

variation called empirical generalization gap, given as the difference between the accuracy in a test dataset and the accuracy in the training dataset, i.e.,

$$\widehat{\text{Gap}}(\mathcal{N}) = \widehat{\text{Acc}}_{\mathcal{D}_{\text{test}}}(\mathcal{N}) - \widehat{\text{Acc}}_{\mathcal{D}_{\text{train}}}(\mathcal{N}).$$

An analysis of methods that try to predict this last measure can be found in Jiang et al. (2021). Also, a large-scale study of generalization bounds and measures of generalization in deep learning can be found in Jiang et al. (2020b).

#### 2.4.7 QUALITY ASSESSMENT OF GENERATIVE MODELS

Neural network quality measurements in classification tasks are more straightforward to define than in generative tasks. In general, there are many interesting factors that can be evaluated to compare the quality of two generative neural networks, and thus deciding which is better is a hard challenge. For example, one of these factors could be the similarity between the supports of the original data distribution  $\mathbb{P}_X$  and the distribution induced by the generation function.

### 3 Analysis of neural networks using topological data analysis

In this section, we discuss the articles that use topological data analysis to study neural networks. This section is divided into the four categories described in the Introduction. Table 1 in the appendix contains a list of all articles included in the survey. For each paper, we specify a one-line summary of it, its categories, and the applications from Section 2.4 explored in the article.

#### 3.1 Structure of a neural network

Recall that a fully connected feedforward neural network  $\mathcal{N}$  is specified by an architecture  $a = (N, \varphi)$  and a set of parameters  $\theta$  depending on the architecture  $a$ . At the same time, recall that the architecture  $a$  defines a directed graph  $G(\mathcal{N})$  that characterizes the computations of the neural network function, where each vertex is associated with some value during the evaluation of  $\phi_{\mathcal{N}}$ . The structure of this graph is relevant for studying the generalization capacity and expressiveness of neural networks. Note that edge directions are important in the neural network graph  $G(\mathcal{N})$  because they characterize how computations are performed on the data. For this reason, it is preferable to avoid analyzing  $G(\mathcal{N})$  forgetting its directions with the general topological tools presented in Section 2.3.

Chowdhury et al. (2019) studied and compared two special homology groups designed for directed graphs in the context of FCFNNs. More specifically, the ranks of these two variants of homology groups are computed. These groups are the path homology (Grigor'yan et al., 2014) and the directed flag complex (DFC) homology (Reimann et al., 2017) groups, denoted by  $\text{PathHom}_k(G(\mathcal{N}))$  and  $\text{DFCHom}_k(G(\mathcal{N}))$ , respectively, where  $k$  denotes the dimension of the homology groups as in the simplicial case. In particular, Chowdhury et al. showed that, given a neural network  $\mathcal{N}$  with architecture  $a = (N, \varphi)$ , where  $N = (N_0, \dots, N_L)$ , and

given a dimension  $k \in \mathbb{N}$ , the following equalities hold:

$$\begin{aligned}\text{rank}(\text{PathHom}_k(G(\mathcal{N}))) &= \delta_{L-1,k} \prod_{i=0}^L (N_i - 1), \\ \text{rank}(\text{DFCHom}_k(G(\mathcal{N}))) &= \text{rank}(H_k(G(\mathcal{N}))),\end{aligned}$$

where  $\text{rank}(H_k(G(\mathcal{N})))$  is the rank of the  $k$ -dimensional simplicial homology group of  $G(\mathcal{N})$  seen as an undirected graph. It can be further shown that

$$\text{rank}(H_k(G(\mathcal{N}))) = \begin{cases} 1 & \text{if } k = 0, \\ 1 - |V(G(\mathcal{N}))| + |E(G(\mathcal{N}))| & \text{if } k = 1, \\ 0 & \text{if } k \geq 2. \end{cases}$$

The equality between the DFC-homology ranks and the standard simplicial homology groups for neural network graphs implies that DFCHom does not capture the information carried out by the directions of neural networks. Note that the ranks for both families of homology groups can be directly computed from the number of neurons and edges contained by the network, and do not reflect most of the structural properties of the neural network graph. For example, path homologies are invariant to layer permutations, although the order of the layers is a key factor in achieving good neural network performance. On the other hand, DFC-homology forgets all the structural information except for the number of vertices and edges, which many neural network configurations with different performances share.

