Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology

Bastian Rieck[1,2,†], **Matteo Togninalli**[1,2,†], **Christian Bock**[1,2,†],
Michael Moor[1,2], **Max Horn**[1,2], **Thomas Gumbsch**[1,2], **Karsten Borgwardt**[1,2]
[1] Department of Biosystems Science and Engineering, ETH Zurich, Switzerland
[2] SIB Swiss Institute of Bioinformatics, Switzerland
[†] These authors contributed equally

Abstract

While many approaches to make neural networks more fathomable have been proposed, they are restricted to interrogating the network with input data. Measures for characterizing and monitoring structural properties, however, have not been developed. In this work, we propose *neural persistence*, a complexity measure for neural network architectures based on topological data analysis on weighted stratified graphs. To demonstrate the usefulness of our approach, we show that *neural persistence* reflects best practices developed in the deep learning community such as dropout and batch normalization. Moreover, we derive a neural persistence-based stopping criterion that shortens the training process while achieving comparable accuracies as early stopping based on validation loss.

1 Introduction

The practical successes of deep learning in various fields such as image processing (Simonyan & Zisserman, 2015; He et al., 2016; Hu et al., 2018), biomedicine (Ching et al., 2018; Rajpurkar et al., 2017; Rajkomar et al., 2018), and language translation (Bahdanau et al., 2015; Sutskever et al., 2014; Wu et al., 2016) still outpace our theoretical understanding. While hyperparameter adjustment strategies exist (Bengio, 2012), formal measures for assessing the generalization capabilities of deep neural networks have yet to be identified (Zhang et al., 2017). Previous approaches for improving theoretical and practical comprehension focus on interrogating networks with input data. These methods include i) feature visualization of deep convolutional neural networks (Zeiler & Fergus, 2014; Springenberg et al., 2015), ii) sensitivity and relevance analysis of features (Montavon et al., 2017), iii) a descriptive analysis of the training process based on information theory (Tishby & Zaslavsky, 2015; Shwartz-Ziv & Tishby, 2017; Saxe et al., 2018; Achille & Soatto, 2018), and iv) a statistical analysis of interactions of the learned weights (Tsang et al., 2018). Additionally, Raghu et al. (2017) develop a measure of *expressivity* of a neural network and use it to explore the empirical success of batch normalization, as well as for the definition of a new regularization method. They note that one key challenge remains, namely to provide meaningful insights while maintaining theoretical generality. This paper presents a method for elucidating neural networks in light of both aspects.

We develop *neural persistence*, a novel measure for characterizing neural network structural complexity. In doing so, we adopt a new perspective that integrates both network weights and connectivity while not relying on interrogating networks through input data. Neural persistence builds on computational techniques from algebraic topology, specifically topological data analysis (TDA), which was already shown to be beneficial for feature extraction in deep learning (Hofer et al., 2017) and describing the complexity of GAN sample spaces (Khrulkov & Oseledets, 2018). More precisely, we rephrase deep networks with fully-connected layers into the language of algebraic topology and develop a measure for assessing the structural complexity of i) individual layers, and ii) the entire network. In this work, we present the following contributions:

- We introduce *neural persistence*, a novel measure for characterizing the structural complexity of neural networks that can be efficiently computed.

- We prove its theoretical properties, such as upper and lower bounds, thereby arriving at a normalization for comparing neural networks of varying sizes.
- We demonstrate the practical utility of neural persistence in two scenarios: i) it correctly captures the benefits of dropout and batch normalization during the training process, and ii) it can be easily used as a competitive early stopping criterion that does not require validation data.

2 Background: Topological data analysis

Topological data analysis (TDA) recently emerged as a field that provides computational tools for analysing complex data within a rigorous mathematical framework that is based on *algebraic topology*. This paper uses persistent homology, a theory that was developed to understand high-dimensional manifolds (Edelsbrunner et al., 2002; Edelsbrunner & Harer, 2010), and has since been successfully employed in characterizing graphs (Sizemore et al., 2017; Rieck et al., 2018), finding relevant features in unstructured data (Lum et al., 2013), and analysing image manifolds (Carlsson et al., 2008). This section gives a brief summary of the key concepts; please refer to Edelsbrunner & Harer (2010) for an extensive introduction.

