Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-021-09761-3

Topological measurement of deep neural networks using persistent homology

Satoru Watanabe[1] · **Hayato Yamana**[1]

Accepted: 10 June 2021Published online: 03 July 2021

Abstract
The inner representation of deep neural networks (DNNs) is indecipherable, which makes it difficult to tune DNN models, control their training process, and interpret their outputs. In this paper, we propose a novel approach to investigate the inner representation of DNNs through topological data analysis (TDA). Persistent homology (PH), one of the outstanding methods in TDA, was employed for investigating the complexities of trained DNNs. We constructed clique complexes on trained DNNs and calculated the one-dimensional PH of DNNs. The PH reveals the combinational effects of multiple neurons in DNNs at different resolutions, which is difficult to be captured without using PH. Evaluations were conducted using fully connected networks (FCNs) and networks combining FCNs and convolutional neural networks (CNNs) trained on the MNIST and CIFAR-10 data sets. Evaluation results demonstrate that the PH of DNNs reflects both the excess of neurons and problem difficulty, making PH one of the prominent methods for investigating the inner representation of DNNs.

Keywords Deep neural network · Convolutional neural network · Persistent Homology · Topological data analysis

Mathematics Subject Classification (2010) 68T07 · 55N31

1 Introduction

Deep neural networks (DNNs) have demonstrated a remarkable performance in various fields including image analysis, speech recognition, and text classification [16, 45]. However, the inner representations of DNNs are indecipherable, which makes it difficult to tune DNN models, control their training process, and interpret their outputs. Many approaches

✉ Satoru Watanabe
satoru.watanabe.aw@hitachi.com

Hayato Yamana
yamana@waseda.jp

[1] Graduate School of Fundamental Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan

enabling the understanding of the inner representation of DNNs have been investigated, including the input identification of specific results [2, 26, 34, 44] and similarity evaluation between different networks [20, 27, 30]. At the same time, the complexity of DNNs is one of the essential subjects, which represents the knowledge in trained DNNs.

In this paper, we propose a novel approach to investigate the inner representation of DNNs using topological data analysis (TDA). TDA employs results from geometry and topology [28, 40], which has provided new insights in various fields such as neuroscience [8, 10, 29, 36, 43], proteomics [7, 14, 42], and material science [18, 21].

Persistent homology (PH) is one of the prominent methods in TDA owing to its three advantages: theoretical foundation, computability in practice, and robustness with small perturbations [28]. These advantages are beneficial for investigating DNNs. Theoretical foundation and computability are fundamental in constructing knowledge from empirical observations, while robustness is indispensable for investigating DNNs involving parameter perturbations.

Bastian et al. investigated the complexity of the inner representation of DNNs using zero-dimensional PH, which counts the number of connected neurons at different resolutions [32]. At the same time, one-dimensional PH can reveal other essential aspects of the knowledge complexity in DNNs because it can examine the combinational effects of multiple neurons. To the best of our knowledge, there is no previous work employing one-dimensional PH for investigating the inner representation of DNNs based on the trained weight parameters except our presentation at a symposium [41].

We constructed clique complexes, which were employed for analyzing brain networks [31], on trained DNNs. Furthermore, we calculated the one-dimensional PH of fully connected networks (FCNs) and networks combining FCNs and convolutional neural networks (CNNs) trained on the MNIST and CIFAR-10 data set to demonstrate the effectiveness of one-dimensional PH.[1]

The remainder of this paper is organized as follows. Section 2 presents the intuition behind this study. Background information is presented in Section 3. Clique complexes are constructed on trained DNNs in Section 4. The evaluation setup and results are provided in Sections 5 and 6, respectively. Section 7 discusses the assumptions and applications of the measurement method. Related work is discussed in Section 8. Conclusions and suggestions for future work are presented in Section 9.

2 Intuition behind topological measurement of DNNs

DNNs work as knowledge distilling pipelines, meaning that the degree of feature abstraction increases with the depth of DNN layers [23]. For example, images of cats are incrementally abstracted from pixels to diagonal lines and ear shapes. Additionally, DNNs can detect cats based on feature combinations [9]. Feature relationships represent the implementation of knowledge in DNNs, which can be investigated from DNN structures.

Previous studies have demonstrated that PH can be used for comparing and characterizing human brains. Cassidy et al. employed PH as a tool for comparing human brains using functional magnetic resonance imaging (fMRI) [8]. Petri et al. demonstrated that psilocybin affects the homological structure of the brain's functional patterns [29]. Furthermore,

[1]The source code used in the evaluation can be accessed at https://github.com/satoru-watanabe-aw/DNNtopology.

Sizemore et al. employed PH to highlight the crucial features of human brains from diffusion spectrum imaging (DSI) [36]. However, it is often difficult to quantify the activation of neurons from fMRIs and DSIs. Hence, PH is more useful for analyzing DNNs because their network structures and the activation of neurons can be described mathematically. In this study, we employed PH to investigate the process of training a DNN and evaluate its knowledge representation complexities.

3 Background

The terms of TDA and PH can be understood based on previous studies [11, 19, 28], while introductory videos explaining TDA and PH can be found on on-demand video services.[2]

3.1 Persistent homology

The homology groups of orders zero and one represent the number of connected components and holes, respectively. PH is a method for computing the homology groups at different resolutions. While the formal definition of PH is provided below, its intuitive understanding is sufficient for interpreting the presented experimental results obtained using some computational libraries.

Definition 1 An abstract simplicial complex is a finite collection of sets $\mathscr{K}$ such that $X \in \mathscr{K}$ and $Y \subseteq X$ implies $Y \in \mathscr{K}$.

The sets X in $\mathscr{K}$ denote its simplices. The dimension of a simplex is $\dim X = \mathrm{card}\, X - 1$, where $\mathrm{card}\, X$ denotes the cardinality of X. The dimension of an abstract simplicial complex is the maximum dimension of any of its simplices. The vertex set is the set consisting of all the simplices of dimension 0, while the face of a simplex X is a non-empty subset $Y \subseteq X$.

A p-chain c of a simplicial complex $\mathscr{K}$ is a formal sum of p-simplices in $\mathscr{K}$, that is, $c = \sum a_i X_i$, where X_i are p-simplices and a_i are coefficients. We employ module-2 coefficients, that is, a_i are either 0 or 1 and $1+1=0$. The binary arithmetic of two p-chains $c = \sum a_i X_i$ and $c' = \sum b_i X_i$ is defined as $c + c' = \sum (a_i + b_i) X_i$, where the coefficients are of modulo-2. The p-chain forms a group denoted as C_p.

A boundary operator ∂_p is a map from a p-simplex to the sum of its $(p-1)$-simplices. Formally, $\partial_p X = \sum_{j=0}^{p} [v_0, \ldots, \hat{v_j}, \ldots, v_p]$, where $[v_0, \ldots, v_p]$ is the simplex with vertices, while the hat indicates that v_j is removed. A chain complex is the sequence of chain groups connected by boundary operators, $\cdots \xrightarrow{\partial_{p+2}} C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \xrightarrow{\partial_{p-1}} \cdots$. A p-cycle is a p-chain with an empty boundary forming a group denoted as $Z_p = \ker \partial_p$. A p-boundary is a p-chain, that is, the image of a $(p+1)$-chain forming a group denoted as $B_p = \mathrm{im}\, \partial_{p+1}$.

