PLOS ONE

RESEARCH ARTICLE

Comparing feedforward neural networks using independent component analysis on hidden units

Seiya Satoh[1]*, **Kenta Yamagishi**[2], **Tatsuji Takahashi**[2]

1 School of Science and Engineering, Tokyo Denki University, Saitama, Japan, **2** Graduate School of Science and Engineering, Tokyo Denki University, Saitama, Japan

* seiya.satoh@mail.dendai.ac.jp

Citation: Satoh S, Yamagishi K, Takahashi T (2023) Comparing feedforward neural networks using independent component analysis on hidden units. PLoS ONE 18(8): e0290435. https://doi.org/10.1371/journal.pone.0290435

Editor: Anas Bilal, Hainan Normal University, CHINA

Received: June 6, 2022

Accepted: August 8, 2023

Published: August 24, 2023

Data Availability Statement: Disentanglement testing Sprites dataset is available at https://github.com/deepmind/dsprites-dataset.

Funding: This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Competing interests: The authors have declared that no competing interests exist.

Abstract

Neural networks are widely used for classification and regression tasks, but they do not always perform well, nor explicitly inform us of the rationale for their predictions. In this study we propose a novel method of comparing a pair of different feedforward neural networks, which draws on independent components obtained by independent component analysis (ICA) on the hidden layers of these networks. It can compare different feedforward neural networks even when they have different structures, as well as feedforward neural networks that learned partially different datasets, yielding insights into their functionality or performance. We evaluate the proposed method by conducting three experiments with feedforward neural networks that have one hidden layer, and verify whether a pair of feedforward neural networks can be compared by the proposed method when the numbers of hidden units in the layer are different, when the datasets are partially different, and when activation functions are different. The results show that similar independent components are extracted from two feedforward neural networks, even when the three circumstances above are different. Our experiments also reveal that mere comparison of weights or activations does not lead to identifying similar relationships. Through the extraction of independent components, the proposed method can assess whether the internal processing of one neural network resembles that of another. This approach has the potential to help understand the performance of neural networks.

Introduction

Neural networks (NNs) have shown high performance in many tasks, such as image processing [1–3] and disease prediction [4, 5]. However, their high performance is not guaranteed, and when they performed poorly it is hard to determine what caused that. One of the reasons for the difficulty of diagnosing the causes is that NNs usually have numerous parameters and the representations are distributed, resulting in failure of revealing the grounds of their prediction result.

Explainable artificial intelligence research, which is the attempt to explicitly render the basis of the prediction results, has been actively conducted in recent years [6, 7]. For

example, integrated gradients [8] and DeepLift (Deep Learning Important FeaTures) [9] are methods to obtain pixel importance based on gradients and give an explanation of where the model focused on to make their prediction. However, there are some shortcomings in the reliability of the explanation [10, 11]. For example in the cases of integrated gradients and DeepLift, it has been shown that it is possible to add a small perturbation to the input image so that the pixel importance significantly changes while the prediction results remain the same [10]. Also, it is difficult to explain the basis for highlighting the pixels as particularly important even when pixel importance can always be calculated well. For example, explaining whether the prediction was made on the basis of pixel color or shape is difficult.

In this study we propose a novel method which performs independent component analysis (ICA) on the outputs of hidden units of feedforward neural networks (FNNs), and compares those networks based on the obtained independent components (ICs). This method can be used even for comparing FNNs with different structures. The proposed method also can compare two FNNs that solve different tasks. These advantages can lead to useful findings. Despite numerous recent proposals for interpreting the internal representations of neural networks [12–14], to the best of our knowledge, no existing method applies ICA to analyze FNNs' hidden layers. In our experiments we use FNNs with one hidden layer, and verify whether FNNs can be compared by the proposed method when the number of hidden units of FNNs is different, when the data is partially different, and when the activation function is different. Additionally, we demonstrate that a mere comparison of weights or activations does not lead to the discovery of similar relationships.

