divide classification problems into several equivalence classes. Linear-separability is an example of such a class. We establish the topological expressive power of a network architecture as the number of different topological classes it is able to express.

Being a novel work in a young research field, most of the thesis is devoted to developing this perspective and creating the tools required. The main objective of this thesis is to tackle neural network's understanding in general and architecture selection in particular, through a novel approach.

Our results show that topological expressiveness has a complex correlation with many features in a neural network's architecture depending weakly on the total number of parameters. Some of our results recapitulate previous research on geometrical properties, while others are unique to this novel topological point of view, sometimes challenging previous research.

Keywords: Topological Data Analysis, Neural Networks, Machine Learning, Decision Boundary, Persistent Homology, Voronoi Diagram.

Resumo

Quantos problemas differentes consegue uma dada rede neuronal resolver? Esta pergunta aberta é central no ramo de aprendizagem profunda e conhecida como o poder expressivo de uma rede neuronal. Tentativas anteriores em resolver este problema fizeram-no usando métodos estatísticos ou geométricos. Este trabalho apresenta um novo método baseado numa prespectiva topologica.

Topologia é o ramo de matemática responsável por descrever espaços e transformações com base em caracteristicas fundamentais. *Topological Data Analysis* (Análise Topológica de Dados) é o recente ramo de investigação desenvolvido para extrair conhecimento Topológico de dados.

Começamos por mostrar que uma caraterização topológica da barreira de decisão é a noção mais próxima da complexidade de um problema de classificação. Estas caracteristicas topoólicas dividas os problemas de classificação em diversas classes de equivalência. O conjunto de problemas separaveis por uma reta são um exemplo de uma destas classes. Establecemos que a expressividade topológica de uma architectura neuronal é equivalente a quantas destas classes consegue resolver.

Dado que é um método novo num ramo de investigação recente, grande parte desta tese foca-se em desenvolver esta perspectiva e em criar as ferramentas necessárias para o seu estudo. O objectivo desta dissertação é, apartir de uma abordagem original, enfrentar a falta de compreensão de redes neuronais no geral e, em particular, informar a escolha de arquitecturas.

Os resultados obtidos mostram que a expressividade topológica tem correlações complexas com diversos elementos da arquitectura de uma rede, mostrando uma depêndencia ténue no número total de parametros. Alguns resultados seguem a mesma linha que a investigação gemétrica anterior, outros são únicos à perspectiva apresentada e complementando resultados anteriores.

Palavras-chave: Análise Topológica de Dados, Redes Neuronais, Aprendizagem Autónoma, Barreiras de Decisão, Homologia Persistente, Diagrama de Voronoi

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

I do things like get in a taxi and say, "The library, and step on it."

-David Foster Wallace (Infinite Jest)

1.1 An Oncoming Plateau

In August 31st 1955, leading information theorists John McCarthy, Marvin Minsky, Nathaniel, Rochester, and the father of information theory himself, Claude Shannon proposed the *Dartmouth Summer Research Project on Artificial Intelligence.* Their objective was "That every aspect of learning can be principled so precisely that a machine can be made to simulate it". In other words, to create Artificial Intelligence[1].

To say they have failed would be inaccurate. Having witnessed the explosive progress of machine learning at the time, such as the first chess playing program and the self-taught checkers program. (Schaeffer [92]) their optimism was not unjustified. Not much later, disappointment and cynicism led to funding costs and loss of interest in the A.I. field.

In the late 1970s renewed interest fuelled new funding and research in the area, prompting the creation the American Association of Artificial Intelligence. In 1984 the *hype* had spiraled out of control, and the funding cutbacks and the downfall in research led to the collapse of the A.I. industry in 1987 (Crevier [33]).

At the expense of being mediatic, data science has accepted several captivating terms that fuel misconceptions. Pseudo-scientific buzzwords such as *Big Data* and *Curse of Dimensionality* symbolize existing dilemmas yet are misleading because they place the blame on the data, when in fact these are symptoms of outdated methodologies.

Big Data conveys the idea that data is immense, when in fact it is our existing methods that cannot handle its complexity. *Curse of Dimensionality* misleads one into thinking that

[1] This was the first instance the term *Artificial Intelligence* was used.

high dimensional data behaves abnormally, when it should instead lead one to question: why care so much about dimensions after all?

The *hype* coming from past results has created a stubbornness in the methods of Data Analysis. Holding back to previous achievements, existing approaches are not being updated to keep up with the exponential increase in data complexity. The inevitable shortcomings of these same approaches have been blamed on the data (using terms previously discussed) making one believe that our ability for analytical insight is somehow asymptomatic. This inevitably leads to cynicism and loss of interest, so much that there is now talk of a third "A. I. winter".

This is disheartening since, historically, mathematics has always been ahead of the curve, solving problems years before their applications have been found (such as ellipses).

The restrictions that *standard* data analysis poses on data (such as the assumption of linearity) have not been relaxed from their early insights. Yet what we come to expect from these same models has increased in complexity by orders of magnitude e.g. predicting the market price of a house versus expecting autonomous driving. There are those who question if the mathematical formalisms of geometry and statistics are sufficient. It might happen that the *nature* of the data cannot be expressed as a summary of pairwise interactions, in this case geometry which is the study of distance functions, captures only *accidents* on the data. There are many arguments to believe in that insufficiency of purely geometrical methods:

1. **Quantitative characterization is insufficient.** The most recurrent problem with big amounts of data is that one does not know *what* to look for. Data has knowledge but the way it is presented assumes many forms. As such the first insight must be to extract a robust characterization of the data. For example Carlsson [20] presents the example of studying diabetes data. Before developing qualitative insights, the necessary first step is to understand that the disease has 2 very different types.

2. **Metrics are not theoretically justified.** A metric completely governs the behaviour of a whole space. It makes no distinction between local and global scales. Real world scenarios rarely display the same rules, specially when human interactions are involved. More often than not, in areas such as biology or economics, the global structure is defined by a juxtaposition of local behaviours (for example Adam Smith's *Invisible Hand*) and not necessarily by pairwise interactions.

3. **Coordinates are not natural.** Although data is *arranged* into vectors, it does not indicate that coordinates are always meaningful, or that these emerge naturally. The idea that coordinates carry intrinsic meaning is sometimes hindering since it is to be expected that fundamental properties resist coordinate changes. The superior performance of Convolutional Neural Networks (CNNs) can be attributed to this. Before CNNs a 28×28 pixel image was handled as a vector of 756 dimensions.

4. **Intrinsic properties should not depend on extrinsic factors.** In geometry, a sphere does not exist in of itself, only lying in an Euclidean space. Geometry considers objects

always within a bigger space and never as a space of their own. This *extrinsic* view, while more intuitive, is not natural for the object in question, since its *intrinsic* properties are defined depending on the surrounding space that it is laying on.

1.2 The Advent of Topology

Topology is a recent field in Mathematics, going back less than 300 years. It is the study of properties of spaces that are invariant under continuous transformations. Topological methods aim to describe spaces and functions through characterizing features without the concept of distance. Therefore, spaces with very different geometries can be considered equivalent. Topology surfaced from the necessity to generalize rigorously structural notions such as continuity, compactness and connectedness

Topological Data Analysis (TDA) is the recent approach to Data Science through the methods and perspective of Topology. The reasoning is that topological methods directly confront the previously exposed handicaps of geometrical methodologies. Below is a brief reasoning why TDA is appropriate to solve each of the points presented above:

1. **Characterization.** Topology was built[2] precisely to study qualitative features of spaces and transformations. Topological Data Analysis extends these methods to points clouds as to extract these very same qualitative properties. Since Topology is a theoretically sound and well structured field, its translation to data analysis generally provides reliable results.

2. **Absence of Metric.** Topology conveys a notion of *nearness* through nested subsets and not through pairwise functions. As such manages to describe robust global identities (shape) through local concepts. This insensitivity to the metric grants topological features an invariance to continuous transformations.

3. **Coordinate free.** Topology only studies properties that do not depend on chosen coordinates. This is due to the fact that Topology considers objects as spaces on their own. For this point of view it is obvious that coordinates can be forfeited in most cases.

4. **Intrinsic point-of-view.** With the advent of topology (and Differential Geometry) geometric objects started being considered as spaces of their own. The previously *extrinsic* point of view only acknowledged the existence of such spaces when embedded in bigger ones, where one could "move outside"the space. While the *extrinsic* view is more visually intuitive, the *intrinsic* is more flexible and descriptive albeit less intuitive.

Topological Data Analysis has increasingly gained interest in research and traction in its applications. From dimensionality reduction (McInnes et al. [67]) to signal analysis (Perea and Harer [84]), from graph reconstruction (Cerri et al. [24], Biasotti et al. [12]) to computer vision. Among many other, it has found widespread success notably on complex network

[2] some argue the correct verb is *discovered*, author included.

analysis and neuroscience (Petri et al. [85], Huang and Ribeiro [52]), and contagion network analysis (Taylor et al. [100]). The reader is pointed to Carlsson [20] and Chazal and Michel [26] for a comprehensive introduction on the perspective and accomplishments of Topological Data Analysis.

1.3 This Contribution

1.3.1 Statement of Originality

The objective of this thesis is to apply the ideas from the new field of Topological Data Analysis to tackle one of the biggest open problems in machine learning: **"What is the expressive power of a Neural Network?"**

To understand how many different problems a Neural Network can solve, one must first ask what makes two problems different? In this thesis we show that Topology is a natural answer to both questions posed above. We partition classification problems into classes based on their topological features. Two binary classification problems belonging in the same class are by no means necessarily equal but **topologically equivalent**. Linear separability is an example of an equivalence class of problems. The Topological Expressiveness of a Neural Network can be understood as the number of these classes it is able to solve.

A considerable body of work has been poured into understanding the expressive power of Neural Networks through inherently **geometrical** approaches (Poole et al. [86]). This work is the first to do it at a Topological level. Besides presenting a novel topological approach, it also gathers, formalizes and translates other geometrical approaches, for instance Olah [80] and Brahma et al. [15], to topological language. On top of this, some corrections are made to other previously drafted topological perspectives on Neural Networks (Guss and Salakhutdinov [48], Naitzat et al. [78] and Ramamurthy et al. [87]).

When this work began, the idea of a topological understanding of Neural Networks was relatively uncharted ground, with notable exceptions such as Bianchini and Scarselli [11]. Although a couple of new directions have appeared since then (Guss and Salakhutdinov [48] and Naitzat et al. [78]), they are very few and far between. That is partly because Topological Data Analysis is a very young field but also due to the fact that the amount of research on the applications of Neural Networks largely outnumbers the work on their understanding.

Aside from Chapter 2 and 3 the whole body of work is original, unless clearly stated otherwise.

1.3.2 Outline of the Thesis

It is not expected from the reader to have any previous knowledge of TDA or Topology. This thesis, however, assumes the reader is familiar with some introductory level Linear Algebra and Real Calculus. The objective of this work is not to teach Applied Topology to the uninitiated but to show what Topology consists of and to illustrate a specific example of how it can be applied to the realm of Data Science.

Chapters 2 and 3 introduce and familiarize the reader with necessary concepts from Topology and how these are applied to Data Analysis. Both are based on several books, papers and lecture notes. The author would like to emphasize the impact of the following textbooks: Ghrist [45], Munkres [77] and Hatcher and Press [49].

Chapter 4 introduces classification problems and their decision boundaries along with a method to accurately sample it.

Chapter 5 introduces Neural Networks, develops a necessary tool for their understanding (UDiPH) and explores their Topological Expressiveness.

The proofs of statements are presented in Appendices A. The proofs regarding clearly elementary concepts were done by the author himself however the author takes no ownership because the proofs of elementary concepts are, more often than not, obvious or their sources untraceable. Of the remaining proofs some are also done by the author, typically for novel statements presented in this work. When that is not the case the source of the proof is clearly stated .

Chapter 2

Topology

One and one and one is three.

-The Beatles (Come together)

Topology is the field in mathematics that concerns itself with spaces and maps between spaces or, informally, shapes and deformations. Its broad definition of structure, through systems of open neighbourhoods (and their intersections), establishes Topological equivalence relations as fundamental properties of spaces. A **Topological invariant** is a qualitative aspect that is assigned to a space, with respect to a certain equivalence relation. This enables us to categorize the space, by comparing it and by distinguishing it from other classes. The number of connected components is an example of a topological invariant.

Topology is the sublimation of metric spaces. It is the generalization of metric spaces agnostic of a distance measure. Its foundations lie on a notion of *nearness* through open sets. This local concept of shape differences itself from Geometry, whose rigid shapes rely on the distance between each pair of objects.

Definition 2.0.1 (Topology)**.** A **topology** on a set X is a collection $\mathcal{T}$ of subsets of X satisfying the properties:

1. $\varnothing, X \in \mathcal{T}$
2. It is closed under unions.
3. It is closed under finite intersections.

To the elements of $\mathcal{T}$ we call **open sets**. A set is said to be **closed** if its complementary is open. Note that while, semantically, open and closed are antonyms their mathematical definition is not mutually exclusive. In a seemingly counter-intuitive way there can be sets that are both open **and** closed, for example both the complete space X and the null set $\varnothing$.

We call a space X endowed with the topology $\mathcal{T}$ a topological space, and denote it by (X,$\mathcal{T}$). Whenever the context is evident, we will simplify the notation to X.

2.1 Continuity

Continuity is a concept that is introductory in any math related subject. It is a fundamental transformation in any field of mathematics, and, as such, there are many different types of continuous functions. The topological definition of continuity is broad and general enough to cover all such different cases.

Definition 2.1.1. Let X and Y be topological spaces. A function $f : X \to Y$ is said to be **continuous** if for each open set $U \subseteq Y$ then $f^{-1}(U)$ is also an open set in X

To $f^{-1}(U)$ one normally refers to **preimage** of U. That is, the object that resulted in that image. A continuous function is then a function where the preimage of an open set is also open. Such nomenclature is often used and most times desirable as it removes any ambiguity that the symbol f^{-1} might bring in terms of existence of inverse or assumptions of bijectivity (which the definitions does not make). Let us now compare it with the familiar definition of continuity of real calculus.

Definition 2.1.2. A function $f : \mathbb{R} \to \mathbb{R}$ is continuous at point p if for all $\epsilon > 0$ there exists $\delta > 0$ such that:

$$\|p - x\| < \delta \implies \|f(p) - f(x)\| < \epsilon \tag{2.1}$$

We say that f is **continuous** if it is continuous in every point of its domain.

Such a definition should be familiar to the reader as it is fundamental to any introductory calculus course. It is generally regarded as the "$\epsilon - \delta$ definition", and it informally states that close points (δ apart) are mapped to close points (ϵ apart). Its also common to find in introductory courses to Topology the exercise of proving the equivalence between both definitions.

Theorem 2.1.3 (Munkres [77])**.** *Definition 2.1.1 and Definition 2.1.2 are equivalent.*

Definition 2.1.4 (Homeomorphism)**.** A function f is said to be an homeomorphism if and only if is a continuous bijection with a continuous inverse.

Let $f : X \to Y$ be a homeomorphism. Since both f and f^{-1} are continuous we have correspondence between not just the elements of X, Y (assured by bijectivity) but also a correspondence between collections of open sets, by the definition of continuity. This means that any property of X that can be expressed in terms of open sets yields the same property in Y (and vice-versa). Such properties are called **topological invariants**. In Topology, structures are studied *up to homeomorphism* because homeomorphisms are the the isomorphisms of Topology, that is they are the structure preserving maps in Topology, and describe an equivalence relation in the field.

2.2 Connectedness and Compactness

In mathematics the behaviours of non-finite sets can be very different and sometimes unintuitive compared to finite sets. There are many behaviours that hold for finite set but not for non-finite sets. For example, for finite sets, all functions are bounded, all functions attain a maximum and a minimum, while these properties are not true for non-finite sets. In a way, compactness is the topological generalization of "finiteness".

Consider a space X and a collection $\mathcal{U} = (U_i)_{i\in I}$ of open sets $U_i \in X$. We say that the collection $\mathcal{U}$ is a **cover** of X if X is equal to the union of all the sets, that is: $X = \bigcup U_i$.

Definition 2.2.1 (Subcover)**.** Let $\mathcal{U} = (U_i)_{i\in I}$ be an open cover of X. We say that $\mathcal{V}$ is a subcover of $\mathcal{U}$ if $\mathcal{V}$ is a cover of X and $\mathcal{V} \subseteq \mathcal{U}$

Definition 2.2.2. A space X is said to be **compact** if every open cover of X has a finite subcover.

Note that to prove that a space is compact its necessary to find a finite subcover for any possible cover. While it is only necessary to find a cover that accepts no finite subcover in order to show a certain space is not compact.

> Example 1. The real line $\mathbb{R}$ is not compact since it is not possible to have a finite subcover from the cover $\mathcal{U} = \{(n-1, n+1) \mid n \in \mathbb{Z}\}$.
>
> Example 2. The set $\{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ is compact, however $\{x \in \mathbb{R} \mid 0 < x < 1\}$ is not since there is no finite subcover from the cover $\mathcal{U} = \{(\frac{1}{n}, 1 - \frac{1}{n}) \mid n \in \mathbb{N}\}$.

Theorem 2.2.3. *The image of a compact space under a continuous transformation is compact.*

Definition 2.2.4. A space X is **disconnected** if there exists open sets U, V whose disjoint union is X. If no such pair of sets exist then X is said to be **connected**

> Example 3. The set $\{x \in \mathbb{R} \mid 1 \leq |x| \leq 2\}$ is disconnected while the set $\{x \in \mathbb{R} \mid 0 \leq |x| \leq 1\}$ is connected.

The notion of connectedness naturally creates a partition of the space X into non-empty sets with disjoint unions. These are what we call connected components of X. In a connected space there exists only one connected component: the space X itself.

Definition 2.2.5. Consider the equivalence relation defined on X as such; x is in relation with y if there is a connected subspace of X containing both x and y. The equivalence classes are called the **connected components** of X.

Theorem 2.2.6. *The image of a connected space under a continuous map is connected.*

Theorem 2.2.7. *Let X, Y be topological spaces and $X = \bigsqcup X_i$ and $Y = \bigsqcup Y_i$ its connected components, if $f : X \to Y$ is a homeomorphism then for all i there exists j such that $f(X_i) = Y_j$*

2.3 Dimension

The geometric concept of Dimension is the minimal number of variables necessary to describe each point of a space. However take for example the plane $\mathbb{R}^2$ and the real line $\mathbb{R}$, since both have the same cardinality it is possible to describe each point in the plane with a correspondent point in the number line.

Then what is the dimension of $\mathbb{R}^2$? It is possible to repeat such a process with any two sets of the same cardinality. Meaning every point in $\mathbb{R}^n$ can be described by an element in $\mathbb{R}$ for **any** n.

Space filling curves, such as the Hilbert curve, are an example of this process. A n-dimensional Hilbert curve defines a map between n-dimensional space and one-dimensional space. Thus one can understand there is an underlying concept in dimension that is not captured by the geometric perspective alone.

The Topological definition of dimension can be motivated by understanding where Hilbert Curves fail. Every space filling curve hits some points multiple times. The multiplicity of these overlaps directly depends on the dimension of the space being mapped. This idea is the basis of the definition of dimension.

Definition 2.3.1 (Refinement)**.** Let $\mathcal{U}$ be an open cover of X. We say that $\mathcal{V}$ is a refinement of $\mathcal{U}$ if $\mathcal{V}$ is a cover of X and every set in $\mathcal{V}$ is contained in some set of $\mathcal{U}$

$$\forall V \in \mathcal{V} \quad \exists U \in \mathcal{U} \quad V \subseteq U \tag{2.2}$$

Note that there is a small difference between a subcover and a refinement. Every subcover is a refinement however the reciprocal is not true. Informally, a subcover "preserves the open sets" while a refinement can make them "smaller".

Corollary 2.3.2. If $\mathcal{V}$ is a subcover of $\mathcal{U}$ then $\mathcal{V}$ is a refinement of $\mathcal{U}$

Intuitively the dimension of a cover is the minimal number of overlaps reached through cover refinements (minus one). For example the covering dimension of a circle is 1 because given an arbitrary cover, it is not possible to find a refinement that **does not** contain a point belonging to 2 open sets.

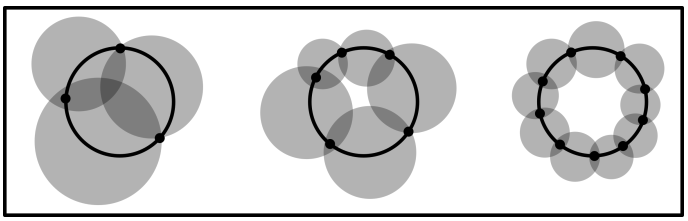


Figure 2.1: Illustration of cover refinements of a 1−dimensional space.

Definition 2.3.3 (Lebesgue's Covering Dimension)**.** The covering dimension of a space X is defined as the minimum number n such that any open cover $\mathcal{U}$ of X has a refinement $\mathcal{V}$ where any point belongs to at most $n+1$ open sets of the cover.

EXAMPLE 4. The covering dimension of $\mathbb{R}^n$ is equal to n.

Theorem 2.3.4. *Homeomorphic spaces have the same covering dimension.*

We turn back to the space filling curves and those overlaps that motivated the definition of Covering Dimension. They are in fact surjective however these same overlaps are the ones that prevent a space-filling curve to become a **bijection**. No space-filing curve is a bijection because that would make it a homeomorphism between spaces of different dimensions.

2.4 Manifolds

Manifolds are some of the most intuitive topological spaces. They serve as higher-order generalizations of familiar structures like curves and surfaces.

Definition 2.4.1. A n-manifold is a topological space[1] that is locally homeomorphic to $\mathbb{R}^n$

Being locally homeomorphic to $\mathbb{R}^n$ means that there exists a cover $\mathcal{U} = (U_i)_{i \in I}$ of M and a set of homeomorphisms $\phi_i : U_i \to V_i \subseteq \mathbb{R}^n$ where V_i is an open set. To each pair (U_i, ϕ_i) we call a **chart** and to the colletion of all charts we call an **atlas**. It is common to work with **smooth** manifolds in order to define derivatives. If such is the case we require that the following maps be smooth for all i, j.

$$\phi_i \circ \phi_j^{-1} : \phi_i(U_i \cap U_j) \to \phi_j(U_i \cap U_j) \tag{2.3}$$

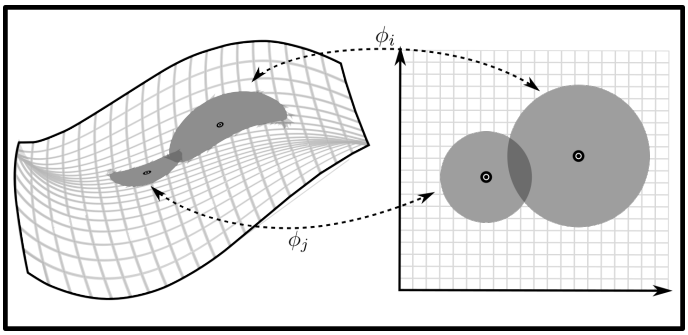


Figure 2.2: Illustration of a Manifold (left) along with homeomorphisms $\phi_{i,j}$ that map open sets (U_i, U_j) from the manifold to an corresponding subsets of the 2−dimensional Euclidean plane $\mathbb{R}^2$ (right).

EXAMPLE 5. A 1−manifold is called a curve and a 2−manifold is called a surface.

Theorem 2.4.2. *The dimension of an n-manifold is equal to n*

EXAMPLE 6. Consider the n-Sphere, denoted by $\mathbb{S}^n$, which is the set of all points in $\mathbb{R}^{n+1}$ that are at the same distance from the origin. $\mathbb{S}^1$ is a circle, $\mathbb{S}^2$ is the surface

[1] The space is assumed to be Hausdorff and second countable. The reader is encouraged to ignore these terms if unfamiliar.

of a sphere. The dimension of $\mathbb{S}^n$ is always n. This fact becomes intuitive when one considers the surface of the Earth. At local scales it *looks like* a flat surface. This relation is actually homeomorphic, meaning at a local level the surface actually *behaves* as the Euclidean plane, where the shortest distance between two point is a straight line. It is possible to map homeomorphically a portion of its surface into a plane, if it is sufficiently small. This is the intuition behind the chart. However its not possible to use that same projection in *every* small portion of the sphere, and definitely not for the whole sphere.

EXAMPLE 7. A very trivial example of a n-manifold is $\mathbb{R}^n$ itself, or any open set of $\mathbb{R}^n$.

Proposition 2.4.3. Homeomorphic manifolds have the same dimension.

2.4.1 Riemannian Manifolds

Definition 2.4.4 (Tangent Space)**.** Let M be a m-manifold embedded in $\mathbb{R}^n$. Consider $\gamma \in \mathcal{C}^\infty$ such that $\gamma :]-\epsilon, \epsilon[\to M$ is a path in M. Let $p \in M$ be the point where $\gamma(0) = p$. The tangent space of M at point p is the set:

$$T_p(M) = \{\gamma'(0)\} \tag{2.4}$$

The tangent space is the set of all vectors of $\mathbb{R}^n$ at p that are the derivatives of any curve passing through p.

It may be hard to understand what does the derivative of $\gamma(0)$ represent when M is not embedded. Let (U, ϕ) be a chart at point p. Meaning U is an open set with $p \in U$ and $\phi : U \to V \subseteq \mathbb{R}^n$ a homeomorphism. Then $\phi \circ \gamma$ defines a curve in $\mathbb{R}^n$ and $(\phi \circ \gamma)'(0)$ is its tangent space.

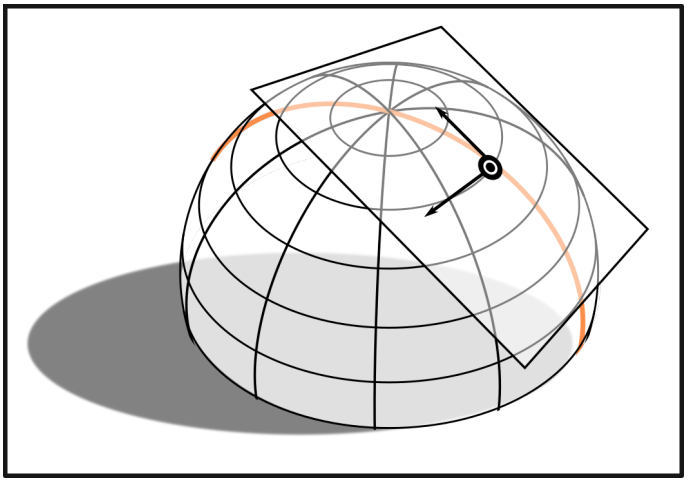


Figure 2.3: Illustration of a Tangent Space

Theorem 2.4.5. *The tangent space $T_p(M)$ of an $n-$manifold M is a $n-$dimensional vector space.*

Within this vector space it is possible to construct an inner product and define a metric g. The construction of these structures are beyond the scope of this thesis (and would require an entire devoted chapter). However the reader is pointed to O'Neill [81] and Carmo [22] for a good introduction in differential geometry, and to Crane et al. [32] for an introduction to its discrete applications.

Definition 2.4.6 (Riemannian Manifold)**.** A Riemannian Manifold is a a smooth n-manifold with a inner product g_p on the tangent space $T_p(M)$ for every p.

The interesting catch of Riemannian Manifolds is that this inner product defines a different metric for each point p in the Manifold. We will use this construction on Chapter 5.

Chapter 3

Topological Data Analysis

Topology! The stratosphere of human though! In the twenty-fourth century it might possibly be of use to someone..."

-Aleksandr Solzhenitsyn (First Circle)

Topological Data Analysis (TDA) is the branch that employs the perspective and tools of Topology into data science. However data is often presented as point clouds or images, never explicitly as topological spaces. For example, if one were to analyze the number of connected components of a point cloud with 100 points, it would come as a conclusion that there are 100 different connected components. As such, the fundamental step in Topological data analysis is to infer and reconstruct, from a discrete sample, the *data manifold*; a continuous space from where the data has been sampled and from where we can extract topological insights.

3.1 Manifold Hypothesis

Inferring a governing mathematical structure from a discrete set of observations is a common approach in data analysis, for example, statistics aims at approximating a distribution from a set of observed samples. It is then no surprise to see the same approach being applied in Topological Data Analysis, summarized under the following assumptions: **1. Data has shape**; we assume that the data, represented as a point cloud, has been sampled from a manifold. **2. Shape matters**; meaning the topological properties of said manifold carry relevant (if not fundamental) information.

These assumptions have been condensed and popularly regarded as *Manifold Hypothesis*.

Proposition 3.1.1 (Manifold Hypothesis)**.** Real-world data lie on low-dimensional manifolds embedded within a high-dimensional space.