Definition 2 The p-th homology group denoted as $H_p (= Z_p / B_p)$ is the p-th cycle group modulo the p-th boundary group. The p-th Betti number β_p is the rank of H_p.

Definition 3 A filtration of the simplicial complex $\mathscr{K}$ is a sequence of simplicial complex such that $\emptyset = K_0 \subset K_1 \subset \cdots \subset K_n = \mathscr{K}$.

[2] https://www.youtube.com/watch?v=akgU8nRNIp0, https://www.youtube.com/watch?v=2PSqWBIrn90

For every $i \leq j$, there is an induced homomorphism in each dimension p, $f_p^{i,j}$ from $H_p(K_i)$ to $H_p(K_j)$. $f_p^{i,j}$ satisfies the condition of $f_p^{k,j} \circ f_p^{i,k} = f_p^{i,j}$ for all $0 \leq i \leq k \leq j \leq n$.

Definition 4 Let $\emptyset = K_0 \subset K_1 \subset \cdots \subset K_n = \mathcal{K}$ be a filtration. The p-th PH of $\mathcal{K}$ is the pair $(\{H_p(K_i)\}_{0 \leq i \leq n}, \{f_p^{i,j}\}_{0 \leq i \leq j \leq n})$, where the homomorphism $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$ represents the maps induced by including maps $K_i \to K_j$.

A homology $\gamma \in H_p(K_i)$ can be said to be born at K_i if $\gamma \notin \mathrm{im} f_p^{i-1,i}$. Furthermore, if γ is born at K_i, then it dies entering K_j if $f_p^{i,j-1}(\gamma) \notin \mathrm{im} f_p^{i-1,j-1}$ but $f_p^{i,j}(\gamma) \in \mathrm{im} f_p^{i-1,j}$. The lifetime of γ is represented by the half-open interval $[i, j)$. If $f_p^{i,j}(\gamma) \neq 0$ $(i \leq \forall j \leq n)$, γ can be said to live forever, and its lifetime is the interval $[i, \infty)$.

3.2 Diagrams

A PH diagram illustrates the birth and death of homologies in a filtration, which was fundamentally introduced in [3]. Figure 1(a) shows points with oblique lined circles in $\mathbb{R}^2$. When the radius of the circles is small, the points are isolated. Two encircled regions appear in $\mathbb{R}^2$ when the circles are gradually enlarged. The appearance of the encircled regions corresponds to the birth of homologies. The regions disappear when the circles are enlarged further, and the disappearances correspond to the death of homologies.

Figure 1b shows the PH diagram of Fig. 1a, in which the X-axis shows the birth of homologies and the Y-axis the death of them. The two points in Fig. 1b correspond to the births and deaths of the two regions. The large region in Fig. 1a is stable with regard to the enlargement of the circles. In contrast, the small region is less stable compared to the large region. The stability of the regions is indicated by the distance from the dialog line in Fig. 1b, i.e., the small region is pointed near the dialog line, whereas the large region is pointed in a distance from the dialog line.

Barcode is another diagram that gives the same information as the PH diagram. Barcode diagram of Fig. 1a is shown in Fig. 1c, in which the start and end points of lines parallel to the X-axis show the birth and death of homologies, respectively. The short and long lines correspond to the small and large regions, respectively. The stability of regions is indicated by the length of the bars in the barcode diagrams.

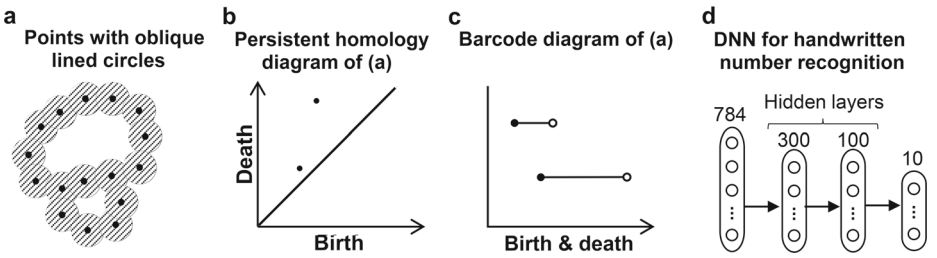

Fig. 1 **a** Examples of persistent homology diagrams; **b** persistent homology diagram of **a**; **c** barcode diagram of **a**; **d** DNN for handwritten number recognition

4 Construction of clique complexes on DNNs

We consider a set of neurons as vertices $V = \{v_0, \ldots, v_n\}$, where $n+1$ is the number of neurons. DNNs are considered as directed graphs with weights w_{ij}, where w_{ij} denotes the weight between v_i and v_j; here, w_{ij} is zero if v_i and v_j are not connected. We set the value of the relevance of identical neurons to one and the relevance R_{ij} between the connected neurons v_i and v_j as the normalized weight. Formally we set

$$R_{ij} = \begin{cases} 1 & (i = j) \\ w_{ij}^{+} / \sum_{i, i \neq j} w_{ij}^{+} & (i \neq j), \end{cases} \tag{1}$$

where w_{ij}^{+} denotes the positive part of the weight, i.e. $w_{ij}^{+} = \max\{0, w_{ij}\}$. R_{ij} indicates the relevance between v_i and v_j because the input to the j-th neuron is calculated by $\sum_i a_i w_{ij} + b_j$ in DNNs, where a_i is the activation of the i-th neuron and b_j is the bias [9]. We employed the positive part of the weight and ignored the bias, in a manner similar to the z^{+}-rule defined in deep Taylor decomposition [26].

To construct clique complexes on DNNs, the relevance was extended to indirectly connected neurons. For example, when v_0 and v_2 are connected to a path $v_0 \to v_1 \to v_2$, the relevance between v_0 and v_2 corresponding to the path is defined as $R_{01}R_{12}$. The intuition behind the definition is as follows: R_{01} and R_{12} indicate the contributions of v_0 and v_1 to the increase in the inputs of v_1 and v_2, respectively; $R_{01}R_{12}$ indicates the contribution of v_0 to the increase in the input of v_2. Formally we set

$$\widetilde{R_{ij}} = \max_{(v_i, v_{m_1} \ldots, v_{m_k}, v_j) \in L_{ij}} R_{v_i v_{m_1}} \cdots R_{v_{m_k} v_j}, \tag{2}$$

where L_{ij} denotes the set of all possible paths from v_i to v_j. It is possible to define $\widetilde{R_{ij}}$ using multiple paths in L_{ij}. However, the maximum was employed in (2) to improve computational efficiency.