FNNs and their properties

Here, we focus on FNNs with one hidden layer. Given the input $\boldsymbol{x}^\mu$, the output of the jth hidden unit is

$$h_j^\mu = g(\boldsymbol{w}_j^{\mathrm{T}} \tilde{\boldsymbol{x}}^\mu) \tag{1}$$

where $\tilde{\boldsymbol{x}}^\mu = (1, \boldsymbol{x}^{\mu\mathrm{T}})^{\mathrm{T}}$, $\boldsymbol{x}^\mu = (x_1^\mu, \ldots, x_K^\mu)^{\mathrm{T}}$, K is the number of input variables, $\boldsymbol{w}_j = (w_{j,0}, \ldots, w_{j,K})^{\mathrm{T}}$ are weights between the input layer and the jth hidden unit, and g is the activation function. Here we use either the logistic sigmoid function or the softplus function (a smooth version of rectified linear unit function) [15] as the activation function of the hidden layer. We use an identity function as the activation functions of the input and output layers. Given the input $\boldsymbol{x}^\mu$, the output of the ith output unit is

$$f_i^\mu = \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{h}^\mu \tag{2}$$

where $\boldsymbol{h}^\mu = (1, h_1^\mu, \ldots, h_J^\mu)^{\mathrm{T}}$, J is the number of hidden units, and $\boldsymbol{v}_i = (v_{i,0}, \ldots, v_{i,J})^{\mathrm{T}}$ are weights between the hidden layer and the ith output unit. The parameters of an FNN with J hidden units are $\boldsymbol{\theta}_J = (\boldsymbol{w}_1^{\mathrm{T}}, \ldots, \boldsymbol{w}_J^{\mathrm{T}}, \boldsymbol{v}_1^{\mathrm{T}}, \ldots, \boldsymbol{v}_I^{\mathrm{T}})^{\mathrm{T}}$.

As the objective function we use the mean squared error (MSE). Given a dataset $D = \{(\boldsymbol{x}^\mu, \boldsymbol{y}^\mu) | \mu \in \{1, \ldots, N\}\}$, the MSE is the following:

$$MSE = \frac{1}{IN} \sum_{\mu=1}^{N} \sum_{i=1}^{I} (f_i^\mu - y_i^\mu)^2 \tag{3}$$

where $\boldsymbol{y}^\mu = (y_1^\mu, \ldots, y_I^\mu)^{\mathrm{T}}$, I is the number of output units, and N is the number of data points.

The solutions of an FNN are usually non-unique and dependent on initial parameters due to the properties of the FNN. For example, the input-output function of an FNN remains unchanged when the order of hidden units is altered. In addition, some activation functions have symmetry and the logistic sigmoid function used in our experiments has the following symmetry:

$$\sigma(x) = 1 - \sigma(-x). \tag{4}$$

Furthermore, one property of FNNs is reducibility [16, 17]. For example, consider the optimal solution of an FNN with J hidden units $\widehat{\boldsymbol{\theta}}_J = (\widehat{\boldsymbol{w}}_1{}^{\mathrm{T}}, \ldots, \widehat{\boldsymbol{w}}_J{}^{\mathrm{T}}, \widehat{v}_{1,0}, \ldots, \widehat{v}_{1,J}, \widehat{v}_{2,0}, \ldots, \widehat{v}_{2,J}, \ldots, \widehat{v}_{I,0}, \ldots, \widehat{v}_{I,J})^{\mathrm{T}}$ and the following region:

$$\begin{aligned}\{\boldsymbol{\theta}_{J+1} | \boldsymbol{w}_j = \widehat{\boldsymbol{w}}_j,\ \boldsymbol{w}_{J+1} = (a, 0, \ldots, 0)^{\mathrm{T}}, v_{ij'} = \widehat{v}_{ij'}, v_{i,J+1} = 0, j \in \{1, \ldots, J\},\\ j' \in \{0, \ldots, J\}, i \in \{1, \ldots, I\}\},\end{aligned} \tag{5}$$

where $\boldsymbol{\theta}_{J+1}$ is an FNN with $J + 1$ hidden units and a is a scalar variable. In this case, the input-output function of an FNN on the region is the same as that of the optimal solution $\widehat{\boldsymbol{\theta}}_J$ regardless of the value of a, and the gradient is zero. Therefore, an FNN with $J + 1$ hidden units on the region is reducible to the optimal solution. For the details of reducibility of FNNs, see [16, 17]. The characteristic of reducibility indicates that even when the weight values differ, the input-output relationship can remain consistent. As such, a simple comparison of FNNs' weights is insufficient to establish similarities in their input-output relationships. The experimental findings discussed later confirm this observation.

