Topological Measurement of Deep Neural Networks Using Persistent Homology

Satoru Watanabe, Hayato Yamana
Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo, Japan
satoru.watanabe.aw@hitachi.com, yamana@waseda.jp

Abstract

The inner representation of deep neural networks (DNNs) is indecipherable, making 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. We constructed simplicial complexes on DNNs based on deep Taylor decomposition and calculated the persistent homology (PH) of DNNs. Evaluation results demonstrated that the PH of DNNs reflected both the excess of neurons and problem difficulty, making PH one of the prominent methods for investigating the inner representation of DNNs.

Introduction

Deep neural networks (DNNs) have demonstrated a remarkable performance in various fields including image analysis, speech recognition, and text classification (Zhang et al. 2018; Hatcher and Yu 2018). However, the inner representations of DNNs are indecipherable, making it difficult to tune DNN models, control their training process, and interpret their outputs. Many approaches that enable the understanding of the inner representation of DNNs have been investigated, including the input identification of specific results (Bach et al. 2015; Zeiler and Fergus 2014; Samek et al. 2016; Montavon et al. 2017) and similarity evaluation between different networks (Raghu et al. 2017; Morcos, Raghu, and Bengio 2018; Kornblith et al. 2019).

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 (Otter et al. 2017; Wasserman 2018), which has provided new insights into various fields such as neuroscience (Sizemore et al. 2018; Cassidy et al. 2018; Curto 2017; Yoo et al. 2016; Petri et al. 2014), proteins (Cang and Wei 2018; Gameiro et al. 2015; Xia and Wei 2014), and material science (Hiraoka et al. 2016; Kramar et al. 2013). However, to the best of our knowledge, there is no previous work to employ TDA for investigating the inner representation of DNNs.

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 (Otter et al. 2017). These advantages are beneficial for investigating DNNs. The theoretical foundation and computability are fundamental in constructing knowledge from empirical observations, while the robustness is indispensable for investigating DNNs involving parameter perturbations (Szegedy et al. 2013).

In this study, we constructed simplicial complexes on a DNN based on deep Taylor decomposition (DTD) (Montavon et al. 2017; Lapuschkin et al. 2019). Further, we calculated the PH of the DNN trained to recognize handwritten digits for the purpose of demonstrating the effectiveness of TDA as a measurement method of DNNs.

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 (LeCun, Bengio, and Hinton 2015). 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 (Chollet 2017). Feature relationships represent the implementation of knowledge in DNNs, which can be investigated from their network 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) (Cassidy et al. 2018). Petri et al. showed that psilocybin affects the homological structure of the brain's functional patterns (Petri et al. 2014). Further, Sizemore et al. employed PH to highlight the crucial features of human brains from diffusion spectrum imaging (DSI) (Sizemore et al. 2018). 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 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.

Background

We introduce the terms of TDA and PH based on previous studies (Edelsbrunner and Harer 2010; Horak, Maletić, and Rajković 2009; Otter et al. 2017). Introductory videos explaining TDA and PH can be found on on-demand video services[1].

Persistent homology

Persistent homology is a method for computing the topological features of a space. Namely, the homology groups of order zero and one represent the number of connected components and holes, respectively. The formal definition of PH is provided in this subsection. However, this subsection can be bypassed with the intuitive understanding because the subject matters are computed by the libraries we employed.

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

We call the sets X in $\mathcal{K}$ as its simplices, and the dimension of a simplex is $\dim X = \mathrm{card} X - 1$ where $\mathrm{card} X$ is the cardinality of X. The dimension of abstract simplicial complex is the maximum dimension of any of its simplices. The vertex set is the set of all elements that lie in at least one simplex, and the face of a simplex X is a non-empty subset $Y \subseteq X$.