Masulli et al. constructed a clique complex $K(G)$ on a finite directed weighted graph $G = (V, E)$ with vertex set V and edge set E with no self-loops and no double edges [25]. They defined the clique complex $K(G)$ as $K(G)_0 = V$ and $K(G)_p = \{(v_{K_0}, \ldots, v_{K_p}) \ ; \ v_{K_i} \in V, (v_{K_i}, v_{K_j}) \in E \text{ for all } K_i < K_j\}$ (for $p \geq 1$), where $K(G)_p$ denotes the set of p-simplices on G.

Correspondingly, $\widetilde{R_{ij}}$ enables the construction of a clique complex and filtration on V. The neurons were numbered in ascending order from the output to input layers. Hence, the numbers of neurons in the closer layer to the output layer are smaller than those in the farther layer, where the distance is indicated by the number of edges from the output layer. Using this numbering, we set p-simpleces on V as

$$K_p^t = \begin{cases} V & (p = 0) \\ \{(v_{k_0}, \ldots, v_{k_p}) \ ; \ v_{k_i} \in V, \widetilde{R_{k_i k_j}} \geq t \ \text{ for all } k_i > k_j\} & (p \geq 1), \end{cases} \tag{3}$$

where t is a threshold value ($0 \leq t \leq 1$).

Proposition 1 *Let $V = (v_0, \ldots, v_n)$ be a finite set, and $\{w_{ij}\}$ ($0 \leq i, j \leq n$) be a set of real numbers. Let $\widetilde{R_{ij}}$ ($0 \leq i, j \leq n$) be the relevance defined by (1) and (2) using $\{w_{ij}\}$. Let K_p^t be the p-simplices defined by (3), where t is a threshold value ($0 \leq t \leq 1$). Then, a finite collection of sets $K^t = K_0^t \cup K_1^t \cup \cdots \cup K_n^t$ is an abstract simplicial complex.*

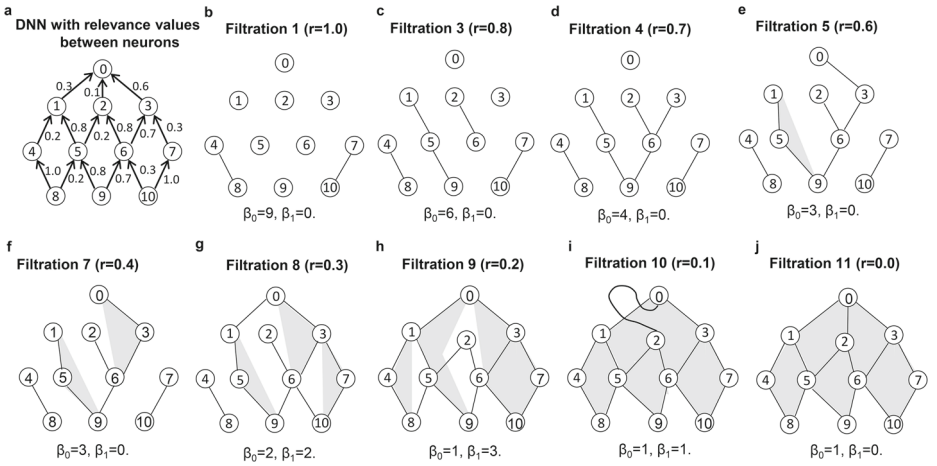

Fig. 2 **a** Example of DNN with weights; **b**–**h** simplicial complexes and betti numbers corresponding to the filtration

Proof Let $X = \{v_{X_0}, \ldots, v_{X_p}\}$ be an element of K^t. Then, $\widetilde{R_{X_i X_j}}$ is greater than or equal to t for all $X_i > X_j$. Let $Y = \{v_{Y_0}, \ldots, v_{Y_q}\}$ be a subset of X. Then, $\widetilde{R_{Y_i Y_j}}$ are greater than or equal to t for all $Y_i > Y_j$. Therefore, $X \in K^t$ and $Y \subseteq X$ imply $Y \in K^t$. □

Proposition 2 *Let* $(t_i)_{i=1}^n$ *be a monotonically decreasing sequence ranging from 1 to 0. Then,* $K_0 = \emptyset$ *and* $K_i = K^{t_i}$ $(1 \leq i \leq n)$ *form a filtration of* K^{t_n}.

Proof $K_p^{t_k}$ is included in $K_p^{t_l}$ $(1 \geq t_k > t_l \geq 0)$ from (3). It implies $\emptyset = K_0 \subset K_1 \subset \cdots \subset K_n = K^{t_n}$. □

Figure 2a illustrates a four-layered DNN with an output neuron v_0. The values adjacent to the arrows denote the weight between two neurons, and the weight matrix is presented in Fig. 3a where the (i,j) element denotes the weight between the i-th and j-th neurons. Figure 2b illustrates the simplicial complex of $K_{r=1.0}$ with Betti number $\beta_0 = 9$. The decrease of the Betti number β_0 according to the filtration can be observed in Fig. 2c to h. Figure 2e illustrates a 2-simplex represented with the gray triangle.

Figure 2g and h illustrate the increase of the Betti number β_1 corresponding to the occurrences of the cycle. If the vertices representing the features of input images are connected

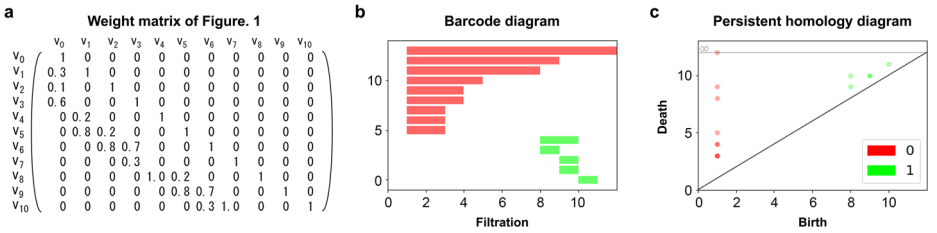

Fig. 3 **a** Weight matrix of Fig. 2a; b,c barcode and PH diagrams illustrated using GUDHI library

Algorithm 1 Algorithm for obtaining simplexes from a vertex s using a threshold t.

procedure GETSIMPLEX(M, s, t) ▷ where M: $n \times n$-matrix, s: array, t: threshold
  $relevance \leftarrow 1.0, result \leftarrow \emptyset, origin \leftarrow s[0]$
  for $dest = s[0]$ to $s[|s|-1]$ **do** ▷ calculate the relevance from $s[0]$ to $s[|s|-1]$.
    $relevance \leftarrow relevance \times M[origin][dest]$ ▷ $s[|s|-1]$ is the last element of s.
    $origin \leftarrow dest$
  if $relevance \geq t$ **then**
    $result.append(combination(s))$ ▷ append all the combinations of the elements in s.
    $lastPoint \leftarrow s[|s|-1]$
    for $i = 0$ to $n-1$ **do** ▷ check if the last point has connections.
      if $M[lastPoint][i] > 0$ and $i \neq lastPoint$ **then**
        $ss \leftarrow$deep copy of s
        $recResult \leftarrow getSimplex(M, ss.append(i), t)$ ▷ recursive call with extended array.
        for e in $recResult$ **do**
          $result.append(combination(e))$ ▷ append all the combinations of the elements in e.
  return $unique(result)$ ▷ return deduplicated array

straightforwardly to the output neurons, the knowledge in the DNN is considered to be simple because it is equivalent to feature detection. In contrast, the increase of the Betti number β_1 indicates that the DNN classifies the input based on the combination of features. From these viewpoints, we can assume the increase in the Betti number β_1 reflects the complexity of knowledge in the DNN. Filtration 10 (Fig. 2i) has Betti number $\beta_1 = 1$. While [0, 2] is a simplex in Filtration 10, it is not included in another simplex [0,. . .,10] and produces $\beta_1 = 1$.