A method to compare FNNs

It is very difficult to compare FNNs to each other due to their properties, such as the ones mentioned in the previous section. In this paper we propose a method that compare FNNs based on independent components (ICs) obtained by independent component analysis (ICA). Algorithm 1 shows the procedure of the proposed method and Fig 1 shows an illustration of the proposed method.

Algorithm 1 Our method to compare FNNs

1: Let two trained FNNs be FNN 1 and FNN 2
2: Input a video to FNN 1, perform ICA on the activations (outputs) of the hidden units of FNN 1, and let the obtained ICs be denoted by $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{J_1}$ where J_1 is the number of the hidden units of FNN 1.
3: Input a video to FNN 2, perform ICA on the activations of the hidden units of FNN 2, and let the obtained ICs be denoted by $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{J_2}$ where J_2 is the number of the hidden units of FNN 2.
4: Calculate dissimilarities between $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{J_1}$ and $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_{J_2}$.

In Steps 2 and 3 of Algorithm 1 we use ICA [18]. Among several ICA methods, we have chosen to use second-order blind identification (SOBI) [19, 20] (source code available at https://github.com/aludnam/MATLAB/blob/master/sobi/sobi.m). Videos used in Steps 2 and 3 need to be selected appropriately depending on what the FNNs have learned. The simplest way to do this is to use the images (input signals) of the dataset used to train FNN 1 (or FNN 2) as a video. We use this straightforward approach in our experiments.

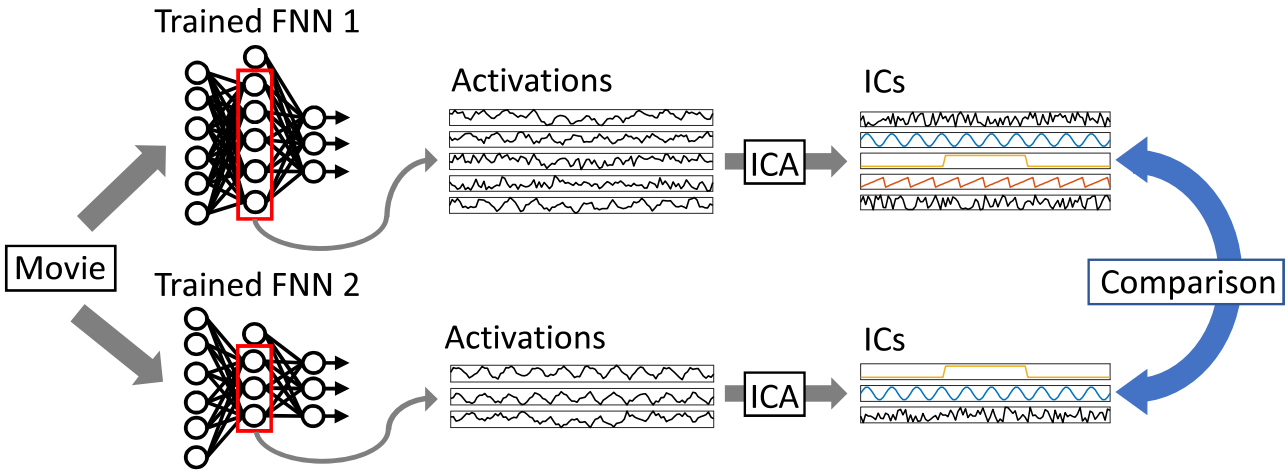


Fig 1. An illustration of our method to compare FNNs.