Path homology theory for neural networks has an important drawback, namely that it forgets the weights of neural networks, which are crucial in the performance of neural networks. However, this can be remedied by studying a *persistent* version of it, called *persistent path homology* (Chowdhury and Mémoli, 2018). To the best of our knowledge, there is no work using persistent path homology to analyze neural networks, and it could be a notable future work for the interested reader.

### 3.2 Input and output spaces

In this section, we analyze how TDA has been used to analyze decision regions, boundaries, decompositions of the input space, and input and output spaces of generative and non-generative models.

#### General output space and decision regions

Let  $f: \mathcal{X} \rightarrow [k]$ ,  $k \in \mathbb{N}_{\geq 2}$  be the unknown function that defines a classification problem as in Section 2.2 with data distribution  $\mathbb{P}_{(X,Y)}$ . A simple remark is that if we train a perfect neural network  $\mathcal{N}$  with projection function  $\pi$ , that is,  $\pi \circ \phi_{\mathcal{N}} = f$ , then the homology groups of the decision regions of each label  $i \in [k]$  must coincide with the homology groups of the space of data with the same label for all dimensions, that is,  $H_k((\pi \circ \phi_{\mathcal{N}})^{-1}(i)) = H_k(f^{-1}(i))$  for all  $k \in \mathbb{N}$  and  $i \in [k]$ . The previous equality suggests that the topology of the decision regions is relevant for understanding the generalization of trained neural networks with respect to the data.

One of the first articles in this survey that studied decision regions using topological data analysis was published by Bianchini and Scarselli (2014). In it, Betti numbers of the decision regions of one of the labels of a binary classification problem are analyzed and theoretically bounded for neural networks with Pfaffian activations (Khovanskiĭ, 1991). A difficulty in the previous paper is that bounds of Betti numbers are not tight and cannot be used easily in practical scenarios. An experimental approach is taken by Guss and Salakhutdinov (2018), who empirically study such Betti numbers and compare them with the Betti numbers of the real region induced by the selected label, that is,  $f^{-1}(i)$ , where  $i \in [2]$  is the label studied.

On a more general setting, Liu et al. (2023a) proposed the GTDA algorithm of Section 2.3.2 as an improved version of Mapper to study the output space of general neural network functions. They studied Reeb networks built using GTDA with input graphs induced from the output vectors  $(\phi_{\mathcal{N}}^{(L)}(x)_1, \dots, \phi_{\mathcal{N}}^{(L)}(x)_{N_L})$  of neural network functions given a dataset  $\mathcal{D}$  and filter functions given by the same output values plus, possibly, some extra values depending on the dataset or the outputs. To give output values a graph structure, they either used previously-known binary relationships between the data or binary relationships given by nearest neighbors algorithms as edges.

Liu et al. were able to gain useful information from GTDA graphs that was not found using Mapper or other alternatives about output values of the Enformer model (Avsec et al., 2021) used as inputs to predict harmful mutations of a human gene. Also, GTDA graphs were used to perform automatic error estimation for machine learning prediction tasks. A numerical error estimation score for a sample  $x$  was calculated by comparing the predicted label of  $x$  with ground truth labels of training and validation samples whose prediction vectors were surrounding the prediction vector of  $x$  in the union of subgraphs of the Reeb network vertices plus the extra edges added in the GTDA algorithm. The error estimation score was shown to successfully correct erroneous predictions in binary classification tasks. Also, it detected regions of the output space with many misclassifications, that allowed them to determine the source of error in those regions for the previous classification problem dealing with the Enformer architecture. Similar experiments were performed on other network architectures and datasets such as ResNet-50 and Imagenette (Howard, 2021), showing that GTDA offers unique insights about the output space that cannot be recovered with other methods such as Mapper.

### Decision boundaries of a single network

When using classification neural networks with  $N_L = k$  and  $\pi(x_1, \dots, x_k) = \operatorname{argmax}_{i \in [k]} x_i$ , decision boundaries  $\mathcal{B}$ , defined in Equation (4), become relevant as they determine the decision regions of the neural network and the spaces in which neural network decisions have more than one valid output. Ramamurthy et al. (2019) propose a modified version of the Čech simplicial complexes such that, given a large enough sample of points in the decision region of a binary classification neural network, can reconstruct a space that is homotopy equivalent to the real decision boundary induced by the neural network under some mild assumptions about the decision boundary. However, Čech complexes are too costly to compute in practical scenarios. For this reason, they propose two computationally feasible variants of the Vietoris–Rips filtrations and complexes that aim to recover the Betti numbers of the decision boundaries of neural networks from a sample of their points using the persistence diagrams induced by these new filtrations. The new filtrations