There are many theoretical (Carlsson et al. [21] and Lui et al. [65]) and empirical (Carlsson et al. [21], Fefferman et al. [41], Lui et al. [65], and Olah [80]) indicators that the Manifold Hypothesis is reasonable. For example, Carlsson et al. [21] showed that the space of natural images is a Klein Bottle, and Fefferman et al. [41] tested the theory to be acceptable up to a given degree of certainty. It is also the basis for many non-linear dimensionality reduction algorithms that first reconstruct the manifold, and only then embed it into lower dimensional spaces such as Locally Linear Embedding (Saul and Roweis [91]), IsoMap (Tenenbaum et al. [101]) and more recently UMAP (McInnes et al. [67]).

The collection of methodologies that aim at reconstructing this lower dimensional manifold from data are called **Manifold Learning** algorithms. Such algorithms attempt to fit data sampled from high dimensions into low dimensional manifolds. An intuitive example would be linear regression where one tries to fit a line (1 dimensional manifold) onto a collection of 2-dimensional data points. Manifold Learning however is more generally used in the case of non-linear algorithms, though the principle is the same.

This first step of manifold learning is essential to TDA. An inaccurate reconstruction (as we will see) might drown any relevant topological features or fail to capture them altogether. Regardless of the approach, manifold reconstruction is generally done through the use of **simplicial complexes**.

3.2 Simplicial Complexes

We now explore the task of creating a continuous topological structure from a discrete set of points. Take two points $x, y \in \mathbb{R}^n$. Each one of them is what we call a 0-dimensional simplex. One can create a continuous object by considering all the points between them:

$$\sigma = t_0 x + t_1 y \quad \text{such that} \quad t_0 + t_1 = 1 \tag{3.1}$$

We have created now a continuous interval by adding the line between x, y, which in turn is called a 1-dimensional simplex. Similarly one can create a 2-dimensional simplex with three points (x, y, z) and considering the convex set created by them:

$$\sigma = t_0 x + t_1 y + t_2 z \quad \text{such that} \quad t_0 + t_1 + t_2 = 1 \tag{3.2}$$

Note that each n-dimensional simplex includes also all other possible simplices of lower dimension.

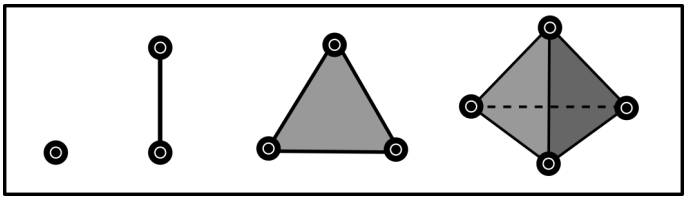


Figure 3.1: Illustration of simplicies of dimension 0,1,2 and 3.

Definition 3.2.1. Let S be an affine subspace of dimension n. A is said to be **convex** if $t_0 x + (1 - t_0)y \in A \quad \forall x, y \in S$ and $0 \leq t_0 \leq 1$. The smallest convex set that contains A is called the **convex hull**.

Definition 3.2.2. We call *n-simplex* the convex hull in $\mathbb{R}^m$ of $n+1$ points $v_0, \ldots, v_n$ in general position[1].

The points $v_0, \ldots, v_n$ are called the **vertices** and the simplex is given by:

$$\Delta^n = \{(t_0, \ldots, t_n) \in \mathbb{R}^{n+1} \mid \textstyle\sum_i t_i = 1 \text{ and } t_i \geq 0\} \tag{3.3}$$

A **simplicial complex** is a structure built by "glueing together" a collection of simplices of different dimensions.

Definition 3.2.3 (Simplicial Complex)**.** A simplicial complex, denoted as Δ-complex, is a collection of n-simplicies closed under the restriction: for each simplex $\sigma \in \Delta$, all its subsets are also in Δ.

One can think of Δ-complexes (simplicial complexes) as higher order generalizations of graphs, or better yet, graphs as Δ-complexes containing only simplices of dimension 0 (vertices) and 1 (edges). When creating a graph the same restriction applies: for an edge (1-simplex) to exist it requires two vertices (0-simplex).

Given a set of elements of a space it is possible to construct many different simplicial complexes (just like graphs). The main approach is to take a finite cover of the space and create higher order simplices depending on the intersections of that cover, i.e. add a 1-dimensional simplex whenever the intersection of two sets if not null. The different possibilities on how to do this process give rise to different famillies of simplicial complexes: alpha (Edelsbrunner et al. [38]), tangential (Boissonnat and Ghosh [14]), witness (Silva and Carlsson [94]) and cover complexes such as Mapper (Singh et al. [95]), and the most commonly used Čech Complex and Vietoris-Rips complex (Vietoris [105]). In light of our later applications we focus on the last two.

3.2.1 Čech Complex

Given a cover $\mathcal{U} = (U_i)_{i \in I}$ we can construct a simplicial complex from $\mathcal{U}$ by adding a 1-dimensional simplex (a line) for each intersection of 2 sets, a 2-dimensional simplex (a face) for each intersection of 3 sets, etc. One can see that this is a simplicial complex because every time we have a non-null intersection of three sets (a 2-simplex) it means that the pairwise intersections of these sets are also non empty (all its 1-simplexes). Informally, this processes can be described as adding a vertex for each set in the cover and then drawing a line if two sets touch, a face if 3 sets touch etc.

The simplicial complex resulting from this construction is called the **nerve** of a cover.

[1] General position is when $n+1$ points do not lie in a hyperplane of dimension less than n. Consider for example 3 colinear points, they are not in general position since they fit in a hyperplane of dimension 2, a line.

Definition 3.2.4 (Nerve)**.** Let $\mathcal{U} = (U_i)_{i \in I}$ be a finite collection of open sets. The simplex $\{t_1, \ldots, t_k\}$ belongs to the *nerve* of $\mathcal{U}$ if $U_{t_1} \cap, \ldots, U_{t_k} \neq \varnothing$.

The nerve of a cover is a simplicial complex generally called **Čech Complex**. The Čech Complex is constructed **from** an existing cover of the space. In order to construct a Čech Complex from a set of points in $\mathbb{R}^n$ we must first build a cover. Let S be a finite set of points in $\mathbb{R}^n$. Let $B_x(r)$ be an open ball of radius r and center $x \in S$:

$$B_x(r) = \{y \in \mathbb{R}^n \mid d(x, y) < r\} \tag{3.4}$$

We now create one open ball centered at each point and with radius r. We then take the collection $\mathcal{B} = (B_x(r))_{x \in S}$. The nerve of this collection is the Čech Complex of radius r created from the sample S, denoted by Čech(r). Note the important property that for $r_0 \leq r_1$ we have Čech$(r_0) \subseteq$ Čech(r_1).

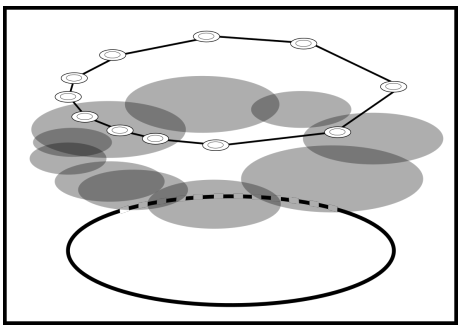


Figure 3.2: Illustration of a Čech Complex. The simplicial complex (top) is built from a cover of open sets (middle), each vertex is a set, each edge is an intersection. These open sets are $\varepsilon-$balls centered in points sampled from the assumed topological space (bottom)

The Čech Complex is central to Topological data analysis because it provides a strong theoretical guarantee because the topology of the nerve of a cover "is related" to the topology of the union of the sets. Informally this means that no topological features are lost or created when considering the cover and its nerve. Which means that the nerve of a "good enough" cover is, for all intents and purposes, a "very good" approximation of the space. Many theorems prove variants of such relations under different conditions, and they are collectively called **nerve theorems**. A good summary can be found in Meunier and Montejano [69].

3.2.2 Vietoris-Rips Complex

Informally in the Vietoris-Rips Complex, simplices are added if the pairwise distance between vertices is less than a certain threshold r. That is, the simplex $\{t_0, t_1\}$ exists if and only if $d(t_o, t_1) \leq 2r$. Higher order simplices are added if all its subsimplices were added too. For example a 2-simplex (triangle) is added to the complex if all its edges also belong to the complex, which in turn belong if the distance between each vertex pair is less than $2r$.

Let (X, d) be a metric space. Consider a sample $S = t_1, ...t_n$ from X, the Vietoris-Rips Complex of radius d is defined as the set:

$$VR_r = \{\langle t_0, \ldots, t_k \rangle \mid d(t_i, t_j) \leq 2r \quad \forall i, j\} \tag{3.5}$$

The difference between both complexes is subtle. Vietoris-Rips complexes require only a metric space, i.e. the pairwise distance between the elements. This makes Vietoris-Rips complexes depend only on the geometry of X and not on an underlying geometry where X maybe be embedded. Consider three points in a plane at the same distance $2d$ from each other. VR_d would contain the face of the triangle as a 2-simplex while Čech(d) would not since there is no point present in all three balls. However if the same points where embedded in a different space where the intersection of the balls was not null, then the Čech complex would include the 2-simplex while the Vietoris-Rips would remain the same.

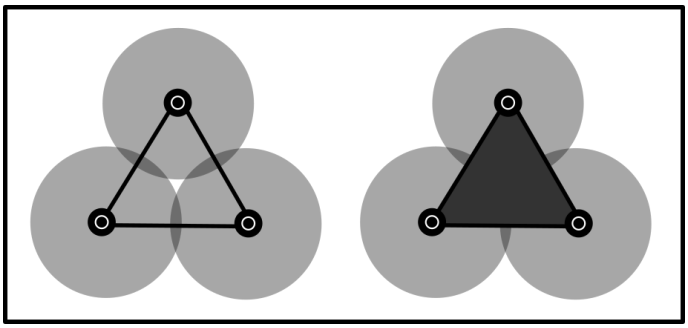


Figure 3.3: Illustration of a Čech Complex (left) and a Vietoris-Rips complex (right) built from the same cover. Note that Vietoris-Rips complex includes the 2−simplex while the Čech doesn't since the intersection of all three sets is null.

In contrast with the Čech complex, the Vietoris-Rips is not the nerve of a cover. Therefore **does not** necessarily inherent the topology of the cover. However since Čech$(r) \subseteq VR_r$ some properties of the nerve theorems still apply for Vietoris-Rips complexes (Latschev [59] and Hausmann [50]).

Since checking for high order intersections is very computationally expensive, Vietoris-Rips complexes are much preferred over Čech Complexes, since the former only require pairwise distances.

So given a dataset S as a set of points in $\mathbb{R}^n$, what are then the topological properties that we can extract from this simplicial complex? How does one proceed after reconstructing the underlying "data manifold"?

3.3 Homology

Informally Homology counts the number of holes of a structure. It defines an equivalence relation in topology. Meaning that it is an inherent feature of a space robust enough to characterize it under continuous deformations. This equivalence relation emerges from the

observation that it is possible to distinguish shapes based on the number of holes that they have in each dimension.

3.3.1 Motivation

Consider the symbols **O**, **Q**, **C**, and **IB**. Let us now imagine we want to categorize these symbols by the number of holes they have in different dimensions.

Denote by β_0 the number of connected components (0-dimensional holes) of each space. in this way we have: $\beta_0(\mathbf{O}) = \beta_0(\mathbf{Q}) = \beta_0(\mathbf{C}) = 1$ and $\beta_0(\mathbf{IB}) = 2$. In a similar fashion let β_1 define the number of 1-dimensional holes, as such: $\beta_1(\mathbf{C}) = 0$, $\beta_1(\mathbf{O}) = \beta_1(\mathbf{Q}) = 1$, $\beta_1(\mathbf{IB}) = 2$.

If we define an equivalence relation based on the value of (β_0, β_1) we observe that **O** and **Q** are equivalent since they have the same characterisation $(\beta_0, \beta_1) = (1, 1)$. We shall say (without definition for now) that these shapes are homology equivalent.

Notice that no matter how much we deform these symbols, as long as it is done in a homeomorphic way, the number of holes does not change. This is why we are able to recognize letters even in the most unconventional fonts. Homology is a topological invariant and as such is subjugate to the homeomorphism class. Meaning that, in general terms, X homeomorphic to $Y \implies X$ homology equivalent to Y.

> EXAMPLE 8. The reciprocal is, however, not true. Take for example the symbols **O** and **Q** although they have the same homology groups they are not homeomorphic. The intuition behind this is that **Q** has a point for which there is no local homeomorphism to $\mathbb{R}$. Informally it has a point that any scale, no matter how small, doesn't look like a line.

Given a simplicial complex, which represents our approximation to the data manifold, how does one compute its homology?

3.3.2 Simplicial Homology

Simplicial homology counts how many closed submanifolds of each dimension are not a boundary of a higher dimensional one. Given that two submanifolds are different if they are not boundaries of the same higher-dimensional submanifold. This defines an equivalence relation where the representatives of each class are the closed submanifolds that are a boundary of a *non-existing* manifold: a hole.

In practice, we first separate the simplicial complex into groups of simplices of the same dimension. Then define a map between the group of the n-simplices to the one of the $n-1$-simplices. Homology is an equivalence relation defined by this map.

Definition 3.3.1. Let $\mathbb{F}^2$ be the finite field with 2 elements. We denote by $C_n(\Delta)$ the vector space over $\mathbb{F}^2$ with basis the $n-$simplices of Δ

We motivate the construction of a map ∂ called the **boundary map**. Consider a $2-$simplex (t_0, t_1, t_2), geometrically it can be represented by a (filled) triangle. The boundary map applied

to this simplex outputs the boundaries of this simplex $\{(t_0, t_1), (t_1, t_2), (t_0, t_2)\}$. These are 1-simplices that geometrically represent the lines that compose the triangle.

Definition 3.3.2. Let $\partial_{n+1} : C_{n+1}(\Delta) \to C_n(\Delta)$ be a linear transformation that sends a $n+1$-simplex to a sum of basis n-simplex faces

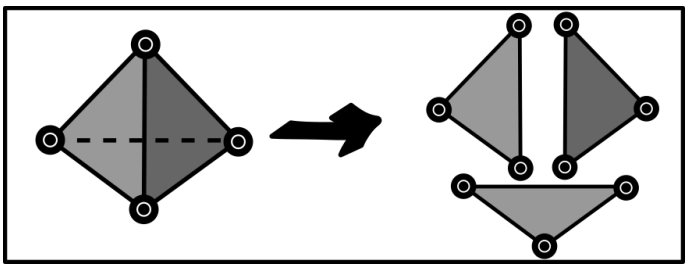


Figure 3.4: Illustration of boundary operator ∂_3 that maps a 3-dimensional simplex to its 2−dimensional boundary.

We define the boundary of zero dimensional simplices to be equal to 0. Note that there are some important elements of $C_n(\Delta)$ whose boundaries is zero. That is let z_n be an n−chain of Δ, z_n is called a n-**cycle** if $\partial_n(z_n) = 0$.

By definition the set of all cycles of $C_n(\Delta)$ are the elements of the kernel of ∂_n. We now consider the construction:

$$\dots C_{n+1}(\Delta) \xrightarrow{\partial_{n+1}} C_n(\Delta) \xrightarrow{\partial_n} \dots C_1(\Delta) \xrightarrow{\partial_1} C_0(\Delta) \to 0 \tag{3.6}$$

Lemma 3.3.3. $\partial^2 = \partial_n \circ \partial_{n+1} = 0 \quad \forall n \in \{0, 1 \dots\}$

The lemma states that the boundary of a boundary is null, equivalently that every boundary is a cycle. The following corollary shows that not every cycle is a boundary.

Corollary 3.3.4. $img(\partial_{n+1}) \subseteq ker(\partial_n) \quad \forall n \in \{0, 1 \dots\}$

Each element of $img(\partial_{n+1})$ is called an n-boundary. The corollary above shows that not all cycles are boundaries of a higher dimensional manifold. Recall the beginning of this section, we can now create a quotient group of all the cycles that are not boundaries of a higher dimensional cycle. **Homology** is that equivalence relation of cycles.

Definition 3.3.5 (Homology)**.** $H_n(\Delta) = \dfrac{ker(\partial_n)}{img(\partial_{n+1})}$

Two n-cycles (elements of $ker(\partial_n)$) are **homologous** if they differ by a an element of $img(\partial_{n+1})$. And thus the dimension of this equivalence class will give us the number of cycles that are not boundaries: the number of holes. This is called **Betti number**.

Definition 3.3.6 (Betti number)**.** $\beta_n = dim(H_n(\Delta)) = dim(ker(\partial_n)) - dim(img(\partial_{n+1}))$

EXAMPLE 9. Consider a circle, it has only a 1-dimensional hole, i.e $dim(H_1(\mathbb{S}^1)) = 1$. Similarly a sphere has only a 2-dimensional hole. In fact, regarding the n-sphere $\mathbb{S}^n$ for $n > 0$ we have:

$$H_k(\mathbb{S}^n) = \begin{cases} \mathbb{Z} & k = n \\ \varnothing & \text{otherwise} \end{cases} \qquad \beta_k(\mathbb{S}^n) = \begin{cases} 1 & k = n \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

EXAMPLE 10. Regarding the n-Torus $\mathbb{T}^n = (\mathbb{S}^1)^n$ we have:

$$\beta_k(\mathbb{T}^n) = \binom{n}{k} \tag{3.8}$$

Given then a simplicial complex we have now the tools to compute its homology in any dimension. In the previous section we explained how to construct a simplicial complex from a discrete set of samples. However the attentive reader will realize that there are many different simplicial complexes that can be created from a discrete set of samples (Fig. 3.5). For example both Vietoris-Rips and Čech Complexes are dependent on a radius parameter. Both generate different complexes for different radii and these complexes might have very different homologies.

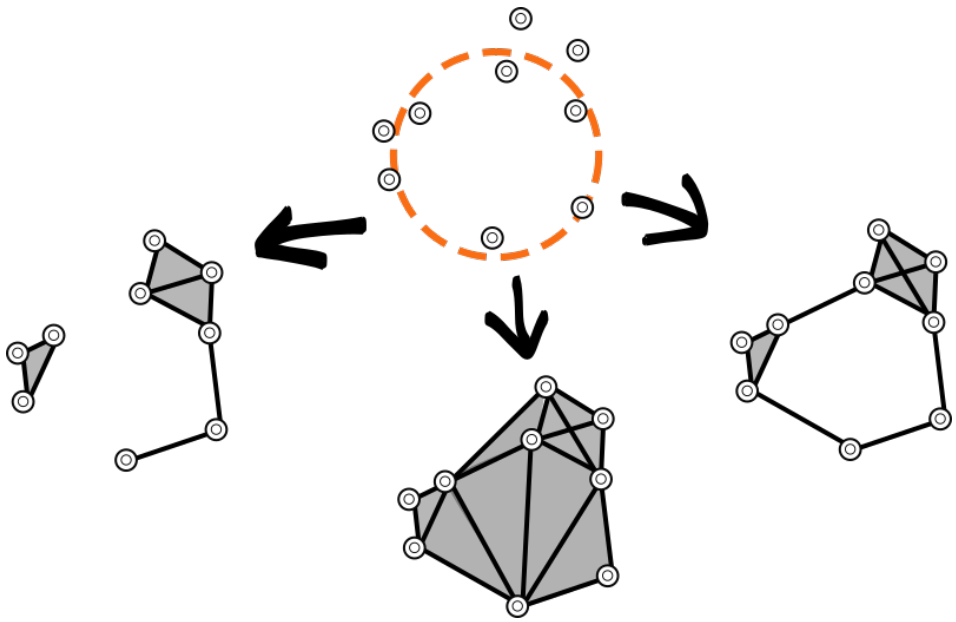


Figure 3.5: Simplicial Complexes constructed from a a discrete set of observations from a underlying data manifold (orange). Note that all constructions are valid however only one captures the correct homology of the data manifold.

3.4 Persistent Homology

3.4.1 Motivation

Both Čech and Vietoris-Rips complexes depend on a parameter (radius) for construction. Different parameter values generate simplicial complexes of varying homologies. How can one infer the homology of the data manifold? Persistent homology is the main tool in TDA that solves this problem. The workflow starts by constructing simplicial complexes for each value of the parameter, one then computes the homology of each complex and evaluate which homology classes persist for the most values of the parameter.

The intuition behind Persistent Homology is to assess the interval within which a homology class is relevant under the assumption that features that persist over a longer interval have more significance. Recall the observation in Section 3.2.1 that for different values of radius $r_0 \leq r_1$ we have $\check{\text{C}}\text{ech}(r_0) \subseteq \check{\text{C}}\text{ech}(r_1)$. If we were a consider a set of increasing values of radius $\{r_0, r_1, r_2, \ldots, r_{n-1}, r_n\}$ we would have the following sequence of nested simplicial complexes: $\check{\text{C}}\text{ech}(r_0) \subseteq \check{\text{C}}\text{ech}(r_1) \subseteq \check{\text{C}}\text{ech}(r_2) \subseteq \cdots \subseteq \check{\text{C}}\text{ech}(r_{n-1}) \subseteq \check{\text{C}}\text{ech}(r_n)$ Such sequence of simplicial complexes ordered by inclusion we call **filtration**. This is the basic setup of persistent homology.

3.4.2 Filtration

Definition 3.4.1. A filtered space (or filtration) is a sequence of subspaces $0 = X_0 \subseteq X_1 \subseteq \ldots X_k = X$ that begins with zero and ends with the space.

$$0 = X_0 \subseteq X_1 \subseteq \ldots X_k = X \quad \xrightarrow{\text{Filtration Parameter } \varepsilon} \tag{3.9}$$

If we consider X to be a simplicial complex, then a filtered simplicial space is the nested sequence of its subcomplexes. Let us consider a **filtration** based on a parameter ε, e.g. a sequence of Čech Complex of increasing radius $(\varepsilon_i)_{i \in \mathbb{N}}$. The inclusion $X_i \subseteq X_j$ defines a linear map $f_i, j : H_n(X_i) \to H_n(X_j)$ on the n–homology groups of both subcomplexes, for all n.

$$H_n(X_0) \xrightarrow{f_{0,1}} H_n(X_1) \xrightarrow{f_{1,2}} H_n(X_2) \longrightarrow \ldots \longrightarrow H_n(X_{k-1}) \xrightarrow{f_{k-1,k}} H_n(X_k) \tag{3.10}$$

A homology class in $H_n(X_i)$ is said to **persist** if its image is in $H_n(X_{i+1})$ otherwise it is said to **die**. Similarly, a homology class $H_n(X_i)$ is said to be **born** if it not in the image of $H_n(X_{i-1})$. To the value $\varepsilon_{death} - \varepsilon_{birth}$ we call the **persistence** of the homology calss $H_n(X_{i+1})$.

For the rest of the work we will consider only filtrations of Vietoris-Ripps simplicial complexes. A Vietoris-Rips filtrations is intuitive to visualize and understand. The filtration parameter is the radius. Increasing this filtration parameter can be seen as increasing the radius of each data point. At a certain radius ε_0 a hole, an element of H_1 appears. As we increase the radius more and more subcomplexes are added. Eventually, at radius value ε_1, drowning out the hole. The value $\varepsilon_1 - \varepsilon_0$ is persistence of the hole; which values of the radius

this H_1 element exists. The bigger the persistence, the more relevant the topological feature is. Small persistence values are normally associated with noise. In case of Vietoris-Rips filtration, where the parameter is the radius, the persistence is also a measure of the radius of the hole.

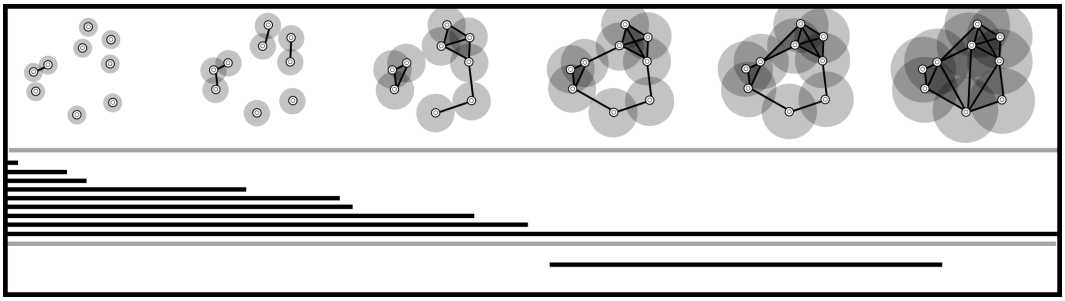


Figure 3.6: Vietoris-Rips filtration of a set of points along with the birth and death value of each homology class, represented as a bar below. The size of the bar, death-birth, is called the persistence. The upper bars represent the 0−dimensional homology classes (connected components), and the bottom represent the 1−dimension (holes).

3.4.3 Persistence Diagrams

We represent the births and deaths of n-dimensional homology classes by a multiset of points in $\mathbb{R}^2$, the n-th **persistence diagram** [37]. For each group we map $(x, y) = (\varepsilon_{birth}, \varepsilon_{death})$. Drawing births along the x axis and deaths on the y axis. The persistence of a given point is equal to its horizontal or vertical distance to the diagonal. Naturally there are no points below the diagonal.

As pointed by Mileyko et al. [70] the space of persistence diagrams is not complete. This means that it is not possible to develop basic statistical inference. To fix this we consider **generalized persistence diagrams** which is the union of the persistence diagram and a set of infinite number of points in the diagonal.

Definition 3.4.2 (Generalized Persistence Diagram)**.** A generalized persistence diagram is a multiset of points in $\mathbb{R}^2$ along with the diagonal $\{(x, y) \mid x = y\}$.

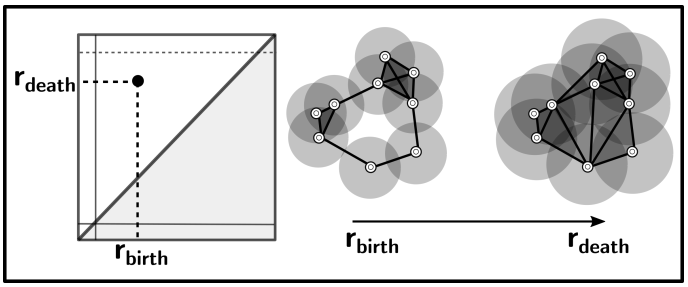


Figure 3.7: Illustration of the construction of a persistence diagram (left), r denotes the parameter of the simplicial filtration (right). In the case of a Vietoris-Rips filtration it denotes the radius of the cover.

For the rest of the work we will only consider generalized persistence diagrams and therefore drop the prefix. We can equip the space of the persistence diagrams with a metric called the **Wasserstein Distance**. The intuition behind this is to make a one-to-one matching between points of two different diagrams. Such a bijection exists due to the existence of points in the diagonal. The bijection η is chosen to be the one that minimizes the L_p distance of each matching. The $p-$Wasserstein distance is that value.

Definition 3.4.3 (Wasserstein distance)**.** The p-th order Wasserstein distance between two persistence diagrams d_1, d_2 is defined as

$$W_p(d_1, d_2) = \inf_{\gamma} \left(\sum_{x \in d_1} \|x - \gamma(x)\|^p \right)^{\frac{1}{p}} \tag{3.11}$$

Just like the L_p norms, there is a different Wasserstein distance for each $p \in (0, \infty]$. For $p = \infty$ we have what is commonly called the **bottleneck distance**. Intuitively, it is the shortest distance for which there is a perfect matching of the two diagrams.

$$W_\infty(d_1, d_2) = \inf_{\eta} \sup_{x} \|x - \eta(x)\| \tag{3.12}$$

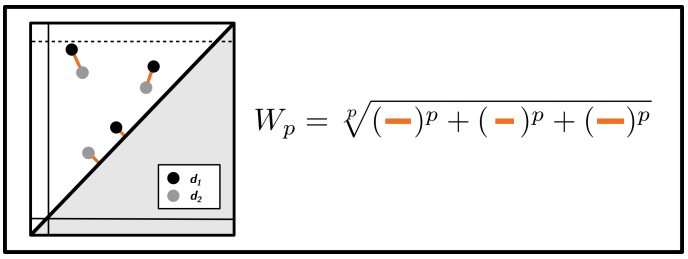


Figure 3.8: Illustration of $p-$Wasserstein distance between two persistence diagrams d_1, d_2. In orange we see the shortest possible pairing of points in d_1 with points in d_2 (the bijection η). Since we work with generalized persistence diagrams it is possible to pair also with the diagonal.

3.4.3.1 Stability

The most important corollary from the bottleneck distance, and a result also from Cohen-Steiner et al. [27], is its **stability**. Intuitively, it means that small variations in the data create small variations in the corresponding persistent diagrams.