https://doi.org/10.1371/journal.pone.0290435.g001

In Step 4 the dissimilarity of the obtained ICs is calculated. Here, we use the following function that uses the Euclidean distance:

$$d(\boldsymbol{p}, \boldsymbol{q}) = \min\{d_E(\boldsymbol{p}, \boldsymbol{q}), d_E(\boldsymbol{p}, -\boldsymbol{q})\}, \tag{6}$$

$$d_E(\boldsymbol{p}, \boldsymbol{q}) \equiv ||\tilde{\boldsymbol{p}} - \tilde{\boldsymbol{q}}||, \tag{7}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are ICs, the length of $\boldsymbol{p}$ is the same as that of $\boldsymbol{q}$, and $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ are vectors normalized so that the means of $\boldsymbol{p}$ and $\boldsymbol{q}$ are 0 and the variances are 1. The reason for using $d_E(\boldsymbol{p}, \boldsymbol{q})$ and $d_E(\boldsymbol{p}, -\boldsymbol{q})$ in Eq (6) is that the significance of an IC does not change even when the sign of the IC is inverted.

Experiments

We conducted three experiments to evaluate the proposed method. In the first experiment (Experiment 1) we verified whether FNNs could be compared to each other by the proposed method when the numbers of hidden units of two FNNs were different. In the next experiment (Experiment 2) we verified whether FNNs that learned partially different teacher signals could also be compared. In the final experiment (Experiment 3) we verified whether FNNs could be compared when the activation functions of two FNNs were different. Across all three experiments, we also demonstrated that simply comparing weight values or activations between two FNNs did not necessarily reveal similarities between them. The comparison of weights was performed by calculating the Euclidean distance between the weights from the input layer to each hidden unit. The comparison of activations was done similarly to the comparison of ICs in our method, utilizing the dissimilarity measure given by Eq (6).

We generated datasets using Disentanglement testing Sprites dataset (dSprites) [21]. We used a computer with an Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz and an NVIDIA GeForce RTX 2080 SUPER with MATLAB R2018b. We normalized input signals $x_1^\mu, \ldots, x_K^\mu$

and teacher signals $y_1^\mu, \ldots, y_I^\mu$ as follows:

$$\tilde{x}_k^\mu \leftarrow \frac{x_k^\mu - \min_\mu(x_k^\mu)}{\max_\mu(x_k^\mu) - \min_\mu(x_k^\mu)}, \tag{8}$$

$$\tilde{y}_i^\mu \leftarrow \frac{y_i^\mu - y_i^{\text{mean}}}{y_i^{\text{std}}}, \tag{9}$$

where

$$y_i^{\text{mean}} = \frac{1}{N}\sum_{\mu=1}^{N} y_i^\mu, \tag{10}$$

$$y_i^{\text{std}} = \sqrt{\frac{1}{N}\sum_{\mu=1}^{N} (y_i^\mu - y_i^{\text{mean}})^2}, \tag{11}$$

$\mu \in \{1, \ldots, N\}$, $k \in \{1, \ldots, K\}$, and $i \in \{1, \ldots, I\}$. We used FNNs whose outputs are written in Eq (2). As a learning method we used scaled conjugate gradient obtained from the Netlab toolbox (available at https://jp.mathworks.com/matlabcentral/fileexchange/2654-netlab) [22]. For each learning trial we set the maximum number of epochs to 1,000.

Experiment 1: FNNs with different numbers of hidden units

In Experiment 1 we verified whether two FNNs could be compared by the proposed method when the numbers of the hidden units of these FNNs were different. We used the logistic sigmoid function as the activation function of the hidden layers. Fig 2 shows a part of the input signals (images) of the dataset. We used the size (0.7 or 1), shape (square or ellipse), and color (white or dim gray) of the figure in an input signal as the teacher signal. We set the number of pairs of an input signal and teacher signal to 200 (= 2 (size) × 2 (shape) × 2 (color) × 25(position)). We call this dataset 1. Table 1 summarizes all datasets used in our experiments.

First, we set the number of hidden units J from 1 to 10, and performed 5 trials for each J changing initial weights. Fig 3 shows MSEs after learning. When J was 3 or more, the minimum value of MSEs was 10^{-5}. Notably, an increase in the number of hidden units from 2 to 3 led to a significant decrease in the MSE, implying a necessity for at least 3 hidden units. We call the FNN with the smallest MSE at $J = 3$ FNN 1 and the FNN with the smallest MSE at $J = 10$ FNN 2. Table 2 encompasses all FNNs compared using our method.