A p-chain c of a simplicial complex $\mathcal{K}$ is a formal sum of p-simplices in $\mathcal{K}$, that is, $c = \sum a_i X_i$ where X_i are p-simplices and a_i are the coefficients. We employ module 2 coefficients, that is, a_i are either 0 or 1 and $1 + 1 = 0$. A 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 modulo 2. The p-chain forms a group and the group is 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 the vertices and 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 empty boundary which forms a group and we denote the group as $Z_p = \ker \partial_p$. A p-boundary is a p-chain that is the boundary of a $(p+1)$-chain which forms a group and we denote the group as $B_p = \mathrm{im}\, \partial_{p+1}$.

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

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

For every $i \leq j$ we have an induced homomorphism in each dimension p, $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$. $f_p^{i,j}$ satisfies that $f_p^{k,j} \circ f_p^{i,k} = f_p^{i,j}$ for all $0 \leq i \leq k \leq j \leq n$ and closes in the filtration.

Definition 4 *Let $\emptyset = K_0 \subset K_1 \subset \cdots \subset K_n = \mathcal{K}$ be a filtration. The p-th persistent homology 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 $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$ are the maps induced by the inclusion maps $K_i \to K_j$.*

We say that a homology $\gamma \in H_p(K_i)$ is 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 $f_p^{i,j}(\gamma) \neq 0$ $(i \leq \forall j \leq n)$, we say that γ lives forever and its lifetime is the interval $[i, \infty)$.

Diagrams

A persistent homology diagram illustrates the birth and death of homologies in a filtration. Fig.1(a) shows points with oblique lined circles in $\mathbb{R}^2$. When the radius of the circles is small, the points are isolated. We gradually enlarge the circles, then two encircled regions appear in $\mathbb{R}^2$. The appearance of the encircled regions corresponds to the birth of homologies. The regions disappear if we continue to enlarge the circles, and the disappearances correspond to the death of homologies.

Fig.1(b) shows the persistent homology diagram of Fig.1(a). The two points in Fig.1(b) correspond to the births and deaths of the two regions. The large region in Fig.1(a) 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.1(b), i.e., the small region is pointed near the dialog line and the large region is pointed in a distance from the dialog line.

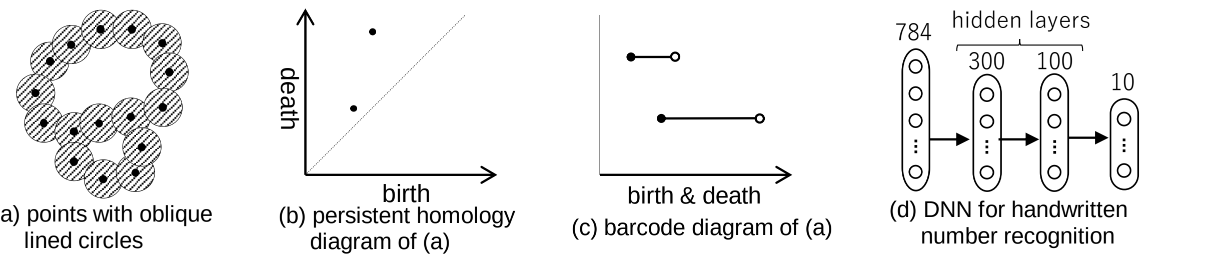
Barcode is another diagram that gives the same information with the persistent homology diagram. Barcode diagram of Fig.1(a) is illustrated in Fig.1(c), which illustrates the births and deaths by lines parallel to the x-axis. The short and long lines correspond to the small and large regions, respectively. The stability of regions is indicated by the length of bars in barcode diagrams.

Construction of simplicial complexes on DNNs

We consider a set of neurons as vertexes ($V = \{v_0, \ldots, v_n\}$) and DNNs as directed graphs with weights w_{ij}. We defined the value of relevance of identical neurons is one. If two neurons are not connected in a directed graph, the value between the two neurons is zero. We define the relevance between two neurons, v_{i_1} and v_{i_k}, as $R_{i_1 i_k} = \max_{L=(v_{i_1}, \ldots, v_{i_k})} (R_{i_1 i_2} R_{i_2 i_3} \cdots R_{i_{k-1} i_k})$, where L denotes possible paths from v_{i_1} to v_{i_k}, and R_{ij} is defined based on DTD formally defined in Eq.(1).