Theorem 3.4.4. *Let f and g two functions and $D(f)$ and $D(g)$ their respective persistence diagrams. Then:*

$$W_\infty(D(f), D(g)) \leq \|f - g\|_\infty \tag{3.13}$$

The proof, quite extensive and off the scope of this project, can be found in Cohen-Steiner et al. [27]. Stability is among the most important result regarding persistence diagrams. While most results phrase stability in regards of $\infty-$norms, Skraba and Turner [96] recently provided new stability results for $p-$Wasserstein norms, especially for Vietori-Rips filtrations.

3.4.3.2 Persistence Diagrams in Machine Learning

The integration of topological summaries, such as persistence diagrams in machine learning is an active and ongoing research field. The major speed bump is that the space of persistence diagrams is not a Hilbert space (Bubenik and Wagner [19]). This means that its not possible to directly apply many of machine learning methods (such as PCA, since they require the existence of an inner product). Commonly used workflows consist in first mapping the diagrams into a Hilbert space. Several approaches include maps into finite vectors (Adams et al. [1], Di Fabio and Ferri [34], and Kališnik [55]) and functional summaries (Bubenik [18] and Rieck et al. [89]) and various others (Le and Yamada [60]). Obviously such map introduces deformation and potential losses of information (Carrière and Bauer [23]) and even for $p > 2$ a map from the space of persistence diagrams and the $p-$Wasserstein distance to a Hilbert space does not even exist (Wagner [106]).

Nevertheless its possible to define probability measures in the space of persistence diagrams (Mileyko et al. [70]), guaranteeing that statistics such as mean and expectations are well-defined. In this work the most complex structure we will consider is the **metric space** of persistence diagrams endowed with $p-$Wasserstein distance in Chapter 5 which is a trivial construction. By assuming a simple structure we do not project on the data behaviours that might be consequence of more complex embeddings. For example, it is common to consider data as vectors in an euclidean vector space. This is a comfortable embedding yet it assumes the existence of many structures (such as inner products, vector norms, dimensions) that might be misleading for example, the embedding dimension can be completely unrelated to the intrinsic dimension of the data.

3.5 Topological Data Analysis Pipeline

The common topological data analysis pipeline is as follows:

1. **Obtain raw data.** It can be images, time-series data, graphs and networks. Most commonly, it is presented as a point cloud, i.e a set of vectors in $\mathbb{R}^n$.

2. **Create filtration.** From the raw data by constructing a nested sequence of simplicial complexes. For point clouds the main filtrations include: Vietoris-Rips(Vietoris [105]), witness (Silva and Carlsson [94]), Cover (Singh et al. [95]), Chech, Tangential (Boissonnat and Ghosh [14]) and Alpha (Edelsbrunner et al. [38]). For images (or other cubical data) it is common to use cubical persistence introduced by Wagner et al. [107]. For graphs, Flag Complexes (Whitney [108]), also commonly called Clique Complexes (or uncommonly, *Whitney Complexes*) are normally used, these are equivalent to the Vietoris-Rips complex.

3. **Topological Summaries.** There are a multitude of possible topological summaries to take from a filtration of simplicial complexes. The most common are the Persistence Diagrams along with the Wasserstein Distance (or Bottleneck). However many others

exist including: Persistence Landscape (Bubenik [18]), Persistence Images (Adams et al. [1]), Persistence FlameletsPadellini and Brutti [83], Heat Vectorizations (Reininghaus et al. [88]), Persistence Entropy (Atienza et al. [4]) and Betti Curves. In Chapter 4 we introduce a novel one: **Topological Complexity**.

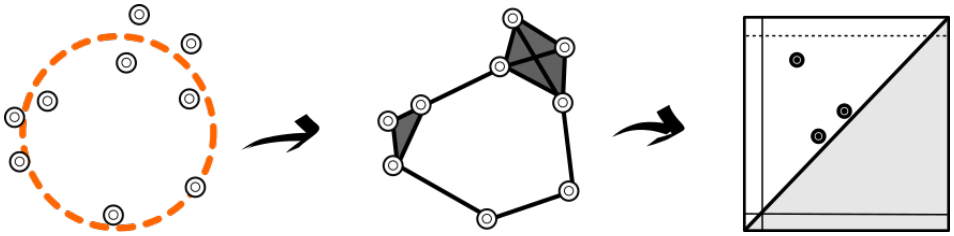


Figure 3.9: Common pipeline of topological data analysis. Consider that data comes from an underlying manifold (left), create a filtration of simplicial complexes (center). Compute persistent homology of filtration and other topological summaries such as the persistence diagram (right).

3.5.1 Computational Overview

The bulk of the computational manpower on this work has been developed in *Python*. There is an increasing number of libraries for topological data analysis available. These include (by order of importance to this work): Ripser (Bauer [9], Tralie et al. [103]), GUDHI (The GUDHI Project [102]) and Giotto (Tauzin et al. [99]). It is important to stress that of the large plethora of available resources, most of them are not included nor required in this project (because they were not used). The reader is pointed to the work of Otter et al. [82] for a computational overview and comparison of most of these projects. Although there is very little standardization in the field, several libraries have started to push TDA computation in that direction (such as GUDHI and Giotto).

Chapter 4

Classification and Topological Complexity

This algorithm has been proved to work, but has never been observed to do so.

-Alexander Barvinok (Illinois Mathematics Colloquium)

In order to understand how many different problems a given Neural Network can solve we first need to understand what it is that makes two problems distinct. This chapter aims to define the basis of a classification problem, how previous geometric approaches have tackled the above question and above all how can we approach them **topologically**. The objective is to establish a baseline and build momentum for understanding topological expressiveness.

4.1 Classification Problem

Let S be a sample of points of $\mathbb{R}^n$. Let S be composed by a disjoint union of sets $S = \bigsqcup C_i$. We call each class C_i. Without loss of generalization and unless explicitly stated, we will only consider binary classification problems. That is, situations where $S = A \sqcup B$.

A binary classification problem can generally be considered to finding a function $f : \mathbb{R}^n \to \{0, 1\}$ such that:

$$f(s) = \begin{cases} 1 & s \in A \\ 0 & s \in B \end{cases} \quad \forall s \in S \tag{4.1}$$

4.1.0.1 Characterizing Classification Problems

The complexity of a problem can range from very simple, such as when the two classes can be separated by a straight line, to very hard, when classes are entangled in non-obvious ways.

Analyzing the complexity of a given problem is the necessary first step in solving it. But how to do it?

A first approach by Ho [51] separates geometric descriptors of complexity into three categories: i) class ambiguity, ii) boundary complexity and iii) sample sparsity and dimensionality. The work has been extended in Lorena et al. [64] and Mollineda et al. [73] with geometrical and statistical descriptors of binary and multiclass problem complexity. In all these works, the authors recognized boundary complexity as the closest notion to the **intrinsic difficulty** of a classification problem. While the other categories are more symptoms of an ill-defined classification problem. However, authors did not agree on any reliable metric for measuring boundary complexity or even defining it. Note that all the approaches described so far are inherently geometrical.

Our approach starts by understanding what makes classification problems similar. While it is not obvious for all types of classification tasks, there is one natural subset of problems of equal complexity: **linearly separable** problems. Note that this is not a measure, but an entire **class** of problems, regardless of data complexity, coordinates and ambient dimension. Such mantra should come familiar to the reader by now because this is describing a topological feature of an intrinsic object in classification: the decision boundary.

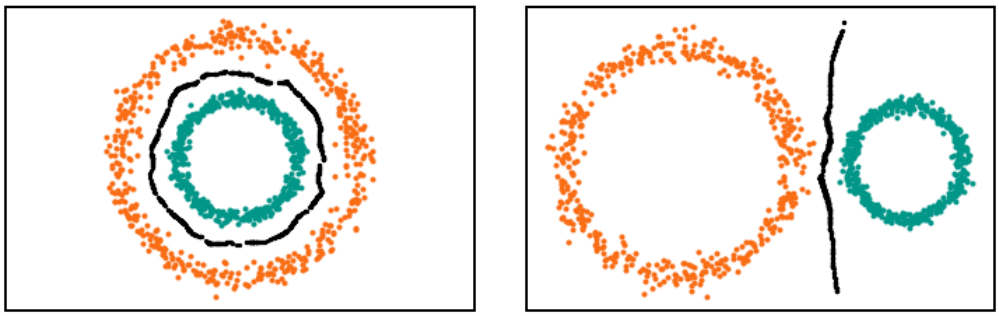
Linearly separable problems are problems whose **decision boundary has trivial homology**. Following the same idea there should be a class of problems that are *bi-linearly* separable, i.e. that can be solved (classes can be separated) using **two lines**. This class would already include the XOR problem. *Bi-linearly* separable problems are problems whose decision boundary has trivial H_1 homology (are lines) but non-trivial H_0 (there are two of them). We will use this to motivate **topological complexity** later.

Our approach is then to use Topology to capture this characterization of complexity that is inherent to its perspective. A topological approach to classification complexity has been previously proposed by Guss and Salakhutdinov [48]. The authors argue that a classification problem is complex if its classes display non-trivial homology. For example, classes with many connected components pose a more difficult challenge that those with just one. However, at difference with Guss and Salakhutdinov [48], we do not focus on the homology of each class as a measure of complexity, but rather on the **homology of the decision boundary**. This is due to the fact that the classes' homology can sometimes be misleading. For example, topologically complex classes might be linearly separable, e.g. the task of classifying between two concentric circles is challenging not because they are circles, but because they are concentric (Fig. 4.1(a) versus (b)).

4.2 Decision Boundary

4.2.1 Necessity of Definition

Characterizing the complexity of a classification problem based on the homological complexity of its decision boundary rises a big statement: all decision boundaries have the same homology.



(a) (b)

Figure 4.1: Two different classification problems where classes have the same topology however (b) is linearly separable while (a) is not.

Without this assurance it is not possible to make our analysis.

The problem above stems for the observation that a decision boundary is **not unique**. For example, support vector machines define a decision boundary that minimizes an empirical margin, while Neural Networks learn one that minimizes a user defined loss.

As a first step we formally define a decision boundary as the hypersurface that maximizes the interclass distance. This construction is reasonable since any other disparity metric (for example, support) is based on distance. Under this assumption, the decision boundary we aim to approximate is the union of the edges of adjacent Voronoi cells corresponding to points of different classes **and is unique.**

Definition 4.2.1 (Voronoi Cell)**.** Let (X, d) be a metric space and $S = \{s_1, ..., s_k\}$ be a set of elements of X. The *Voronoi cell* associated with point s_i is the set:

$$V_{s_i} = \{x \in X \mid d(x, s_i) \leq d(x, s_j) \forall i \neq j\} \tag{4.2}$$

We call **Voronoi Cover** (or Voronoi Diagram) to the collection $(V_s)_{s \in S}$.

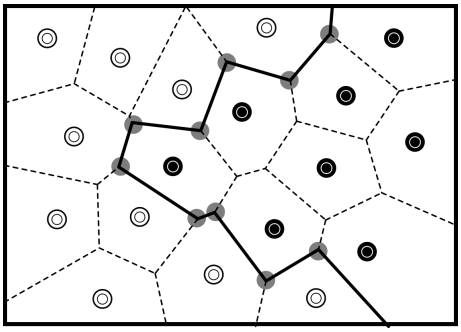


Figure 4.2: Illustration of a Voronoi diagram (dotted line) along with the decision boundary (solid line) of two classes (black and white).

Intuitively, the Voronoi cell of a point s_i is the set of all the points that are the closest to s_i than to any other datapoint. Given two points of different classes a_i, b_i their edge is the

intersection of both cells. This intersection is empty unless both points have adjacent cells. In that case the edge defines a bounded hyperplane.

Definition 4.2.2 (Decision Boundary)**.** Let $S = A \sqcup B$, given points of different classes $a \in A$ and $b \in B$ their decision boundary is the set:

$$DB_{a,b} = V_a \cap V_b = \{x \in \mathbb{R}^n \mid d(x,a) = d(x,b) \leq d(x,s)\} \tag{4.3}$$

We call **Decision Boundary** the union: $\bigcup_{a\in A, b\in B} DB_{ab}$

We call **decision boundary** the collection of all these edges. Therefore, for a classification problem in $\mathbb{R}^n$ with c classes, the decision boundary is a $n-1$-manifold since it is a union of bounded hyperplanes.

4.2.2 Sampling the decision boundary.

While Voronoi diagrams are fundamental structures, computing one on n points in $\mathbb{R}^d$ requires $O(n \log n + n^{\lceil d/2 \rceil})$[5], making it prohibitive in high dimensions. One can bypass this hurdle by taking advantage of the fact that we do not require the complete boundary but only enough to compute its topology. Thus we introduce an algorithm to sample the decision boundary that is theoretically exact and computationally feasible for high dimensions. The central idea is to sample randomly a point and then "push" it into the decision boundary, by iteratively projecting it to the hyperspace orthogonal to its closest points belonging to different classes.

Algorithm 1: Sample the decision boundary

Input: $A \leftarrow$ list of points class A
$B \leftarrow$ list of points of class B
$n \leftarrow$ number of points to sample from boundary
$iteration \leftarrow$ number of iterations

Output: $Q \leftarrow$ list of n points in the decision boundary of A and B

$Q \leftarrow$ Sample n points uniformly.;
for *each iteration* **do**
  for *each point* p *in* Q **do**
    $p_A \leftarrow$ Nearest Neighbour of p in A ;
    $p_B \leftarrow$ Nearest Neighbour of p in B ;
    project p *to the hyperplane orthogonal to* $p_A - p_B$;
    $p \leftarrow proj_{(p_A - p_B)_\perp}(p)$;
  end
end

Proposition 4.2.3. The algorithm converges to the edges of adjacent Voronoi cells corresponding to points of different classes.

Proof. By definition the Voronoi cell associated with point $s_i \in S$ is the set $\{x \in \mathbb{R}^n \mid d(x,s_i) \leq d(x,s_j) \forall i \neq j\}$. Given a point a_i belonging to a class, and b_i belonging to another class, we

have that the set of points in the common edge of their Voronoi cells is given by: $DB_{ab} = \{x \in \mathbb{R}^n \mid d(x, a_i) = d(x, b_i) \leq d(x, s_j)\}$.
Therefore, at a given iteration of the algorithm, if point P does not belong to the set DB_{ab} then, by definition of Voronoi cell, there has to exist a point a_j (or b_j) such that $d(P, a_j) < d(P, a_i) = d(P, b_i)$. Therefore this point is considered the new closest neighbor in the next iteration. It follows that the algorithm only stops when all points reach the decision boundary. □

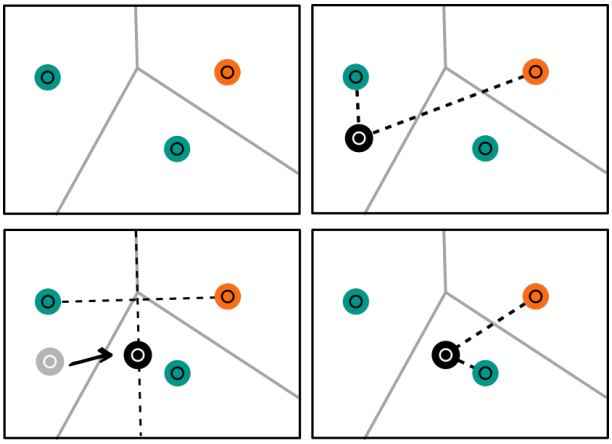
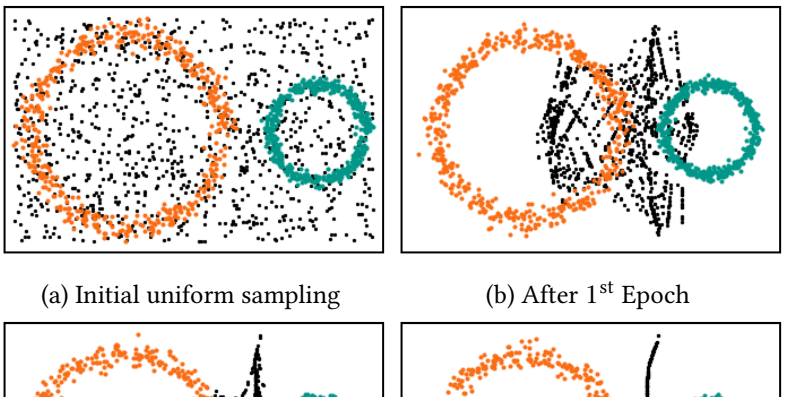


Figure 4.3: Example of the algorithm with just one sampled point (black), orange and green represent points of different classes. The Voronoi Diagram represented in grey. It first finds it closest neighbours of different classes and then projects the point to the orthogonal hyperplane.



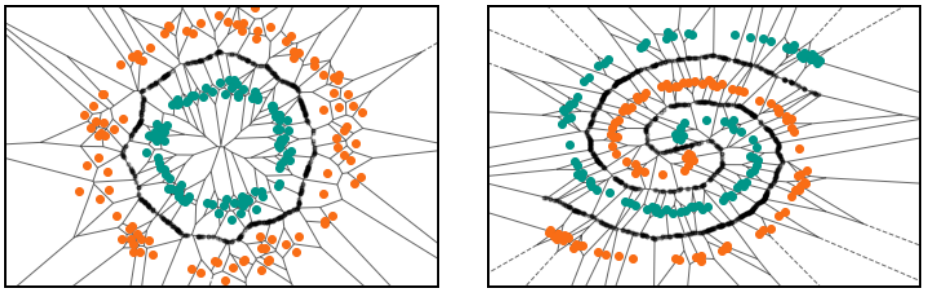
(a) Initial uniform sampling

(b) After 1st Epoch

(c) After 2nd Epoch

(d) After 3rd Epoch

Figure 4.4: Example of the decision boundary sampling algorithm over 3 epochs. First we sample N points uniformly distributed (black) (a). Each point is then pushed to the hyperplane orthogonal to the closest neighbours of each class (orange and green). The process is repeated 3 times (a)→(b), (b)→(c), (c)→(d).



(a) Concentric Circles (b) Spirals

Figure 4.5: Sampled points in the decision boundary (black). Along with the Voronoi diagram. Note that all points lie on the edges of the Voronoi cells belonging to points of different classes (orange and green).

4.2.3 Complexity

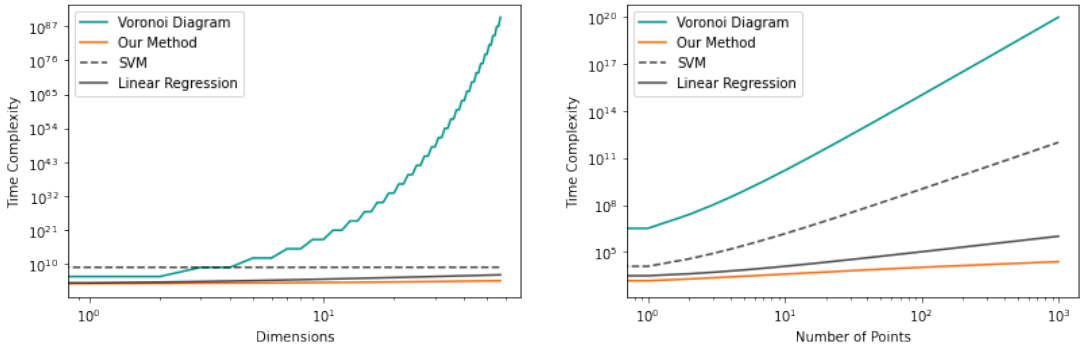
As stated before, calculating the Voronoi diagram of n points in $\mathbb{R}^d$ has a time complexity of $O(n \log n + n^{\lceil d/2 \rceil})$[5].

Below we study the complexity of our algorithm. The method is composed of two main steps:

1. **The (conditional) Neighbour Search.** This step has an average complexity of $O(\log n)$, although this can be reduced using approximate nearest neighbour methods such as *Nearest Neighbour Descent* (Dong et al. [35]) and *HNSWLIB* (Malkov and Yashunin [66]), Fig. 4.6.

2. **Getting the orthogonal hyperplane.** Calculating a orthonormal basis of the subspace orthogonal to a vector $w \in \mathbb{R}^d$ is normally done by taking the null-space of a matrix $[w \mid 0]$. That is, the square matrix $d \times d$ with only one non-zero column, w. Such operation has average complexity of $O(d^3)$, which is good but still above that sub-quadratic sweet-spot. It is possible however to calculate it using **QR-Decomposition** which approximates the orthonormal basis using a least-squares method. This method lowers the complexity to $O\left(\sqrt{d}\right)$, Fig. 4.6.

Running our method through k epochs to calculate the decision boundary using n points in $\mathbb{R}^d$ has average complexity of $O\left(\left(\sqrt{d} + \log n\right)^k\right)$. This is **orders of magnitude** faster than calculating the Voronoi diagram, making the proposed algorithm very reliable even high for dimensions (above 1000). For perspective, standard (Support Vector Machine) SVM has $O(n^2 d + n^3)$ complexity, which is known to be underperforming for high number of points. On the other hand we have Linear Regression, with its misleading name, with $O(nd^2 + d^3)$ complexity. This might seem surprising but Linear Regression relies on matrix inversion which is a complex task for high dimensions Fig (4.6).



(a) (b)

Figure 4.6: Comparison of the time complexity of our proposed sampling method and the common calculation of the Voronoi diagram for varying dimensions (a) and number of points (b). For comparison there are also the time complexities of the standard SVM algorithm $O(n^2d + n^3)$ and Linear Regression $O(nd^2 + d^3)$. Note the logarithmic scale on **both** axes.

4.2.4 Stability

The sampling algorithm was designed to enable us to compute the persistent homology of the decision boundary. As such, stability denotes how many point are needed to accurately evaluate the homology of the decision boundary. In this regard our method is inherently robust, by construction, due to two factors:

1. **Fewer points are required to capture topological properties** compared to geometric properties. To accurately describe a certain topological feature one does not require all the points. Consider for example a circle. It is possible to capture its topological properties with only 4 points, given that they are uniformly distributed for example this is the basis of the *Witness Complex* (Silva and Carlsson [94]). In practice, this means in a dataset with 1000 points we should require much fewer to accurately capture the topology of its decision boundary.

2. **The initial sample covers the space uniformly.** If we condition the initial cover of points, in the algorithms initialization, to be uniformly distributed, it is easy to see that they will approximately cover of the space uniformly[1].

In conclusion, by construction our method captures the accurate topological features of the decision boundary using a small number of points (Fig. 4.7). This is a desirable property especially in case of very large datasets. As a rule of thumb we recommend sampling the same number of points as the cardinality of the sparsest class.

4.2.5 Scalability

Another major improvement of our method relative to the one developed by Ramamurthy et al. [87] is its scalability. Their method allows only binary classification while our method is

[1] This is ongoing work

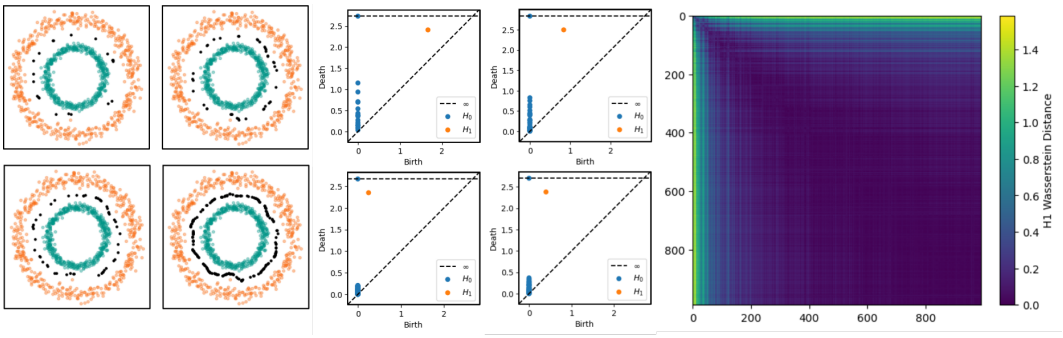


Figure 4.7: Left: Sampling 20,50,100 and 200 points (black) in the decision boundary of two classes (orange and green). Center: The persistence diagrams associated to each set of sampled points (from the decision boundary). Right: The Wasserstein Distance matrix of the $H1$ persistence diagrams of decision boundaries sampled from 10 to 1000 points.

scalable to **multiclassification** problems. At a given iteration for each point p we compute its closest neighbours p_A and p_B. The only requirement is that these belong to different classes, disregarding the total number of classes. Consider a multiclassification problem, given by a dataset $S = \{s_0, s_1, \dots s_n\} = A_0 \bigsqcup A_1 \bigsqcup \dots A_k$ where k is the number of classes. At a given iteration for each points p we compute its closest neighbours p_{A_i} and p_{A_j} here the only requirement is that $i \neq j$. Nothing else is affected, including complexity.

4.3 Topological Complexity

Recall the thought experiment of linearly separable problems and *bi-linearly* separable problems. The first has trivial H_0 class while the second one does not. We can say that classification problems in the latter class are more complex than in the former because their decision boundaries have trivial H_0 homology. The affirmation is obvious when one realizes that any linearly separable problem is also *bi-linearly separable*, meaning that a model that solves the latter category also solves the former.

Intuitively a decision boundary with more connected components is more complex than another with just one. We propose that a classification problem is complex if and only if its decision boundary has non-trivial topology. But how to quantify this complexity?

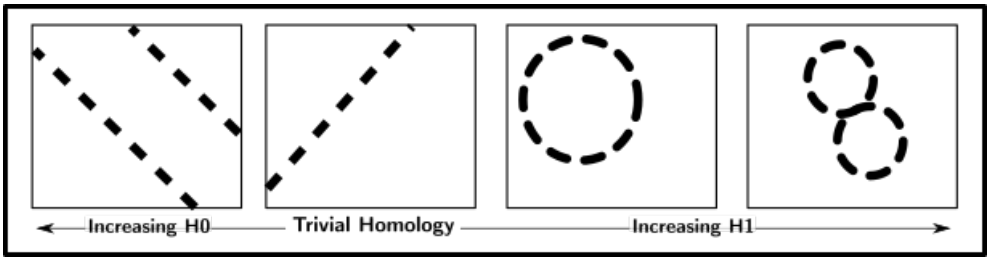


Figure 4.8: Illustration of decision boundary with trivial homology and decision boundaries with increasing H_0 and H_1 homology groups.

The definition of topological complexity has been first introduced by Farber [40] in the

context of motion planning problems. It formalizes this general idea we have been introducing so far. However it is based on a topological features different than homology called **homotopy**, that is much more difficult to work with in practice. While it has garnered a lot of attention, including an application to simplicial complexes (Fernández-Ternero et al. [42]), a computationally feasible workflow has not yet been achieved.

On the other hand Bianchini and Scarselli [11] present an intuitive **heuristic** for topological complexity as the sum of its Betti numbers. However, with real-world data, one has to take persistent homology as a measure of the homology groups of each dimension. Ramamurthy et al. [87] also tackle this problem using only persistent homology. In their approach they develop a heuristic for topological complexity as the total sum of persistence.

Our approach fits somewhere in the middle. It has direct topological meaning beyond persistence, while managing to leverage only persistence diagrams. We define topological complexity as the $p-$Wasserstein Distance to a contractible space [2].

Definition 4.3.1 (Topological Complexity)**.** Given a k-persistence diagram D_k, we call $p-$Topological Complexity $\mathcal{T}(D_k)$ to its $p-$Wasserstein Distance to the empty diagram. That is, let γ be such that it maps each point $x \in D_k$ to its closest point in the diagonal. Then the Topological Complexity is given by:

$$\mathcal{T}_p(D_k) = \left(\sum_{x \in D_k} \|x - \gamma(x)\|_p \right)^{\frac{1}{p}} \tag{4.4}$$

Throughout all experiments, for both topological complexity and Wasserstein distance we considered $p = 1$.

In this sense topological complexity can be understood as the topological distance to something we know to be not complex. This has direct application to classification problems. We can characterize their complexity by measuring how topologically far away they are from being linearly separable, i.e. not complex.

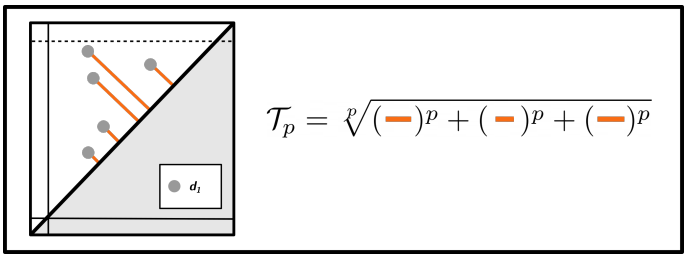


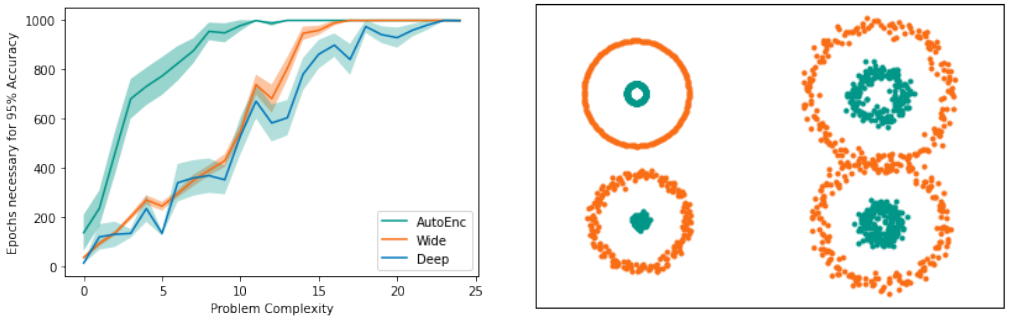
Figure 4.9: Illustration of topological complexity measure.