We compared FNN 1 and FNN 2 using the proposed method. As the video for Steps 2 and 3 of Algorithm 1 we used a video in which the images of dataset 1 were arranged in the order of data numbers. Fig 4 shows the activations $\boldsymbol{h}_1^1, \boldsymbol{h}_2^1, \boldsymbol{h}_3^1$ of the hidden units of FNN 1 and their ICs $\boldsymbol{p}_1^1, \boldsymbol{p}_2^1, \boldsymbol{p}_3^1$; Fig 5 shows the activations $\boldsymbol{h}_1^2, \ldots, \boldsymbol{h}_{10}^2$ of the hidden units of FNN 2 and their ICs $\boldsymbol{p}_1^2, \ldots, \boldsymbol{p}_{10}^2$. Note that the magnitude of the values of an IC shown in Figs 4b and 5b has no meaning, since each IC was normalized before calculating dissimilarities.

Fig 6c shows the dissimilarities between the ICs of FNNs 1 and 2. In addition, Fig 6a shows the Euclidean distances between the weights $\boldsymbol{w}_1^1, \boldsymbol{w}_2^1, \boldsymbol{w}_3^1$ of FNN 1 and those $\boldsymbol{w}_1^2, \ldots, \boldsymbol{w}_{10}^2$ of FNN 2. Meanwhile, Fig 6b shows the dissimilarities between the activations of FNNs 1 and 2. The values in Fig 6c are the values of dissimilarities. For example, the value in the first column of the third row is the value of $d(\boldsymbol{p}_3^1, \boldsymbol{p}_1^2)$, which is 2.28. As shown in Fig 6b and 6c, all the dissimilarity values were greater than 5 except for $d(\boldsymbol{h}_1^1, \boldsymbol{h}_8^2)$ in the comparisons of activations. Alternatively, $d(\boldsymbol{p}_1^1, \boldsymbol{p}_3^2)$, $d(\boldsymbol{p}_2^1, \boldsymbol{p}_2^2)$, and $d(\boldsymbol{p}_3^1, \boldsymbol{p}_1^2)$ were less than 5 in the comparisons of ICs.

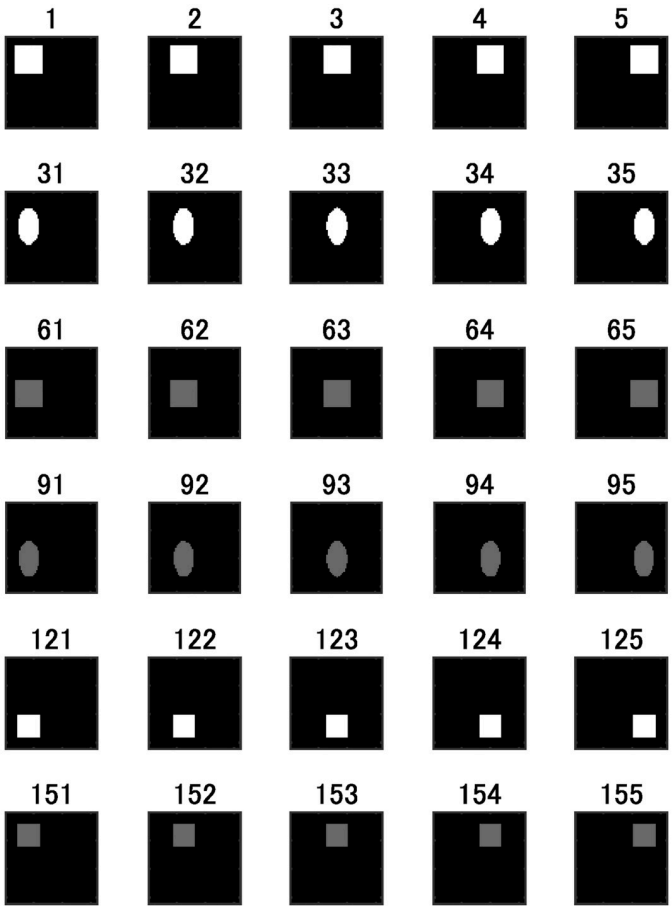


Fig 2. Input images. The number above each image represents the data number.

https://doi.org/10.1371/journal.pone.0290435.g002

Activations of FNN 1 were not similar to those of FNN 2 in most cases, but three ICs of FNN 1 were similar to three ICs of FNN 2, each of which is considered an IC that contributes to size, color, and shape recognition.