Fig.2(a) illustrates a four-layered DNN with an output neuron v_0. The values adjacent to the arrows denote the relevance between two neurons, and the relevance matrix is presented in Fig.3(a). Fig.2(b) illustrates the simplicial complex of $K_{r=1.0}$ with Betti number $\beta_0 = 9$. We observe the decrease of Betti number β_0 according to the filtration from

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



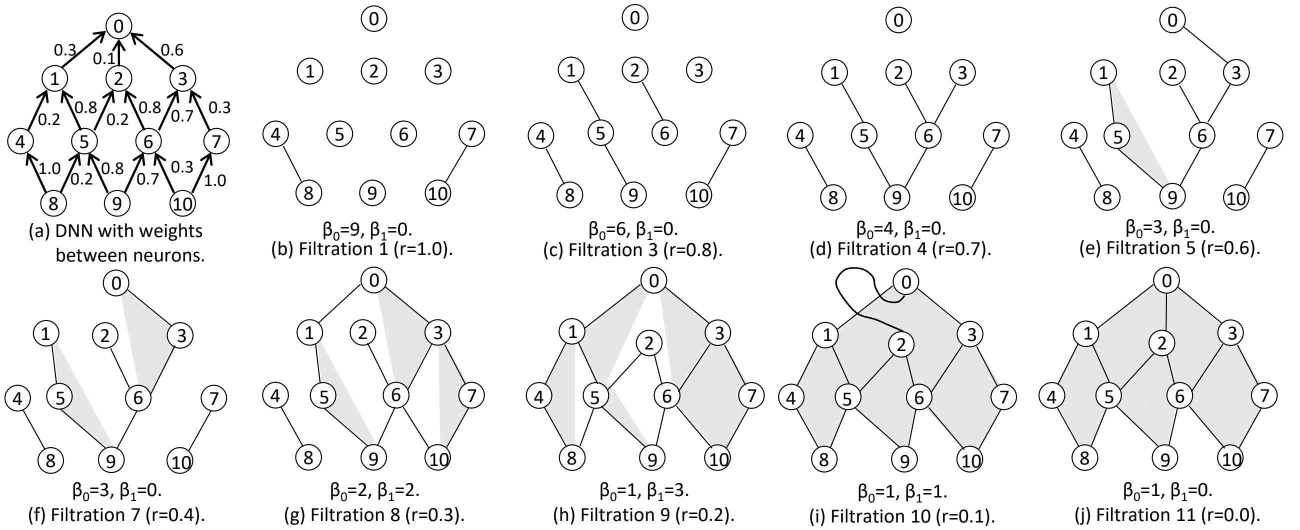
(a) points with oblique lined circles

(b) persistent homology diagram of (a)

(c) barcode diagram of (a)

(d) DNN for handwritten number recognition

Figure 1: (a) Examples of persistent homology diagrams; (b) persistent homology diagram of (a); (c) barcode diagram of (a); (d) DNN for handwritten number recognition.



(a) DNN with weights between neurons.

β_0=9, β_1=0. (b) Filtration 1 (r=1.0).

β_0=6, β_1=0. (c) Filtration 3 (r=0.8).

β_0=4, β_1=0. (d) Filtration 4 (r=0.7).

β_0=3, β_1=0. (e) Filtration 5 (r=0.6).

β_0=3, β_1=0. (f) Filtration 7 (r=0.4).

β_0=2, β_1=2. (g) Filtration 8 (r=0.3).

β_0=1, β_1=3. (h) Filtration 9 (r=0.2).

β_0=1, β_1=1. (i) Filtration 10 (r=0.1).

β_0=1, β_1=0. (j) Filtration 11 (r=0.0).

Figure 2: (a) Example of DNN with weights; (b–h) simplicial complexes and betti numbers corresponding to the filtration.