Simplicial homology The central object in algebraic topology is a simplicial complex K, i.e. a high-dimensional generalization of a graph, which is typically used to describe complex objects such as manifolds. Various notions to describe the connectivity of K exist, one of them being simplicial homology. Briefly put, simplicial homology uses matrix reduction algorithms (Munkres, 1996) to derive a set of groups, the homology groups, for a given simplicial complex K. Homology groups describe topological features—colloquially also referred to as holes—of a certain dimension d, such as connected components ($d = 0$), tunnels ($d = 1$), and voids ($d = 2$). The information from the dth homology group is summarized in a simple complexity measure, the dth Betti number β_d, which merely counts the number of d-dimensional features: a circle, for example, has Betti numbers $(1, 1)$, i.e. one connected component and one tunnel, while a filled circle has Betti numbers $(1, 0)$, i.e. one connected component but no tunnel. In the context of analysing simple feedforward neural networks for two classes, Bianchini & Scarselli (2014) calculated bounds of Betti numbers of the decision region belonging to the positive class, and were thus able to show the implications of different activation functions. These ideas were extended by Guss & Salakhutdinov (2018) to obtain a measure of the topological complexity of decision boundaries.

Persistent homology For the analysis of real-world data sets, however, Betti numbers turn out to be of limited use because their representation is too coarse and unstable. This prompted the development of persistent homology. Given a simplicial complex K with an additional set of weights $a_0 \leq a_1 \leq \cdots \leq a_{m-1} \leq a_m$, which are commonly thought to represent the idea of a scale, it is possible to put K in a filtration, i.e. a nested sequence of simplicial complexes $\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{m-1} \subseteq \mathrm{K}_m = \mathrm{K}$. This filtration is thought to represent the 'growth' of K as the scale is being changed. During this growth process, topological features can be *created* (new vertices may be added, for example, which creates a new connected component) or *destroyed* (two connected components may merge into one). Persistent homology tracks these changes and represents the creation and destruction of a feature as a point $(a_i, a_j) \in \mathbb{R}^2$ for indices $i \leq j$ with respect to the filtration. The collection of all points corresponding to d-dimensional topological features is called the dth persistence diagram $\mathcal{D}_d$. It can be seen as a collection of Betti numbers at multiple scales. Given a point $(x, y) \in \mathcal{D}_d$, the quantity $\operatorname{pers}(x, y) := |y - x|$ is referred to as its *persistence*. Typically, high persistence is considered to correspond to features, while low persistence is considered to indicate noise (Edelsbrunner et al., 2002).

3 A novel measure for neural network complexity

This section details *neural persistence*, our novel measure for assessing the structural complexity of neural networks. By exploiting both network structure and weight information through persistent homology, our measure captures network expressiveness and goes beyond mere connectivity properties. Subsequently, we describe its calculation, provide theorems for theoretical and empirical bounds, and show the existence of neural networks complexity regimes. To summarize this section, Figure 1 illustrates how our method treats a neural network.

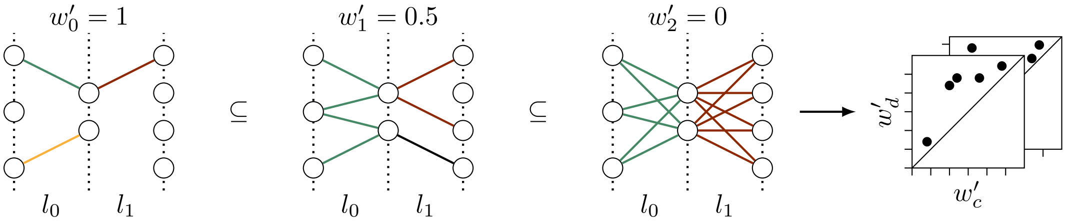

Figure 1: Illustrating the neural persistence calculation of a network with two layers (l_0 and l_1). Colours indicate connected components per layer. The filtration process is depicted by colouring connected components that are created or merged when the respective weights are greater than or equal to the threshold w'_i. As w'_i decreases, network connectivity increases. Creation and destruction thresholds are collected in one persistence diagram per layer (right), and summarized according to Equation 1 for calculating neural persistence.

