Neural Networks 143 (2021) 42–51

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Experimental stability analysis of neural networks in classification problems with confidence sets for persistence diagrams

Naoki Akai [a,*], Takatsugu Hirayama [b], Hiroshi Murase [c]

[a] Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan
[b] Institute of Innovation for Future Society, Nagoya University, Nagoya 464-8601, Japan
[c] Graduate School of Informatics, Nagoya University, Nagoya 464-8603, Japan

ARTICLE INFO

Article history:
Received 23 October 2020
Received in revised form 29 April 2021
Accepted 6 May 2021
Available online 12 May 2021

Keywords:
Neural networks
Topological data analysis
Persistent homology

ABSTRACT

We investigate classification performance of neural networks (NNs) based on topological insight in an attempt to guarantee stability of their inference. NNs which can accurately classify a dataset map it into a hidden space while disentangling intertwined data. NNs sometimes acquire forcible mapping to disentangle the data, and this forcible mapping generates outliers. The mapping around the outliers is unstable because the outputs change drastically. Hence, we define stable NNs to mean that they do not generate outliers. To investigate the possibility of the existence of outliers, we use persistent homology and a method to estimate the confidence set for persistence diagrams. The combined use enables us to test whether the focused geometry is topologically simple, that is, no outliers. In this work, we use the MNIST and CIFAR-10 datasets and investigate the relationship between the classification performance and the topological characteristics with several NNs. Investigation results with the MNIST dataset show that the test accuracy of all the networks is superior, exceeding 98%, even though the transformed dataset is not topologically simple. Results with the CIFAR-10 dataset also show that the possibility of the existence of outliers is shown in the mapping by the accurate convolutional NNs. Therefore, we conclude that the presented investigation is necessary to guarantee that the NNs, in particular deep NNs, do not acquire unstable mapping for forcible classification.

1. Introduction

Guaranteeing stability of inference by neural networks (NNs), in particular deep NNs, is difficult because they involve a large number of computational processes. This problem is known as a *black box problem*. Consequently, the use of NNs in applications that require high stability, such as automated driving (Akai, Morales, Takeuchi et al., 2017; Akai, Morales, Yamaguchi et al., 2017), is challenging. The NNs are expected to have major role in various autonomous systems; however, it is necessary to guarantee the stability of their inference. In this work, we focus on classification problems with NNs and investigate their performance from topological insight to analyze the stability. We define *stable NNs* to mean that they do not generate outliers as their inference because the inference around the outliers changes drastically. In this investigation, we use persistent homology (PH) (Edelsbrunner & Zomorodian, 2002), which is widely used in topological data analysis (TDA), in combination with a method to estimate the confidence set for persistence diagrams (PDs) (Fasy et al., 2014). The PH is a method for computing topological features of a space at different spatial resolutions and the PD is a representation method of the PH (these are detailed in Section 3). The investigation is able to show that the NNs do not acquire unstable mapping for forcible classification, that is, it could help to guarantee stability of applications using NNs.

The output layers of NNs classify hidden features of the previous layer with an affine transformation. In this work, we refer to the previous layer and its space as the *hidden layer* and *hidden space*, respectively. It is almost impossible to directly classify a dataset based on the output layers because the data of different classes are typically intertwined. NNs which can accurately classify the dataset map it into the hidden space while disentangling the intertwined data. NNs sometimes acquire forcible mapping to disentangle the data, and this forcible mapping generates outliers. To guarantee the stability of their inference, data of the same class must be transformed into the hidden space as *topologically simple* one.

We first provide an accurate definition of topologically simple. The Betti numbers, $\beta_0, \beta_1, \ldots, \beta_{D-1}$, are used to distinguish between topological spaces based on the connectivity of

* Correspondence to: Graduate School of Engineering, Nagoya University, Department of Engineering, 2nd Building, Furo, Chikusa, Nagoya, 464-8603, Aichi, Japan.
E-mail address: akai@nagoya-u.jp (N. Akai).
URL: https://sites.google.com/view/naokiakaigoo/home (N. Akai).

https://doi.org/10.1016/j.neunet.2021.05.007
0893-6080/

D-dimensional simplicial complexes. The ith Betti number represents the rank of the ith homology group, H_i. Topologically simple is defined as that the Betti number of the 0th order, β_0, is 1 and that of orders higher than the 1st order, β_k ($k \geq 1$), are 0, that is $(\beta_0, \beta_1, \ldots, \beta_{D-1}) = (1, 0, \ldots, 0)$, otherwise topologically complex. Figures of the top row of Fig. 1 illustrate dataset examples distributed on topological simple (left) and complex (middle and right) manifolds in $\mathbb{R}^2$. It should be noted that we assume that a dataset (point cloud) is distributed on a manifold in this work. Informally, the ith Betti number refers to the number of i-dimensional holes on a topological surface, that is, β_0 and β_1 are numbers of connected components and tunnels. Hence, the Betti numbers of the top row figures (β_0, β_1) are $(1, 0)$, $(2, 0)$, and $(1, 4)$, respectively. It can be argued that NNs do not generate outliers if NNs transform a dataset into topologically simple one.

Naitzat et al. (2020) argued that well-trained NNs transform a topologically complicated dataset into a topologically simple one as it passes through the layers. They used the PH to determine whether the transformed dataset is topologically simple. However, the PH cannot be used to count the Betti numbers of the transformed dataset. In their work, they generated a simulated dataset and assumed that the use of the PH with fixed parameters to calculate the Betti numbers of the dataset can be used to confirm the way in which the Betti numbers change. However, the Betti numbers of a dataset are always unknown in real problems.

Hamada and Goto (2018) presented a data-driven analysis method to test whether a point cloud is distributed on a topologically simple manifold. This method also cannot count the Betti numbers of the manifold; however, it can show whether the Betti number of the 0th order and that of higher than the 1st order of the manifold are larger than 1 and 0, that is, it can test whether the manifold is topologically simple or not. Therefore, we use this method to analyze how the dataset is transformed by NNs and compare the results of the analysis with the classification performance. Our investigation can provide more clear evidence of whether the transformed dataset is topologically simple because the method is based on the method to estimate the confidence set for the PDs (Fasy et al., 2014). To the best of our knowledge, this is the first study in which the method for confidence set estimation is used to analyze NN performance.

Figures of the bottom row of Fig. 1 show examples of the analysis. The figures are the PDs corresponding to the datasets in the top row, and the estimated confidence bands are depicted in gray. The points on the PDs are referred to as *birth–death pairs* that are topological features extracted by the PH (see Section 3.1 for more details). A few pairs are located outside the confidence band if the manifold is topologically complex. The analysis using the confidence set tells us whether the manifold that the dataset is distributed on is topologically simple.

In this work, we use the MNIST and CIFAR-10 datasets. In the investigation with the MNIST dataset, we analyze the transformed dataset by six NNs and show that the test accuracy of all the networks is superior, exceeding 98%, in both cases where the transformed dataset is topologically simple and complex. In the investigation with the CIFAR-10 dataset, we use convolutional NNs (CNNs) and show that CNNs with higher classification accuracy transform the dataset into topologically complex one more than a simple CNN. Therefore, we conclude that the possibility of the existence of outliers, that is, stability, cannot be discussed in terms of only the classification accuracy and the presented investigation is necessary for guaranteeing the stability.

The contribution of this work is twofold:

- Application of the method for confidence set estimation to analyze the classification performance of the NNs
- Suggestion of the necessity of the presented investigation to guarantee the stability of the mapping acquired by the NNs.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 describes the methods used in this work. Section 4 presents the investigation that is conducted using the MNIST and CIFAR-10 datasets. Section 5 concludes this work.