As depicted in Fig 6a, when merely comparing weights, the weights $\boldsymbol{w}_1^2$ of FNN 2 were most similar to the weights $\boldsymbol{w}_1^1$ of FNN 1, with the Euclidean distance being 0.437. However, the smallest Euclidean distance to the weights $\boldsymbol{w}_2^1$ was 2.85, which was 6.5 times the Euclidean

Table 1. Datasets used in the experiments.

		Dataset 1	Dataset 2
Input signals	**Size**	2 types (0.7 or 1)	
	Shape	2 types (square or ellipse)	
	Color	2 types (white or dim gray)	
	Position	25 types	
Teacher signals	**Size**	✓	
	Shape	✓	✓
	Color	✓	
	Position		✓

https://doi.org/10.1371/journal.pone.0290435.t001

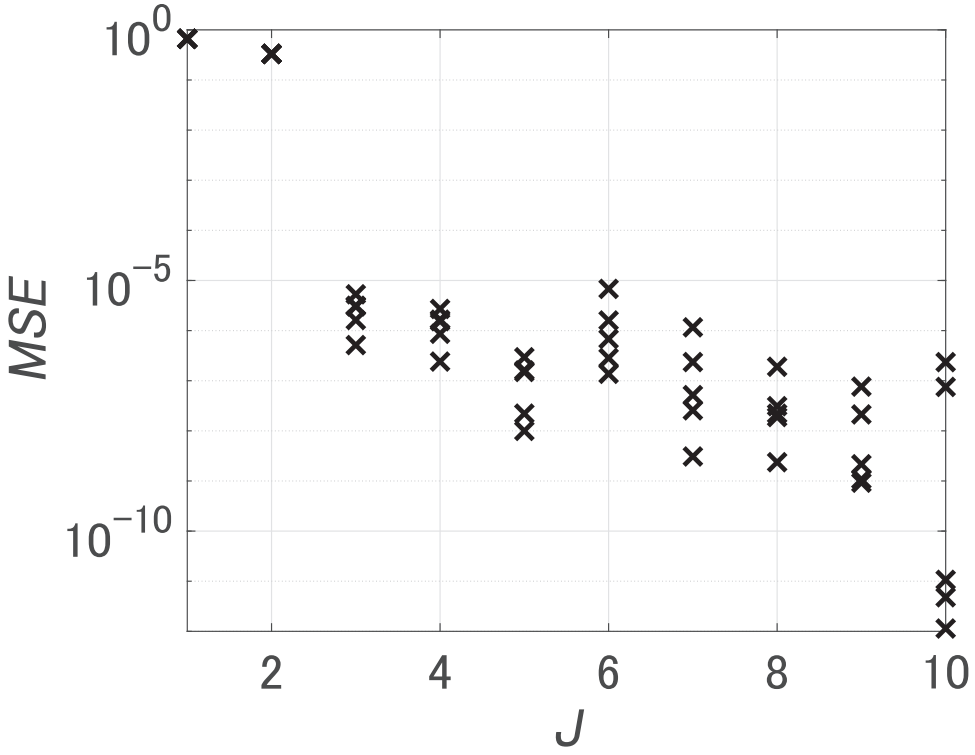


Fig 3. MSEs for dataset 1. J is the number of hidden units.

https://doi.org/10.1371/journal.pone.0290435.g003

distance between $\boldsymbol{w}_1^1$ and $\boldsymbol{w}_1^2$. Similarly, the smallest Euclidean distance to $\boldsymbol{w}_3^1$ was 4.28, which was 9.8 times larger than the Euclidean distance between $\boldsymbol{w}_1^1$ and $\boldsymbol{w}_1^2$. Consequently, we were unable to find weights similar to $\boldsymbol{w}_2^1$ and $\boldsymbol{w}_3^1$.