3.1 Neural persistence

Given a feedforward neural network with an arrangement of neurons and their connections E, let $\mathcal{W}$ refer to the set of weights. Since $\mathcal{W}$ is typically changing during training, we require a function $\varphi\colon E \to \mathcal{W}$ that maps a specific edge to a weight. Fixing an activation function, the connections form a *stratified graph*.

Definition 1 (Stratified graph and layers)**.** *A* stratified graph *is a multipartite graph $G = (V, E)$ satisfying $V = V_0 \sqcup V_1 \sqcup \ldots$, such that if $u \in V_i$, $v \in V_j$, and $(u, v) \in E$, we have $j = i + 1$. Hence, edges are only permitted between adjacent vertex sets. Given $k \in \mathbb{N}$, the kth* layer *of a stratified graph is the unique subgraph $G_k := (V_k \sqcup V_{k+1}, E_k := E \cap \{V_k \times V_{k+1}\})$.*

This enables calculating the persistent homology of G and each G_k, using the filtration induced by sorting all weights, which is common practice in topology-based network analysis (Carstens & Horadam, 2013; Horak et al., 2009) where weights often represent closeness or node similarity. However, our context requires a novel filtration because the weights arise from an incremental fitting procedure, namely the training, which could theoretically lead to unbounded values. When analysing geometrical data with persistent homology, one typically selects a filtration based on the (Euclidean) distance between data points (Bubenik, 2015). The filtration then connects points that are increasingly distant from each other, starting from points that are direct neighbours. Our network filtration aims to mimic this behaviour in the context of fully-connected neural networks. Our framework does not *explicitly* take activation functions into account; however, activation functions influence the evolution of weights during training.

Filtration Given the set of weights $\mathcal{W}$ for one training step, let $w_{\max} := \max_{w \in \mathcal{W}} |w|$. Furthermore, let $\mathcal{W}' := \{|w|/w_{\max} \mid w \in \mathcal{W}\}$ be the set of transformed weights, indexed in non-ascending order, such that $1 = w'_0 \geq w'_1 \geq \cdots \geq 0$. This permits us to define a filtration for the kth layer G_k as $G_k^{(0)} \subseteq G_k^{(1)} \subseteq \ldots$, where $G_k^{(i)} := (V_k \sqcup V_{k+1}, \{(u, v) \mid (u, v) \in E_k \wedge \varphi'(u, v) \geq w'_i\})$ and $\varphi'(u, v) \in \mathcal{W}'$ denotes the transformed weight of an edge. We tailored this filtration towards the analysis of neural networks, for which large (absolute) weights indicate that certain neurons exert a larger influence over the final activation of a layer. The strength of a connection is thus preserved by the filtration, and weaker weights with $|w| \approx 0$ remain close to 0. Moreover, since $w' \in [0, 1]$ holds for the transformed weights, this filtration makes the network invariant to scaling, which simplifies the comparison of different networks.

Persistence diagrams Having set up the filtration, we can calculate persistent homology for every layer G_k. As the filtration contains at most 1-simplices (edges), we capture zero-dimensional topological information, i.e. how connected components are created and merged during the filtration. These information are structurally equivalent to calculating a maximum spanning tree using the weights, or performing hierarchical clustering with a specific setup (Carlsson & Mémoli, 2010). While it would theoretically be possible to include higher-dimensional information about each layer G_k, for example in the form of cliques (Rieck et al., 2018), we focus on zero-dimensional information in this paper, because of the following advantages: i) the resulting values are easily