2. Related work

Guaranteeing the stability of inference by NNs and the interpretation thereof has been widely studied in recent years. Many researchers have tried to understand the reason for the inference and have proposed methods based on activation maximization, sensitivity analysis, and deconvolution (Le et al., 2011; Selvaraju et al., 2016; Smilkov et al., 2017). The explainability of these approaches was evaluated in terms of consistency and validity (Montavona et al., 2018). Our work aims to understand whether the mapping acquired by NNs is stable with the TDA approaches.

Topological complexity can be quantified in terms of the Betti numbers and these numbers have been used for the analysis of NNs. Bianchini and Scarselli (2014) proposed a purely theoretical measure based on the Betti numbers to evaluate the complexity of functions implemented by NNs. They showed that NNs can realize mapping with a higher complexity than a shallow mapping. Guss and Salakhutdinov (2018) proposed a measure for dataset complexity with the PH and studied the empirical relation between the complexity and minimal network capacity for classifying the dataset. Ramamurthy et al. (2019) provided theoretical conditions and an analysis for recovering the homology of a decision boundary in classification tasks with the PH. Rieck et al. (2019) proposed a complexity measure for NN architectures with reference to neural persistence and derived a neural persistence-based early stopping criterion. In this work, we use the PH to analyze datasets that have been transformed by NNs and investigate whether these are transformed to topologically simple.

As mentioned in Section 1, our work is inspired by previous research by Naitzat et al. (2020). They investigated the changes in the topology of datasets as it passes through the NNs. Their work was also inspired by a Google Brain blog post (Olah, 2014). The rectified linear unit (ReLU) (Glorot et al., 2011), which has become a widely used activation function in recent years, is considered to be more effective than other activation functions because it can avoid the vanishing gradient problem (LeCun et al., 2015). Naitzat et al. (2020) further argued that the ReLU quickly changes the topology of a dataset more than other activation functions such as sigmoid and hyperbolic tangent, and it contributes to more effective inference. They used the persistence Betti numbers to show that the topology changes; however, their investigation was unable to determine whether the transformed dataset was topologically simple.

In our work, we use the data-driven method presented in Hamada and Goto (2018), which can test whether a point cloud is distributed on a simplicial manifold, and it has more strong evidence than the method that Naitzat et al. (2020) used. This enables us to discuss whether the mapping acquired by NNs is stable because it reveals the possibility of the existence of outliers in the mapping.

3. Methods

This section contains brief descriptions of the methods used in this work, that is, persistent homology (PH), persistence diagrams (PDs), confidence set for PDs, and the subsampling method for calculating the confidence set.

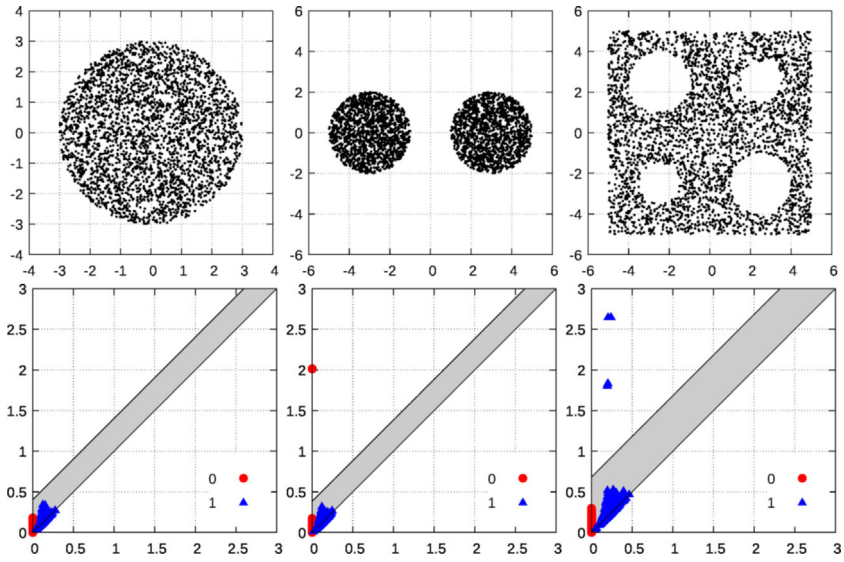

Fig. 1. Top: Point clouds sampled from topologically simple (left) and complex (middle and right) manifolds in $\mathbb{R}^2$. The corresponding Betti numbers (β_0, β_1) are $(1, 0)$, $(2, 0)$, and $(1, 4)$. Bottom: Persistence diagrams corresponding to the point clouds in the top row with the confidence band (gray). Birth–death pairs locate outside the band if the manifold is a topologically complex.

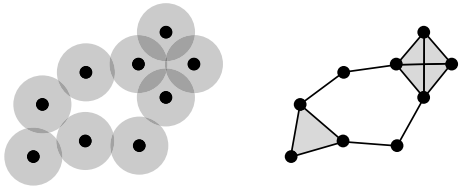

Fig. 2. Point cloud X (black dots) with radius α in $\mathbb{R}^2$ and its Vietoris–Rips complex $\mathrm{Rips}_\alpha(X)$.

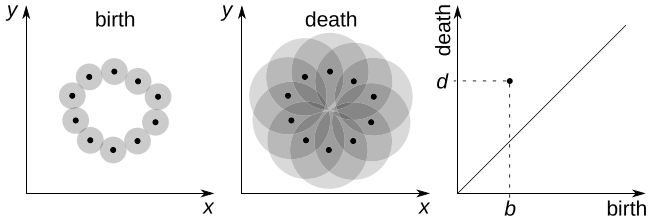

Fig. 3. Topological feature tracked by the PH in $\mathbb{R}^2$ and the PD.

3.1. Persistent homology and persistence diagrams

To calculate the PH, simplicial complexes need to be constructed from a dataset (point cloud) $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ $\mathbf{x}_i \in \mathbb{R}^D$, that is, simplicial complexes of which the vertex set is X must be defined. We assume that the dataset forms a point cloud in D-dimensional space. Fig. 2 illustrates the point cloud X with radius α (left) and the Vietoris–Rips complex $\mathrm{Rips}_\alpha(X)$ (right) corresponding to the point cloud on the left. Fig. 2 shows the case in $\mathbb{R}^2$; however, we discuss the case in $\mathbb{R}^D$ below for the sake of generality. The regions covered by gray circles, i.e., the union, of the left-hand figure can be denoted as

$$\bigcup_{\mathbf{x}\in X} B(\mathbf{x}, \alpha) = \{\mathbf{x} : d_X(\mathbf{x}) \leq \alpha\}, \tag{1}$$

where $B(\mathbf{x}, \alpha)$ is a D-dimensional ball with radius α centered at $\mathbf{x}$ and $d_X(\mathbf{x})$ is the Euclidean distance function denoted by $d_X(\mathbf{x}) = \inf_{\mathbf{y}\in X} \|\mathbf{x} - \mathbf{y}\|_2$.

$\mathrm{Rips}_\alpha(X)$ is the set of simplices $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ such that $d_X(\mathbf{x}_i, \mathbf{x}_j) \leq \alpha$ for all (i, j). $\mathrm{Rips}_\alpha(X)$ is non-decreasing with α, that is, $\mathrm{Rips}_{\alpha_1}(X)$ is included in $\mathrm{Rips}_{\alpha_2}(X)$, that is, $\mathrm{Rips}_{\alpha_1}(X) \subset \mathrm{Rips}_{\alpha_2}(X)$, where $\alpha_1 \leq \alpha_2$. These sequences of inclusions are referred to as *filtrations*. It should be noted that there are other simplicial complexes such as Čech and alpha complexes, but we use the Vietoris–Rips complex in this work because it offers the advantage of being computationally efficient.