From the above we consider that similar ICs could be extracted in two FNNs with different numbers of hidden units using our method. In contrast, a simple comparison of the weight values or activations between the two FNNs did not yield any significant findings of similar relationships.

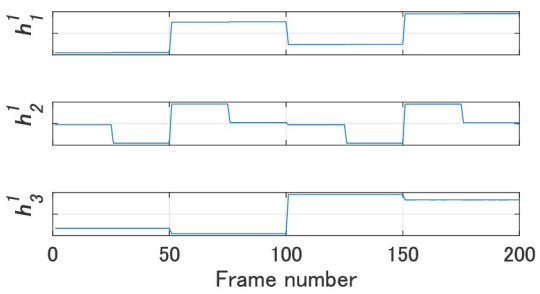
Experiment 2: Partially different teacher signals

In Experiment 2 we verified whether two FNNs that learned partially different teacher signals could also be compared by our method. We used an FNN that learned a different dataset (dataset 2) than dataset 1 and compared that FNN and another FNN that learned dataset 1. Specifically, the input images of dataset 2 were the same as those of dataset 1, and the teacher signal was changed to the shape (square, or ellipse) and the position of the figure for learning.

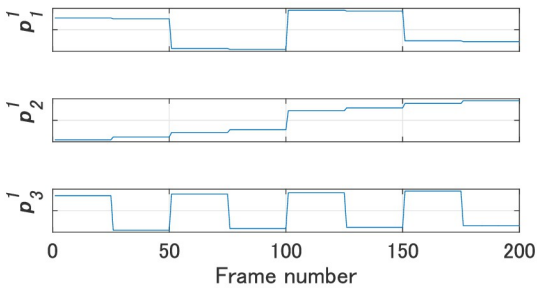
Table 2. FNNs.

	Number of hidden units	Activation function	Dataset
FNN 1	3	Logistic sigmoid	Dataset 1
FNN 2	10	Logistic sigmoid	Dataset 1
FNN 3	10	Logistic sigmoid	Dataset 2
FNN 4	3	Softplus	Dataset 1

https://doi.org/10.1371/journal.pone.0290435.t002



(a) Activations. $\boldsymbol{h}_j^1$ represents the activations of the jth hidden unit.



(b) ICs $\boldsymbol{p}_1^1, \boldsymbol{p}_2^1, \boldsymbol{p}_3^1$.

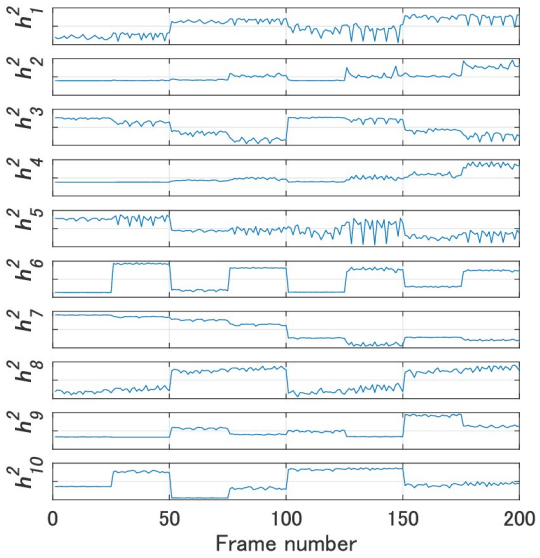
Fig 4. Activations of the hidden units of FNN 1 and their ICs for Experiment 1.