Fig.2(c) to (h). Fig.2(e) has a 2-simplex and we illustrate it with the gray triangle.

Fig.2(g) and (h) illustrate the increase of Betti number β_1 corresponding to the occurrences of cycle. If the vertexes which represent features of input images are connected straightforwardly to output neurons, the knowledge in the DNN is considered simple because it is equivalent to feature detection. Contrary, the increase of Betti number β_1 indicates that the DNN classifies the input based on the combination of features. From these viewpoints, we can consider the increase of Betti number β_1 reflects the complexity of knowledge in the DNN. Filtration 10 (Fig.2(i)) has Betti number $\beta_1 = 1$. Although [0, 2] is a simplex in Filtration 10, it is not included in another simplex [0,. . .,10], and it produces $\beta_1 = 1$.

The computation of PH involves the explosion of complexity caused by the increase of vertexes, and several implementations are publicly available (Otter et al. 2017). We employed GIDHI (Rouvreau 2016; The GUDHI Project 2015; Boissonnat and Maria 2014), JavaPlex (Tausz, Vejdemo-Johansson, and Adams 2014), and dyonysus 2 (Edelsbrunner and Morozov 2012; Edelsbrunner, Letscher, and Zomorodian 2000; Morozov 2005) libraries for our computation and visualization. These libraries require to register simplexies in each filtration to calculate PH.

The algorithm shown in Fig.4 identifies all simplexis from a vertex v up to the limit of a threshold t of relevance using recursive procedure call. We identifies all simplexes in each filtration by the procedure and register them to the libraries. Fig.3(b) and (c) are barcode and persistent homology diagrams illustrated by GUDHI. Betti numbers correspond to the number of intersections between the bars and perpendicular lines to x-axis in Fig.3(b) (remembering that the lifetime of homologies is defined by the half-open interval $[birth, death)$). GUDHI illustrates Betti numbers using a shade of color in PH diagrams as shown in Fig.3(c). We also calculate the PH with Dionysus 2 and JavaPlex, and confirmed that same diagrams are obtained.

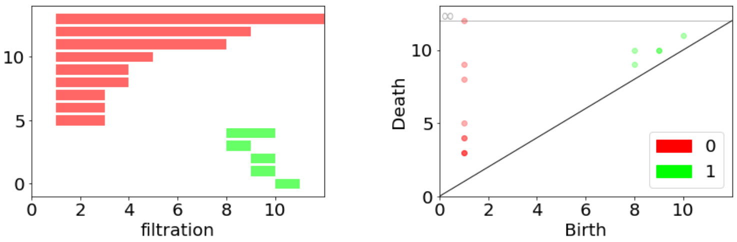
Evaluation setup

The MNIST data set of handwritten digits was used in this study (LeCun et al. 1998). We considered a fully-connected neural network with two hidden layers of sizes 300 and 100, ReLU activation function in the hidden layers and ten output neurons with sigmoid activation function illustrated in Fig.1(d).

The relevance among neurons is formulated by Deep

$$\begin{array}{c|ccccccccccc} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \\ \hline v_0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_1 & 0.3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0.1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_3 & 0.6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0.2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_5 & 0 & 0.8 & 0.2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_6 & 0 & 0 & 0.8 & 0.7 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ v_7 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ v_8 & 0 & 0 & 0 & 0 & 1.0 & 0.2 & 0 & 0 & 1 & 0 & 0 \\ v_9 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.7 & 0 & 0 & 1 & 0 \\ v_{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 1.0 & 0 & 0 & 1 \end{array}$$

(a) Relevance value matrix of Figure. 1 before propagation.



(b) Barcode diagram.

(c) Persistent homology diagram.

Figure 3: (a) Relevance matrix of Fig.2(a); (b,c) barcode and persistent homology diagrams illustrated with GUDIHI.