The overlapping areas of the circles increase as α increases. The topology of $\mathrm{Rips}_\alpha(X)$ also changes as α increases because $\mathrm{Rips}_\alpha(X)$ is defined based on the overlaps. The PH tracks the change in the homology. Fig. 3 illustrates a topological feature that is tracked by the PH. On the left of Fig. 3, the circles are connected, and they create a hole. The time at which the hole is appeared is referred to as its *birth*. The value of α continues to increase and the hole disappears as shown in the middle of Fig. 3. The time at which the hole disappears is referred to as its *death*. By using filtrations, we obtain a set of birth–death pairs (b_k, d_k) $k = 1, \ldots, K$ $(b_k < d_k)$. The birth–death pairs can be plotted on a 2D surface, as shown on the right of Fig. 3, and these diagrams are referred to as *persistence diagrams (PDs)*.

More precisely, the PH tracks changes in the homology group, H, as α increases. In total, D PH groups can be calculated because $H_0, \ldots, H_{D-1}$ can be defined in $\mathbb{R}^D$ (Fig. 3 illustrates the 1st PH group), where H_i is the ith homology group and $D_{X,i}$ denotes the ith PD of X. However, a significant amount of time is required to calculate high-order PDs. Fortunately, $D_{X,0}$ and $D_{X,1}$ are sufficient for our investigation and we discuss this in Section 3.4.

3.2. Confidence sets for PDs

In this work, we assume that the dataset X (or the transformed dataset X') is distributed according to the manifold $\mathbb{M}$ (or $\mathbb{M}'$). Ideally, we would like to discuss the topology of the manifold; however, it is almost impossible to know the shape of the manifold. Hence, we need to consider discussing the topology of the manifold by using the dataset.

As previously described, the PDs of the dataset D_X summarize the topological characteristics of the dataset. The birth–death pairs close to the diagonal are regarded as topological noise. On the other hand, the birth–death pairs far from the diagonal are regarded as stronger topological signals, as shown in Fig. 3. Hence,

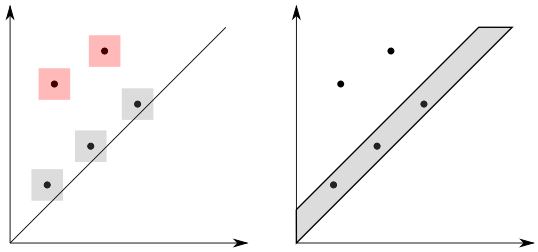

Fig. 4. Visualization of the confidence set.

we can conclude that existing PH groups do not topologically characterize the manifold well if all the birth–death pairs are close to the diagonal, that is, the manifold is topologically simple.

However, it is difficult to distinguish whether a birth–death pair is close to the diagonal by only focusing on the PDs. Here, we introduce the distance between the PDs of X and $\mathbb{M}$, referred to as the *bottleneck distance*, which is denoted as

$$d_B(D_X, D_{\mathbb{M}}) = \inf_{\gamma} \sup_{\mathbf{x} \in D_X \cup \Delta} \|\mathbf{x} - \gamma(\mathbf{x})\|_\infty, \tag{2}$$

where $\gamma : D_X \cup \Delta \to D_{\mathbb{M}} \cup \Delta$ is a bijection, and Δ is a set of diagonals $\{(a, a) | a \in \mathbb{R}\}$. We can distinguish whether a birth–death pair is close to the diagonal by comparing the distance from the diagonal to the birth–death pair with the bottleneck distance. Thus, we consider the following probability

$$\limsup_{N \to \infty} p(d_B(D_X, D_{\mathbb{M}}) > c_N) \leq p_{\text{th}}, \tag{3}$$

where $c_N \equiv c_N(\mathbf{x}_1, \ldots, \mathbf{x}_N)$, and $0 \leq p_{\text{th}} \leq 1$. Then, we define a range denoted as $[0, c_N]$. An asymptotic $1 - p_{\text{th}}$ confidence set for the bottleneck distance can be defined as

$$\liminf_{N \to \infty} p(d_B(D_X, D_{\mathbb{M}}) \in [0, c_N]) \geq 1 - p_{\text{th}}. \tag{4}$$

Finally, the confidence set C_N is defined as a subset of all PDs of which the distance to D_X is at most c_N

$$C_N = \{\hat{D}_X | d_B(D_X, \hat{D}_X) \leq c_N\}, \tag{5}$$

However, calculation of the PDs is significantly time consuming. Instead, the following theorem (Fasy et al., 2014), which respects the bottleneck stability theorem (Cohen-Steiner et al., 2007), was presented:

$$d_B(D_X, D_{\mathbb{M}}) \leq \|d_X - d_{\mathbb{M}}\|_\infty = d_H(X, \mathbb{M}), \tag{6}$$

where D_X and $D_{\mathbb{M}}$ are the PDs based on the lower-level sets $\{\mathbf{x} : d_X(\mathbf{x}) \leq \alpha\}$ and $\{\mathbf{x} : d_{\mathbb{M}}(\mathbf{x}) \leq \alpha\}$, and $d_H(X, \mathbb{M})$ is the Hausdorff distance denoted as

$$d_H(X, \mathbb{M}) = \max \left\{ \max_{\mathbf{x} \in X} \min_{\mathbf{y} \in \mathbb{M}} d_E(\mathbf{x}, \mathbf{y}), \max_{\mathbf{y} \in \mathbb{M}} \min_{\mathbf{x} \in X} d_E(\mathbf{x}, \mathbf{y}) \right\}, \tag{7}$$

where $d_E(\mathbf{x}, \mathbf{y})$ is the Euclidean distance. Hence, we consider the following probability instead of Eq. (3)

$$\limsup_{N \to \infty} p(d_H(X, \mathbb{M}) > c_N) \leq p_{\text{th}}. \tag{8}$$

Fig. 4 shows an example of the visualization of the confidence set. In the left hand figure, the confidence set is centered at each birth–death pair and has the side range $2c_N$. The pairs are distinguished as topological noise if the range crosses to the diagonal. Hence, we only need to check whether birth–death pairs exist on the inside the band from the diagonal, as shown in the right hand figure. The range of the band is $\sqrt{2}c_N$.

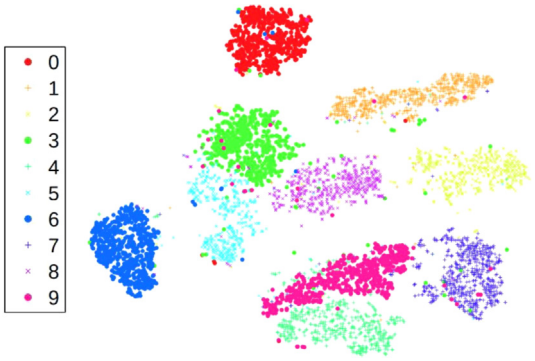

Fig. 5. Visualization of the MNIST dataset using t-SNE (van der Maaten & Hinton, 2008).

3.3. Subsampling method for estimating confidence sets

To estimate the confidence sets, we need to calculate $d_H(X, \mathbb{M})$. However, this distance cannot be calculated because manifold $\mathbb{M}$ is included as the argument. Here, we introduce a subsample set of the dataset denoted as $X_B^{(m)} = (\mathbf{x}_1, \ldots, \mathbf{x}_B)$ and $m = 1, \ldots, M := \binom{N}{B}$, where $B = B_N$ and $B_N = O(N/\log N)$ if $N \to \infty$. The following condition is almost completely satisfied if N is sufficiently large

$$d_H(X, \mathbb{M}) \leq d_H(X, X_B^{(m)}). \tag{9}$$

The right-hand term can be calculated since the manifold is removed from the arguments.

We define the cumulative distribution function (CDF) for the above distance as

$$L_B(d) = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}\left(d_H(X, X_B^{(m)}) > d\right), \tag{10}$$

where $\mathbb{I}(\cdot)$ is an indicator function that is equal to 1 when the condition within the bracket is true, and 0 otherwise. From the CDF, we define the range of the confidence band c_B as

$$c_B = \sqrt{2} L_B^{-1}(p_{\text{th}}), \tag{11}$$

where $L_B^{-1}(\cdot)$ is the inverse function of L_B. The following theorem was also proven (Fasy et al., 2014):

$$\begin{aligned} &p(d_B(D_X, D_{\mathbb{M}}) > c_B) \leq p(d_H(X, \mathbb{M}) > c_B) \\ &\leq p(d_H(X, X_B^{(m)}) > c_B) \leq p_{\text{th}} + O\left(\frac{B}{N}\right)^{1/4}. \end{aligned} \tag{12}$$

Thus, we use Eq. (11) to calculate the confidence band.