The computation of PH involves the explosion of the complexity caused by the increase of vertices, several implementations of which are publicly available [28]. We employed the GUDHI [6, 33, 39], JavaPlex [38], and Dionysus 2 [12, 13] libraries for the computation and visualization. These libraries require registering simplexes in each filtration to calculate PH.

Algorithm 1 identifies all simplexes from a vertex s up to the limit of the threshold of relevance t using the recursive procedure call. All simplexes in each filtration are identified using this procedure and registered to the libraries. Figure 3b and c are barcode and PH diagrams illustrated by the GUDHI library, respectively. The library employed red and green for indicating zero- and one-dimensional homologies, respectively. The Betti numbers in Fig. 3b correspond to the number of the intersections between the bars and the perpendicular lines to the X-axis (remembering that the lifetime of homologies is defined by the half-open interval $[birth, death)$). The GUDHI library illustrates Betti numbers using color shades in PH diagrams shown in Fig. 3c. PH was calculated using the Dionysus 2 and JavaPlex libraries, resulting in the same diagrams.

A filtration is defined using thresholds of relevance. This study considered 64 threshold values composed with $(1.0^0, \ldots, 1.0^{-7})$ and eight interval values between the adjacent values. Formally, we considered the simplicial complexes $K_{n(r=(1-0.1\times(l-1))\times 10^{-m})} (1 \leq n \leq 64)$, where m and l are the quotient and remainder when n is divided by 9, respectively. And the filtration was defined as $K_{1(r=1.0)} \subset K_{2(r=0.9)} \subset \cdots \subset K_{10(r=1.0^{-1})} \subset K_{11(r=0.09)} \subset \cdots \subset K_{64(r=1.0^{-7})}$. While the thresholds should be considered depending on the network

Table 1 Overview of the data sets and network types employed in this study

Data set	Content	Data size	Network type
MNIST	handwritten digits	784 (28 × 28 grayscale)	FCN
CIFER-10	photographs	3072 (32 × 32 color)	CNN, FCN

structure of DNNs, we set this aside as a task for future work; this study only examined the prominence of the topological measurement of DNNs.

5 Evaluation setup

The MNIST and CIFAR-10 data sets were employed in the evaluation [22, 24]. As shown in Table 1, the contents of the MNIST and CIFAR-10 data sets are 28×28 grayscale handwritten digits and 32×32 color photographs, respectively. The CIFAR-10 data set comprises the photographs of 10 types of objects such as airplanes, automobiles, birds, etc. All experiments were conducted using Keras and Tensorflow [1, 9], and DNNs were developed based on the examples in Keras 2.3.0.

For the classification of the MNIST data set, we employed an FCN with two hidden layers of sizes 300 and 100, the ReLU activation function in the hidden layers and 10 output neurons with the sigmoid activation function (Fig. 1d). The models were traind for 10 epochs with a batch size of 64, and all models achieved an accuracy of over 97% on the test data.

For the classification of the CIFAR-10 data set, we employed DNNs consisting of a CNN and an FCN. The CNN was used to extract features from the photographs, while the FCN was used to classify the photographs based on the combination of the features. The proposed method was applied to the FCN since the purpose of this study was to examine the complexity of the knowledge in DNNs represented in the combination of features.

We employed the CNN from an example network included in Keras 2.3.0 without modifications. This CNN comprises multiple layers, including two-dimensional convolution, max pooling, and dropout layers. Two FCNs with sizes of (300, 100, 10) and (512, 512, 10) were used for examining the sensitivity of the proposed method to the network structures.[3] The DNNs were trained for 30 epochs with a batch size of 32.

6 Evaluation results

6.1 MNIST data set

Figures 4a–j illustrate PH diagrams of the FCNs produced using the Dionysus 2 library, where the number of input digits used to train the FCN models was varied. In particular, we extracted the images of the target digits from the MNIST data set and trained FCN models

[3]The following network structures are employed: input(3072)–Conv2D(32 filters, 3 × 3 kernel, ReLu activation)–Conv2D(32 filters, 3 × 3 kernel, ReLu activation)–MaxPooling2D(2×2 pool)–Dropout(dropout ratio 0.25)–Conv2D(64 filters, 3 × 3 kernel, ReLu activation)–Conv2D(64 filters, 3 × 3 kernel, ReLu activation)–MaxPooling2D(2 × 2 pool)–Dropout(dropout ratio 0.25)–Flatten–Dense(300 or 512, ReLu activation)–Dropout(dropout ratio 0.5)–Dense(100 or 512, ReLu activation)–Dense(10, softmax activation).