Figure 4 Algorithm for getting simplexes from vertex s with threshold t.

```
procedure GETSIMPLEX($M$, $s$, $t$)                ▷ where $M$: $n \times n$-matrix, $s$: array, $t$: threshold
    $relevance \leftarrow 1.0$, $result \leftarrow \emptyset$, $startPoint \leftarrow s[0]$
    for $point$ in $s$ do
        $relevance \leftarrow relevance \times M[startPoint][point]$
        $startPoint \leftarrow point$
    if $relevance \geq t$ then
        $result.append(combination(s))$
        $lastPoint \leftarrow s[-1]$                  ▷ $s[-1]$ is the last element of s
        for i in $[0, \cdots, n-1]$ do
            if $M[lastPoint][i] > 0$ and $i \neq lastPoint$ then
                $ss$ ←deep copy of $s$
                $recResult \leftarrow getSimplex(M, ss.append(i), t)$      ▷ recursive call of procedure
                for $e$ in $recResult$ do
                    $result.append(combination(e))$
    return $unique(result)$                          ▷ return deduplicated array
```

Taylor Decomposition (DTD) which propagates the activation of output neurons by their z^+-rule, $R_i = \sum_j \{x_i w_{ij}^+ / \sum_i x_i w_{ij}^+\} R_j$, where w_{ij}^+ denotes the positive part of the weight, and x_i is the input of neurons (Montavon et al. 2017; Lapuschkin et al. 2019). Our purpose is to investigate network structures, and we focus on the relevance between two neurons that do not depend on input images. We made two modifications to their formulation; (i) calculate the relevance between i-th and j-th neurons; (ii) remove the effects of input images. Formally, we define the relevance between i-th and j-th neurons by

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

R_{ij} defines the relevance matrix, and we construct simplicial complexes on the DNN with the procedure described in Fig.4. Although $z^{\mathcal{B}}$-rule of DTD propagates the activation to input pixels, we define the relevance up to the first hidden layer for investigating the inner representations of DNNs. Therefore, the size of the relevance matrix is $410 \times 410 (= 300 + 100 + 10)$ unless noted otherwise.