It should be noted that Fasy et al. (2014) presented four methods, three of which are based on the distance function to the data, and the last of which is based on the density estimation. We selected to use the method based on the distance function, that is, the subsampling method, because it is more directly connected to the raw data.

3.4. Orders of PH groups

As previously described, the homology groups can be calculated from the 0th to $D-1$th order in D-dimensional space. However, calculating the PH groups of high-dimensional orders is highly time consuming and is usually infeasible. Hence, we use only the 0th and 1st order PH groups in this work. In this subsection, we discuss the reason for these PH groups being sufficient for our investigation.

The 0th and 1st order homologies correspond to connected components and tunnels. Fig. 5 shows the visualization result

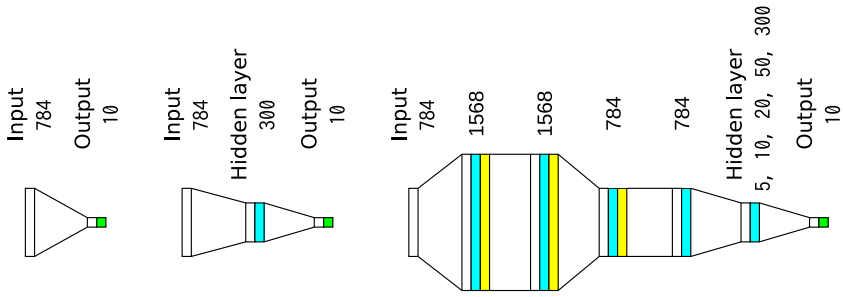

Fig. 6. Network architectures of the simple classifier (left), NN (middle), and NNs (right). The white, cyan, yellow, and green layers depict the fully connected, ReLU, dropout, and softmax layers, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the MNIST dataset using t-SNE (this figure is a reproduction from van der Maaten & Hinton, 2008). As shown in the figure, data with the same labels form a connected component, and clear holes cannot be seen. Note that some data are plotted at far from the main component and we define such data as outliers. If a hole is visible, this means that the data forming the hole cyclically change according to the position of the circumference of the hole. It would not be possible to detect these cyclic changes if the dataset did not contain cyclic or reversed data. Hence, we would have to determine the number of connected components in the dataset (or the transformed dataset) via the 0th order PH group. In the next section, we present the results of the calculation of the 1st PH group and show that cyclic changes do not occur in the MNIST dataset.

4. Investigation

We use the MNIST and CIFAR-10 datasets to investigate the relationship between the classification performance and the topological characteristics of the datasets transformed by NNs. This section describes the investigation results while showing their classification performance.

4.1. Investigation with the MNIST dataset

4.1.1. Training, validation, and test of NNs

We built a simple classifier network composed of only input and output layers, an NN with three layers, and five deep neural networks (DNNs). The architectures of these networks are shown in Fig. 6.These networks are only composed of fully connected layers. The ReLU was used as the activation function because it was suggested to rapidly change the topology of the dataset (Naitzat et al., 2020). A dropout layer was applied to the DNNs, and the dropout rate was set to 10%. The size of the hidden layers in the DNNs was set to 5, 10, 20, 50, and 300, and these DNNs are denoted as DNN5, DNN10, DNN20, DNN50, and DNN300, respectively. The simple classifier network and the NN with three layers are denoted as SC and NN, respectively.

We trained these networks on 55,000 samples with 50 epochs, and 5000 samples were used for the validation. Categorical cross-entropy was used as the loss function and the Adam optimizer (Kingma & Ba, 2015) was employed. Fig. 7 shows the training (solid) and validation (dashed) losses. It should be noted that the results for the SC are not shown, because it is only used to confirm the complexity of the dataset. All the training and validation losses were appropriately converged. Overfitting might not occur in any of the networks because the validation losses did not increase during the training. Hence, we consider that all the networks were successfully trained in terms of optimization of the loss function.

The classification results for the 10,000 samples are presented in Table 1 (values "smaller" than 98% are highlighted in bold font). Because the results from 0 to 9 are the binary classification results, that is, the target label is positive and other labels are negative, other metrics in addition to accuracy were used to assess the performance and are also listed in the table. The results for all the labels (in the rightmost column) were obtained by only assessing the accuracy because these are multi-class classification results. All the networks achieved 98% classification accuracy for all the labels, whereas the accuracy with the SC was 92.71%. Note that we adjusted the size of the hidden layer of the NN to exceed 98% accuracy.

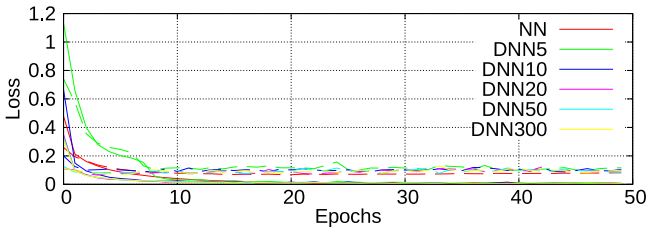

Fig. 7. Training (solid lines) and validation (dashed lines) losses.

Based on the results, all the networks were considered to have been successfully trained since all the classification accuracy exceed 98%. This is also suggested by the results shown in Fig. 7 because the validation losses decreased appropriately. It is also considered that all the networks acquired similar mapping because performances are similar. However, our investigation on the basis of topological insight provides new considerations.

4.1.2. PDs with confidence sets

Fig. 8 shows the PDs with the confidence set.[1] Note that these results were obtained from the training dataset because we aimed to investigate the mapping acquired by the networks. The red and blue dots are the 0th and 1st order PH groups, and the gray, cyan, and magenta areas are the 5%, 50%, and 95% confidence bands calculated using Eq. (11). Below, we focus only on the 0th order PH group because all the 1st order PH groups exist inside the 5% confidence band.

The figures in the first row are calculated from the raw input images. Note that the images were normalized because normalization is necessary before the images are input into the networks. All the birth–death pairs exist inside the 5% confidence band. Hence, data with the same label are considered to be distributed on a topologically simple manifold; however, data with different labels are slightly intertwined.

The NN achieved 98.08% classification accuracy; however, the PDs, shown in the second row, with the confidence set are similar

[1] We used HomCloud to calculate the PDs https://www.wpi-aimr.tohoku.ac.jp/hiraoka_labo/homcloud/index.en.html. The software employs Risper (Bauer, 2019) for the Vietoris–Rips computation.

Table 1
Classification results for the MNIST dataset (unit is %). Values smaller than 98% are highlighted in bold.