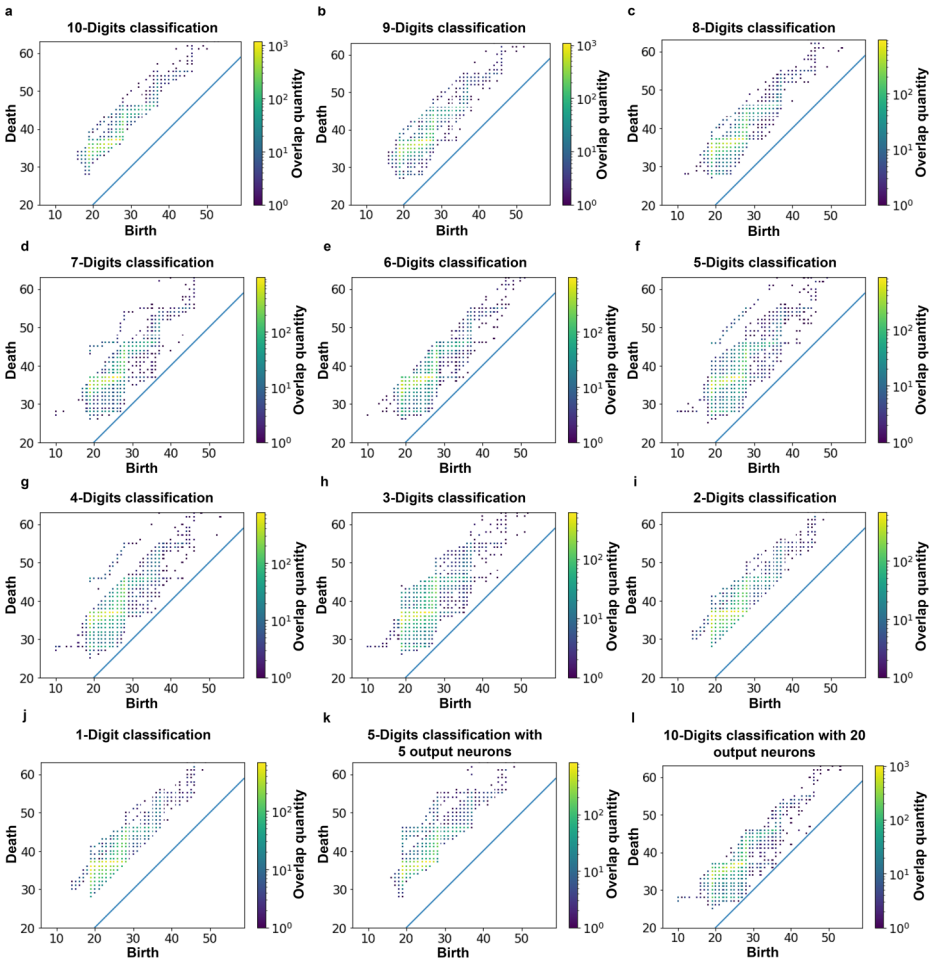

Fig. 4 **a**–**j** PH diagrams of the FNC models trained to classify handwritten digits based on a varying number of input digits from 10 to 1; **k** persistent diagram of the FCN model trained to classify five digits using five output neurons; **l** persistent diagram of the FCN model trained to classify 10 digits using 20 output neurons

using the images of digits 0–9 (Fig. 4a), digits 0–8 (Fig. 4b), and so on. The Dionysus 2 library allows to visualize the overlapping quantity of homologies using different colors as indicated by the legends in Fig. 4. The values of birth and death in the axes on PH diagrams indicate the order of the 64 threshold values defined in Section 4. Let m and l are the quotient and remainder when the values of birth and death are divided by 9, respectively, the threshold values corresponding to the values in the axes on PH diagrams are $(1 - 0.1 \times (l - 1)) \times 10^{-m}$. This correspondence is consistent through the paper.

The following three observations can be made from Fig. 4a–j: (1) points are plotted in the belt-like area ($birth + 5 < death < birth + 20$) parallel to the dialog line; (2) some figures have points below the belt-like area; and (3) some figures have points over the belt-like area.

With respect to observation (2), the number of points below the belt-like area increases from Fig. 4a to g and decreases from Fig. 4h to j. This pattern reflects both the excess of the output neurons and problem difficulty. It can be further observed that the diagrams

Table 2 Number of points in Figs. 4a–e, i, and j

	(a)	(b)	(c)	(d)	(e)	(i)	(j)
Total number	16,420	16,399	16,150	16,222	16,133	15,857	15,531
(c1)	N/A	1,317	2,034	1,700	2,972	8,226	13,123
(c2)	0	45	26	254	273	0	0
(c1) and (c2)	N/A	45	26	254	40	0	0

seem to reflect the degree of confidence of the FCN models, i.e., the excess of the output neurons reduced the confidence, whereas the simplicity of the problem increases it. For further investigation, we classified five digits using five output neurons (Fig. 4k) and 10 digits using 20 output neurons (Fig. 4l). In contrast to Fig. 4f, the points below the belt-like area disappeared in Fig. 4k. The opposite can be observed in Figs. 4a and l.

Table 2 lists the number of points plotted in Fig. 4a–e, i, and j. We categorized the points using the representative cycles calculated by the JavaPlex based on the following two conditions: (c1) the homology includes unused output neurons and (c2) the points are under the belt-like area ($death \leq birth + 5$). While the number of points that include unused output neurons in Fig. 4i and j is more than twice of that in Fig. 4e, these points are not plotted below the belt-like area. The simplicity of the problem led to no points being plotted under the belt-like area.

6.2 CIFAR-10 data set

Figure 5a–j illustrate PH diagrams of the DNN models combining a CNN and an FCN (300, 100, 10), where the number of classes used to train the models was varied. In particular, we extracted photographs of the target classes from the CIFAR-10 data set and trained the DNN models using the photographs of 10 classes (Fig. 5a), nine classes (Fig. 5b), and so on.

As described in Section 5, the contents of the CIFAR-10 data set differs from that of the MNIST data set in terms of the image size, tone, and represented object. Unlike FCN-based models traind on the MNIST data set, CNNs were employed in addition to FCNs to classify the CIFAR-10 data set.

Despite these differences, Fig. 5 demonstrate similar patterns to those in Fig. 4. In particular, the points under the belt-like area appear only in Fig. 5d–h; k, where the photographs of five classes are classified using five output neurons, has no points under the belt-like area, whereas Fig. 5l, where the photographs of 10 classes are classified using 20 output neurons, has points under the belt-like area.

A further experiment was conducted using the DNN models combining a CNN and an FCN (512, 512, 10). The results of this experiment are illustrated in Fig. 6. A similar patterns regarding the appearance and disappearance of points under the belt-like area can be observed from Fig. 6; that is, only Fig. 6d–h and l have the points under the belt-like area. This result suggests that the observation is robust to not only the network type and content of data sets but also number of neurons in FCNs.

Two additional observations can be made from Figs. 5 and 6: (i) the numbers of points in Fig. 6 are larger than those in Fig. 5; (ii) the sizes of the areas that points are plotted in Fig. 6 are larger than those in Fig. 5. Tables 3 and 4 list the numbers of points and sizes of the convex hull of the points plotted in Figs. 5a–j and 6a–j, respectively. The numbers of