We generated 25 binary classification problems in $\mathbb{R}^2$ with increasing complexity. In order to map topological complexity to one dimension we assured that $\dim(H_0)$=$\dim(H_1)$ (Fig 4.10 b). This makes it possible to order the 25 problems from least complex, ($\dim(H_0)$=$\dim(H_1)$=1), to the most complex, $\dim(H_0)$=$\dim(H_1)$=50.

[2] A space with trivial homology

We then took 3 different Neural Network architectures. All architectures had a total of 5000 parameters, however they had very different structure. One was shallow and wide (79, 59 neurons), another very deep (44, 34, 33, 24, 18, 17,15, 13, 6), the third had a strong middle bottleneck similar to an autoencoder (50, 36, 14, 2, 14, 36, 50).

The justification for this experiment is twofold. First, we wanted to cement the topological complexity of the decision boundary as an adequate measure of a classification problems intrinsic difficulty. Second to motivate the next section. Notice that even though all architectures have the same number of parameters, they have very different behaviours towards tackling complex classification problems (Fig. 4.10).



(a) (b)

Figure 4.10: (a) Average (30 runs) number of epochs to reach 95% accuracy on binary classification problems of different complexity, for different architectures. (b) Example problem $H_0 = H_1 = 4$.

For all architectures, a more complex problem (under our metric) directly relates to difficulty in classification (Fig. 4.10 (a)). Even though all architectures have the same number of parameters, the autoencoder severely underperforms compared to the wide and the deep one, who seems to be the least influenced by the increase in difficulty..

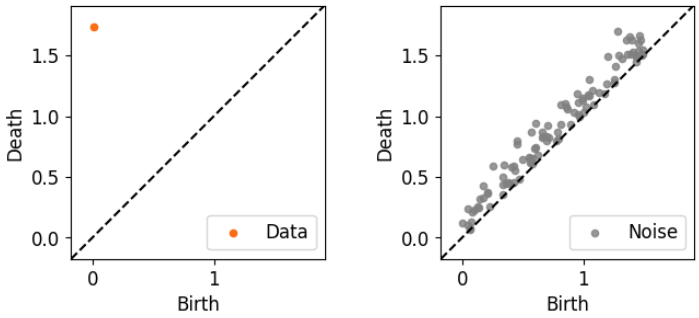
In the next section we will explore the idea of topological expressiveness of a neural network. After having identified, in this section, what makes classification problems different, one can now measure expressiveness by how topologically different and complex are the decision boundaries that a given architecture is able to express.

4.3.1 Stability

The main concern on the definition of the metric is the stability to noise. One might have that noise, i.e. points close to the diagonal, end up increasing the topological complexity, when in high numbers (Fig.4.11), because Topological Complexity is defined as a geometric mean of all the distances to the diagonal.

Since topological complexity is defined over the $p-$Wasserstein distance, the stability results of topological complexity **are the same** as the ones of Wasserstein Distance. Most importantly for when $p = \infty$ we have:

$$\mathcal{T}_\infty(D(f)_k) \leq \|f\|_\infty \tag{4.5}$$

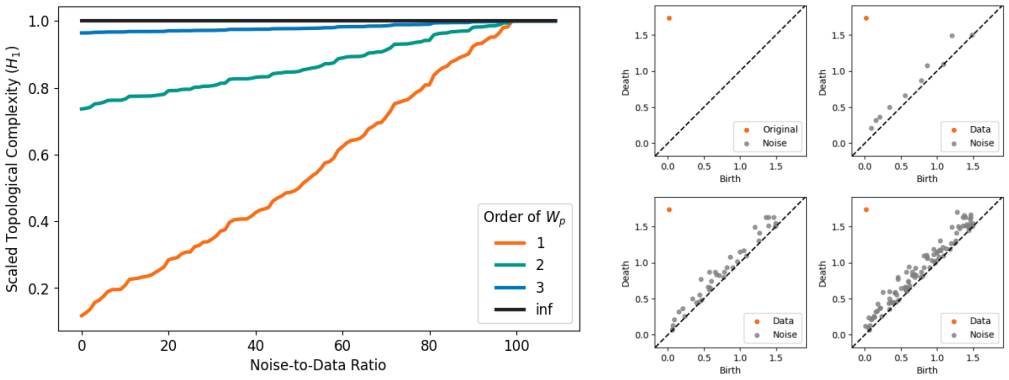


(a) Data (b) Noise

Figure 4.11: Comparison of two persistence diagrams, a) has one very persistent homology class, while b) has many but with non-relevant persistence, normally regarded as noise.

where $D(f)_k$ is the $k-$persistence diagram of f. Given a persistence diagram its topological complexity is invariant to the addition of noise, for $p = \infty$. At least as long as the persistence of the noise is smaller than that of the data. This conclusion is obvious by realizing that $\mathcal{T}_\infty(D)$ is equal the maximum of the distances to the diagonal.

But what happens to the stability of $\mathcal{T}_p$ as we consider different values of p? As an experiment we took a persistence diagram with one homology class (Fig.4.11 a)) and began to add noise to its diagonal (Fig.4.11 b)). This was done incrementally up to a Noise-to-data ratio of 100 to 1, meaning 100 noise classes for each data class (Fig.4.12 b)). For each persistence diagram we computed its topological complexity for different values of $p = 1, 2, 3$ and $p = \infty$ (Fig.4.12 a)).



(a) (b)

Figure 4.12: Topological complexity of a series of persistence diagrams with increasing noise-to-data ratio. (a) compares the stability for different p values of the $p-$Wasserstein distance used to compute topological complexity. (b) is a sample of the persistence diagrams corresponding to the noise-to-data ratios: 0, 1, 20 and 100

The values of topological complexity were normalized to the same scale as to be comparable. Unsurprisingly $\mathcal{T}_p$ is remarkably stable as p increases, ending up being completely invariant to noise for $p = \infty$.

Chapter 5

Topological Expressiveness of Neural Networks

Computers are useless. They can only give you answers.

-Pablo Picasso

5.1 Neural Networks

A fully connected dense Neural Network of n-layers can be seen as a chain:

$$X_0 \xrightarrow{L_0} X_1 \xrightarrow{L_1} \dots X_{n-1} \xrightarrow{L_{n-1}} X_n \tag{5.1}$$

Where X_0 is called the **input** and X_n the **output**. Each L_i is generally defined to be the composition of an affine transformation with a non-linear continuous monotonous function, called **activation** function.

$$L_i(x) = a(W_i x + B_i) \tag{5.2}$$

Each X_i is a set of points in $\mathbb{R}^k$, where k is the number of neurons in each layer i.e. the number of rows in the matrix W_{i-1}. One normally regards Neural Networks as a single function $f : X_0 \to X_n$ given by:

$$f = L_{n-1} \circ \dots \circ L_1 \circ L_0 \tag{5.3}$$

5.1.1 Topological Perspective

The realization that feed forward Neural Networks are but a sequence of maps between spaces has opened the door for topological tools that explain them. For example one would want to know what morphism ϕ_i is induced by the map L_i in the homology classes of X_i.

$$H_\bullet(X_0) \xrightarrow{\phi_0} H_\bullet(X_1) \xrightarrow{\phi_1} \dots H_\bullet(X_{n-1}) \xrightarrow{\phi_{n-1}} H_\bullet(X_n) \tag{5.4}$$

There are some situations where each transformation L_i is a homeomorphism meaning that both X_i and $L_i(X_i)$ have the same topology, and ϕ_i is an isomorphism.

Proposition 5.1.1. Let W_i be a $m \times n$ matrix such that $m \geq n$ and $rank(W_i) = n$, and let a be a continuous bijection with continuous inverse. The transformation:

$$L_i(x) = a(W_i x + B_i) \quad \forall x \in X_i \tag{5.5}$$

defines a homeomorphism between X_i and $X_{i+1} = L_i(X_i)$.

The above result states that, in theory, all topological features of the data are **preserved** from one layer to the next as long as the final layer has at least the same amount of neurons as the initial and the activation function is a homeomorphism (such as hyperbolic tangent or the sigmoid function). There is, however a caveat. Computationally speaking, as pointed by Naitzat et al. [78], functions like the hyperbolic tangent lose their otherwise homeomorphic behavior due to **rounding errors** of floating point precision numbers. In truth it is a piecewise function defined as:

$$\tanh_\delta(x) = \begin{cases} 1 & \text{if } \mathrm{fl}(\tanh(x)) > 1 - \delta \\ \mathrm{fl}(\tanh(x)) & \text{otherwise} \\ -1 & \text{if } \mathrm{fl}(\tanh(x)) < -1 + \delta \end{cases} \tag{5.6}$$

Where $\mathrm{fl}(x)$ is the floating point representation of x and δ the unit round-off. This means that while tanh: $\mathbb{R}^n \to (-1,1)^n$ is a homeomorphism, $\tanh_\delta : \mathbb{R}^n \to [-1,1]^n$ is not.

Even though most architectures of dense Neural Networks are classified as homeomorphisms, in reality they are not. This poses a huge problem to say, study the topological properties of data through training, or as they travel through the network. This topic is explored further in section 5.2.4 and central in section 5.3.

As the field of Topological Data Analysis develops, approaches to Neural Networks have become progressively more common in recent years. Bianchini and Scarselli [11] evaluate the maximum complexity certain architetures can express. They use the sum of Betti numbers as an heuristic for topological complexity. Guss and Salakhutdinov [48] showed that persistent homology can be used to characterize the capacity of neural architectures in direct relation to their ability to generalize on data. Rieck et al. [90] pointed that the 0-dimensional homology class of the weights and their connections is an adequate indicator of a given Neural Network's learning performance. Naitzat et al. [78] and Brahma et al. [15] propose an early topological understanding to the impact of Neural Networks in the topology of the data.

In order to advance on the work proposed by the above authors we must first develop one last tool to help us capture the topological features of Neural Networks. Each network is a map $f : X_0 \to X_n$ (made of the composition of functions L_i). Understanding the topological characteristics of a Neural Network is understanding the topological characteristics of a **transformation**.

This is fundamentally different from what we have until now. Proposition 5.1.1 provides both with a strong baseline but also the assurance that such baseline is only theoretical, not computationally verifiable.

Thus in order to understand how each function $L_i : X_i \to X_j$ acts topologically the common pipeline is to compare the persistent homology of both X_i and X_j. This is however not so trivial since X_i and X_j are different metric spaces and their persistent homology might not be comparable. We now present a novel method to solve this problem.

5.2 UDiPH: Uniform Distributed Persistent Homology

5.2.1 Motivation

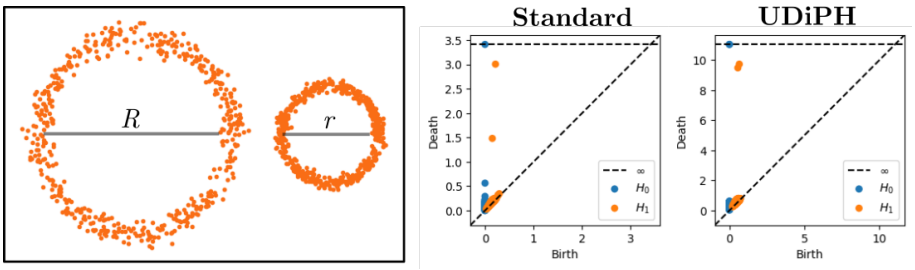
The study of the homology of the spaces X_i induced by a Neural Network, i.e the homology of the activations of each layer,

$$H_\bullet(X_0) \xrightarrow{\phi_0} H_\bullet(X_1) \xrightarrow{\phi_1} \dots H_\bullet(X_{n-1}) \xrightarrow{\phi_{n-1}} H_\bullet(X_n) \tag{5.7}$$

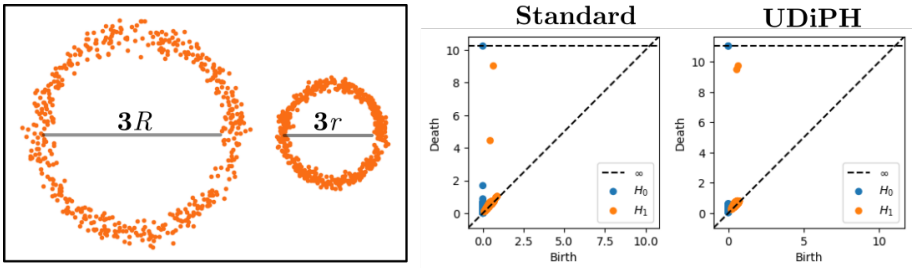
pose a speed bump in the common pipeline of topological data analysis. Since X_i and X_j are different metric spaces (for $i \neq j$) the persistence values are not comparable. Vietoris-Rips filtrations use the metric of the space as a filtration parameter and hence spaces with different metrics will necessarily have different persistent homology regardless if they have the same homology or not. This is because a standard Vietoris-Rips filtration (through the use of the metric) captures always some geometry of space. Thus a change in geometry will affect the topological summary of persistence homology. Even though no change in topology happened.

Take for example a space X_i and a map $L_i : X_i \to X_j$. One way to gain insight on the map L_i is to compare the topology of both X_i and X_j. However if L_j is a transformation that simply scales all pairwise distances by a certain factor, this function is obviously an homeomorphism and, as such it preserves all topological features. This implies that both X_i and X_j have the same homology. However, the persistence values of X_j are going to be scaled by that same factor (Fig 5.1) and thus different from X_i. If one were then to calculate the Wasserstein distance between the diagrams of both metric spaces, one would find the distance to be different than zero, even though the homology had not changed. In this case Wasserstein distance is measuring the change in persistence and not the change in homology. This is a major inconsistency in the perspective of topological data analysis. The tools we use to capture geometric invariant features, are dependent on said geometry.

In this chapter we present **UDiPH** (Uniform Distributed Persistent Homology). This novel approach creates a new metric space that both preserves the topology, and is independent of geometric changes. The objective is to then do a regular Vietoris-Rips filtration over this new metric space, which will highlight the topological properties (homology) and less the geometric ones (sizes). Notice on Fig. 5.1 a) that the persistence homology of a filtration using this UDiPH method still captures the 2 different H_1 classes (holes), however their persistence value is very similar.



(a) Example metric space.



(b) Example metric space where distances are scaled by a factor of 3

Figure 5.1: Comparison of persistence diagrams of a metric space (a) and the same but scaled by a factor of 3. The persistence diagrams are calculated using standard Vietoris-Rips filtration, and also using the presented novel approach UDiPH (Uniform).

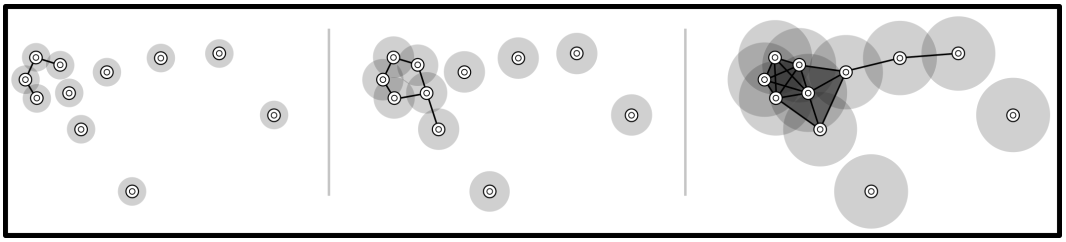
Compare it with regular persistent homology where the persistence of both H_1 classes is different. This is due to the fact that the Vietoris-Rips filtration uses the ambient metric as a filtration parameter. Thus the different persistence values represent the size of each H_1 class: one smaller, the other bigger.

After applying a simple scale, as described in the example above, the persistence values obtained through the standard Vietoris-Rips filtration have also been scaled by the same factor. On the other, hand the persistent homology captures through our UDiPH method has remained the same.

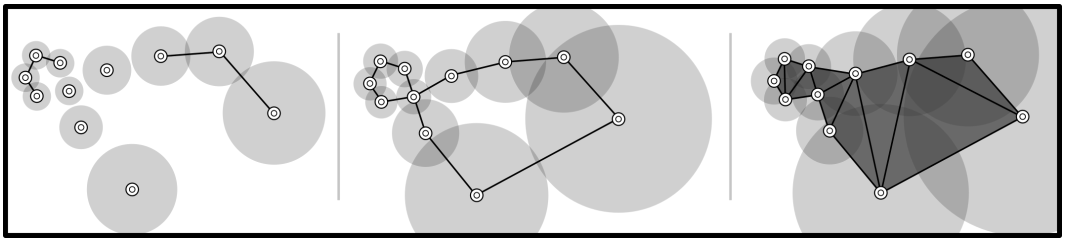
Such features can be desirable of course. The persistence values are normally used to validate the existence of the homology classes they represent. This is fundamental when working with **static** point clouds. In a different scenario, take a Neural Network for example, this dependence on persistence is undesirable since we have a sequence of **dynamic** metric spaces, and this we require a new way to compare their persistences. We wish to push persistence homology back towards its topological roots and remove some of its quantitative features in favor of a **more qualitative topological characterization**.

5.2.2 Background

Fuzzy mathematics deals with common mathematical structures that accept some uncertainty measurement. It has found its way into set theory, metric spaces (Kramosil and Michálek [58]) and topology (Chang [25])



(a) Standard Vietoris-Rips filtration.



(b) Simplicial filtration using UDiPH.

Figure 5.2: Illustration of a standard Vietoris-Rips filtration (a) where the filtration parameter is the ambient metric, or (b) the metric constructed using UDiPH.

Definition 5.2.1 (Fuzzy Set)**.** Let X be a set and. A fuzzy set A is the set:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \tag{5.8}$$

where $\mu_A : X \to [0, 1]$ is called the membership function.

Let A, B be two fuzzy sets and μ_A, μ_B their respective membership functions.

1. The **complement** of A is defined by the membership function:

$$\mu_{X \setminus A} = 1 - \mu_A \tag{5.9}$$

2. The **intersection** $A \cap B$ is given by:

$$\mu_{A \cap B} = \min\{\mu_A, \mu_B\} \tag{5.10}$$

3. The **union** $A \cup B$ is given by:

$$\mu_{A \cup B} = \max\{\mu_A, \mu_B\} \tag{5.11}$$

Definition 5.2.2 (Fuzzy Topology)**.** A fuzzy topology $\mathcal{T}$ on a set X is a collection of fuzzy sets of X satisfying:

1. $\varnothing, X \in \mathcal{T}$
2. It is closed under unions.
3. It is closed under finite intersections.

5.2.3 Construction

5.2.3.1 Intuition

UDiPH is novel approach to the computation of persistent homology built on a new metric space that better focuses on the topological features of the data. Given two metric spaces X and Y, our objective is two create new metric spaces $\mathcal{X}$ and $\mathcal{Y}$ where the persistence values of the respective persistence diagrams are comparable.

$$\begin{array}{ccc} \mathcal{X} & \xrightarrow{W_p} & \mathcal{Y} \\ U\uparrow & & \uparrow U \\ X & \xrightarrow{W_p} & Y \end{array}$$

This new metric space is built as to preserve a fuzzy topology that is constructed in an analogous manner for both spaces.

This method is the same as used by McInnes et al. [67] who, in turn, is built over the theoretical guarantees of Spivak [97]. The construction steps, including the construction of the fuzzy topology, are the same used in McInnes et al. [67]. The main pipeline is as follows:

1. **Assume uniform sampling.** Assume that the data has been uniformly sampled from a data manifold with respect to a Riemannian metric.

2. **Approximate metric**. For each data point, create a local metric space that approximates this Riemannian metric thus validating the uniformity assumption.

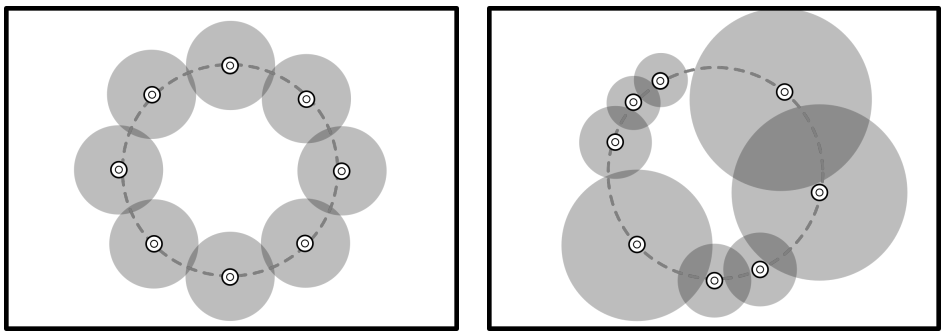
3. **Fuzzy Topology.** Merge different metric spaces into a fuzzy topological space.

4. **Create new metric space that preserves topology.** Create global metric based on topological structure.

5.2.3.2 Uniform Distribution

Similarly to Niyogi et al. [79] and McInnes et al. [67], this algorithm's main assumption is that the data is uniformly distributed in the data manifold. Naturally, real world data is rarely that well behaved from the perpective of ambient metrics. However we assume that the manifold from which the data has been sampled has a Riemannian metric that makes this assumption true.

In Fig.5.3 a) we have a set of points that are uniformly distributed with respect to both the ambient dimension **and** the geodesic. In this case, all the drawn circles have the same radius. Consider now Fig. 5.3 b) where we have some data that is **not** uniformly distributed with regards to the ambient metric. Under this assumption we create different metric spaces for each point such that the data is now uniformly distributed. Here each circle also has the same radius (but not in ambient dimensions). By creating a different notion of distance for each point, we enforce our assumption. We then unify all these metric spaces into a fuzzy topological space.



(a) (b)

Figure 5.3: Example of the assumption of uniform distribution. In both scenarios the balls represent have all radius one. with respect to the ambient dimension a), or to a Riemannian metric b).

5.2.3.3 Approximate Metric

Informally, if a sample S is uniformly distributed in X with relation to a certain metric, then in a ball of fixed radius around each $x_i \in S$ there is approximately the same number of points. Conversely, a ball centered at a certain point that contains its k-nearest neighbours should have the same radius for all $x_i \in S$.

From the previous step we already assumed that there exists a metric such that S is uniformly distributed in X. We now need to approximate that metric using the ambient metric. The following lemma guarantees that we can approximate geodesic distances from a point to its neighbours by normalizing distances with respect to the distance to its k-th nearest neighbour.

Lemma 5.2.3 (McInnes et al. [67])**.** Let (M, g) be a Riemannian manifold in $\mathbb{R}^n$, and let $p \in M$ be a point. If g is locally constant about p in an open neighbourhood U such that g is a constant diagonal matrix in ambient coordinates, then in a ball $B \subseteq U$ centered at p with volume $\frac{\pi^{n/2}}{2\Gamma(n/2n+1)}$with respect to g, the geodesic distance from p to any point $q \in B$ is $\frac{1}{r} d_{\mathbb{R}^n}(p, q)$, where r is the radius of the ball in the ambient space and $d_{\mathbb{R}^n}$ is the existing metric on the ambient space.

In practice this means that if $\{x_i^1, x_i^2, \dots x_i^k\}$ are the k-nearest neighbours of $x_i \in S$, then we take,

$$d'(x_i, x_i^j) = \frac{d(x_i, x_i^j)}{\max\limits_{m \in \{1,\dots,k\}} d(x_i, x_i^m)} \quad \forall x_i \in S. \tag{5.12}$$

The reader is pointed again to the definition of Manifold in order to have an informal intuition of lemma 5.2.3. Since manifolds are **locally** homeomorphic to an euclidean space.

5.2.3.4 Fuzzy Topology

We now create a cover of X using fuzzy sets. For each point x_i we create a fuzzy set. For each of its $k-$nearest neighbours $\{x_i^1, x_i^2, \dots x_i^k\}$ the membership function μ_i is given by:

$$\mu_i(x_i^j) = \exp\left(-\frac{d'(x_i, x_i^j)}{\sigma_i}\right) \tag{5.13}$$

Where σ_i is such that:

$$\sum_j^k \mu_i^j = \log_2 k \quad \forall x_i \in S \tag{5.14}$$

The value of σ_i is a normalizating factor useful to unite all different metric spaces. The reader is once again pointed to Spivak [97] and McInnes et al. [67] for a thorough explanation of this process. The proofs and derivations are well into *Category Theory* which is beyond the scope of this thesis.

We wish to build a global metric through the membership functions μ_i. Notice however, that the membership functions are not necessarily symmetric. That is, its **not** true that

$$\mu_i(x_j) = \mu_j(x_i) \quad \forall x_i, x_j \in S \tag{5.15}$$

The next step is then to take care of this asymmetry. Let x_i be the m^{th}-neighbour of x_j. And x_j be the n^{th}-neighbour of x_i, we consider the new value:

$$\delta_{ij} = \mu_i(x_j) + \mu_j(x_i) - \mu_i(x_j)\mu_j(x_i) \tag{5.16}$$

If the intuition of a membership function is that of a probability of belonging to that set then δ_{ij} can be informally understood as the probability of x_i or x_j belonging to each other's set minus the probability of belonging to the intersection:

$$\delta_{ij} = \underbrace{\mu_i(x_j)}_{P(x_j \in X_i)} + \underbrace{\mu_j(x_i)}_{P(x_i \in X_j)} - \underbrace{\mu_i(x_j)\mu_j(x_i)}_{P(x_{ij} \in X_i \cap X_j)} \tag{5.17}$$

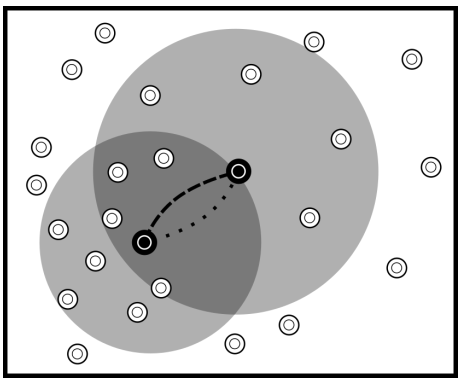


Figure 5.4: Illustration of different local metric spaces (grey circles) resulting in non symmetric distances between points (dotted lines) since both circles have the same radius.

5.2.3.5 Metric Space

The resulting construction is a 1-dimensional fuzzy simplicial complex (a.k.a. weighted graph)defined by the adjacency matrix $A = [\delta_{ij}]$. This topological space is the approximation of our data manifold in which, by construction, the data is uniformly distributed.

As our objective is to be able to calculate persistence homology we need to create a metric space. However, we wish to take advantage of the newfound metrics and manifold in order to do this. This can be done in two steps:

1. **Local metric.** If we consider distance of x_i to x_j as the probability of x_i **not** belonging in the fuzzy set X_j we create the dissimilarity matrix $A' = [1 - \delta_{ij}]$, which is symmetric by construction.

2. **Global metric.** We then extend this notion and define the distance between **any** two points $x_i, x_j \in S$ as the shortest path in the simplicial complex A', which is an approximation of the geodesic distances in the manifold.

5.2.4 Stability

5.2.4.1 Of a Transformation

Consider a function $f : X \to Y$ on two metric spaces X, Y. We compare the stability of the persistence diagrams of both X and Y using standard Vietoris-Rips filtration and our method. Despite the abuse of language, for ease of reading, we will denote by $W_p(X_u, Y_u)$ the $p-$Wasserstein distance between the persistence diagrams of both X and Y after applying the UDiPH method.

In Fig. 5.1 we have seen that homeomorphic transformations such as scaling have a strong impact in the persistence diagrams of Y. UDiPH was developed with the following implication in mind:

$$X \text{ homeomorphic to } Y \implies W_p(X_u, Y_u) = 0 \tag{5.18}$$

We will show that in **some** situations that is actually the case. We will also present situations where conjecture 5.18 only approximately holds, meaning that:

$$W_p(X_u, Y_u) \neq 0 \text{ however } W_p(X_u, Y_u) \leq W_p(X, Y) \tag{5.19}$$

and provide a possible explanation as to why there is not a complete invariance. The author believes that conjecture 5.18 holds under mild assumptions and an adapted proof can be achieved. Proving that *the neighbour structure of uniformly distributed manifolds is invariant under homeomorphisms* is a ongoing future work direction. Consider the following types of homeomorphisms.