		0	1	2	3	4	5	6	7	8	9	All
SC	Accuracy	99.30	99.42	98.28	98.16	98.72	98.13	98.96	98.46	**97.74**	98.25	**92.71**
	Recall	98.26	**97.79**	**89.24**	**91.18**	**93.78**	**87.44**	**94.78**	**92.50**	**90.24**	**90.88**	–
	Specificity	99.41	99.62	99.31	98.94	99.25	99.17	99.40	99.14	98.54	99.07	–
	Precision	**94.78**	**97.11**	**93.78**	**90.64**	**93.21**	**91.22**	**94.38**	**92.50**	**87.02**	**91.70**	–
	F-measure	**96.49**	**97.45**	**91.45**	**90.91**	**93.50**	**89.29**	**94.58**	**92.50**	**88.60**	**91.28**	–
NN	Accuracy	99.74	99.78	99.57	99.54	99.62	99.66	99.64	99.64	99.51	99.46	98.08
	Recall	99.18	99.11	**97.77**	**97.72**	**97.96**	**97.64**	98.12	98.05	**97.02**	98.01	–
	Specificity	99.80	99.86	99.77	99.74	99.80	99.85	99.80	99.82	99.77	99.62	–
	Precision	98.18	98.94	98.05	**97.72**	98.16	98.52	98.12	98.43	**97.92**	**96.67**	–
	F-measure	98.68	99.03	**97.91**	**97.72**	98.06	98.08	98.12	98.24	**97.47**	**97.34**	–
DNN5	Accuracy	99.87	99.87	99.63	99.61	99.74	99.68	99.78	99.67	99.61	99.62	98.54
	Recall	99.18	99.38	98.25	98.31	98.57	98.09	98.95	98.34	98.76	**97.42**	–
	Specificity	99.94	99.93	99.78	99.75	99.86	99.83	99.86	99.82	99.70	99.86	–
	Precision	99.48	99.47	98.16	**97.83**	98.77	98.31	98.75	98.44	**97.26**	98.79	–
	F-measure	99.33	99.42	98.20	98.07	98.67	98.20	98.85	98.39	98.01	98.10	–
DNN10	Accuracy	99.83	99.84	99.78	99.65	99.63	99.71	99.74	99.65	99.61	99.48	98.46
	Recall	99.38	99.38	98.83	98.31	98.57	**97.98**	98.95	98.34	**96.81**	**97.81**	–
	Specificity	99.87	99.89	99.88	99.79	99.74	99.87	99.82	99.79	99.91	99.66	–
	Precision	98.88	99.20	99.02	98.21	**97.67**	98.75	98.34	98.25	99.15	**97.05**	–
	F-measure	99.13	99.29	98.93	98.26	98.12	98.36	98.64	98.29	**97.97**	**97.43**	–
DNN20	Accuracy	99.77	99.84	99.66	99.72	99.68	99.73	99.75	99.58	99.57	99.64	98.47
	Recall	99.48	99.11	98.35	98.31	98.87	**97.75**	98.53	98.63	98.15	**97.32**	–
	Specificity	99.80	99.93	99.81	99.87	99.76	99.92	99.87	99.68	99.72	99.89	–
	Precision	98.18	99.46	98.35	98.90	**97.88**	99.20	98.84	**97.31**	**97.45**	99.09	–
	F-measure	98.83	99.29	98.35	98.60	98.37	98.47	98.69	**97.97**	**97.80**	98.20	–
DNN50	Accuracy	99.77	99.77	99.70	99.69	99.71	99.66	99.66	99.66	99.51	99.59	98.36
	Recall	99.38	99.29	98.54	98.71	98.47	**97.86**	**96.86**	98.15	98.66	**97.42**	–
	Specificity	99.81	99.83	99.83	99.79	99.84	99.83	99.95	99.83	99.60	99.83	–
	Precision	98.28	98.68	98.54	98.22	98.57	98.31	99.57	98.53	**96.38**	98.49	–
	F-measure	98.83	98.98	98.54	98.46	98.52	98.08	98.20	98.34	**97.51**	**97.95**	–
DNN300	Accuracy	99.84	99.83	99.76	99.62	99.65	99.67	99.80	99.70	99.66	99.55	98.54
	Recall	99.28	99.64	98.93	98.71	**97.96**	**96.97**	99.06	98.54	98.15	**97.81**	–
	Specificity	99.90	99.85	99.85	99.72	99.83	99.93	99.87	99.83	99.82	99.74	–
	Precision	99.08	98.86	98.74	**97.55**	98.46	99.31	98.85	98.54	98.35	**97.72**	–
	F-measure	99.18	99.25	98.83	98.12	98.21	98.12	98.95	98.54	98.25	**97.77**	–

to that of the dataset. These results show that the overlapped data could be disentangled while retaining the topologically simple geometry. Hence, the mapping acquired by the NN is stable because all the classes are mapped onto the topologically simple manifolds, namely, they do not generate outliers in their output.

The results produced by the DNNs, shown in from the third to seventh rows, show a few birth–death pairs that are located far from the origin, for example, label 0 of DNN5. These results show that some data points are transformed far from the main component, that is, they are outliers, such as shown in Fig. 5. Because the Hausdorff distance is weak to outliers, a large bias exists in the CDF of the Hausdorff distance (the 50% and 95% confidence bands are obviously larger than that of the top two rows). Hence, we consider it to be more useful to focus only on the 5% confidence band to distinguish whether the transformed dataset is topologically simple.

Focusing on the 5% confidence band, we notice that a few birth–death pairs can be seen outside the band. Hence, this could be considered to signify that mapping by the DNNs forcibly disentangles overlaps among the different labels, thereby resulting in the formation of outliers. This forcible mapping is not directly related to the classification accuracy because the classification boundary can be exactly drawn even though the outliers are included. In fact, the classification performance of the DNNs was superior. However, such forcible mapping causes unstable mapping around the outliers. Thus, we do not regard these DNNs as the stable NNs even though they accurately classify the dataset.

We compared the classification performance and the PDs; however, it is difficult to identify certain relationships. Hence, we do not consider it possible to discuss the classification performance on the basis of the PDs for each label, and vice versa. At least to discuss the performance with the PDs, we need to calculate the PDs with two (or more) classes; however, this would result in significant computational complexity and cause a combinatorial explosion if many classes are employed. However, investigation on the basis of topological insight enables us to know whether outliers are included in the mapping acquired by the networks. As a result, we can discuss the stability of the mapping. Therefore, we conclude that the investigation is necessary to guarantee the stability of the NNs even though their performance is superior.

4.2. Investigation with the CIFAR-10 dataset

4.2.1. Training, validation, and test of CNNs

In the investigation with the CIFAR-10 dataset, we built three CNNs and these architectures are illustrated in Fig. 9. The network shown in Fig. 9(a) is a simple CNN and we refer it to *benchmark model*. The network shown in Fig. 9(b) is deeper than the benchmark model and we refer it to *deep model*. The network shown in Fig. 9(c) contains the residual connections presented by He et al. (2016) (the branched arrows depict the residual connections) and we referred it to *ResNet model*. The size of the fully connected layer of all the networks was set to 1024. Features of the fully connected layer are used as the hidden features and are analyzed.

The benchmark model was trained with a plain manner, that is, any tricks to improve the classification accuracy such as data augmentation (Shorten & Khoshgoftaar, 2019) were not used. The deep and ResNet models were trained with such tricks. All the networks were trained with the categorical cross entropy loss and the Adam optimizer. The activation function of the output layer of all the CNNs is the softmax function. Fig. 10 shows the