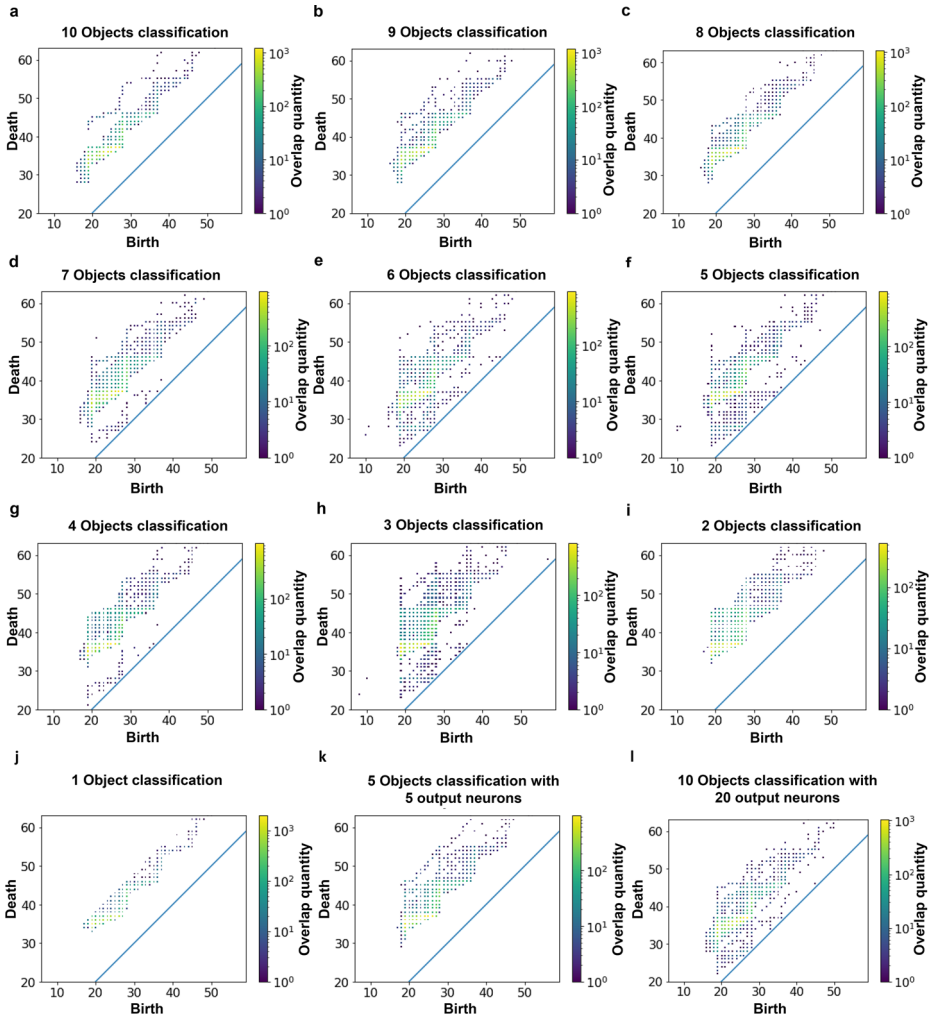

Fig. 5 **a**–**j** PH diagrams of the DNNs using the FCN (300, 100, 10) trained to classify photographs based on a varying number of input classes from 10 to 1; **k** PH diagram of the DNN using the FCN (300, 100, 10) trained to classify five classes using five output neurons; **l** PH diagram of the DNN using the FCN (300, 100, 10) trained to classify 10 classes using 20 output neurons

points in Fig. 6 are 8.81 to 9.31 times larger than those in Fig. 5. The sizes of the convex hulls in Fig. 6 are 1.05 to 2.57 times larger than those in Fig. 5.

The number of points reflects the difference of expressiveness of the FCN (512, 512, 10) and FCN (300, 100, 10). The FCN (512, 512, 10) has more parameters compared to the FCN (300, 100, 10), which results in the ability of the FCN (512, 512, 10) to learn knowledge is higher than that of the the FCN (300, 100, 10) and produces many homologies. As a rough approximation, the FCN (512,512,10) has $512 \times 512 + 512 \times 10$ of weight parameters, whereas the FCN (300, 100, 10) has $300 \times 100 + 100 \times 10$ of them. The ratio $8.62\ (= (512 \times 512 + 512 \times 10)/(300 \times 100 + 100 \times 10))$ provides the explanation for the increase in the values listed in Table 3.

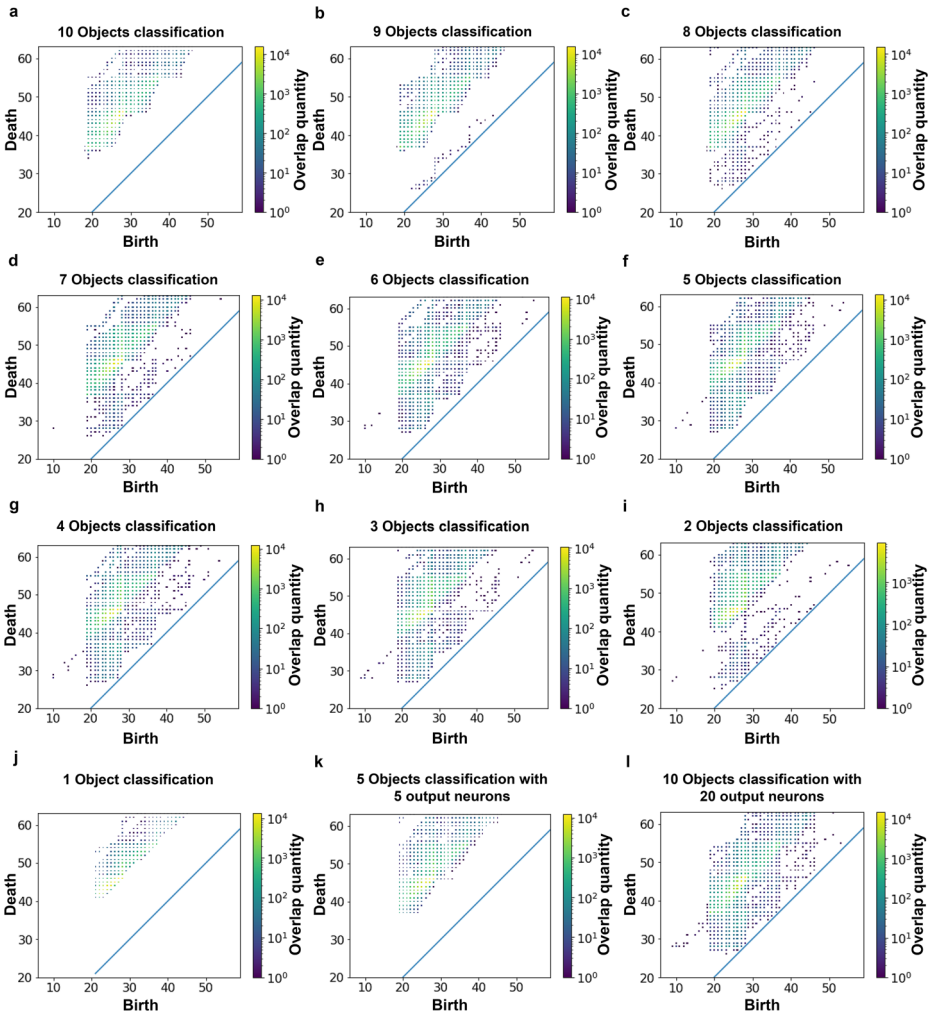

Fig. 6 **a**–**j** PH diagrams of the DNN using the FCN (512, 512, 10) trained to classify photographs based on a varying number of input classes from 10 to 1; **k** PH diagram of DNN using the FCN (512, 512, 10) trained to classify five classes using five output neurons; **l** PH diagram of DNN using the FCN (512, 512, 10) trained to classify 10 classes using 20 output neurons

The increase in the size of convex hull is smaller than that of the number of points, which indicates that the FCNs (512, 512, 10) have duplicated homologies approximately 4 to 8 times more often compared to the FCNs (300, 100, 10). It implies that the FCNs (512, 512, 10) have duplicated homologies with different neurons, which can be achieved with expressive training to the data set. The interpretation of the PH diagrams requires further investigation, which we left as a task for future work because the purpose of this study was only to examine the prominence of the topological measurement of DNNs.