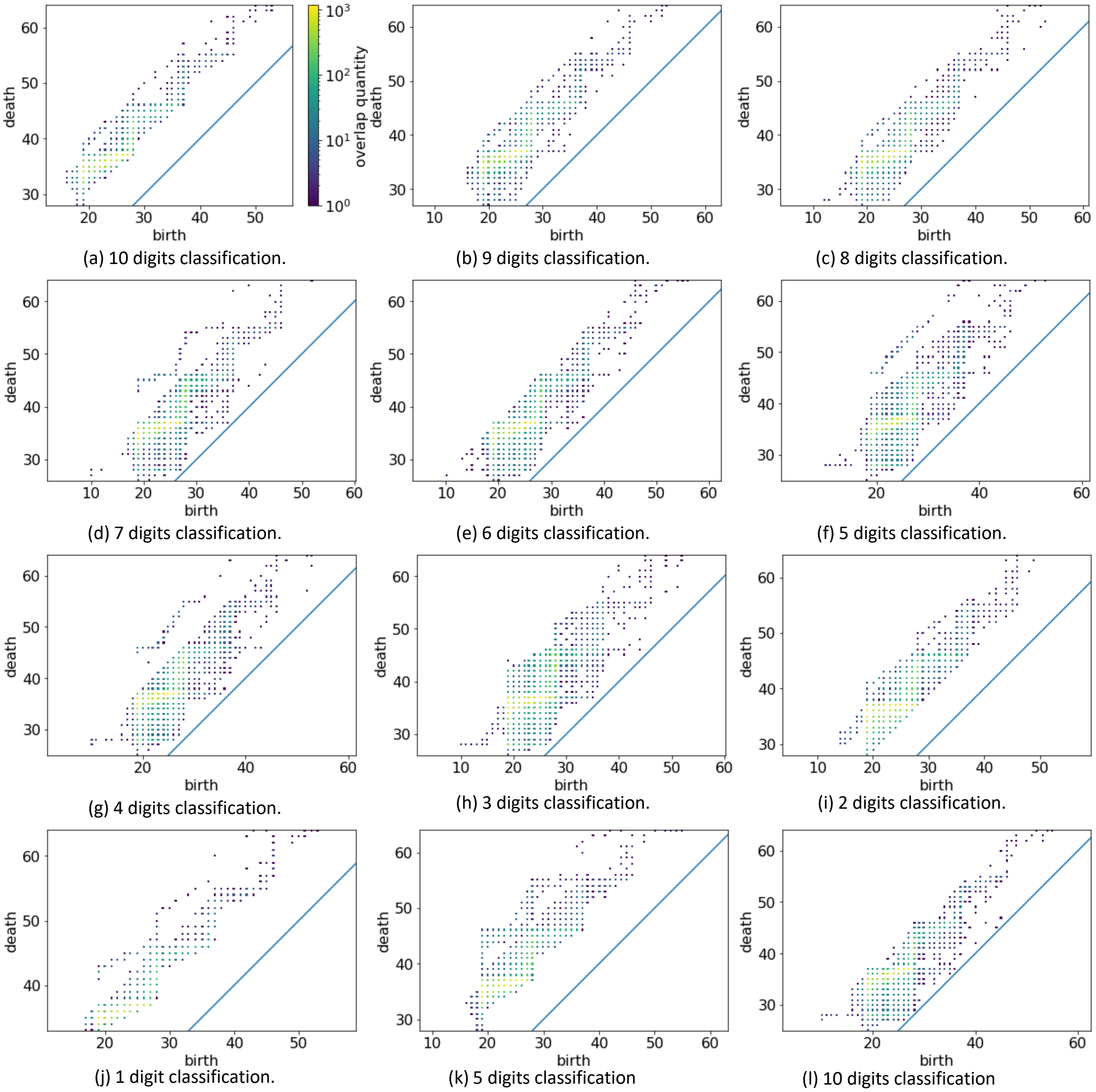
We defined a filtration using thresholds of relevance. The threshold took eight values $1.0^0, \ldots, 1.0^{-7}$ and eight interval values between the adjacent values. Overall, the filtration was defined as $K_{1(r=1.0)} \subset K_{2(r=0.9)} \subset \cdots \subset K_{10(r=1.0^{-1})} \subset K_{11(r=0.09)} \subset \cdots \subset K_{64(r=1.0^{-7})}$.

Evaluation results

Figs.5(a–j) illustrate PH diagrams produced using Dyonysus 2, where the number of digits was varied. We extracted images of the target digits from MNIST data set and trained using images of digits 0–9 represented in Fig.5(a), digits 0–8 represented in Fig.5(b), and so on. We trained models for ten epochs with a batch size of 64, and all models achieved over 97% test data accuracy. PH diagram indicates the stability of homologies by the distance from the dialog line, i.e. stable homologies are pointed in a distance from the dialog line, and unstable homologies are pointed near the dialog line. Dyonysus 2 illustrates the overlapping quantity of homologies using different colors, and the color legend provided for Fig.5(a) is applicable to all diagrams.

The following three observations were made from Figs.5(a–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; (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.5(a) to Fig.5(g) and decreases from Fig.5(h) to Fig.5(j). It reflects both the excess of the output neurons and problem difficulty. We observed that the diagrams seem to reflect the degree of confidence of the DNN, i.e., the excess of the output neurons harms the confidence, whereas the simplicity of problems mitigates it. For further investigation, we classified five digits using five



(a) 10 digits classification.

(b) 9 digits classification.

(c) 8 digits classification.

(d) 7 digits classification.

(e) 6 digits classification.

(f) 5 digits classification.

(g) 4 digits classification.

(h) 3 digits classification.

(i) 2 digits classification.

(j) 1 digit classification.

(k) 5 digits classification with 5 output neurons.

(l) 10 digits classification with 20 output neurons.