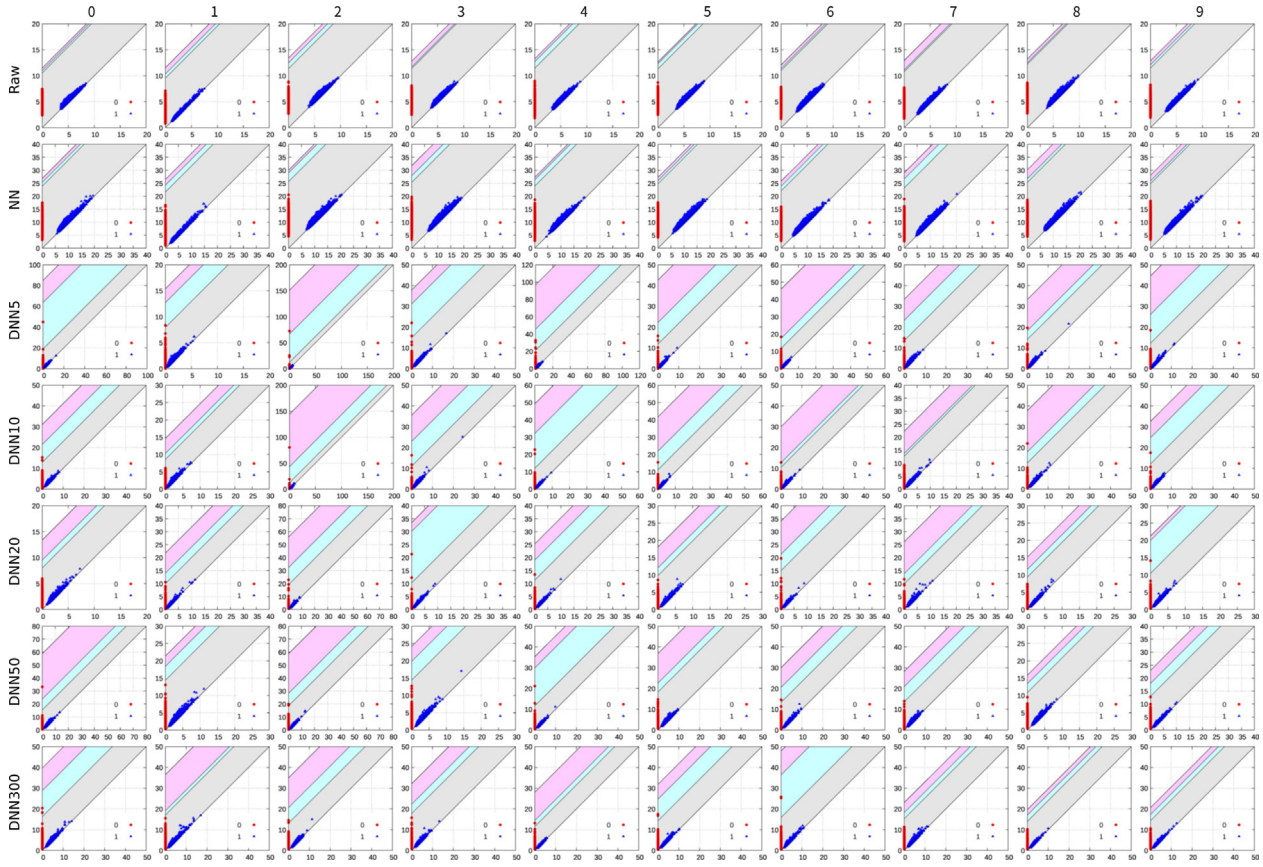

Fig. 8. PDs with the confidence sets obtained from the investigation with the MNIST dataset. The red and blue dots are the 0th and 1st order PH groups, and the gray, cyan, and magenta areas are the 5%, 50%, and 95% confidence bands. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

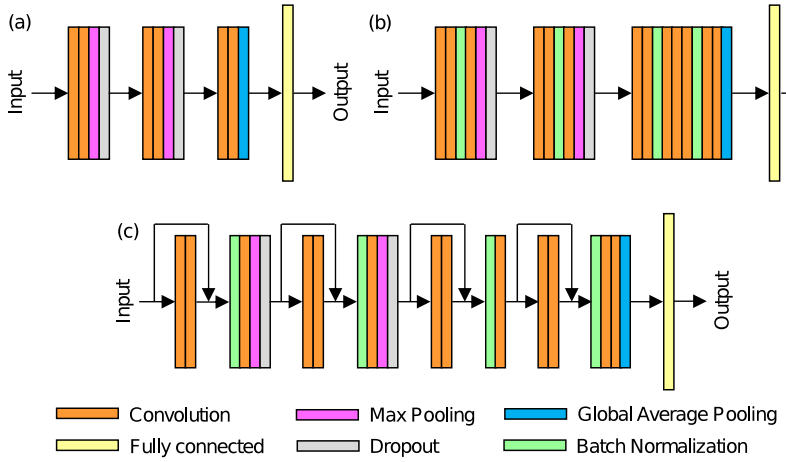

Fig. 9. Convolutional neural networks used in the investigation with the CIFAR-10 dataset. The branched arrows shown in (c) depict the residual connections.

training (solid) and validation (dashed) losses of the CNNs. The training loss of the benchmark model is the smallest; however, its validation loss is the highest in the final epoch. The validation loss of the deep model is unstable; however, it is finally smaller than that of the benchmark model. The training and validation losses of the ResNet model stably converged and its validation loss is the smallest.

Table 2 shows the classification results. The format of this table is almost the same as that of Table 1; however, unlike Table 1, the values higher than 90% are highlighted in bold font. The total classification accuracy of the deep and ResNet models

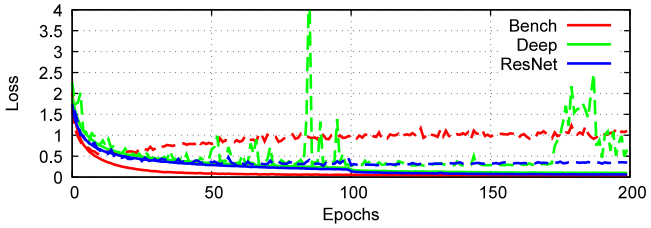

Fig. 10. Training (solid) and validation (dashed) losses of the CNNs shown in Fig. 9.

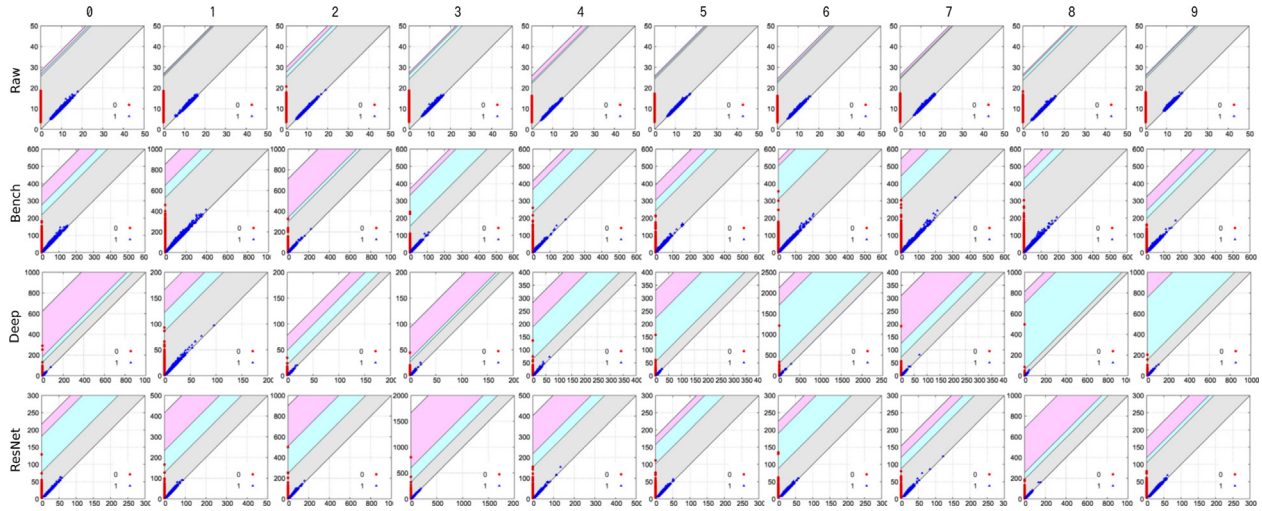

Fig. 11. PDs with the confidence sets obtained from the investigation with the CIFAR-10 dataset. Format of the figures is the same as that of Fig. 8.

is higher than that of the benchmark model. The classification accuracy by the ResNet model is slightly higher than that of the deep model. In following, we discuss their mapping while considering the confidence set for the PDs.

4.2.2. PDs with confidence sets

Fig. 11 shows the confidence sets for the PDs in each label. The format of this figure is the same as that of Fig. 8. The top row figures show the results of the raw images. All the birth–death pairs are included in the 5% confidence band, depicted in gray. In the results by the benchmark model, almost all the birth–death pairs are also included in the 5% confidence band. However, the benchmark model cannot accurately classify the dataset. These results show that different classes of the CIFAR-10 dataset are intertwined and a simple network such as the benchmark model cannot appropriately disentangle data. In other words, simple models do not have enough representability for the classification.