Table 3 Number of points in Figs. 5a–j and 6a–j

	(A) Fig. 5: FCN (300, 100, 10)	(B) Fig. 6: FCN (512, 512, 10)	(B) / (A)
(a)	16,214	142,768	8.81
(b)	16,278	139,783	8.59
(c)	15,702	142,016	9.04
(d)	15,421	141,027	9.15
(e)	15,274	138,732	9.08
(f)	15,759	136,508	8.66
(g)	14,878	133,503	8.97
(h)	14,348	124,919	8.71
(i)	11,496	106,983	9.31
(j)	15,073	132,775	8.81

6.3 Robustness on weight initialization

We conducted additional experiments by varying the initial values of network weights to investigate the robustness of the PH diagrams' transitions described in Sections 6.1 and 6.2. Keras framework starts the training with random initial values of network weights [9]. We repeated each experiment 10 times by varying the number of input classes from 10 to 1 with the three network types, MNIST (300, 100, 100), CIFAR-10 (300, 100, 100), and CIFAR-10 (512, 512, 10), resulting in a total of 300 additional experiments.

Figure 7 shows the minimum, average, and maximum size of convex hulls of the points in the PH diagrams. The differences between the maximum and minimum values indicate the degree of vibration of the experiment results. All the three graphs are approximately convex upward, indicating that the PH diagrams transit the shape in a similar manner to those described in Sections 6.1 and 6.2, and the transitions are robust on the initial values of network weights.

Table 4 Size of the convex hull in Figs. 5a–j and 6a–j

	(A) Fig. 5: FCN (300, 100, 10)	(B) Fig. 6: FCN (512, 512, 10)	(B) / (A)
(a)	445.5	492.5	1.11
(b)	477.0	737.5	1.55
(c)	406.0	881.0	2.17
(d)	710.5	1029.5	1.45
(e)	823.0	959.5	1.17
(f)	836.0	904.8	1.08
(g)	634.5	964.5	1.52
(h)	992.0	1041.5	1.05
(i)	413.5	1061.0	2.57
(j)	232.5	254.0	1.09

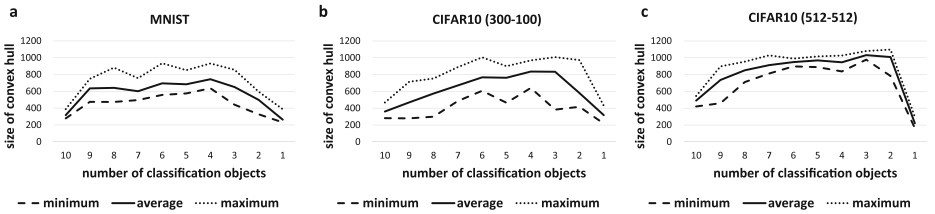

Fig. 7 **a–c** Size of the convex hull of points in the PH diagrams with MNIST using the FCN (300, 100, 10), CIFAR-10 using the FCN (300, 100, 10), and CIFAR-10 using FCN (512, 512, 10) by varying the number of input classes, respectively

In Sections 6.1 and 6.2, we observed the transition of the PH diagrams that the number of points near the dialog line ($death \leq birth + 5$) changes by varying the number of input classes. No point near the dialog line appeared when the number of input classes was set to 10 and 1. Additionally, the number of points near the dialog line increased and decreased with the decrease in the number of input classes from 10 to 8 and 3 to 1, respectively.

Table 5 lists the minimum, average, and maximum numbers of points near the dialog line regarding the additional experiments. We observed that no point appeared near the dialog line when the number of input classes was set to 10 and 1 in all the additional experiments. Additionally, the increase and decrease followed the same trend in the additional experiments, shown in Table 5, meaning that the observations obtained in Sections 6.1 and 6.2 are robust on the initial values of network weights.

7 Discussion

In this section, the assumptions used in this study are explained and the application of the topological measurement of DNNs is discussed.

Table 5 Number of points near the dialog line ($death \leq birth + 5$)

Number of input classes	MNIST			CIFAR-10 (300-100)			CIFAR-10 (512-512)		
	min.	avg.	max.	min.	avg.	max.	min.	avg.	max.
10	0	0	0	0	0	0	0	0	0
9	57	96	132	0	11	59	0	115	234
8	110	150	199	0	33	102	79	273	497
7	141	209	297	0	78	143	278	375	451
6	141	269	348	0	136	284	209	376	571
5	137	332	528	0	142	334	52	380	620
4	111	308	524	48	196	321	13	423	823
3	46	131	207	0	158	365	591	764	909
2	0	0	1	0	36	252	145	581	936
1	0	0	0	0	0	0	0	0	0

7.1 Assumptions

The assumptions of this study include the follows: (1) the knowledge in DNNs can be investigated from their network weights among neurons and (2) PH reveals the knowledge complexity of DNNs. The first assumption is acceptable because the weights are the outcome of the training process. The second assumption is based on the observations from previous works described in Section 2 [9, 23]. PH reveals the births and deaths of feature combinations, which are difficult to be captured without using PH. The effectiveness of the second assumption can be evaluated from the usability, which changes depending on the application.

7.2 Applications

One of the most important applications of the proposed method is recognizing the quality of DNN training. In particular, the performance of DNNs can deteriorate for many reasons, including a shortage of data, overfitting, and improper hyper-parameter setting [4, 37]. Our results imply that the shortage of data can be indicated by the PH, that is the excess of the output neurons produces homologies near the dialog line. Furthermore, the proposed method is beneficial for selecting appropriate DNN architectures, which is one of the major challenges when utilizing DNNs [35, 46].

8 Related work

Bianchini et al. investigated the upper and lower bounds of network complexity from the viewpoint of PH [5]. Based on their results, Guss et al. empirically analyzed the relationship between the upper bound of network complexity and data complexity measured by PH to determine appropriate network architecture for a given data set [15]. However, these two types of complexities are not homogeneous, and their comparability is uncertain. Under these considerations, we addressed the inner representations of DNNs with small perturbations. Our evaluation results revealed that small perturbations such as the number of output neurons and a variety of input data have significant impact on PH. Thus, the sensitivity of PH requires a careful investigation for securing comparability.

Bastian et al. investigated the complexity of the inner representation of DNNs using zero-dimensional PH [32]. Zero-dimensional PH counts the number of connected components in DNNs. Figure 2f and g have $\beta_0 = 3$ and $\beta_0 = 2$ corresponding to the connected components, respectively. In contrast, the Betti number β_1 reveals the combinations among neurons illustrated in Fig. 2g, where the neurons one and three collaborate to increase the Betti number β_1. Thus, we believe that one-dimensional PH can reveal the combination of neurons and access essential aspects of DNNs that are difficult to be accessed using other methods.

9 Conclusion

This paper introduced a novel approach to investigate the inner representation of DNNs using PH. Evaluations were conducted using FCNs and networks combining a CNN and an FCN trained on the MNIST and CIFAR-10 data sets. The evaluation results demonstrated that the one-dimensional PH of DNNs can reflect both the excess of neurons and problem

difficulty, which implies that PH can become one of the prominent methods for investigating the inner representation of DNNs.