1. **Isometry** is a map that preserves all pairwise distances. It preserves the metric of X in Y. Rotations, reflections and translations (very common in Neural Networks) are examples of isometries. Isometries are the isomorphisms of geometry, meaning they preserve the geometric structure.

2. **Similarity** transformations preserve angles and ratio between distances, such as scaling Fig. 5.1 and Fig. 5.5.

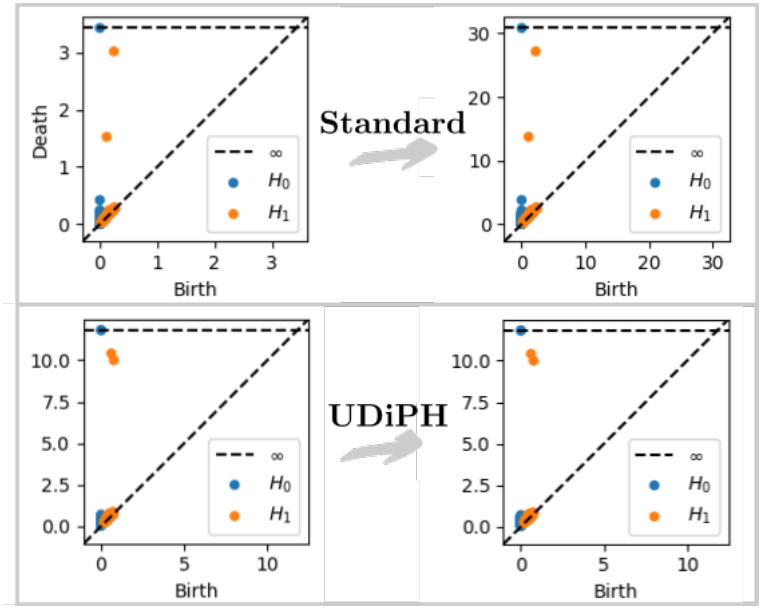


Figure 5.5: Persistence diagrams before and after a **similarity** transformation, using standard Vietoris-Rips filtration (top), and UDiPH (bottom).

3. **Affine** transformations preserve parallel vectors. Since we are considering both X and Y to be embedded in some vector space in $\mathbb{R}^n$ and $\mathbb{R}^m$ these vectors are well defined. A linear **projection to a higher dimensional** space using a non-singular matrix is an example (actually the one that is most common in Neural Networks). (Fig. 5.6).

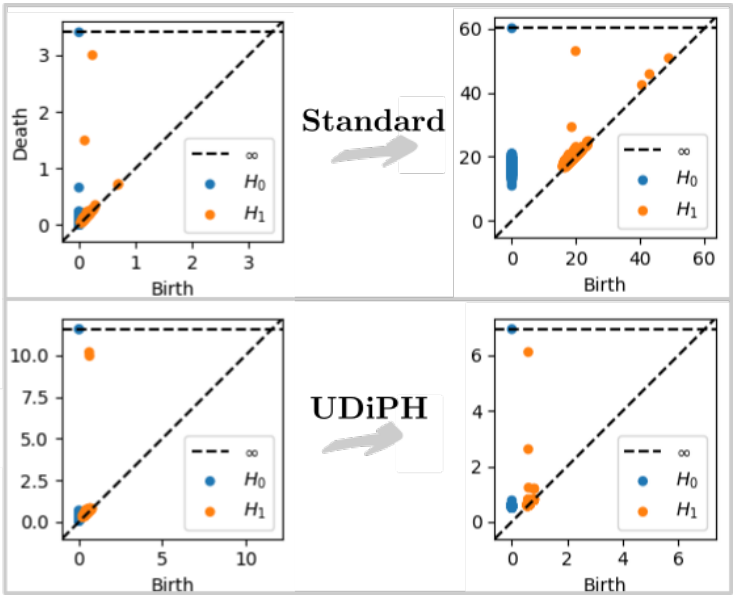


Figure 5.6: Persistence diagrams before and after an **affine** transformation, using standard Vietoris-Rips filtration (top), and UDiPH (bottom).

Transformation		H0		H1	
		Standard	UDiPH	Standard	UDiPH
Isometry	Rotation	0	0	0	0
	Translation	0	0	0	0
Scale (factor)	5	3056.1	5.3	318.4	6.1
	25	15174.7	5.3	1579.8	6.1
	100	60615.2	5.3	6305.2	6.1
Affine	Random	$8642.9_{\pm 2131.9}$	$220.12_{\pm 5.2}$	$501.9_{\pm 174.6}$	$73.5_{\pm 28.3}$
Continuous Bijection	exp	60723.6	26.1	4182.7	7.9
	$sigmoid$	11371.2	218.1	590.5	57.8
	$tanh$	6041.5	202.5	237.8	32.2

Table 5.1: Wasserstein distance between X and Y for different homeomorphic functions $f : X \to Y$. We compare the H_0 and H_1 homology as the functions always had input domain $\mathbb{R}^2$. For affine transformations random matrixes were used and such the result is the mean of 20 runs and along is the standard deviation.

4. **Continuous bijections** are the definition of homeomorphisms, with continuous inverse of course(e.g. functions such as exp, log etc, Fig. 5.7).

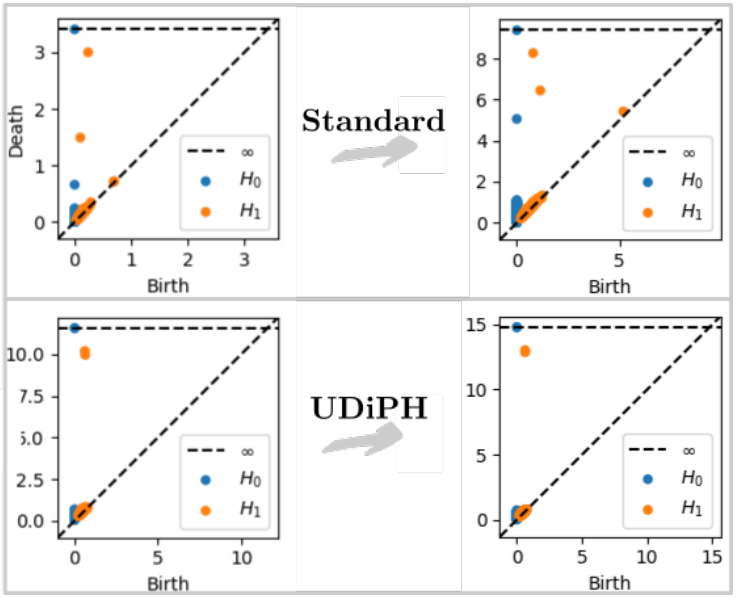


Figure 5.7: Persistence diagrams before and after an **continuous** transformation (Hyperbolic Tangent), using standard Vietoris-Rips filtration (top), and UDiPH (bottom).

5. **Compositions.** Any composition of the above maps is also a homeomorphism. If f, g are homeomorphisms then $f \circ g$ and $g \circ f$ are also homeomorphisms.

In Table 5.1, we present different homeomorphic transformations $f : X \to Y$ along with the $p-$Wasserstein distance $W_p(X, Y)$ between the persistence diagrams of the domain (X) and codomain (Y). Note that since f is a homeomorphisms, in theory $W_p(X, Y) = 0$.

Invariant to scaling. While both methods achieve complete invariance to isometries, the standard Vietoris-Rips filtration still suffers from resizing. Note that as we increase the scale factor, the Wasserstein distance between input and output increases proportionally. With UDiPH, however, even though it is not precisely zero, it is **completely invariant** to the scale of the metric space. This is arguably the most important property since it allows us to compare persistent homology of spaces with very different scales, for example, Euclidean spaces of **different dimensions**.

More stable for other transformations. UDiPH shows that it preserves much more the homology of the data in the presence of other strong transformations, such as exponential and affine transformations. It is, unfortunately, not invariant but much stabler nonetheless. The reader is pointed to the composition of affine transformation, translation and sigmoid or hyperbolic tangent as it is a very common layer transformation in Neural Networks.

As invariant as it gets. The search for homeomorphism invariance might be a futile one. Recall the fact that homeomorphisms are impossible for computers to compute. As such functions such as exponential and sigmoid etc. still display some error when computed even with UDiPH.

5.2.4.2 Of Hyper-parameters.

UDiPH has one hyper-parameter: the number of neighbours used. This parameter controls the trade-off between global and local structure. Lower values favor local structure over global. It controls the number of nearest neighbours used to approximate the Riemannian metric of the data manifold, and thus the connectivity of the fuzzy simplicial complex.

We wish to examine the stability of this parameter. Consider the very common layer transformation: $L(x) = tanh(Wx+B)$ for a non-singular matrix W. We then compute various X^i for different values of nearest neighbours. In Fig 5.8 we have the $p-$Wasserstein distance matrix, $W_{ij} = W_p(X_i, X_j)$. The data has a total of 1000 points. We see that even going as far as

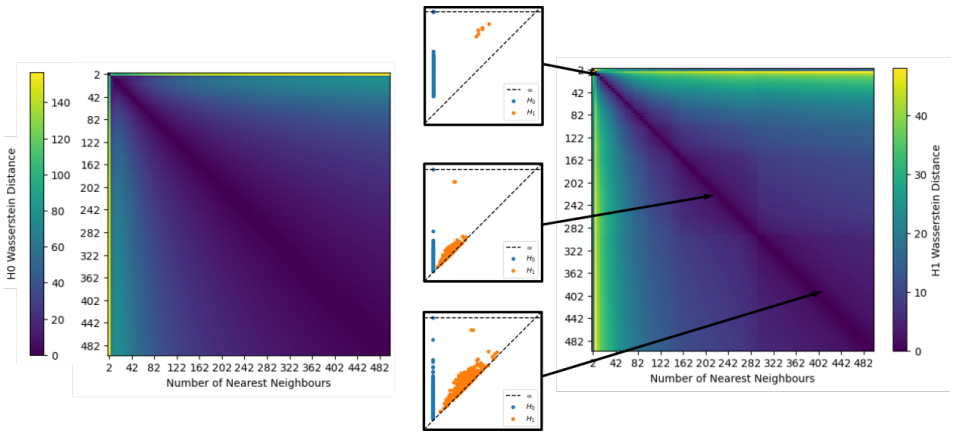


Figure 5.8: Wasserstein distance matrix of persistence diagrams computed using UDiPH for H_0 (left) and H_1(right).

500 nearest neighbours the homology of the data is still robustly captured. The only suboptimal

values seem to be below 5 nearest neighbors. Regardless, there is considerable stability for all values, and although there are some mild fluctuations, the homology is always well described for values in $[5, 400]$. Even the small fluctuations are irrelevant when comparing different spaces since one can use the same value of nearest neighbors throughout.

5.3 Disentanglement

Previously (Chapter 4) we developed the idea that a classification problem is presented as an entanglement of different manifolds. Olah [80] shows that Neural Networks deform these manifolds in such a way as to make them linearly separable, **disentangling** them.

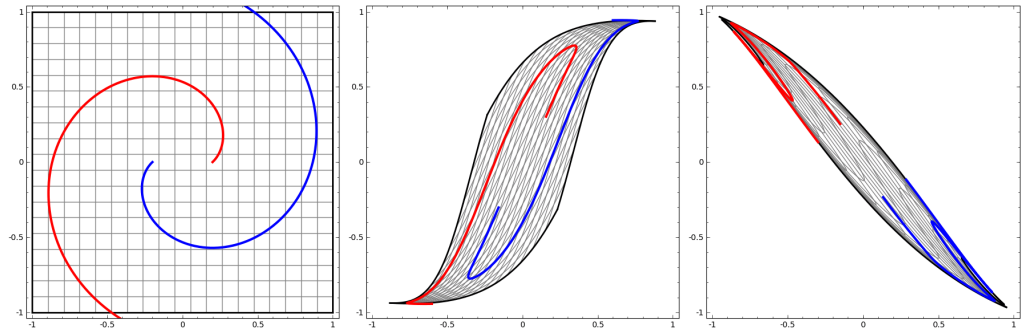


Figure 5.9: A (shallow) Neural Network learning a linearly separable embedding of the input space. Images courtesy of Olah [80].

The architecture of a Neural Network is therefore directly involved in how effective it is in disentangling. His work was then picked up by Brahma et al. [15] who build a rough framework to test a Neural Networks effect on manifold entanglement, yet the concepts of **curvature** and **entanglement** are loosely defined, often interchanged and presented as equivalent, measuring flattening as disentanglement.

A topological perspective of this scenario was first drafted by Naitzat et al. [78], who empirically showed that Neural Networks deform these manifolds to simplify their topology (Fig. 5.10). However, they inevitably encounter two problems:

1. **Inadequate definition of complexity.** They follow Bianchini and Scarselli [11] definition of topological complexity as the sum of the Betti numbers of each dimension. This is adequate for theoretical environments since the manifold is known. For practical cases, it is inadequate because the real world manifolds are unknown, and thus the Betti numbers have to be empirically inferred.

2. **The complexity of the classes is irrelevant.** Similarly to Guss and Salakhutdinov [48], they focus solely on the homology of each class. We have shown that the homology alone does not reflect the complexity of the classification problem, which can be better expressed by the complexity of the decision boundary. As such, they end up associating disentanglement through the observation of the simplification of the homology of each class, which by no means implies disentanglement.

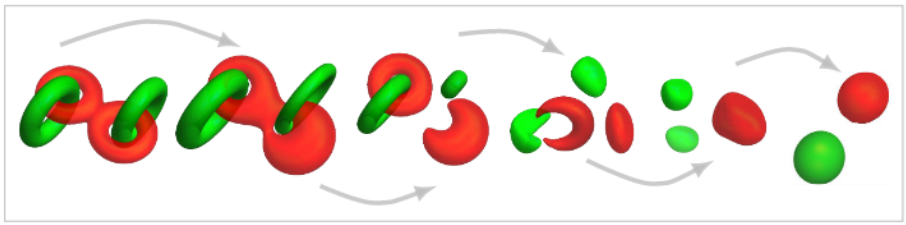


Figure 5.10: The Neural Network learning an embedding that separates two entangled manifolds (classes). Image from Naitzat et al. [78]

The reader may have noticed already that the previous chapters have prepared us to tackle this same problem from a controlled topological perspective. We have shifted the aim to the decision boundaries, have defined topological complexity in terms of persistence homology (enabling us to apply it to real world data) and have tackled the issue of computing persistent homology on different metric spaces.

Using our work so far we can support, formalize and enhance the approaches of Olah [80], Naitzat et al. [78] and Brahma et al. [15]. Take a trained Neural Network $f : X_0 \to X_n$:

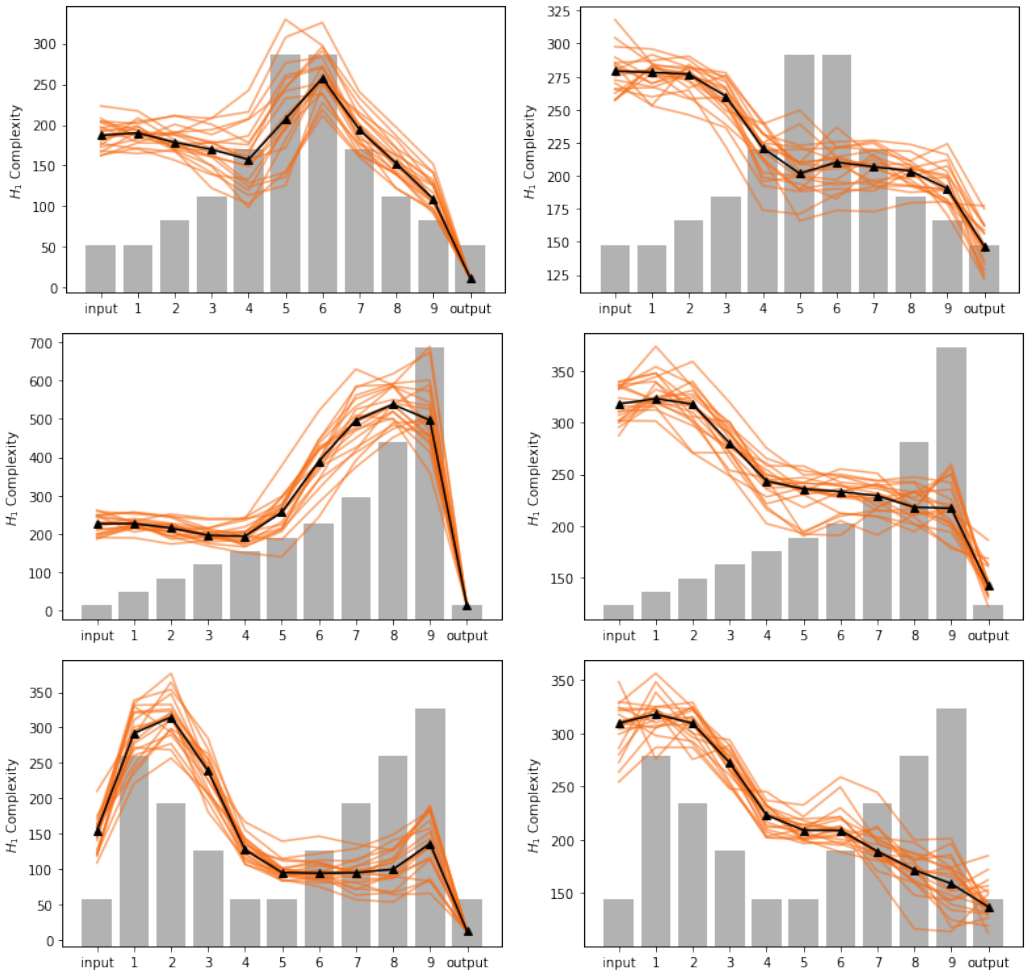
$$X_0 \xrightarrow{L_0} X_1 \xrightarrow{L_1} \dots X_{n-1} \xrightarrow{L_{n-1}} X_n \tag{5.20}$$

Using UDiPH we can now compare the homologies of X_i for all i. This step requires UDiPH because X_i and X_j are different metric spaces for $i \neq j$. Standard Vietoris-Rips filtrations that are dependent on ambient metric will generally provide higher persistence values in high dimensional spaces, such as when a layer has a lot of neurons, and lower values in low dimensional spaces, when the number of neurons is low (Fig. 5.11 a)). With UDiPH we are able to compare these spaces regardless of their metric and dimension (Fig 5.12 b)). We train several Neural Networks on the MNIST and Fashion-MNIST datasets, until they reach 95% accuracy, and compute the topological complexity of the decision boundary after each layer using UDiPH. We can see that in trained Neural Networks the complexity of the decision boundary decreases with depth. This behaviour is displayed regardless of activation function (Fig. 5.12) and of architecture (Fig. 5.11).

Note that the topological properties of the decision boundary (and classes) are changing. This can only happen when L_i is **not** a homeomorphism, such as in the case of ReLU activation function. However, we have seen that this non-homeomorphic behaviour also happens with functions such as $tanh$ and $sigmoid$ because computers do not compute exact homeomorphisms.

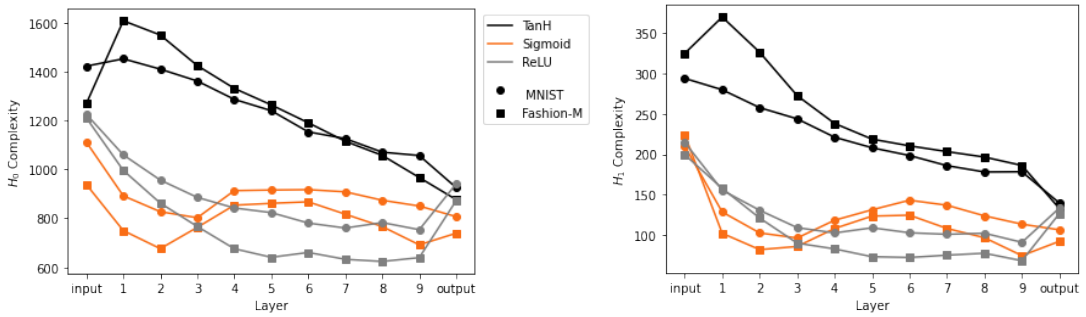
Naitzat et al. [78] argue that since Neural Networks work by disentangling the classes then non-homeomorphic transformations, such as ReLUs, are preferable for network training. They also state it as the reason for their superiority over homeomorphic transformations such as sigmoids or hyperbolic tangents.

Naitzat et al. [78] also affirm that a drop in floating point precision separate *tanh* and other homeomorphic transformations further from a homeomorphism. Stating that they *"(...) suspect that this may account for the paradoxical superior performance of lower precision arithmetic in deep Neural Networks [30, 47, 53]."*



(a) Standard Filtration (b) UDiPH

Figure 5.11: Topological complexity H_1 of the decision boundary in different layers of Neural Networks of different architectures (each row). The number of neurons in each layer is represented by the grey bars. Each orange line corresponds a Neural Network trained to 95% accuracy on the MNIST dataset, the black line represents the median of 30 runs. Note that when using standard Vietoris-Rips filtration, the persistence homology depends on the metric, as such persistence values are higher in higher dimensional spaces. This explains why the number of neurons heavily influences the persistence values (**left**). UDiPH creates a metric-invariant simplicial filtration allowing us to safely observe the correct trend. (**right**).



(a) H_0 Topological Complexity

(b) H_1 Topological Complexity

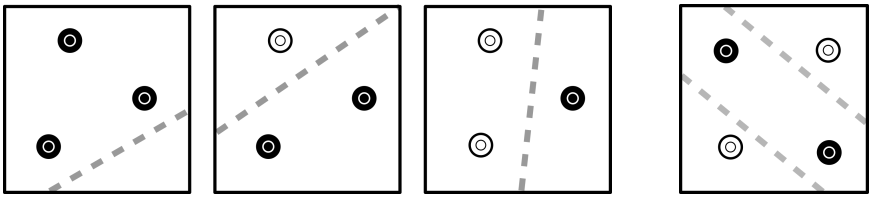
Figure 5.12: Topological complexity (y-axis) of the decision boundary of two classification problems: MNIST (circles), Fashion-MNIST (squares). Each color represents one different activation function.

5.4 Topological Expressiveness of Neural Networks

5.4.1 Previous Measures of Expressive Power

Given a Neural Network, how many different problems can it solve? This important and open question in deep learning is usually referred to as the problem of the expressive power of a Neural Network. Former research tackled this issue through either statistical or geometrical methods. We propose a new method based on a **topological** perspective.

The earliest research on expressive power has been through the scope of Vapnik–Chervonenkis dimension (VC-dimension) (Vapnik and Chervonenkis [104]). It is defined over a family of functions $\mathcal{F}$ defined over some set $D \to \{0,1\}$. If **every** function $h : D \to \{0,1\}$ can be emulated by a function $f \in \mathcal{F}$ then the VC-dimension of $\mathcal{F}$ is equal to the cardinality of D.



(a) (b)

Figure 5.13: A family of functions that can only represent a straight line can solve any permutation of 3 points (a), but fail at 4 points (b). It's VC-Dimension is equal to 3.

By considering the family $\mathcal{F}$ to be the set of Neural Networks of a specific architecture, Karpinski and Macintyre [56] developed the first bounds for the VC-Dimension of Neural Networks. This work was later extended by Bartlett and Maass [8] who developed bounds for polynomial activated architectures. Since then these bounds have been improved until recently Bartlett et al. [7] proved that the VC-Dimension of an architecture with W weights and L layers and piecewise linear activations is $O(WL\log W)$.

More recently, and considering a different approach to neural expressive power, Montúfar et al. [74] showed that Neural Networks with piecewise linear activation functions, such as

ReLU, describe a piecewise-linear function by dividing the input space into linear regions. In so doing, they acquire the capacity to build complex decision boundaries (Fig. 5.14). In particular, the authors showed that the number of linear regions increases exponentially with the number of layers, leading to a natural measure of network expressive power.

Poole et al. [86] proposed an elegant and –currently the most influential– measure of expressive power. Building on the results of Montúfar et al. [74], they realized that as it passes through successive layers, the trajectory over a closed curve crosses exponentially more linear regions. They tracked the *trajectory growth* of a closed curve through successive layers of a deep Neural Network, showing that the length of such curve grows exponentially with network depth (Fig. 5.14). This observation has also some other relevant implications, for example, based on this, it is possible to predict how a perturbation inserted at a certain depth will propagate throughout the rest of the network.

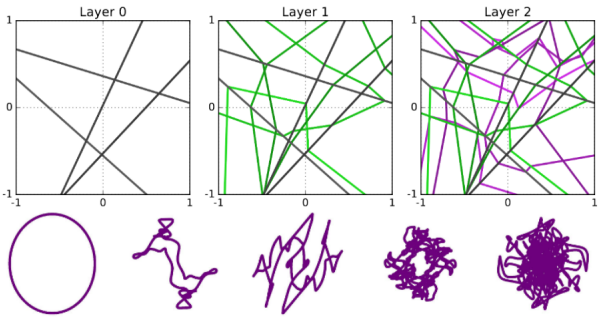


Figure 5.14: Illustration of increase of linear regions defined by successive layers of a Neural Network as described by Montúfar et al. [74], along with an illustration of the trajectory growth on of a circle along successive layers (bottom). Images from Poole et al. [86].

Interestingly, another concept related to expressive power is that of capacity of Neural Network. This has been studied since the early days of Neural Networks: Cover [31] and Gardner and Derrida [44] explored the architecture's capacity to memorize a certain number of uncorrelated samples; more recently, Collins et al. [29] explored the same capacity-per-parameter for RNNs, and Baldi and Vershynin [6] approach an architecture's capacity as the logarithm of the cardinality of the set of functions that can be generated. However, there is an important difference between expressive power and capacity. The former quantifies the number and breadth of different problems that a given architecture could solve. The latter focuses on predicting what architectures –typically minimal ones– can solve a given problem. In the first the focus is on the network, in the second it is on the problem. Note that all the approaches described so far are inherently geometrical.

5.4.2 Topological Approach

5.4.2.1 Motivation

Informally, we evaluate an architecture's expressive power by considering how many and how different, in terms of topology, are the decision boundaries it can generate. Under the

assumption that more expressive architectures are going to generate more diverse and complex decision boundaries.

We use the word **architecture** of a Neural Network in its usual sense, that is, to refer to the number and type of layers, the number of neurons per layer, the activation functions, presence of bias etc. An architecture $\mathcal{F}$ is a family of Neural Networks of fixed number of parameters n. Each element in $\mathcal{F}$ can be identified uniquely by a vector in $\mathbb{R}^n$, it is trivial to see that there is a direct mapping $\phi : \mathbb{R}^n \to \mathcal{F}$. Let $\phi : \mathbb{R}^n \to \mathcal{F}$ be the function that maps a vector of parameters to a Neural Network. We denote by $f_w = \phi(w) \in \mathcal{F}$ the Neural Network corresponding to the parameter vector $w \in \mathbb{R}^n$. Whenever the parameter vector is irrelevant or obvious from the context we will denote just by f. Consider the following example, take two architectures $\mathcal{F}_0$ and $\mathcal{F}_1$. Let $\mathcal{F}_0 = \{\sigma(w_0 x + w_1 y + w_2) \mid (w_0, w_1, w_2) \in \mathbb{R}^3\}$ where σ is the sigmoid function, and $\mathcal{F}_1$ be a more complex network with one hidden layer of multiple neurons. Obviously, **for all** $w \in \mathbb{R}^3$ the decision boundary of $\phi(w) = f_w \in \mathcal{F}_0$ **is a straight line** (Fig.5.15 (top)). While maybe different geometrically, every element of $\mathcal{F}_0$ produces a decision boundary with equal topology. The same cannot be said regarding $\mathcal{F}_1$ (Fig.5.15 (bottom)).

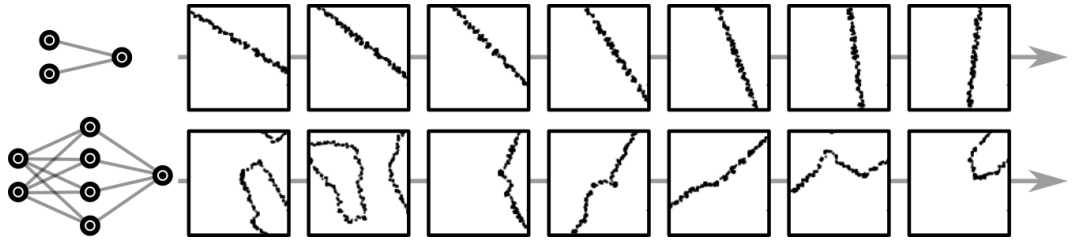


Figure 5.15: The decision boundaries of two paths in the parameter space of two different architectures. A very simple one $\mathcal{F}_0$ (top row) and one with one hidden layer (bottom row).