As shown in Table 2, the deep model achieved more accurate classification than the benchmark model. However, the relative size of the 5% confidence band against its 50% and 95% confidence bands, depicted in cyan and magenta, is smaller than that of the benchmark model. Several birth–death pairs can be seen inside the 50% and 95% confidence bands. In particular, the obvious signal indicating that the transformed dataset is topologically complex is found in the result of label 9. Fig. 12 shows the zoom-out figure of label 9 by the deep model (the result shown in Fig. 11 is enlarged to show the 5% and 50% confidence bands). The existence of the birth–death pair far from the origin can be seen. Hence, it could be considered that the deep model cannot acquire the stable mapping even though the classification performance is better than the benchmark model.

The classification accuracy by the ResNet model is slightly higher than that of the deep model, but the obvious far birth–death pairs as shown in Fig. 12 are not found. However, several birth–death pairs are also seen inside the 50% and 95% confidence bands. These results show that we cannot deny the existence of outliers in the mapping by the ResNet model; however, it could be considered that the mapping by the ResNet model is more stable than that of the deep model. These results also suggest that better models could acquire stable mapping more than cheep models. Additionally, it is shown that discussing stability of the mapping is difficult while focusing only on the classification accuracy.

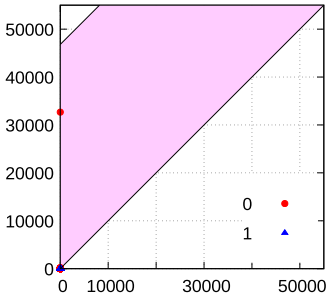

Fig. 12. The birth–death pairs and 95% confidence band in the training result of label 9 by the deep model.

4.3. Limitation

Distance metrics cannot be intuitively used when the dimensionality of the space is quite high, as discussed in Aggarwal et al. (2001). For example, the following equation is satisfied in high-dimensional space

$$\lim_{D\to\infty}\left|\frac{d^D_{\max}(\mathbf{x})-d^D_{\min}(\mathbf{x})}{d^D_{\min}(\mathbf{x})}\right|=0, \tag{13}$$

where $d^D_{\max}(\mathbf{x})$ and $d^D_{\min}(\mathbf{x})$ are the maximum and minimum distances in D-dimensional space, respectively. This indicates that the nearest and farthest distances are the same, and that it is not meaningful to discuss the distance in high-dimensional space. Hence, the PH is ineffective because the calculation thereof is based on the distance. Phenomena such as this are referred to as the *curse of dimensionality*, which is reason for concern because the dimensionality of a dataset is typically high.

Figs. 13 and 14 show the histograms of Eq. (13) for the raw images of the MNIST and CIFAR-10 datasets, respectively. To build the histograms, we calculated the minimum and maximum distances for all data in each class. As can be seen from the figures, the histograms do not converge around zero. Hence, we consider the distance metrics to be effective in both the datasets and the smaller transformed datasets. In fact, the results for NN and DNN300 are different even though the number of hidden features is the same. These results suggest that the results of the investigation described in this paper are valid. If the distance metrics cannot be used, the investigation presented in this paper is invalid. This is a limitation of the method.

Table 2
The classification results on the CIFAR-10 dataset (unit is %). Values larger than 90% are highlighted in bold font.

		0	1	2	3	4	5	6	7	8	9	All
Benchmark	Accuracy	**97.42**	**98.24**	**95.71**	**93.98**	**96.58**	**95.44**	**96.84**	**97.38**	**98.16**	**98.05**	83.90
	Recall	86.30	**94.50**	80.00	70.60	83.10	62.60	**91.60**	89.50	**91.60**	89.20	–
	Specificity	**98.65**	**98.65**	**97.45**	**96.57**	**98.07**	**99.08**	**97.42**	**98.25**	**98.88**	**99.03**	–
	Precision	87.70	88.64	77.74	69.62	82.76	88.41	79.79	85.07	**90.15**	**91.11**	–
	F-measure	86.99	**91.48**	78.85	70.10	82.93	73.30	85.28	87.23	**90.87**	**90.14**	–
Deep	Accuracy	**98.67**	**99.09**	**98.36**	**97.03**	**98.58**	**97.73**	**98.66**	**98.98**	**98.95**	**98.81**	**92.43**
	Recall	**90.40**	**97.30**	**91.70**	80.60	**94.60**	85.90	**97.00**	**95.80**	**94.90**	**96.10**	–
	Specificity	**99.58**	**99.28**	**99.10**	**98.85**	**99.02**	**99.04**	**98.84**	**99.33**	**99.40**	**99.11**	–
	Precision	**96.06**	**93.82**	**91.88**	88.66	**91.48**	**90.89**	**90.31**	**94.10**	**94.61**	**92.31**	–
	F-measure	**93.14**	**95.53**	**91.79**	84.44	**93.01**	88.32	**93.53**	**94.94**	**94.75**	**94.16**	–
ResNet	Accuracy	**98.80**	**99.27**	**98.63**	**97.20**	**98.69**	**97.88**	**98.73**	**99.16**	**99.08**	**98.82**	**93.13**
	Recall	**91.80**	**97.80**	**92.20**	83.70	**94.70**	86.20	**97.80**	**95.20**	**96.20**	**95.70**	–
	Specificity	**99.57**	**99.43**	**99.34**	**98.70**	**99.13**	**99.17**	**98.83**	**99.60**	**99.40**	**99.16**	–
	Precision	**96.02**	**95.04**	**93.98**	87.73	**92.39**	**92.09**	**90.30**	**96.35**	**94.68**	**92.73**	–
	F-measure	**93.86**	**96.40**	**93.08**	85.67	**93.53**	89.04	**93.90**	**95.77**	**95.43**	**94.19**	–

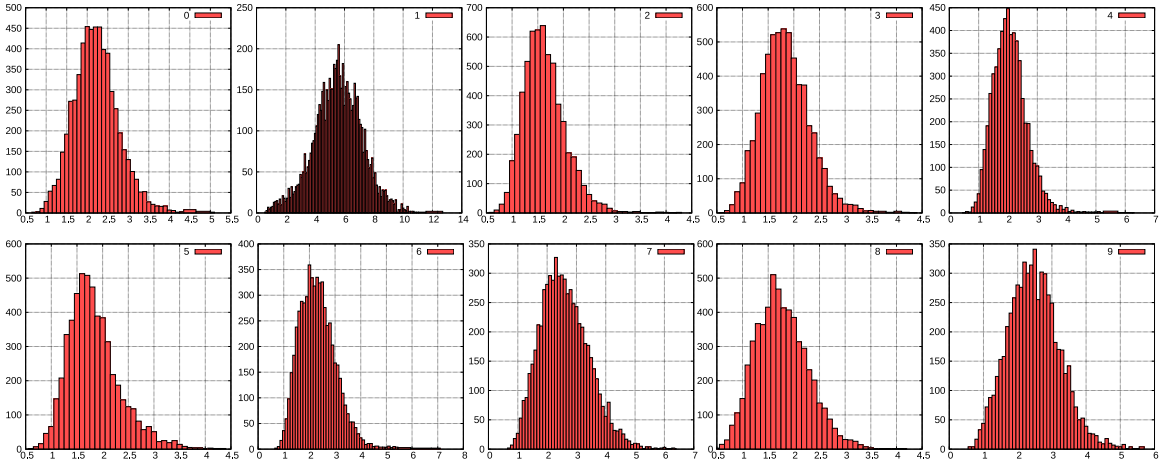

Fig. 13. Histograms obtained with Eq. (13) for each class of the MNIST dataset.