The methods for constructing simplicial complexes and defining the filtration are developed on the basis of our attempts. The development of these methods will, however, include many research areas, especially due to large variety of network types, including CNNs and recursive neural networks (RNNs). Furthermore, with regard to computation, the development would require considerable efforts in applying the topological measurement to enlarged neural networks, which can have more than 1,000 layers [17]. At the same time, we believe that the topological measurement of DNNs is worth further investigation.

Acknowledgements This work was supported in part by JST CREST, Japan, under Grant JPMJCR1503. We are also grateful to Hitachi, Ltd. for the tuition subsidy. The founder had no role in study design and technical investigation in this paper.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {$USENIX$} Symposium on Operating Systems Design and Implementation ({$OSDI$ 16}), pp. 265–283 (2016)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
3. Barannikov, S.: The framed morse complex and its invariants (1994)
4. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 24, pp. 2546–2554. Curran Associates, Inc. (2011)
5. Bianchini, M., Scarselli, F.: On the complexity of neural network classifiers: A comparison between shallow and deep architectures. IEEE Trans. Neural Netw. Learn. Sys. **25**(8), 1553–1565 (2014)
6. Boissonnat, J.-D., Maria, C.: The simplex tree: An efficient data structure for general simplicial complexes. Algorithmica **70**(3), 406–427 (2014)
7. Cang, Z., Wei, G.-W.: Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. Int. J. Numer. Methods Biomed. Eng. **34**(2), e2914 (2018)
8. Cassidy, B., Bowman, F.D., Rae, C., Solo, V.: On the reliability of individual brain activity networks. IEEE Trans. Med. Imaging **37**(2), 649–662 (2018)
9. Chollet, F. Deep Learning with Python, 1st edn. Manning Publications Co., Greenwich (2017)
10. Curto, C.: What can topology tell us about the neural code? Bull. Am. Math. Soc. **54**(1), 63–78 (2017)
11. Edelsbrunner, H., Harer, J.: Computational topology: an introduction. American Mathematical Soc. (2010)
12. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 454–463. IEEE (2000)
13. Edelsbrunner, H., Morozov, D.: Persistent homology: theory and practice. Technical report, Lawrence Berkeley National Lab.(LBNL). Berkeley, CA (United States) (2012)

14. Gameiro, M., Hiraoka, Y., Izumi, S., Kramar, M., Mischaikow, K., Nanda, V.: A topological measurement of protein compressibility. Jpn. J. Ind. Appl. Math. **32**(1), 1–17 (2015)
15. Guss, W.H., Salakhutdinov, R.: On characterizing the capacity of neural networks using algebraic topology. arXiv:1802.04443 (2018)
16. Hatcher, W.G., Yu, W.: A survey of deep learning: platforms, applications and emerging research trends. IEEE Access **6**, 24411–24432 (2018)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
18. Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E.G., Matsue, K., Nishiura, Y.: Hierarchical structures of amorphous solids characterized by persistent homology. Proc. Natl. Acad. Sci. **113**(26), 7035–7040 (2016)
19. Horak, D., Maletić, S., Rajković, M.: Persistent homology of complex networks. Journal of Statistical Mechanics: Theory and Experiment **2009**(03), P03034 (2009)
20. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. arXiv:1905.00414 (2019)
21. Kramar, M., Goullet, A., Kondic, L., Mischaikow, K.: Persistence of force networks in compressed granular media. Phys. Rev. E **87**(4), 042207 (2013)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
24. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
25. Masulli, P., Villa, A.EP.: The topology of the directed clique complex as a network invariant. SpringerPlus **5**(1), 388 (2016)
26. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.-R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recogn. **65**, 211–222 (2017)
27. Morcos, A., Raghu, M., Bengio, S.: Insights on representational similarity in neural networks with canonical correlation. In: Advances in Neural Information Processing Systems, pp. 5727–5736 (2018)
28. Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.: A roadmap for the computation of persistent homology. EPJ Data Science **6**(1), 17 (2017)
29. Petri, G., Expert, P., Turkheimer, F., Carhart-Harris, R., Nutt, D., Hellyer, P.J., Vaccarino, F.: Homological scaffolds of brain functional networks. Journal of The Royal Society Interface **11**(101), 20140873 (2014)
30. Raghu, M., Gilmer, J., Yosinski, J., Sohl-Dickstein, J.: Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In: Advances in Neural Information Processing Systems, pp. 6076–6085 (2017)
31. Reimann, M.W., Nolte, M., Scolamiero, M., Turner, K., Perin, R., Chindemi, G., Dłotko, P., Levi, R., Hess, K., Markram, H.: Cliques of neurons bound into cavities provide a missing link between structure and function. Frontiers in computational neuroscience **11**, 48 (2017)
32. Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., Borgwardt, K.: Neural persistence: A complexity measure for deep neural networks using algebraic topology. arXiv:1812.09764 (2018)
33. Rouvreau, V.: Cython interface. In: GUDHI User and Reference Manual. GUDHI Editorial Board (2016)
34. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R.: Evaluating the visualization of what a deep neural network has learned. IEEE Trans. Neural Netw. Learn. Sys. **28**(11), 2660–2673 (2016)
35. Saxena, S., Verbeek, J.: Convolutional neural fabrics. In: Advances in Neural Information Processing Systems, pp. 4053–4061 (2016)
36. Sizemore, A.E., Giusti, C., Kahn, A., Vettel, J.M., Betzel, R.F., Bassett, D.S.: Cliques and cavities in the human connectome. J. Comput. Neurosci. **44**(1), 115–145 (2018)
37. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
38. Tausz, A., Vejdemo-Johansson, M., Adams, H.: JavaPlex: A research software package for persistent (co)homology. In: Hong, H., Yap, C. (eds.) Proceedings of ICMS 2014, Lecture Notes in Computer Science 8592, pp. 129–136. Software available at http://appliedtopology.github.io/javaplex/ (2014)
39. The GUDHI Project: GUDHI User and Reference Manual. GUDHI Editorial Board (2015)
40. Wasserman, L.: Topological data analysis. Annual Rev. Stat. Appl. **5**, 501–532 (2018)
41. Watanabe, S., Yamana, H.: Topological measurement of deep neural networks using persistent homology. International Symposium on Artificial Intelligence and Mathematics (2020)
42. Xia, K., Wei, G.-W.: Persistent homology analysis of protein structure, flexibility, and folding. Int. J. Numer. Methods Biomed. Eng. **30**(8), 814–844 (2014)

43. Yoo, J., Kim, E.Y., Ahn, Y.M., Ye, J.C.: Topological persistence vineyard for dynamic functional brain connectivity during resting and gaming stages. J. Neurosci. Methods **267**, 1–13 (2016)
44. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision, pp. 818–833. Springer (2014)
45. Zhang, Q., Yang, L.T., Chen, Z., Li, P.: A survey on deep learning for big data. Information Fusion **42**, 146–157 (2018)
46. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv:1611.01578 (2016)