5.4.2.2 Construction

Our method starts by building a metric space for a given architecture $\mathcal{F}$. The elements of this metric space are the persistence diagrams of decision boundaries of elements of $\mathcal{F}$. Each element of $\mathcal{F}$ is a Neural Network obtained by sampling the parameter space (Fig. 5.16). Along with the $p-$Wasserstein distance between the persistence diagrams this creates a metric space. Following the above intuition, where an architecture is not very "versatile" if it can only produce straight lines, the inverse would be true. A Neural Network's architecture can be then characterized by how "topologically diverse" the set of all its decision boundaries is. We develop a novel pipeline for evaluating the topological expressive power of a Neural Network's architecture $\mathcal{F}$, Fig. 5.17.

1. **Sample** $\mathcal{F}$. Let n be the number of parameters in $\mathcal{F}$. We start by sampling a set of parameter vectors $w_i \in \mathbb{R}^n$. Since each parameter vector describes a Neural Network we consider the set of Neural Networks $\{f_i \in \mathcal{F} \mid f_i = \phi(w_i)\}$.

2. **Compute decision boundary** of each Neural Network in the sampled set $\{f_i \in \mathcal{F} \mid f_i = \phi(w_i)\}$

3. **Evaluate Homology** of the decision boundary of each sampled element of $\mathcal{F}$. We denote by P the set of all these persistence diagrams. That is $\mathcal{D}_i \in P$ is the dimension k persistence diagram of the decision boundary of the Neural Network $f_i = \phi(w_i) \in \mathcal{F}$ whose parameter vector is $w_i \in \mathbb{R}^n$ (Fig. 5.16).

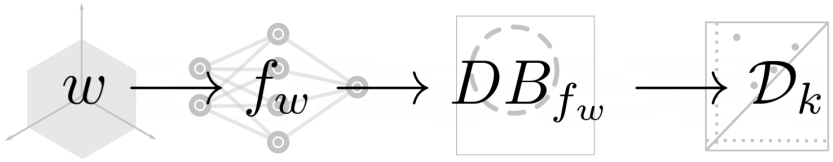


Figure 5.16: Illustration of the pipeline to obtain the persistence diagram of the decision boundary of one element belonging to some architecture $\mathcal{F}$

4. **Compute metric space** Let P be the set of persistence diagrams and W_p the p-th order Wasserstein distance, then (P, W_p) is a metric space. For every Neural Network architecture we can consider the metric space (P, W_p) associated with the persistent diagrams of its decision boundaries (Fig. 5.18 and Fig.5.17). Intuitively, the information that this metric space encodes is how topologically different are the decision boundaries of a specific architecture, given an uniform sample.

5. **Measure the Spread** of the metric space (Willerton [109]).

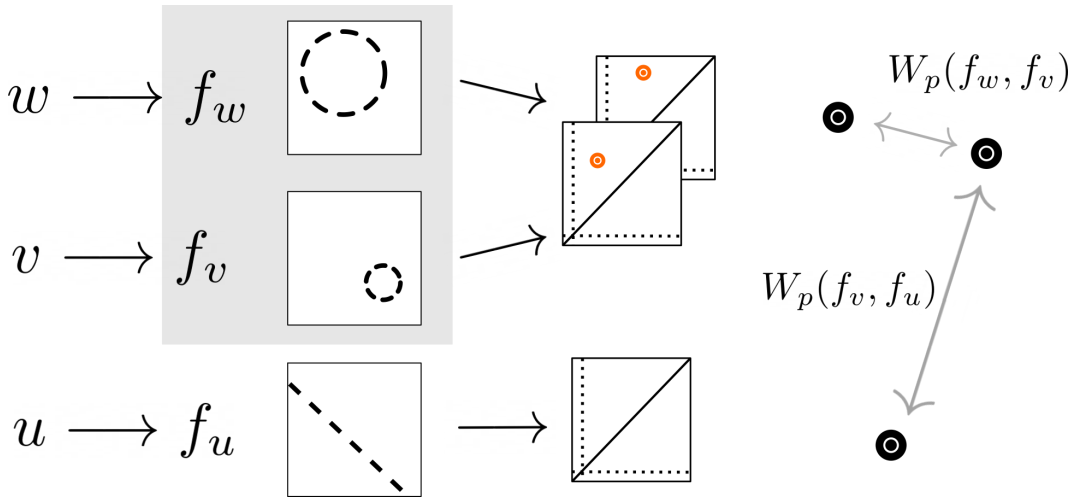


Figure 5.17: Illustration of the pipeline for evaluating the topological expressive power of an architecture. We sample parameter vectors w, u, v. From the resulting Neural Networks f_w, f_u and f_v we compute the persistence diagram of their decision boundaries and create a metric space.

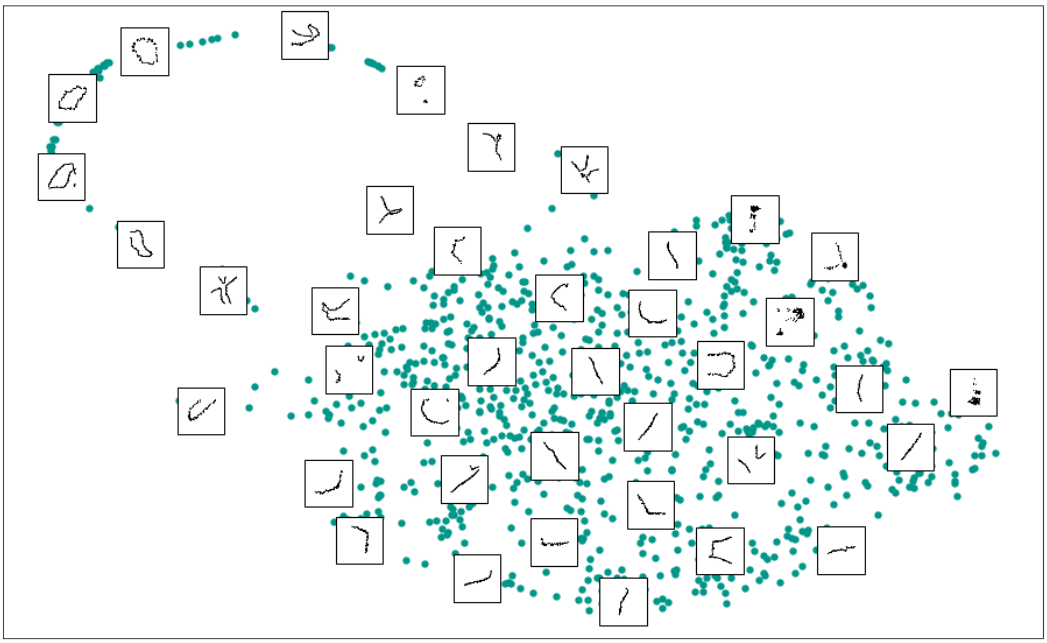


Figure 5.18: Embedding of the metric space (P, W_2) corresponding to a specific architecture. The metric space is composed of the persistent diagrams of the decision boundaries of a Neural Network along with the Wasserstein distance. The architecture has two hidden relu-activated layers of 10 neurons. The spread of this metric space is our measure of the topological expressive power of this architecture.

We now formalize the idea of "diversity" and "spread" of a metric space, that we have been using loosely above. Willerton [109] defines the notion of **spread** of a metric space. The definition is based on the idea of diversity on a measure space. It is defined as follows,

Definition 5.4.1 (Spread)**.** Given a (X, d) a metric space we define spread by

$$E_0(tX) = \sum_{x \in X} \left(\sum_{y \in X} e^{-td(x,y)} \right)^{-1} \tag{5.21}$$

EXAMPLE 11. The measure is aptly named and is intuitive to understand. A sample of points in a straight line has a lower value of spread E_0 than a sample of points uniformly in a plane. Through spread Willerton [109] manages to convey a sense of dimension within metric spaces.

Theorem 5.4.2. *For a metric space (X, d) where X has N elements, we have the following properties:*

- $t_0 \leq t_1 \Rightarrow E_0(t_0 X) \leq E(t_1 X)$
- $\lim_{t \to 0} E_0(tX) = 1$
- $\lim_{t \to \infty} E_0(tX) = N$

5.5 Results and Discussion

5.5.1 Results

In the following results we restrict ourselves to fully connected dense Neural Networks, with parameters defined in the unitary hypercube in $\mathbb{R}^n$. That is, we uniformly sample a set of parameter vectors $\{w_0, w_1 \dots w_k \mid w_i \in [-1,1]^n\}$ and consider the set of Neural Networks $\{f_i \in \mathcal{F} \mid f_i = \phi(w_i)\}$.

For each and every architecture, we uniformly sample 2000 vectors w_i in the interval $[-1,1]^n \subseteq \mathbb{R}^n$. For each parameter sample vector w_i we consider the Neural Network $f_{w_i} = \phi(w_i)$. With the exception of subsection 5.5.1.5 all architectures have input dimension 2, such that an analysis of only H_0 and H_1 homology classes is exhaustive. All architectures have output dimension 1, since (w.l.g.) we will only explore binary classification. All layers have *ReLU* activation except for the output layer which has *sigmoid* activation function, as per usual for binary classification problems.

5.5.1.1 Stability of Spread

The first step is to make sure that the spread measure is intrinsic to the architecture and not dependent on the number of points. Meaning that it remains constant no matter how many points are sampled, or at least converge to a stable value. Fig. 5.19 plots the spread values for two different architectures: one shallow and wide (79, 59 neurons), another very deep (44, 34, 33, 24, 18, 17,15, 13, 6) for different sample sizes.

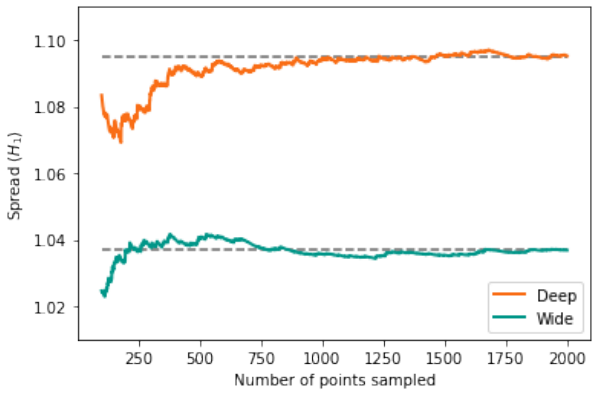


Figure 5.19: Spread of two metric spaces corresponding to a shallow and wide architecture and a deep one. The x-axis represents the number of points sampled and used to calculate the spread.

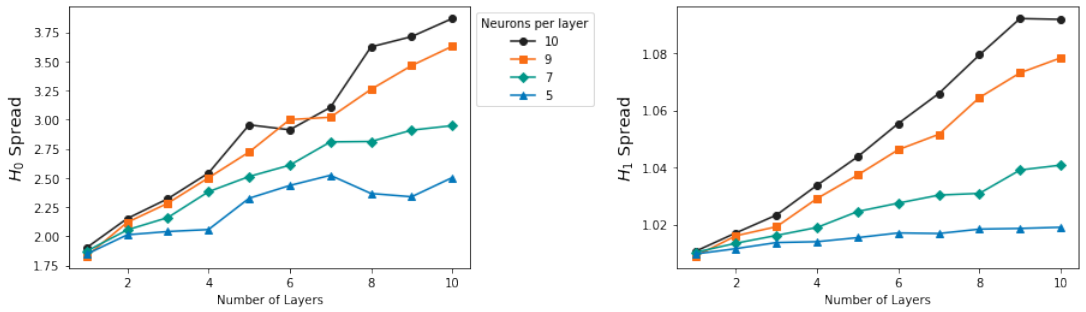
Note that the spread remains relatively stable specially when more than 800 points are sampled. Note that even though both architecture have the **same number of parameters**[1] the wide architecture displays a higher value of spread. Thus confirming that this architecture is more topologically expressive than a shallow and wide one.

[1] The reader is pointed to Appendix B for details.

5.5.1.2 Depth and Width

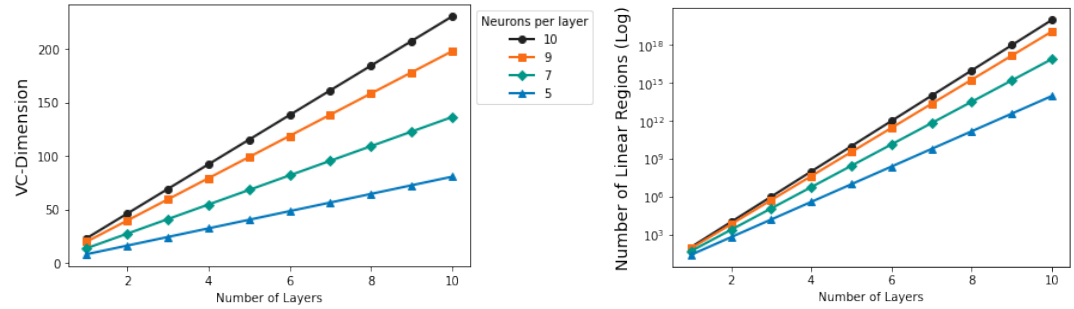
To observe the effect of depth and width in the topological expressiveness power we measured the spread of 4 sets of 10 different architectures Fig. (5.20). Each set consists of 10 architectures with increasing number of layers 1-10. The number of neurons in each layer is constant for each set. For example the first set has Neural Networks with layers 1 to 10 all with 5 neurons: $(2, 5, 1)$, $(2, 5, 5, 1)$, $(2, 5, 5, 5, 1)\ldots$

We compare our observations with the previous measures of expressive power of Neural Network architectures Fig. 5.23, the VC-Dimension [7] and the number of linear regions [74, 86] on the same architectures.



(a) H_0 Spread (b) H_1 Spread

Figure 5.20: Spread values for Neural Networks as a function of their width (number of neurons) and depth (number of layers).



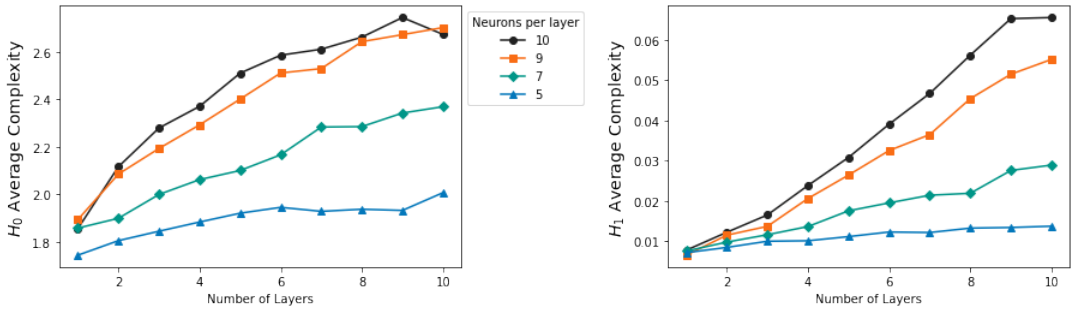
(a) VC-Dimension (b) Number of linear regions

Figure 5.21: Comparison with previous measures of expressive power. a) The VC-Dimension of the previous architectures as computed by Bartlett et al. [7]. b) The upper bound of the number of linear regions expressed by the same architectures as computed by Montúfar et al. [74] and Poole et al. [86], note the logarithm scale on the y-axis

It should come as no surprise to see that the spread increases with both depth and width (Fig. 5.20). We observe a linear growth with respect to the number of layers in accordance with the VC-Dimension bounds by Bartlett et al. [7] (Fig. 5.23 (a)) and in contrast with the exponential increase in the upper bound of linear regions as found Poole et al. [86] and Montúfar et al. [74] (Fig. 5.23 (b)).

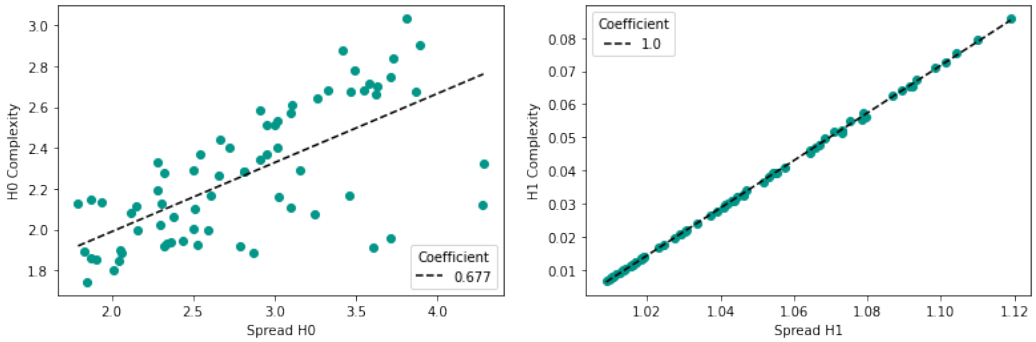
5.5.1.3 Spread and Complexity

In Chapter 4 we introduced Topological Complexity as a summary of a persistence diagram. In Fig. 5.22 instead of considering the spread we take the average Topological Complexity of each of the 2000 sampled decision boundaries of each architecture. Note that for the spread we require the pairwise Wasserstein distance between each diagram but for the Topological complexity we do not, making the latter much faster to compute.



(a) H_0 (b) H_1

Figure 5.22: Average topological complexity over the 2000 sampled decision boundaries for each architecture. Notice the correlation with the spread values in Fig. 5.20



(a) H_0 (b) H_1

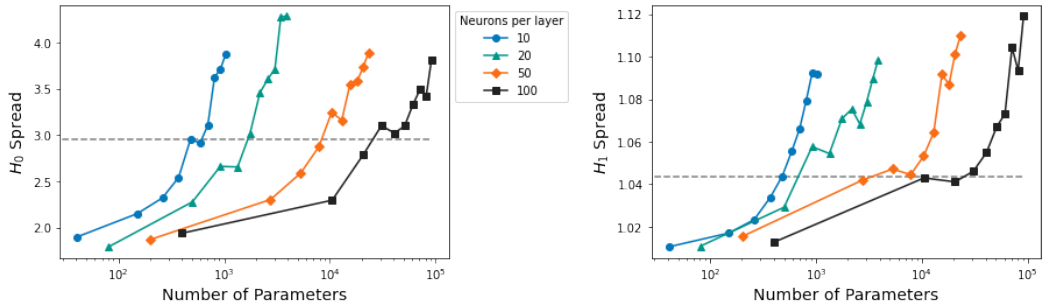
Figure 5.23: Average topological complexity versus spread. (Pearson's correlation value in the legend)

Intuitively, the average topological complexity represent the expected topological complexity that a decision boundary sampled from an architecture will have. According to Fig. 5.22, Neural Networks of more complex and expressive architecture also tend to describe more topologically complex decision boundaries.

Note the similarities between average topological complexity and the spread values in Fig. 5.20. Given the strong correlation between spread and the average topological complexity (Fig. 5.23) it would be interesting to study if Topological Complexity could be computed as a proxy for spread given its lighter computational complexity.

5.5.1.4 Fixed Number of Parameters

So far we have seen that the spread (our measure of topological expressive power) increases with depth and width. However, increasing depth and width also increases the number of parameters of an architecture. In Fig. 5.20 architectures have all different architectures but also different number of parameters.



(a) H_0 (b) H_1

Figure 5.24: How the spread (y-axis) changes with respect to the total number of parameters (x-axis). Each line represents architectures of 2 - 10 layers of 10, 20, 50 and 100 neurons each (colors). Note the logarithm scale on the x-axis. The grey line highlights the fact that there are Neural Networks with the same spread but vastly different number of parameters.

Note the gray line in the Fig. 5.24 and how architectures of different number of parameters have the same spread. The correlation between the number of parameters (weights and biases) with the number of layers and neurons is explored in Appendix B along with a method to sample all possible architectures given a fixed number of parameters. Following such method we sampled 200 different architectures **all with the same fixed number of parameters** (5000).

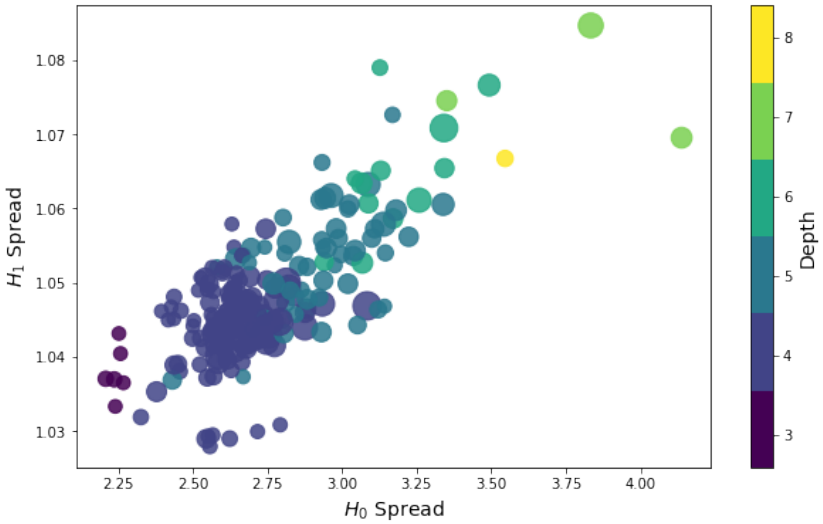
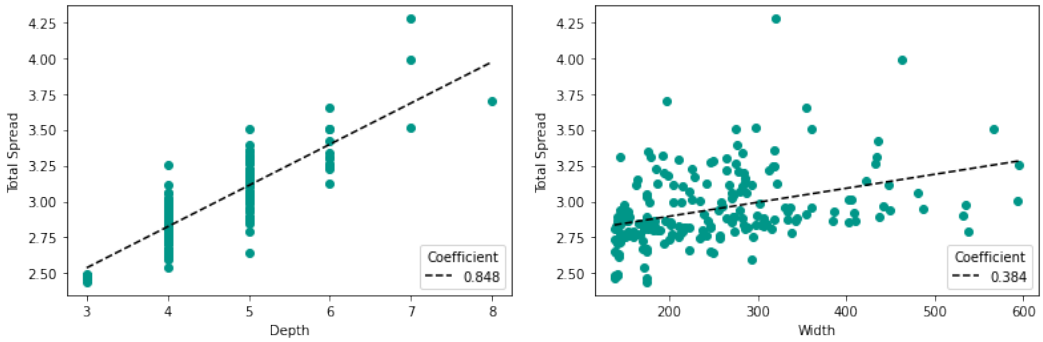


Figure 5.25: Spread values for 200 architectures with the same number of parameters (5000) along with their depth (color of each point) and width (size of each point).



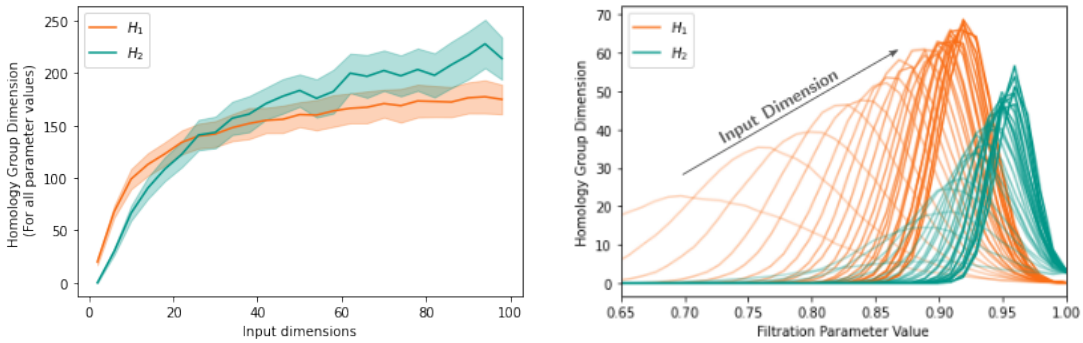
(a) Depth (b) Width

Figure 5.26: Pearson correlation values between the depth of a network (a) and width (b) with both H_0 and H_1 spread, where total spread is equal to $\sqrt{H_0^2 + H_1^2}$.

Combining spread in both homology dimensions, in Fig. 5.25 the most expressive architectures would be on the upper right. One could try to condense this information by taking the norm of each point (Fig. 5.26). The weaker correlation between width and spread can be justified by realizing that increasing the number of neurons directly increases the number of parameters in the network, which is not necessarily true with regarding the number of layers. Figure 5.25 and Fig. 5.26 (a) further cement the importance of depth as the strongest factor for expressive power of neural architectures, even when the number of parameters is fixed.

5.5.1.5 Input Dimension

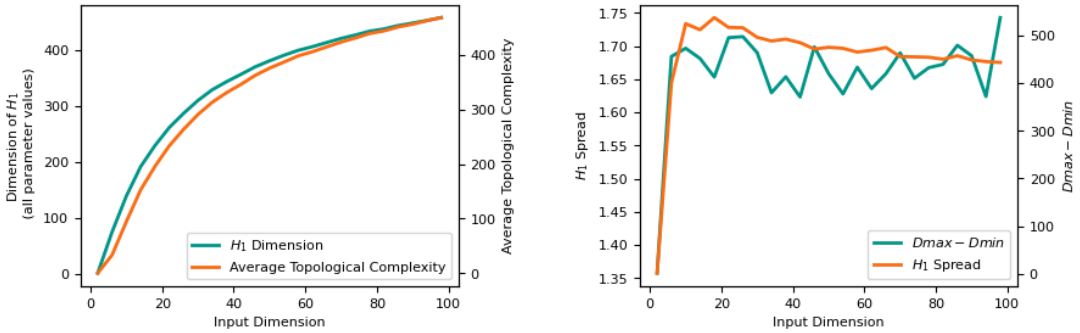
With respect to the input dimension one has to take into consideration its effect on two specific aspects. **First**, it is on homology itself: for higher dimensions, the dimension of each homology class also tends to increase as also observed in Giusti et al. [46] (Fig. 5.27). This means that concepts like the topological complexity which depend on the number of homology groups (as explained in Chapter 4) will follow the upwards trend caused by the increase of input dimensions (Fig 5.28 (a)).



(a) (b)

Figure 5.27: Effect of input dimension on the total number of homology classes H_1 and H_2. For each dimension we generated 100 samples of 1000 randomly generated points. (b) show the number for each value of the filtration parameter, while (a) shows the sum over all values.

Secondly, as already discussed in the previous section, using the ambient metric as a filtration parameter yields persistence diagrams that are not comparable between spaces of different dimensions. An extensive study on the behavior of metrics in high dimensional spaces can be found in Aggarwal et al. [2]. This in turn makes spread not comparable since it is computed using the Wasserstein Distance between persistence diagrams (Fig 5.28 (b)). Contrary to the first, issue we can solve this problem by using UDiPH to create a filtration (Fig. 5.29).

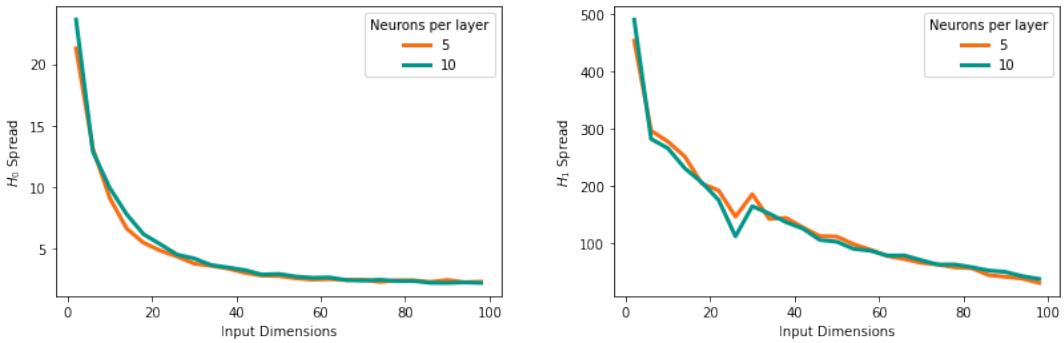


(a) (b)

Figure 5.28: (a) Influence of number of homology classes on topological complexity. (b) How the behaviour of the Euclidean metric influences the spread for different dimensions.

In their paper, Aggarwal et al. [2], describe a method to visualize the behaviour of different metrics on higher dimensional spaces. Given a set of points $\{w_i\}$ in $\mathbb{R}^n$ we measure the difference of $D_{max} = \max_i \|w_i\|$ and $D_{min} = \min_i \|w_i\|$, that is the difference between the closest point to the origin and the farthest. In Fig. 5.28 (b) we use this very same approach to measure the behaviour of the Euclidean metric. In our case the set $\{w_i\}$ is the set of points sampled from the decision boundary, which is described by the input space.

Based on Fig. 5.28, topological complexity does not serve as an accurate proxy for spread and the use of UDiPH becomes imperative.



(a) H_0 Spread (b) H_1 Spread

Figure 5.29: Spread values for Neural Networks with different input dimensions. Both Neural Networks have 5 layers of 5 and 10 neurons each. The persistence diagrams were computed using UDiPH so that the spread values are comparable.