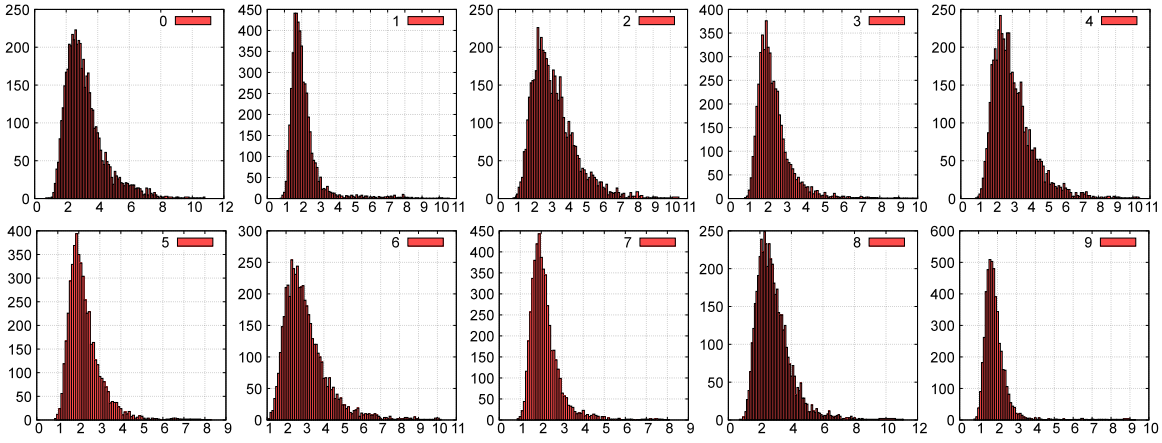

Fig. 14. Histograms obtained with Eq. (13) for each class of the CIFAR-10 dataset.

5. Conclusion

We investigated the relationship between the classification performance of neural networks (NNs) and considered the topological characteristics of a dataset transformed by these NNs. To investigate the topological characteristics, we used the persistent homology (PH), which yields the persistence diagrams (PDs) of a point cloud (we assumed that the dataset or transformed dataset forms a point cloud in high-dimensional space). The PDs are composed of birth–death pairs, and these pairs represent the topological characteristics of the point cloud. However, it is difficult to extract birth–death pairs that topologically characterize the point cloud well from the PDs. To extract stronger topological signals, we used a method for confidence set estimation for the PDs (Fasy et al., 2014). Our decision to use this method was inspired by Hamada and Goto (2018), who showed that the

method can be used to determine whether the point cloud is distributed on a topologically simple manifold. Using this method, we distinguished whether the mapping acquired by the NNs is stable because the method enables us to determine whether the outliers are included in the mapping.

We used the MNIST and CIFAR-10 datasets to conduct our investigation and compared several NNs. As a result of this work, we showed that

- the relationship between the classification performance and the topological characteristics of the transformed dataset is uncertain even though their classification performances are similar,
- we cannot deny the possibility of the existence of outliers in the mapping acquired by the networks by focusing only on the classification performances,
- the investigation based on topological insight is necessary to guarantee that the networks, in particular deep networks, do not acquire an unstable mapping for forcible classification.

We previously studied hybrid localization systems using model- and NN-based methods (Akai et al., 2020a, 2020b, 2018). These methods integrate the NNs into probabilistic localization framework, that is, it is necessary to consider their probabilistic models. To consider the models, guaranteeing the stability is important ability. In future, we aim to apply the presented method to NNs for modeling their inferences and establish methods that can guarantee the safety of the localization. In addition, we would like to find better applications using PH for autonomous navigation as presented in Akai et al. (2021).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by JST COI, Japan under Grant JPMJCE1317 and KAKENHI, Japan under Grant 18K13727.

References

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the international conference on database theory* (pp. 420–434).

Akai, N., Hirayama, T., & Murase, H. (2020a). Hybrid localization using model- and learning-based methods: Fusion of Monte Carlo and E2E localizations via importance sampling. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 6469–6475).

Akai, N., Hirayama, T., & Murase, H. (2020b). Semantic localization considering uncertainty of object recognition. *IEEE Robotics and Automation Letters*, *5*(3), 4384–4391.

Akai, N., Hirayama, T., & Murase, H. (2021). Persistent homology in LiDAR-based ego-vehicle localization. In *Proceedings of the IEEE intelligent vehicles symposium* (accepted).

Akai, N., Morales, L. Y., & Murase, H. (2018). Simultaneous pose and reliability estimation using convolutional neural network and Rao-Blackwellized particle filter. *Advanced Robotics*, *32*(17), 930–944. http://dx.doi.org/10.1080/01691864.2018.1509726.

Akai, N., Morales, L. Y., Takeuchi, E., Yoshihara, Y., & Ninomiya, Y. (2017). Robust localization using 3D NDT scan matching with experimentally determined uncertainty and road marker matching. In *Proceedings of the IEEE intelligent vehicles symposium* (pp. 1357–1364).

Akai, N., Morales, L. Y., Yamaguchi, T., Takeuchi, E., Yoshihara, Y., Okuda, H., Suzuki, T., & Ninomiya, Y. (2017). Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning. In *Proceedings of the IEEE international conference on intelligent transportation systems* (pp. 1147–1152).

Bauer, U. (2019). Ripser: Efficient computation of Vietoris-Rips persistence barcodes. arXiv:1908.02518. Preprint.

Bianchini, M., & 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.

Cohen-Steiner, D., Edelsbrunner, H., & Harer, J. (2007). Stability of persistence diagrams. *Discrete & Computational Geometry*, *37*, 103–120, URL: https://doi.org/10.1007/s00454-006-1276-5.

Edelsbrunner, L., & Zomorodian (2002). Topological persistence and simplification. *Discrete & Computational Geometry*, *28*, 511–533.

Fasy, B. T., Lecci, F., Rinaldo, A., Wasserman, L., Balakrishnan, S., & Singh, A. (2014). Confidence sets for persistence diagrams. *The Annals of Statistics*, *42*(6), 2301–2339.

Glorot, X., Bordes, A., & Bengio, Y. (2011). *Proceedings of machine learning research*: *vol. 15*, *Deep sparse rectifier neural networks* (pp. 315–323). URL: http://proceedings.mlr.press/v15/glorot11a.html.

Guss, W. H., & Salakhutdinov, R. (2018). On characterizing the capacity of neural networks using algebraic topology. arXiv:1802.04443.

Hamada, N., & Goto, K. (2018). Data-driven analysis of pareto set topology. In *Proceedings of the genetic and evolutionary computation conference* (pp. 657–664).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations*. URL: http://arxiv.org/abs/1412.6980.

Le, Q. V., Monga, R., Devin, M., Corrado, G., Chen, K., Ranzato, M., Dean, J., & Ng, A. (2011). Building high-level features using large scale unsupervised learning. arXiv:1112.6209. URL: http://arxiv.org/abs/1112.6209.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605, URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Montavona, G., Samekb, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15.

Naitzat, G., Zhitnikov, A., & Lim, L.-H. (2020). Topology of deep neural networks. arXiv:2004.06093.

Olah, C. (2014). Neural networks, manifolds, and topology. http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/.

Ramamurthy, K. N., Varshney, K., & Mody, K. (2019). *Proceedings of machine learning research*: *vol. 97*, *Topological data analysis of decision boundaries with application to model selection* (pp. 5351–5360). URL: http://proceedings.mlr.press/v97/ramamurthy19a.html.

Rieck, B., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., & Borgwardt, K. (2019). Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International conference on learning representations*. URL: https://openreview.net/forum?id=ByxkijC5FQ.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2016). Grad-CAM: Why did you say that? Visual explanations from deep networks via gradient-based localization. arXiv:1610.02391. URL: http://arxiv.org/abs/1610.02391.

Shorten, C., & Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(60).

Smilkov, D., Thorat, N., Kim, B., Viégas, F. B., & Wattenberg, M. (2017). SmoothGrad: removing noise by adding noise. arXiv:1706.03825. URL: http://arxiv.org/abs/1706.03825.