Figure 5: (a–j) Persistent diagrams of DNN trained to classify handwritten digits based on a varying number of input digits from 10 to 1; (k) persistent diagram of DNN trained to classify five digits using five output neurons; (l) persistent diagram of DNN trained to classify ten digits using 20 output neurons.

Table 1: Number of points in Figs. 5(a)–5(e), 5(i), and 5(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

output neurons (Fig.5(k)) and ten digits using 20 output neurons (Fig.5(l)). In contrast to Fig.5(f), the points below the belt-like area disappeared in Fig.5(k). The opposite can be observed in Figs.5(a) and (l).

Table 1 shows the number of points in Fig.5. We categorized the points based on the following two conditions: (c1) the homology includes unused output neurons; (c2) the points are under the belt-like area ($death \leq birth + 5$). While Figs.5(i) and 5(j) have points that include unused output neurons more than twice of Fig.5(e), these points are not plotted below the belt-like area. The simplicity of the problem results in no points being plotted under the belt-like area.

Discussion

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

Our assumptions

Our assumptions are summarized as follows: (1) the knowledge in DNNs can be investigated from their network weights among neurons; (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 of previous works described in Section "Intuition behind topological measurement of DNNs." PH reveals the births and deaths of feature combinations, which are difficult to be overviewed without using PH. The effectiveness of the second assumption can be evaluated from the usability in applications, and it changes depending on them.

Application

One of the most important applications is the distinguishing of properly trained DNNs. The performance of DNNs can deteriorate for many reasons, including a shortage of data, overfitting, and improper hyper-parameter setting (Bergstra et al. 2011; Srivastava et al. 2014). Our results imply that the shortage of data can be indicated by the PH, that is the excess of output neurons produces homologies near the dialog line. Furthermore, it will be beneficial to the selection of proper DNN architecture, which is one of the major challenges for utilizing DNNs (Saxena and Verbeek 2016; Zoph and Le 2016).

Future work

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 convolutional neural networks (CNNs) and recursive neural networks (RNNs). Furthermore, with regard to computation, the development will also require considerable efforts for applying the topological measurement to enlarged neural networks, which can have more than 1,000 layers (He et al. 2016). However, our assumptions, we believe, support that topological measurement of DNNs is worth further investigation.

Related work

Bianchini et al. investigated the upper and lower bounds of network complexity from the viewpoint of PH (Bianchini and Scarselli 2014). 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 the proper network architecture for a given data (Guss and Salakhutdinov 2018). 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 or a variety of input data, have significant impact on PH. Thus, the sensitivity of PH requires careful investigation for securing comparability.

Conclusion

We proposed a novel approach to investigate the inner representation of DNNs using PH. Evaluation results demonstrated that the PH of DNNs reflected both the excess of neurons and problem difficulty. They implied that PH can be used as one of the prominent methods for investigating the inner representation of DNNs.

References

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7):e0130140.

Bergstra, J. S.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for hyper-parameter optimization. In Shawe-Taylor, J.; Zemel, R. S.; Bartlett, P. L.; Pereira, F.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 24*. Curran Associates, Inc. 2546–2554.

Bianchini, M., and Scarselli, F. 2014. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems* 25(8):1553–1565.

Boissonnat, J.-D., and Maria, C. 2014. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica* 70(3):406–427.

Cang, Z., and Wei, G.-W. 2018. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International journal for numerical methods in biomedical engineering* 34(2):e2914.

Cassidy, B.; Bowman, F. D.; Rae, C.; and Solo, V. 2018. On the reliability of individual brain activity networks. *IEEE Transactions on Medical Imaging* 37(2):649–662.

Chollet, F. 2017. *Deep Learning with Python*. Greenwich, CT, USA: Manning Publications Co., 1st edition.

Curto, C. 2017. What can topology tell us about the neural code? *Bulletin of the American Mathematical Society* 54(1):63–78.

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

Edelsbrunner, H., and Morozov, D. 2012. Persistent homology: theory and practice. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States).

Edelsbrunner, H.; Letscher, D.; and Zomorodian, A. 2000. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 454–463. IEEE.