It is fundamental to realize that, as a corollary of Fig. 5.27, increasing input dimensions results in an increase in cardinality of all homology groups, in particular higher dimensional ones. For example, data coming from $\mathbb{R}^2$ normally does not display relevant H_2 classes, while data say in $\mathbb{R}^5$ not only might have a sizeable H_2 class but also H_3 and H_4. Note that in Fig. 5.29 we are only taking into account H_0 and H_1. Although we observe a decrease the spread in H_0 and H_1 with respect to an increase in input dimensions, it is reasonable to assume that architectures with higher input dimension will present greater spread in higher dimensional homology groups such as H_2 and H_3, as discussed above, but are not present in the graph. Meaning that, as the input dimensions increase, spread will be lower in each homology dimension but will be present in more dimensions.

The decrease of spread observed in Fig. 5.29 might be a consequence of an analysis limited to just H_0 and H_1. One could conjecture that each architecture has a "constant spread" summed over all dimension and, the more dimensions it has available, the smaller is the spread allocated for each one. A further analysis for H_2 and H_3 would be relevant, however, given the sizeable number of persistence diagrams we calculate per architecture (2000) it would require a tremendous computational effort.

5.5.2 Discussion

Spread grows with network depth and width. We first investigated how the spread depended on the structural properties of $\mathcal{F}$. We studied architectures with two input neurons, a fixed number of neurons per layers (width, $5, 7, 9, 10$) and increasing number of layers (depth). For each architecture, we sampled 2000 points from the corresponding $\mathbb{R}^n$, where n is the number of parameters of $\mathcal{F}$. We find that spreads for both H_0 and H_1 grow monotonically with the number of layers, with the slope monotonically increasing with the number of neurons per layer too (Fig. 5.20). It seems to follow the behaviour of VC-Dimension rather than the exponential growth observed in other geometric properties.

Spread is summarised by complexity. For the same type of architectures, we also ask how the average topological complexities for H_0 and H_1 grow with depth and width (total number of neurons) (Fig. 5.22). We find that complexity too increases with both quantities, similarly to spread (Fig. 5.22). In addition, we also find that complexity and spread correlate strongly with each other (Fig. 5.23). This might perhaps appear unsurprising, but we believe it to be interesting, because it implies that architectures producing richer topological can produce many different types of richer topologies. That is, it is not only that more complex topologies have more ways in which to be different, but that they actually take all these ways, increasing the spread. As a final comment, complexity is computationally much lighter to compute than spread. In fact, computing the spread is quadratic to the number of persistence diagrams while computing the average complexity is linear. The latter therefore constitutes a promising proxy observable in cases where computing spread becomes prohibitive.

Spread depends weakly on the total number of parameters. We found that the total number of parameters in networks with fixed number of neurons (width) and increasing

number of layers (depth) is indeed correlated with the topological measures (spread and complexity), but surprisingly the correlation is weaker than with other architectural quantities, like n_w and l. This is surprising because it implies that the expressive power of a network can be increased more by choosing its structure carefully than just by adding more degrees of freedom. For instance, the architectures $(10, 10, 10, 10, 10)$, $(20, 20, 20)$, $(50, 50)$ and $(100, 100)$ all display the same H_1 Spread (Fig. 5.24). To investigate further this point, we sampled 200 arbitrary architectures with the only constrain: that the total number of parameters was fixed to $n = 5000$. Note that in this case we also allowed networks with variable number of neurons, which also included autoencoder-like bottlenecks and other complex architectures. Appendix B explores and explains how the architecture is related to the number of parameters. We find again that depth correlates with spread and complexity and so does –more weakly but significantly– the total number of neurons, showing that even at fixed number of parameters, different architectures can have very different expressive power.

More input dimensions, less low-dimensional spread. Using UDiPH we were able to compare spread from different input dimensions. We observed that H_0 and H_1 drastically decrease with the increase in input dimensions (Fig. 5.29). In accordance with the literature (Fig.5.27) we also observed an increase in the number of homology classes as the input dimension increases. Together these observations suggest that the topology of the decision boundaries is getting more complex but also more similar, as the input dimension rises. This, joined by the fact that higher input dimensions also provide higher-dimensional homology groups, points us to the possibility that while the spread lowers for low dimensional homologies groups, it also spreads over more homology dimensions.

5.5.3 Open problems and future directions.

The results reported here are interesting but leave many questions open. Mainly regarding:

1. **Classification Problems and Decision Boundaries:** It would be interesting to relate the topological complexity of a problem's decision boundary with the average accuracy achieved through different machine learning algorithms. More specifically, to expand the experiment described in Fig. 4.10 for more problems (preferably real world examples) and different state-of-the-art classification methods such as Gradient Boosting. The objective would be to confirm topological complexity of decision boundaries as a fundamental measure of intrinsic difficulty of a classification problem.

2. **Voronoi Diagram sampling method:** A methods for fast calculation of high-dimensional Voronoi Diagrams opens the possibility for new insights in areas where the Voronoi Diagram plays a fundamental role, not just in Topology (Edelsbrunner et al. [38]) but also in Computational Geometry (Wolfram and Media [110]), Machine Learning (Mitchell [72]), Material Science (Mulheran and Blackman [75]), Biology (Bock et al. [13]) and even Urban Planning (Lopez et al. [63]) among many others.

3. **Neural Networks:** The proposed measure of topological expressiveness has theoretical understanding in mind but further practical applications should follow. More specifically for architecture selection. Furthermore a future direction would be understand how does spread relate with accuracy and how do other architecture aspects (such as activation function and the presence of a bottleneck) affects spread.

4. **Uniform Distributed Persistent Homology:** UDiPH provides an early approach towards finding a topological summary that is invariant under homeomorphisms. In its current iteration it requires a bit more work in the theory as well as the implementation. More precisely it requires the proof of a result equivalent to: "*The neighbour structure of a uniformly distributed set of points in a Riemannian Manifold is a Topological invariant*". Nevertheless, the preliminary results show promise and enable us to study the Topological impact of not only Neural Networks but of arbitrary functions in general.

5. **Topological Expressive Power:** The metric space created by taking the persistence diagrams of decision boundaries along with the Wasserstein distance between them, seems to capture intrinsic properties of Neural Architecture that are cannot described by just the spread. For example notice the rich topology visible in Fig. 5.18. Further work would be on extracting other descriptors from these spaces (such as persistent homology) aside from only the spread.

   The main limitation of this work is the computational cost of calculating higher dimensional homology dimensions. A more thorough understanding could have been achieved if we had considered homology groups beyond H_0 and H_1. This is an ongoing research direction in Topological Data Analysis.

6. **Broader implementations:** Our proposed pipeline for evaluating the topological expressive power is not bound to Neural Networks. It could be adapted to explore other models such as Random Forests or Decision Trees. It requires only a map $f : \mathcal{F} \to \mathbb{R}^n$ from a family of models to a space of parameters. Finally, it would be interesting to investigate how this construction could be generalized to other types of problems that are not strictly classifications.

Chapter 6

Conclusion

I don't exactly know what I mean by that, but I mean it.

-J. D. Salinger (The Catcher in the Rye)

This thesis provides a novel pipeline for the topological understanding of the expressive power of Neural Networks. Being a convoluted task, several other novel tools and approaches had to be crafted to accommodate this thesis's objective. Namely the creation of an efficient way to sample subsets of the Voronoi Diagram (the decision boundaries) in arbitrarily high dimensions, an adequate measure of topological complexity and an original simplicial filtration for finite metric spaces along with a general new perspective.

This work's primary objective was to bridge the areas of Topological Data Analysis (TDA) and Neural Networks. The tools provided and results therein obtained not only achieve that goal but further cement Topological insight as an unalienable step in Machine Learning and Data Analysis.

Taken as a whole, this project builds its path on an ongoing migration to stronger broader methodologies, such as Topological Data Analysis and Differential Geometry, to accommodate an increase in complexity of implementations such as Neuroscience and Complex Systems analysis. In practice this insight can be beneficial in certain applications such as:

1. Situations where the increase in model complexity has escalated the opportunity cost of training. To avoid the cost of suboptimal models, in these scenarios it is critical to know beforehand which architectures are more expressive given a fixed allocation of computational resources.

2. Combat task-specific models and aim for the development of architectures and networks that are adaptive for multiple tasks. Models that can, with minimal training, be shifted towards a different problems without requiring a complete redesign.

Taken in parts, the impact of this thesis on the ongoing research areas of Machine Learning, Neural Networks and Topological Data Analysis is not reserved to the final pipeline. The ability to efficiently sample subsets of Voronoi-Diagram can enable many new insights in areas where the Voronoi Diagram plays a fundamental role: not just in Topology (Edelsbrunner et al. [38]) but also in Computational Geometry (Wolfram and Media [110]), Machine Learning (Mitchell [72]), Material Science (Mulheran and Blackman [75]), Biology (Bock et al. [13]) and even Urban Planning (Lopez et al. [63]) among many others . Along the same lines UDiPH enables topological insight of functions without requiring multi-parameter simplicial filtrations.

Chapter 4 introduces an original method to sample and further study the decision boundary of any classification problem and formalizes the idea of entanglement: topologically complex decision boundaries. Chapter 5 develops an essential tool that enables the transition of standard TDA methods from data points to functions. It further provides new results and confirmation of previous attempts at formalizing topological understanding of Neural Networks.

The developed tools enable the formalization and analysis of topological expressiveness of Neural Networks. In the end of Chapter 5 this work provides a pipeline along with results. The impact of this thesis on the past work and future directions of the scientific community can be summarised as follows:

Opens-up the door for using standard Topological Data Analysis (TDA) methods such as persistent homology to evaluate functions in general and Neural Networks in particular. Although it is an active research field in Discrete Morse Theory (Ghrist [45]), this thesis provides a strong approach without demanding multi-parameter filtrations (Cohen-Steiner et al. [27], Edelsbrunner and Harer [37]).

Provides an outlook into Neural Networks study at an architecture level. An analysis that is inherent to the architecture and completely agnostic of a given application or problem instance. The vast majority of studies rely on specific instances such as Rieck et al. [90] and even though works such as Bianchini and Scarselli [11] and Baldi and Vershynin [6] also work under this perspective, this thesis is the first to provide an empirical workflow.

Corrects previous approaches namely Ramamurthy et al. [87]'s work on the persistent homology of decision boundaries by introducing a new definition along with a faster and scalable algorithm. It also amends Brahma et al. [15] concept of disentanglement.

Validates Olah [80] perspective and Naitzat et al. [78] experiments on neural networks effect in manifold disentanglement, whose shortcomings in methodology prevented the validation of their statements.

Challenges the most prevalent perspective that the complexity of classification problems being based on data complexity. By taking the topological complexity of decision boundary this work disputes results such as Guss and Salakhutdinov [48] regarding classification of data complexity.

Translates the perspective of many previous geometric results regarding expressiveness of Neural Networks such as Montúfar et al. [74] and Poole et al. [86] to Topological.

Bibliography

[1] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. "Persistence Images: A Stable Vector Representation of Persistent Homology." In: *Journal of Machine Learning Research* 18.8 (2017), pp. 1–35. URL: http://jmlr.org/papers/v18/16-337.html.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. "On the Surprising Behavior of Distance Metrics in High Dimensional Space." In: *Database Theory — ICDT 2001*. Ed. by J. Van den Bussche and V. Vianu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 420–434. ISBN: 978-3-540-44503-6.

[3] A. Ansuini, A. Laio, J. H. Macke, and D. Zoccolan. "Intrinsic dimension of data representations in deep neural networks." In: *CoRR* abs/1905.12784 (2019). arXiv: 1905.12784. URL: http://arxiv.org/abs/1905.12784.

[4] N. Atienza, R. Gonzalez-Díaz, and M. Rucco. "Separating Topological Noise from Features using Persistent Entropy." In: (May 2016).

[5] F. Aurenhammer and R. Klein. *Voronoi Diagrams*. Informatik-Berichte. Karl-Franzens-Univ. Graz & Techn. Univ. Graz, 1996. URL: https://books.google.pt/books?id=27vkSgAACAAJ.

[6] P. Baldi and R. Vershynin. *The capacity of feedforward neural networks*. 2019. arXiv: 1901.00434 [cs.LG].

[7] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. "Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks." In: *Journal of Machine Learning Research* 20.63 (2019), pp. 1–17. URL: http://jmlr.org/papers/v20/17-612.html.

[8] P. L. Bartlett and W. Maass. "Vapnik-Chervonenkis dimension of neural nets." In: *The handbook of brain theory and neural networks* (2003), pp. 1188–1192.

[9] U. Bauer. *Ripser: efficient computation of Vietoris-Rips persistence barcodes*. 2021. arXiv: 1908.02518 [math.AT].

[10] J. Bernstein, A. Vahdat, Y. Yue, and M.-Y. Liu. *On the distance between two neural networks and the stability of learning*. 2020. arXiv: 2002.03432 [cs.LG].

[11] M. Bianchini and F. Scarselli. "On the complexity of neural network classifiers: A comparison between shallow and deep architectures." In: *IEEE transactions on neural networks and learning systems* 25.8 (2014), pp. 1553–1565.

[12] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. "Size functions for comparing 3D models." In: *Pattern Recognition* 41.9 (2008), pp. 2855–2873. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2008.02.003. URL: https://www.sciencedirect.com/science/article/pii/S0031320308000733.

[13] M. Bock, A. K. Tyagi, J.-U. Kreft, and W. Alt. *Generalized Voronoi Tessellation as a Model of Two-dimensional Cell Tissue Dynamics*. 2009.

[14] J.-D. Boissonnat and A. Ghosh. "Manifold reconstruction using tangential Delaunay complexes." In: *Discrete and Computational Geometry* 51.1 (Jan. 2014), pp. 221–267. DOI: 10.1007/s00454-013-9557-2. URL: https://hal.inria.fr/hal-00932209.

[15] P. P. Brahma, D. Wu, and Y. She. "Why deep learning works: A manifold disentanglement perspective." In: *IEEE transactions on neural networks and learning systems* 27.10 (2015), pp. 1997–2008.

[16] G. Bredon. *Topology and Geometry*. Graduate texts in mathematics. Springer-Verlag, 1993. ISBN: 9783540979265. URL: https://books.google.pt/books?id=vsSjQgAACAAJ.

[17] D. Bremner, E. Demaine, J. Erickson, J. Iacono, S. Langerman, P. Morin, and G. Toussaint. "Output-sensitive algorithms for computing nearest-neighbour decision boundaries." In: *Discrete and Computational Geometry* 33.4 (2005), pp. 593–604.

[18] P. Bubenik. "Statistical Topological Data Analysis using Persistence Landscapes." In: *Journal of Machine Learning Research* 16.3 (2015), pp. 77–102. URL: http://jmlr.org/papers/v16/bubenik15a.html.

[19] P. Bubenik and A. Wagner. "Embeddings of persistence diagrams into Hilbert spaces." In: *Journal of Applied and Computational Topology* 4.3 (2020), 339–351. ISSN: 2367-1734. DOI: 10.1007/s41468-020-00056-w. URL: http://dx.doi.org/10.1007/s41468-020-00056-w.

[20] G. Carlsson. "Topology and data." In: *Bulletin of the American Mathematical Society* 46.2 (2009), pp. 255–308.

[21] G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian. "On the local behavior of spaces of natural images." In: *International journal of computer vision* 76.1 (2008), pp. 1–12.

[22] M. do Carmo. *Riemannian Geometry*. Mathematics (Boston, Mass.) Birkhäuser, 1992. ISBN: 9783764334901. URL: https://books.google.it/books?id=uXJQQgAACAAJ.

[23] M. Carrière and U. Bauer. "On the Metric Distortion of Embedding Persistence Diagrams into Reproducing Kernel Hilbert Spaces." In: *CoRR* abs/1806.06924 (2018). arXiv: 1806.06924. URL: http://arxiv.org/abs/1806.06924.

[24] A. Cerri, M. Ferri, and D. Giorgi. "Retrieval of trademark images by means of size functions." In: *Graphical Models* 68.5 (2006). Special Issue on the Vision, Video and Graphics Conference 2005, pp. 451–471. ISSN: 1524-0703. DOI: https://doi.org/10.1016/j.gmod.2006.07.001. URL: https://www.sciencedirect.com/science/article/pii/S1524070306000592.

[25] C. Chang. "Fuzzy topological spaces." In: *Journal of Mathematical Analysis and Applications* 24.1 (1968), pp. 182 –190. ISSN: 0022-247X. DOI: https://doi.org/10.1016/0022-247X(68)90057-7. URL: http://www.sciencedirect.com/science/article/pii/0022247X68900577.

[26] F. Chazal and B. Michel. *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*. 2017. arXiv: 1710.04019 [math.ST].

[27] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. "Stability of Persistence Diagrams." In: *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*. SCG '05. Pisa, Italy: ACM, 2005, pp. 263–271. ISBN: 1-58113-991-8.

[28] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov. "Vines and Vineyards by Updating Persistence in Linear Time." In: *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*. SCG '06. Sedona, Arizona, USA: ACM, 2006, pp. 119–126. ISBN: 1-59593-340-9. DOI: 10.1145/1137856.1137877. URL: http://doi.acm.org/10.1145/1137856.1137877.

[29] J. Collins, J. Sohl-Dickstein, and D. Sussillo. *Capacity and Trainability in Recurrent Neural Networks*. 2016. arXiv: 1611.09913 [stat.ML].

[30] M. Courbariaux, Y. Bengio, and J.-P. David. *Training deep neural networks with low precision multiplications*. 2014. arXiv: 1412.7024 [cs.LG].

[31] T. M. Cover. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition." In: *IEEE Trans. on Electronic Computers* EC-14 (1965), pp. 326–334.

[32] K. Crane, F. de Goes, M. Desbrun, and P. Schröder. "Digital Geometry Processing with Discrete Exterior Calculus." In: *ACM SIGGRAPH 2013 courses*. SIGGRAPH '13. Anaheim, California: ACM, 2013.

[33] D. Crevier. *Ai: The Tumultuous History Of The Search For Artificial Intelligence*. Basic Books, 1993. ISBN: 9780465029976. URL: https://books.google.pt/books?id=QJNQAAAAMAAJ.

[34] B. Di Fabio and M. Ferri. "Comparing Persistence Diagrams Through Complex Vectors." In: *Image Analysis and Processing — ICIAP 2015*. Ed. by V. Murino and E. Puppo. Cham: Springer International Publishing, 2015, pp. 294–305. ISBN: 978-3-319-23231-7.

[35] W. Dong, M. Charikar, and K. Li. "Efficient K-nearest neighbor graph construction for generic similarity measures." In: Jan. 2011, pp. 577–586. DOI: 10.1145/1963405.1963487.

[36] R. A. Dwyer. "Higher-dimensional Voronoi diagrams in linear expected time." In: *Discrete and Computational Geometry* 6.3 (1991), pp. 343–367.

[37] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010. ISBN: 9780821849255. URL: https://books.google.pt/books?id=MDXa6gFRZuIC.

[38] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. "On the shape of a set of points in the plane." In: *IEEE Transactions on information theory* 29.4 (1983), pp. 551–559.

[39] E. Facco, M. d'Errico, A. Rodriguez, and A. Laio. "Estimating the intrinsic dimension of datasets by a minimal neighborhood information." In: *Scientific reports* 7.1 (2017), p. 12140.

[40] M. Farber. "Topological Complexity of Motion Planning." In: *Discrete & Computational Geometry* 29 (Mar. 2003), pp. 211–221. DOI: 10.1007/s00454-002-0760-9.

[41] C. Fefferman, S. Mitter, and H. Narayanan. "Testing the manifold hypothesis." In: *Journal of the American Mathematical Society* 29.4 (2016), pp. 983–1049.

[42] D. Fernández-Ternero, E. Macías-Virgós, E. Minuz, and J. A. Vilches. *Discrete Topological complexity*. 2017. arXiv: 1706.02894 [math.AT].

[43] M. Gabella, N. Afambo, S. Ebli, and G. Spreemann. "Topology of Learning in Artificial Neural Networks." In: *CoRR* abs/1902.08160 (2019). arXiv: 1902.08160. URL: http://arxiv.org/abs/1902.08160.

[44] E Gardner and B Derrida. "Optimal storage properties of neural network models." In: *Journal of Physics A: Mathematical and General* 21.1 (1988), pp. 271–284. DOI: 10.1088/0305-4470/21/1/031. URL: https://doi.org/10.1088\%2F0305-4470\%2F21\%2F1\%2F031.

[45] R. Ghrist. *Elementary Applied Topology*. CreateSpace Independent Publishing Platform, 2014. ISBN: 9781502880857.

[46] C. Giusti, E. Pastalkova, C. Curto, and V. Itskov. "Clique topology reveals intrinsic geometric structure in neural correlations." In: *Proceedings of the National Academy of Sciences* 112.44 (2015), 13455–13460. ISSN: 1091-6490. DOI: 10.1073/pnas.1506407112. URL: http://dx.doi.org/10.1073/pnas.1506407112.

[47] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. *Deep Learning with Limited Numerical Precision*. 2015. arXiv: 1502.02551 [cs.LG].

[48] W. H. Guss and R. Salakhutdinov. "On characterizing the capacity of neural networks using algebraic topology." In: *arXiv preprint arXiv:1802.04443* (2018).

[49] A. Hatcher and C. U. Press. *Algebraic Topology*. Algebraic Topology. Cambridge University Press, 2002. ISBN: 9780521795401.

[50] J.-C. Hausmann. "On the Vietoris-Rips complexes and a cohomology theory for metric spaces." eng. In: ed. by F. Quinn. Prospects in topology : proceedings of a conference in honor of William Browder. ID: unige:12821. Princeton, N.J.: Princeton University Press, 1995, pp. 175–188. ISBN: 978-0-691-02729-6. URL: https://archive-ouverte.unige.ch/unige:12821.

[51] T. K. Ho. *Geometrical Complexity of Classification Problems*. 2004. arXiv: cs/0402020 [cs.CV].

[52] W. Huang and A. Ribeiro. "Persistent Homology Lower Bounds on High Order Network Distances." In: *IEEE Transactions on Signal Processing* PP (July 2015). DOI: 10.1109/TSP.2016.2620963.

[53] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. *Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations*. 2016. arXiv: 1609.07061 [cs.NE].

[54] P. Joharinad and J. Jost. "Topology and curvature of metric spaces." In: *Advances in Mathematics* 356 (2019), p. 106813. ISSN: 0001-8708. DOI: 10.1016/j.aim.2019.106813. URL: http://dx.doi.org/10.1016/j.aim.2019.106813.

[55] S. Kališnik. "Tropical Coordinates on the Space of Persistence Barcodes." In: *Foundations of Computational Mathematics* 19.1 (2018), 101–129. ISSN: 1615-3383. DOI: 10.1007/s10208-018-9379-y. URL: http://dx.doi.org/10.1007/s10208-018-9379-y.

[56] M. Karpinski and A. Macintyre. "Bounding VC-dimension for neural networks: progress and prospects." In: *European Conference on Computational Learning Theory*. Springer. 1995, pp. 337–341.

[57] T. Kato and T. Wada. "Direct condensing: An efficient Voronoi condensing algorithm for nearest neighbor classifiers." In: 3 (Sept. 2004), 474 –477 Vol.3. DOI: 10.1109/ICPR.2004.1334569.

[58] I. Kramosil and J. Michálek. "Fuzzy metrics and statistical metric spaces." eng. In: *Kybernetika* 11.5 (1975), (336)–344. URL: http://eudml.org/doc/28711.

[59] J. Latschev. "Vietoris-Rips complexes of metric spaces near a closed Riemannian manifold." In: *Archiv der Mathematik* 77 (Dec. 2001), pp. 522–528. DOI: 10.1007/PL00000526.

[60] T. Le and M. Yamada. *Persistence Fisher Kernel: A Riemannian Manifold Kernel for Persistence Diagrams*. 2018. arXiv: 1802.03569 [stat.ML].

[61] T. Leinster. "The magnitude of metric spaces." In: (2010). arXiv: 1012.5857 [math.MG].

[62] T. Leinster and E. Roff. *The maximum entropy of a metric space*. 2019. arXiv: 1908.11184 [math.MG].

[63] C. Lopez, C.-L. Zhao, S. Magniol, N. Chiabaut, and L. Leclercq. "Microscopic Simulation of Cruising for Parking of Trucks as a Measure to Manage Freight Loading Zone." In: *Sustainability* 11.5 (2019). ISSN: 2071-1050. DOI: 10.3390/su11051276. URL: https://www.mdpi.com/2071-1050/11/5/1276.

[64] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, and T. K. Ho. *How Complex is your classification problem? A survey on measuring classification complexity.* 2018. arXiv: 1808.03591 [cs.LG].

[65] K. Y. C. Lui, Y. Cao, M. Gazeau, and K. S. Zhang. "Implicit manifold learning on generative adversarial networks." In: *arXiv preprint arXiv:1710.11260* (2017).

[66] Y. A. Malkov and D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs." In: *CoRR* abs/1603.09320 (2016). arXiv: 1603.09320. URL: http://arxiv.org/abs/1603.09320.

[67] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." In: *arXiv preprint arXiv:1802.03426* (2018).

[68] M. W. Meckes. "Magnitude, Diversity, Capacities, and Dimensions of Metric Spaces." In: *Potential Analysis* 42.2 (2014), 549–572. ISSN: 1572-929X. DOI: 10.1007/s11118-014-9444-3. URL: http://dx.doi.org/10.1007/s11118-014-9444-3.

[69] F. Meunier and L. Montejano. "Different versions of the nerve theorem and colourful simplices." In: *Journal of Combinatorial Theory, Series A* 169 (2020), p. 105125. ISSN: 0097-3165. DOI: https://doi.org/10.1016/j.jcta.2019.105125. URL: http://www.sciencedirect.com/science/article/pii/S0097316519300998.

[70] Y. Mileyko, S. Mukherjee, and J. Harer. "Probability measures on the space of persistence diagrams." In: *Inverse Problems* 27.12 (2011), p. 124007.

[71] J. van Mill. *Infinite-Dimensional Topology. Prerequisites and Introduction.* 1st. North-Holland Mathematical Library 43. North Holland, 1988. ISBN: 0444871330,9780444871336. URL: http://gen.lib.rus.ec/book/index.php?md5=8ca549662db78125115d56e3261e85ce.

[72] T. Mitchell. *Machine Learning.* McGraw-Hill International Editions. McGraw-Hill, 1997. ISBN: 9780071154673. URL: https://books.google.pt/books?id=EoYBngEACAAJ.

[73] R. Mollineda, J. Sánchez, and J. Sotoca. "A meta-learning framework for pattern classification by means of data complexity measures." In: *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial, ISSN 1137-3601, Nº. 29, 2006 (Ejemplar dedicado a: Minería de Datos), pags. 31-38* 10 (Dec. 2006). DOI: 10.4114/ia.v10i29.875.

[74] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. *On the Number of Linear Regions of Deep Neural Networks.* 2014. arXiv: 1402.1869 [stat.ML].

[75] P. A. Mulheran and J. A. Blackman. "Capture zones and scaling in homogeneous thin-film growth." In: *Phys. Rev. B* 53 (15 1996), pp. 10261–10267. DOI: 10.1103/PhysRevB.53.10261. URL: https://link.aps.org/doi/10.1103/PhysRevB.53.10261.

[76] E. Munch. "Applications of persistent homology to time varying systems." In: (2013).

[77] J. Munkres. *Topology*. Featured Titles for Topology Series. Prentice Hall, Incorporated, 2000. ISBN: 9780131816299.

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

[79] P. Niyogi, S. Smale, and S. Weinberger. "Finding the homology of submanifolds with high confidence from random samples." In: *Discrete & Computational Geometry* 39.1-3 (2008), pp. 419–441.

[80] C. Olah. *Neural Networks, Manifolds, and Topology*. 2014. URL: https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/ (visited on 01/06/2020).

[81] B. O'Neill. *Semi-Riemannian Geometry With Applications to Relativity*. ISSN. Elsevier Science, 1983. ISBN: 9780080570570. URL: https://books.google.it/books?id=CGk1eRSjFIIC.

[82] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington. "A roadmap for the computation of persistent homology." In: *EPJ Data Science* 6.1 (2017). ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-017-0109-5. URL: http://dx.doi.org/10.1140/epjds/s13688-017-0109-5.

[83] T. Padellini and P. Brutti. *Persistence Flamelets: multiscale Persistent Homology for kernel density exploration*. 2017. arXiv: 1709.07097 [stat.ML].

[84] J. Perea and J. Harer. *Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis*. 2013. arXiv: 1307.6188 [math.AT].

[85] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. Hellyer, and F. Vaccarino. "Homological scaffolds of brain functional networks." In: *Journal of The Royal Society Interface* 11 (2014), p. 20140873. DOI: 10.1098/rsif.2014.0873.