Gameiro, M.; Hiraoka, Y.; Izumi, S.; Kramar, M.; Mischaikow, K.; and Nanda, V. 2015. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics* 32(1):1–17.

Guss, W. H., and Salakhutdinov, R. 2018. On characterizing the capacity of neural networks using algebraic topology. *ArXiv* abs/1802.04443.

Hatcher, W. G., and Yu, W. 2018. A survey of deep learning: platforms, applications and emerging research trends. *IEEE Access* 6:24411–24432.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hiraoka, Y.; Nakamura, T.; Hirata, A.; Escolar, E. G.; Matsue, K.; and Nishiura, Y. 2016. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences* 113(26):7035–7040.

Horak, D.; Maletić, S.; and Rajković, M. 2009. Persistent homology of complex networks. *Journal of Statistical Mechanics: Theory and Experiment* 2009(03):P03034.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*.

Kramar, M.; Goullet, A.; Kondic, L.; and Mischaikow, K. 2013. Persistence of force networks in compressed granular media. *Physical Review E* 87(4):042207.

Lapuschkin, S.; Wäldchen, S.; Binder, A.; Montavon, G.; Samek, W.; and Müller, K.-R. 2019. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications* 10:1096.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553):436.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Montavon, G.; Lapuschkin, S.; Binder, A.; Samek, W.; and Müller, K.-R. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition* 65:211–222.

Morcos, A.; Raghu, M.; and Bengio, S. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, 5727–5736.

Morozov, D. 2005. Persistence algorithm takes cubic time in worst case. biogeometry news, dept. *Comput. Sci., Duke Univ., Durham, North Carolina*.

Otter, N.; Porter, M. A.; Tillmann, U.; Grindrod, P.; and Harrington, H. A. 2017. A roadmap for the computation of persistent homology. *EPJ Data Science* 6(1):17.

Petri, G.; Expert, P.; Turkheimer, F.; Carhart-Harris, R.; Nutt, D.; Hellyer, P. J.; and Vaccarino, F. 2014. Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface* 11(101):20140873.

Raghu, M.; Gilmer, J.; Yosinski, J.; and Sohl-Dickstein, J. 2017. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, 6076–6085.

Rouvreau, V. 2016. Cython interface. In *GUDHI User and Reference Manual*. GUDHI Editorial Board.

Samek, W.; Binder, A.; Montavon, G.; Lapuschkin, S.; and Müller, K.-R. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems* 28(11):2660–2673.

Saxena, S., and Verbeek, J. 2016. Convolutional neural fabrics. In *Advances in Neural Information Processing Systems*, 4053–4061.

Sizemore, A. E.; Giusti, C.; Kahn, A.; Vettel, J. M.; Betzel, R. F.; and Bassett, D. S. 2018. Cliques and cavities in the human connectome. *Journal of computational neuroscience* 44(1):115–145.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tausz, A.; Vejdemo-Johansson, M.; and Adams, H. 2014. JavaPlex: A research software package for persistent (co)homology. In Hong, H., and Yap, C., eds., *Proceedings of ICMS 2014*, Lecture Notes in Computer Science 8592, 129–136. Software available at http://appliedtopology.github.io/javaplex/.

The GUDHI Project. 2015. *GUDHI User and Reference Manual*. GUDHI Editorial Board.

Wasserman, L. 2018. Topological data analysis. *Annual Review of Statistics and Its Application* 5:501–532.

Xia, K., and Wei, G.-W. 2014. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering* 30(8):814–844.

Yoo, J.; Kim, E. Y.; Ahn, Y. M.; and Ye, J. C. 2016. Topological persistence vineyard for dynamic functional brain connectivity during resting and gaming stages. *Journal of neuroscience methods* 267:1–13.

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

Zhang, Q.; Yang, L. T.; Chen, Z.; and Li, P. 2018. A survey on deep learning for big data. *Information Fusion* 42:146–157.

Zoph, B., and Le, Q. V. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.