[86] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. "Exponential expressivity in deep neural networks through transient chaos." In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 3360–3368. URL: http://papers.nips.cc/paper/6322-exponential-expressivity-in-deep-neural-networks-through-transient-chaos.pdf.

[87] K. N. Ramamurthy, K. R. Varshney, and K. Mody. "Topological Data Analysis of Decision Boundaries with Application to Model Selection." In: *arXiv preprint arXiv:1805.09949* (2018).

[88] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt. "A stable multi-scale kernel for topological machine learning." In: June 2015, pp. 4741–4748. DOI: 10.1109/CVPR.2015.7299106.

[89] B. Rieck, F. Sadlo, and H. Leitte. *Topological Machine Learning with Persistence Indicator Functions*. 2019. arXiv: 1907.13496 [math.AT].

[90] B. Rieck, M. Togninalli, C. Bock, M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt. "Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology." In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=ByxkijC5FQ.

[91] L. K. Saul and S. T. Roweis. "An introduction to locally linear embedding." In: *unpublished. Available at: http://www. cs. toronto. edu/~ roweis/lle/publications. html* (2000).

[92] J. Schaeffer. *One Jump Ahead: Challenging Human Supremacy in Checkers*. Copernicus Series. Springer, 1997. ISBN: 9780387949307. URL: https://books.google.com.gi/books?id=SLvpkTVhsIsC.

[93] V. Silva. "A weak characterisation of the Delaunay triangulation." In: *Geom. Dedicata* 135 (Aug. 2008), pp. 39–64. DOI: 10.1007/s10711-008-9261-1.

[94] V. Silva and G. Carlsson. "Topological estimation using witness complexes." In: *Proc. Sympos. Point-Based Graphics* (June 2004). DOI: 10.2312/SPBG/SPBG04/157-166.

[95] G. Singh, F. Mémoli, and G. E. Carlsson. "Topological methods for the analysis of high dimensional data sets and 3d object recognition." In: *SPBG* 91 (2007), p. 100.

[96] P. Skraba and K. Turner. *Wasserstein Stability for Persistence Diagrams*. 2020. arXiv: 2006.16824 [math.AT].

[97] D. I. Spivak. "METRIC REALIZATION OF FUZZY SIMPLICIAL SETS." In: 2009.

[98] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[99] G. Tauzin, U. Lupo, L. Tunstall, J. B. Pérez, M. Caorsi, A. Medina-Mardones, A. Dassatti, and K. Hess. *giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration*. 2020. arXiv: 2004.02551 [cs.LG].

[100] D. Taylor, F. Klimm, H. A. Harrington, M. Kramár, K. Mischaikow, M. A. Porter, and P. J. Mucha. "Topological data analysis of contagion maps for examining spreading processes on networks." In: *Nature Communications* 6.1 (2015). ISSN: 2041-1723. DOI: 10.1038/ncomms8723. URL: http://dx.doi.org/10.1038/ncomms8723.

[101] J. B. Tenenbaum, V. De Silva, and J. C. Langford. "A global geometric framework for nonlinear dimensionality reduction." In: *science* 290.5500 (2000), pp. 2319–2323.

[102] The GUDHI Project. *GUDHI User and Reference Manual*. 3.3.0. GUDHI Editorial Board, 2020. URL: https://gudhi.inria.fr/doc/3.3.0/.

[103] C. Tralie, N. Saul, and R. Bar-On. "Ripser.py: A Lean Persistent Homology Library for Python." In: *The Journal of Open Source Software* 3.29 (2018), p. 925. DOI: 10.21105/joss.00925. URL: https://doi.org/10.21105/joss.00925.

[104] V. N. Vapnik and A. Y. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities." In: *Theory of Probability and its Applications*. 1971, 16:264–280.

[105] L. Vietoris. "Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen." In: *Mathematische Annalen* 97.1 (1927), pp. 454–472.

[106] A. Wagner. *Nonembeddability of Persistence Diagrams with $p > 2$ Wasserstein Metric*. 2019. arXiv: 1910.13935 [math.FA].

[107] H. Wagner, C. Chen, and E. Vuçini. "Efficient Computation of Persistent Homology for Cubical Data." In: *Topological Methods in Data Analysis and Visualization II: Theory, Algorithms, and Applications*. Ed. by R. Peikert, H. Hauser, H. Carr, and R. Fuchs. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 91–106. ISBN: 978-3-642-23175-9. DOI: 10.1007/978-3-642-23175-9_7. URL: https://doi.org/10.1007/978-3-642-23175-9_7.

[108] H. Whitney. "Congruent Graphs and the Connectivity of Graphs." In: *American Journal of Mathematics* 54.1 (1932), pp. 150–168. ISSN: 00029327, 10806377. URL: http://www.jstor.org/stable/2371086.

[109] S. Willerton. *Spread: a measure of the size of metric spaces*. 2012. arXiv: 1209.2300 [math.MG].

[110] S. Wolfram and W. Media. *A New Kind of Science*. Wolfram Media, 2002. ISBN: 9781579550080.

APPENDIX A

Proofs

It is my experience that proofs involving matrices can be shortened by 50% if one throws the matrices out.

-Emil Artin (Geometric Algebra)

Proof of theorem 2.1.3.

THEOREM. The following definitions are equivalent

1. Let X and Y be topological spaces. Let $f : X \to Y$ be such that if for each open set $U \subseteq Y$ then $f^{-1}(U)$ is also an open set in X
2. $\forall \epsilon > 0, \exists \delta > 0$ such that $\|p - x\| < \delta \implies \|f(p) - f(x)\| < \epsilon$

Proof. We prove (1) $\implies$ (2). Let $f : X \to Y$ be such that if for each open set $U \subseteq Y$ then $f^{-1}(U)$ is also an open set in X. We want to show that for every $\epsilon > 0$ it is possible to find $\delta > 0$ such that $\|p - x\| < \delta \implies \|f(p) - f(x)\| < \epsilon$. For a fixed $x \in X$ and $\epsilon > 0$ let V be the set

$$V = B_\epsilon(f(p)) = \{y \in Y \mid \|y - f(p)\| < \epsilon\} \tag{A.1}$$

which is by definition an open set. By assumption if V is open, $f^{-1}(V)$ is also an open set, hence there exists a $\delta > 0$ such that $x \in f^{-1}(V)$ if $\|x - f(p)\| \leq \delta$ which implies

$$x \in= B_\delta(p) \implies f(x) \in B_\epsilon(f(p)) \tag{A.2}$$

We prove (2) $\implies$ (1).

Let $f : X \to Y$ such that $\forall \epsilon > 0, \exists \delta > 0$ such that $\|p - x\| < \delta \implies \|f(p) - f(x)\| < \epsilon$. Let V be an open set in Y. Suppose $p \in X$ and $f(p) \in V$ since V is open there exists a $\epsilon > 0$ such that

$$B_\epsilon(f(p)) = \{y \in Y \mid \|y - f(p)\| < \epsilon\} \subseteq V \tag{A.3}$$

. By assumption there exists a $\delta > 0$ such that

$$B_\delta(p) = \{x \in X \mid \|x - p\| < \epsilon\} \subseteq f^{-1}(V) \tag{A.4}$$

which is an open set in X.

□

Proof of theorem 2.2.3

THEOREM. The image of a compact space under a continuous map is compact.

Proof. Let $f : X \to Y$ be a continuous map and X be compact. Let $\mathcal{U}$ be a cover of $f(X) \subseteq Y$. The collection:

$$\{f^{-1}(U) \mid U \in \mathcal{U}\} \tag{A.5}$$

is a collection of sets covering X and are open since f is continuous.If there is a finte number of sets $f^{-1}(U_i)$ $i \in \{1, 2, \ldots, n\}$ that cover X then U_i $i \in \{1, 2, \ldots, n\}$ also cover $f(X)$. □

Proof of theorem 2.2.6

THEOREM. The image of a connected space under a continuous map is connected

Proof adapted from Munkres [77]. Let $f : X \to Y$ be a continuous map, let X be connected. We wish to proof the image space $Z = f(X)$ is connected. Consider $g : X \to Z$ the map obtained from f by restricting its range, obviously it is also continuous. Suppose that $Z = A \bigsqcup B$ is a separation of Z into disjoint nonempty open sets in Z. Then $g^{-1}(A)$ and $g^{-1}(B)$ are disjoints sets whose union is X. They are open because g is continuous and nonempty because g is surjective. Therefore they are a separation of X, contradicting the assumption that X is connected. □

Proof of theorem 2.2.7

THEOREM. Let X, Y be topological spaces and $X = \bigsqcup X_i$ and $Y = \bigsqcup Y_i$ its connected components, if $f : X \to Y$ is a homeomorphism then for all i there exists j such that $f(X_i) = Y_j$

Proof. Let $f : X \to Y$ be a homeomorphism. From theorem 2.2.6 we have that $f(X_i)$ is connected for all i since f is continuous. Then there exists j such that $f(X_i) \subseteq Y_j$. Since f is a homeomorphism we have that $f^{-1}(Y_j) \subseteq X_i$ and thus $Y_j \subseteq f(X_i)$. □

Proof of corollary 2.3.2

THEOREM. If $\mathcal{V}$ is a subcover of $\mathcal{U}$ then $\mathcal{V}$ is a refinement of $\mathcal{U}$.

Proof. Trivial □

Proof of theorem 2.3.4

THEOREM. Homeomorphic spaces have the same covering dimension.

Proof. Let X, Y be spaces such that $dim(X) = n$ and $dim(Y) = m$ and $f : X \to Y$ be a homeomorphism. Since $dim(Y) = m$ there exists $m+1$ open sets U_i and an element $y \in Y$ such that

$$y \in \bigcap_{i}^{m+1} U_i \tag{A.6}$$

since f is a continuous bijection there exists $f^{-1}(y) \in X$ and open m open sets $f^{-1}(U_i) \subseteq X$ such that

$$f^{-1}(y) \in \bigcap_{i}^{m+1} f^{-1}(U_i) \tag{A.7}$$

which proves that $dim(X) \geq dim(Y)$. Since f is an homeomorphism f^{-1} is also a continuous bijection and a reversed reasoning would provide us with $dim(Y) \geq dim(X)$, concluding that $dim(X) = dim(Y)$. □

Proof of theorem 2.4.2

THEOREM. The dimension of an n-manifold is equal to n.

Proof. This statement might appear harmless but the proof is far beyond the scope of this thesis. The proof is however accessible and rather elegant if the reader is interested. A sketch of the proof follows: Mill [71] proves that the Lebesgue covering dimension of a manifold is equal to a concept called small inductive dimension. The proof is completed by understanding that the small inductive dimension is the same as the dimension of the manifold. □

Proof of theorem 2.4.3

THEOREM. Homeomorphic manifolds have the same dimension.

Proof. The proof is obvious considering theorem 2.3.4 and 2.4.3. Below is another proof directly from the definition. It is an interesting exercise to familiarize oneself with the concepts of manifold and local homeomorphisms.

Let M and N be manifolds of dimension m and n respectively. It is sufficient[1] to show that if M and N are homeomorphic then there exists a homeomorphism between any two open sets $U \subseteq \mathbb{R}^m$, $V \subseteq \mathbb{R}^n$ and that implies $m = n$. Let $\theta : M \to N$ a homeomorphism. Let $x \in M$ there exists $A \subset M$ open and homeomorphism $\phi_1 : A \subset M \to \mathbb{R}^m$.

$$\begin{array}{c} x \in A \subset M \\ \big\downarrow \phi_1 \\ \mathbb{R}^m \end{array} \tag{A.8}$$

The homeomorphism maps to an open set of $\mathbb{R}^m$, by abuse of language we will denote it just like this. Consider $\theta(x) \in N$, then there exists a $B \subset N$ open and homeomorphism $\phi_2 : B \subset N \to \mathbb{R}^n$

[1] Consequence of the Domain Invariance Theorem

$$\begin{array}{ccc} x \in A \subset M & \xrightarrow{\theta} & \theta(x) \in B \subset N \\ \downarrow \phi_1 & & \downarrow \phi_2 \\ \mathbb{R}^m & & \mathbb{R}^n \end{array} \tag{A.9}$$

Note that $\phi_2(\theta(A) \cap B)$ is an open set in $\mathbb{R}^n$. By restricting the domain of ϕ_2 to $\theta(A) \cap B$ and the range of ϕ_1 to $\theta^{-1}(\theta(A) \cap B))$ we get that the composition $\phi_2 \circ \theta \circ \phi_1^{-1}$ is an homeomorphism between an open set of $\mathbb{R}^m$ to an open set in $\mathbb{R}^n$.

$$\begin{array}{ccc} x \in A \subset M & \xrightarrow{\theta} & \theta(x) \in B \subset N \\ \downarrow \phi_1 & & \downarrow \phi_2 \\ \mathbb{R}^m & \xrightarrow{\phi_2 \circ \theta \circ \phi_1^{-1}} & \mathbb{R}^n \end{array} \tag{A.10}$$

□

Proof of Lemma 5.2.3[McInnes et al. [67]]

> THEOREM. Let (M, g) be a Riemannian manifold in $\mathbb{R}^n$, and let $p \in M$ be a point. If g is locally constant about p in an open neighbourhood U such that g is a constant diagonal matrix in ambient coordinates, then in a ball $B \subseteq U$ centered at p with volume $\frac{\pi^{n/2}}{2\Gamma(n/2n+1)}$ with respect to g, the geodesic distance from p to any point $q \in B$ is $\frac{1}{r} d_{\mathbb{R}^n}(p, q)$, where r is the radius of the ball in the ambient space and $d_{\mathbb{R}^n}$ is the existing metric on the ambient space.

Proof from McInnes et al. [67]. Let $x^1, \ldots, x^n$ be the coordinate system for the ambient space. A ball B in M under Riemmannian metric g has volume given by

$$\int_B \sqrt{\det(g)} dx^1 \wedge \cdots \wedge dx^n. \tag{A.11}$$

If B is contained in U, then g is constant in B and hence $\sqrt{\det(g)}$ is constant and can be brought outside the integral. Thus the volume of B is

$$\sqrt{\det(g)} \int_B dx^1 \wedge \cdots \wedge dx^n = \sqrt{\det(g)} \frac{\pi^{n/2}}{2\Gamma(n/2n+1)}, \tag{A.12}$$

where r is the radius o f the call in the ambient $\mathbb{R}^n$. If we fix the volume of the ball to be $\frac{\pi^{n/2}}{2\Gamma(n/2n+1)}$ we arrive at the requirement that

$$\det(g) = \frac{1}{r^{2n}} \tag{A.13}$$

Now since g is assumed to be diagonal with constant entries we can solve for g itself as

$$g_{ij} = \begin{cases} \frac{1}{r^2} & if\, i = j \\ 0 & \text{otherwise} \end{cases} \tag{A.14}$$

The geodesic distance on M under g from p to q (where $p, q \in B$) is defined as

$$\inf_{c \in C} \int_a^b \sqrt{g(c'(t), c'(t))} dt, \tag{A.15}$$

where C is the class of smooth curves c on M such that $c(a) = p$ and $c(b) = q$ and c' denotes the first derivative of c on M. We can now simply g to

$$\begin{aligned} &\frac{1}{r} \inf_{c \in C} \int_a^b \sqrt{\langle c'(t), c'(t) \rangle} dt \\ =&\frac{1}{r} \inf_{c \in C} \int_a^b \|c'(t), c'(t)\| dt \\ =&\frac{1}{r} d_{\mathbb{R}^n}(p, q) \end{aligned}$$

□

Proof of proposition 5.1.1

> THEOREM. Let W_i be a $m \times n$ matrix such that $m \geq n$ and $rank(W_i) = n$, and let a be a continuous bijection with continuous inverse. The transformation:
>
> $$L_i(x) = a(W_i x + B_i) \quad \forall x \in X_i \tag{A.16}$$
>
> defines a homeomorphism between X_i and $X_{i+1} = L_i(X_i)$.

Proof. Let us first consider the case when $m = n$. In these conditions W_i is a linear function with a linear inverse. Since linear functions are continuous W_i is a homeomorphism. If a is a continuous bijection with a continuous inverse then L_i is a composition of homeomorphisms which is also a homeomorphism.

Let us take now $m > n$. Without loss of generality we only need to find a homeomorphism between X_i and $W_i X_i$

Since $rank(W_i) = n$, $dim(W_i x) = n$. Since $W_i x \subset \mathbb{R}^m$ there exist $m - n$ linearly independent vectors $\{e_1, \dots e_{m-n}\}$ such that the matrix $W_i' = \left[W_i | \, e_1 \, \dots \, e_{m-n} \right]$ has $rank(W_i') = m$. We have then that:

$$W_i x = W_i' x' = \left[\begin{array}{ccccc} & e_1^1 & \dots & e_{m-n}^1 \\ W_i & \vdots & & \\ & e_1^m & \dots & e_{m-n}^m \end{array} \right] \left[\begin{array}{c} x_1 \\ \vdots \\ x_n \\ \hline 0 \\ \vdots \\ 0 \end{array} \right] \begin{array}{l} \Big\} \text{n} \\ \\ \Big\} \text{m-n} \end{array} \tag{A.17}$$

By construction W_i' is a linear function with a linear inverse such that

$$L_i(X_i) = a(W_i x + B_i) = a(W_i' x' + B_i) \quad \forall x \in X \tag{A.18}$$

Along with the first case we conclude that X_i and $X_{i+1} = L_i(X)$ are homeomorphic. □

Proof of proposition 5.4.2

THEOREM. For a metric space (X, d) where X has N elements, we have the following properties:

1. $t_0 \leq t_1 \Rightarrow E_0(t_0 X) \leq E(t_1 X)$
2. $\lim_{t\to 0} E_0(tX) = 1$
3. $\lim_{t\to\infty} E_0(tX) = N$

Proof. Statement 1:

$$\begin{aligned}
t_0 \leq t_1 &\Rightarrow e^{-t_0 d(x,y)} \geq e^{-t_1 d(x,y)} \\
&\Rightarrow \sum_{y\in X} e^{-t_0 d(x,y)} \geq \sum_{y\in X} e^{-t_1 d(x,y)} \\
&\Rightarrow \left(\sum_{y\in X} e^{-t_0 d(x,y)}\right)^{-1} \leq \left(\sum_{y\in X} e^{-t_1 d(x,y)}\right)^{-1} \\
&\Rightarrow \sum_{x\in X}\left(\sum_{y\in X} e^{-t_0 d(x,y)}\right)^{-1} \leq \sum_{x\in X}\left(\sum_{y\in X} e^{-t_1 d(x,y)}\right)^{-1} \\
&\Rightarrow E_0(t_0 X) \leq E(t_1 X)
\end{aligned}$$

Statement 2:

$$\begin{aligned}
\lim_{t\to 0} E_0(tX) &= \sum_{x\in X}\left(\sum_{y\in X} \lim_{t\to 0} e^{-t d(x,y)}\right)^{-1} \\
&= \sum_{x\in X} (N)^{-1} \\
&= \frac{N}{N} = 1
\end{aligned}$$

Statement 3:

$$\begin{aligned}
\lim_{t\to\infty} E_0(tX) &= \lim_{t\to\infty} \sum_{x\in X}\left(\sum_{y\in X} e^{-t_1 d(x,y)}\right)^{-1} \\
&= \lim_{t\to\infty} \sum_{x\in X}\left(1 + \sum_{y\neq x} e^{-t_1 d(x,y)}\right)^{-1} \\
&= \sum_{x\in X} 1 = N
\end{aligned}$$

□

Appendix B

On The Combinatoria of Neural Architectures

What, you're still here? Go home.

-Deadpool (Post-credits scene)

This section is motivated by the following combinatorial problem: how many different neural network architectures have p parameters? We will only consider architectures that are dense and fully connected and disregard the choice of activation functions.

Let's start, for simplicity, by not considering bias now. The only parameters are then the weights between connections. Take a neural network with 3 layers of $3, 4, 5$ neurons respectively. The number of weights, and therefore parameters, is equal to $3 \times 4 + 4 \times 5 = 32$. In the general case, for an architecture we can associate the ordered set $(n_0, n_1, n_2 ... n_k)$ where n_i is the number of neurons in the i-th layer, the number of parameters p is given by:

$$p = n_0 n_1 + n_1 n_2 + \cdots + n_{k-1} n_k = \sum_{i=0}^{k-1} n_i n_{i+1} \tag{B.1}$$

Let us consider the sequence $(l_0, l_1, l_2, \dots l_{k-1})$ such that $l_i = n_i n_{i+1}$. One can think of it as something like the number of parameters per layer. And so now we have that:

$$p = \sum_{i=0}^{k-1} l_i \tag{B.2}$$

Assuming we are given a value of p and we want to find all possible neural network architectures. We start by considering only all the integer partitions of p, i.e all the possible sequence of integers such that their sum is equal to p. The integer partitions of p are the only possible combinations of l_i.

Therefore for each possible partition $(l_0, l_1, l_2 \dots l_n)$ of p we consider the system:

$$\begin{aligned} n_0 n_1 &= l_0 \\ n_1 n_2 &= l_1 \\ &\vdots \\ n_{k-1} n_k &= l_{k-1} \end{aligned}$$

This system is underdetermined since we have k variables and $k-1$ equations. Yet the total number of solutions is still finite since we have the restriction that each n_i has to be an integer. Therefore let α be a divisor of l_0. By setting $n_0 = \alpha$ we get a determined system:

$$\begin{cases} n_0 = \alpha \\ n_{i+1} = \dfrac{l_i}{n_i} \quad \forall i > 0 \end{cases} \tag{B.3}$$

And the ordered set $(n_0, n_1, n_2, \dots n_k)$ is a valid architecture of p parameters if each n_i is an integer. Note that each n_{i+1} only exists if it is a common divisor of both l_i and l_{i+1}.

B.1 Adding bias to the mix.

The addition of bias changes minimally the system of equations described above, because the effect of biases on the number of parameters is also uniquely dependent on the number of neurons in each layer. For an architecture $(n_0, n_1, n_2 \dots n_k)$ the number of parameters p, *including bias* is given by:

$$\begin{aligned} p &= n_0 n_1 + n_1 + n_1 n_2 + n_2 \dots n_{k-1} n_k + n_k \\ p &= \sum_{i=0}^{k-1} n_i n_{i+1} + n_{i+1} = \sum_{i=0}^{k-1} n_{i+1}(n_i + 1) \end{aligned}$$

In an analogous way, for a $(l_0, \dots l_k - 1)$ partition of p we have that $l_i = n_{i+1}(n_i + 1)$ And for each divisor α of l_0 we get the determined system:

$$\begin{cases} n_0 = \alpha \\ n_{i+1} = \dfrac{l_i}{n_i + 1} \quad \forall i > 0 \end{cases} \tag{B.4}$$

B.2 A computational view.

Note that we have a possible architecture for every partition of p, times every divisor of the first element of each partition. If $\mathcal{N}_p$ is the number of neural network architectures with p parameters then we have that:

$$\mathcal{N}_p \leq C(p) \sum_{i,j} \alpha_{ij} \tag{B.5}$$

Where α_{ij} is the i-th divisor of the first element of the j-partition of p and $C(p)$ is number of ways to partition p. Note that the maximum value in each any partition is $p-1$ and the sum of all the divisors of $p-1$, denoted by $\sigma(p-1)$ is bounded by:

$$\sigma(p-1) < e^{\gamma}(p-1)\log\log(p-1) + \frac{0.6483(p-1)}{\log\log(p-1)} \tag{B.6}$$

Where $\gamma = 0.577215...$ is the Euler's constant. We also have that the an upper bound for the different partitions of p:

$$C(p) < e^{\pi\sqrt{\frac{2}{3}}\sqrt{p}} \tag{B.7}$$

Proposition B.2.1. Let $\mathcal{N}_p$ be the number of different architectures with p parameters, we have that:

$$\mathcal{N}_p < e^{\pi\sqrt{\frac{2}{3}}\sqrt{p}}\left(e^{\gamma}(p-1)\log\log(p-1) + \frac{0.6483(p-1)}{\log\log(p-1)}\right) \tag{B.8}$$

ON the practical side, the biggest hurdle is definitely the number of partitions which exponentially increases the number of possible architectures, while the number of divisors has a constant effect on the number of possibilities. This is mainly because there are a lot of partitions that have no practical value in our study such as the trivial ones $p = \sum^{p} 1$. We will touch on this point later.

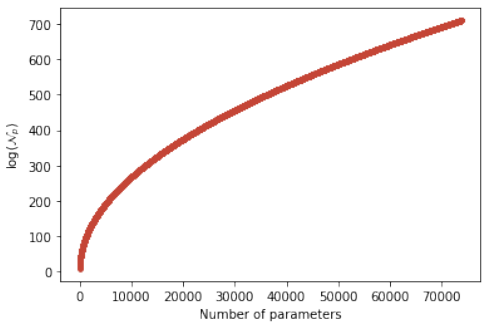
Based on the analysis done above we can create and algorithm to find all possible architectures given a parameter value p. Such an algorithm obviously runs on exponential

time.

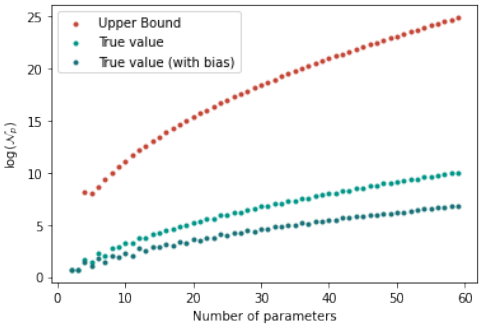
Algorithm 2: Find all architectures of p parameters

input : Integer p number of parameters
input : $b = 1$ if bias, $b = 0$ otherwise
output: List of lists $A = (N_0, N_1 ... N_m)$
where each N_i is a list of integers $(n_0, n_1, \dots n_l)$

```
A ← initialize empty list;
for each partition q of p do
    for each factor f of q_0 do
        N ← initialize empty list;
        z ← f;
        add z to N;
        i ← 0;
        while z is an integer do
            z ← q_i / (z + b);
            add z to N;
            i++;
            if i = size of q then
                add N to A;
            end
        end
    end
end
```



(a) Upper bound for the number of possible architectures



(b) Upper bound compared with the architectures computed using Algo. 2

Figure B.1: Logarithm of total number of possible architectures (y-axis $\log \mathcal{N}_p$) of p-parameters (x-axis) compared with the upper bound (red)

Proposition B.2.2. Let $\mathcal{N}_p$ be the number of possible architectures of p parameters without considering bias and $\mathcal{N}_p^b$ be the number of architectures with bias. We have that

$$\mathcal{N}_p^b \leq \mathcal{N}_p \quad \forall p \in \mathbb{N} \tag{B.9}$$

Proof. Consider N_p as the set of all architectures of p parameters with no bias. Then $\mathcal{N}_p = |N_p|$. equivalently for the set of architectures with bias $\mathcal{N}_p^b = |N_p^b|$. We will prove the proposition by showing that:

- $\forall n \in N_p^b \quad \exists q < p$ such that $n \in N_q$
- thus justifying that $\sum_i^p \mathcal{N}_i^b \leq \sum_i^p \mathcal{N}_i$
- which along with the implication $q \leq p \Rightarrow \mathcal{N}_q \leq \mathcal{N}_p$ concludes the proof.

Let $(n_0, n_1, \dots n_k)$ be a p-parameter architecture with bias, meaning that:

$$\begin{aligned} p &= \sum_{i=0}^{k-1} n_i n_{i+1} + n_{i+1} \\ &= \sum_{i=0}^{k-1} n_i n_{i+1} + \sum_{i=1}^{k} n_i \end{aligned}$$

Which is equivalent to

$$p - \sum_{i=1}^{k} n_i = \sum_{i=0}^{k-1} n_i n_{i+1} \tag{B.10}$$

Let $q = p - \sum_{i=1}^{k} n_i$ and since every n_i is a positive integer then q is also a positive integer. Thus we have that:

$$q = \sum_{i=0}^{k-1} n_i n_{i+1} \tag{B.11}$$

Which an acceptable q parameter neural network architecture with no bias. Thus if a sequence $(n_0, n_1, \dots n_k) \in N_p^b$ then it also belongs to N_q for some $q < p$. Hence, we now have that

$$\sum_{i}^{p} \mathcal{N}_i^b \leq \sum_{i}^{p} \mathcal{N}_i \tag{B.12}$$

By noting that $q \leq p \Rightarrow \mathcal{N}_q \leq \mathcal{N}_p$ we have that:

$$\mathcal{N}_p^b \leq \mathcal{N}_p \quad \forall p \in \mathbb{N} \tag{B.13}$$

□

The proof hangs on the (small) assumption that there is a valid architecture for every $p \in \mathbb{N}$. We leave this unproven, albeit being a interesting problem all by itself.