https://doi.org/10.1371/journal.pone.0290435.g004

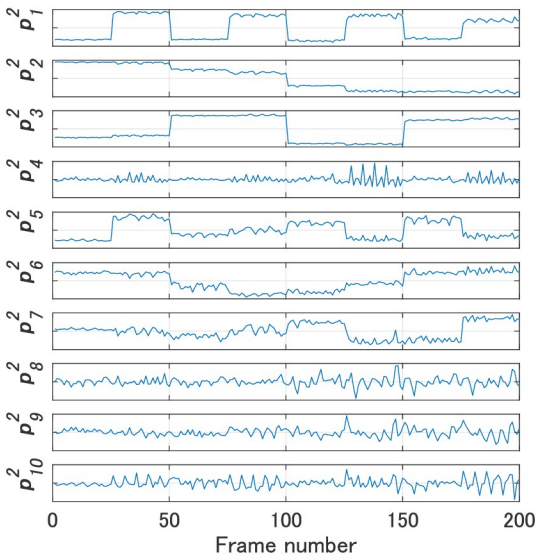
Therefore, only the shape is included in the teacher signals of both dataset 1 and dataset 2. We used the same activation functions as in Experiment 1.

First, we set the number of hidden units J from 1 to 10, and performed 5 trials to learn dataset 2 for each J changing initial weights. Fig 7 shows MSEs after learning. We used the FNN with the smallest MSE at $J = 10$ as FNN 3.

We compared FNN 2, which we used in Experiment 1, and FNN 3 by our method. We used the same video used in Experiment 1 for Steps 2 and 3 of Algorithm 1. Fig 8 shows the activations $\boldsymbol{h}_1^3, \ldots, \boldsymbol{h}_{10}^3$ of the hidden units of FNN 3 and their ICs $\boldsymbol{p}_1^3, \ldots, \boldsymbol{p}_{10}^3$. Fig 9c shows the dissimilarities between the ICs of FNNs 2 and 3. In addition, Fig 9a shows the Euclidean distances between the weights $\boldsymbol{w}_1^2, \ldots, \boldsymbol{w}_{10}^2$ of FNN 2 and those $\boldsymbol{w}_1^3, \ldots, \boldsymbol{w}_{10}^3$ of FNN 3, and Fig 9b shows the dissimilarities between the activations of FNNs 2 and 3. As shown in Fig 9b and



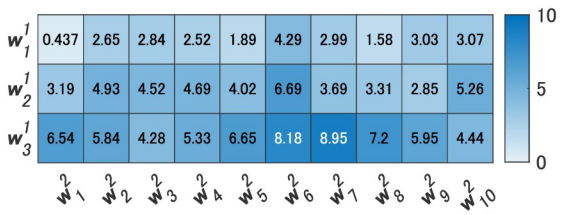
(a) Activations. $\boldsymbol{h}_j^2$ represents the activations of the jth hidden unit.



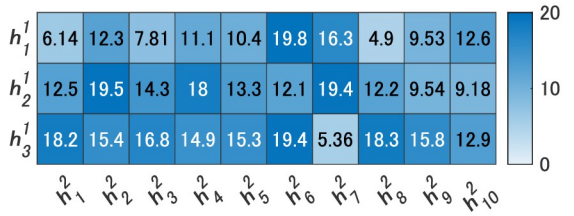
(b) ICs $\boldsymbol{p}_1^2, \ldots, \boldsymbol{p}_{10}^2$.

Fig 5. Activations of the hidden units of FNN 2 and their ICs for Experiment 1.

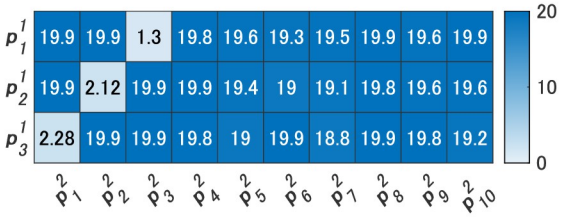
https://doi.org/10.1371/journal.pone.0290435.g005



(a) Euclidean distances between the weights of FNNs 1 and 2.



(b) Dissimilarities between the activations of FNNs 1 and 2.



(c) Dissimilarities between the ICs for FNNs 1 and 2 (our method).

Fig 6. Heatmaps of the Euclidean distances and dissimilarities for Experiment 1.

https://doi.org/10.1371/journal.pone.0290435.g006

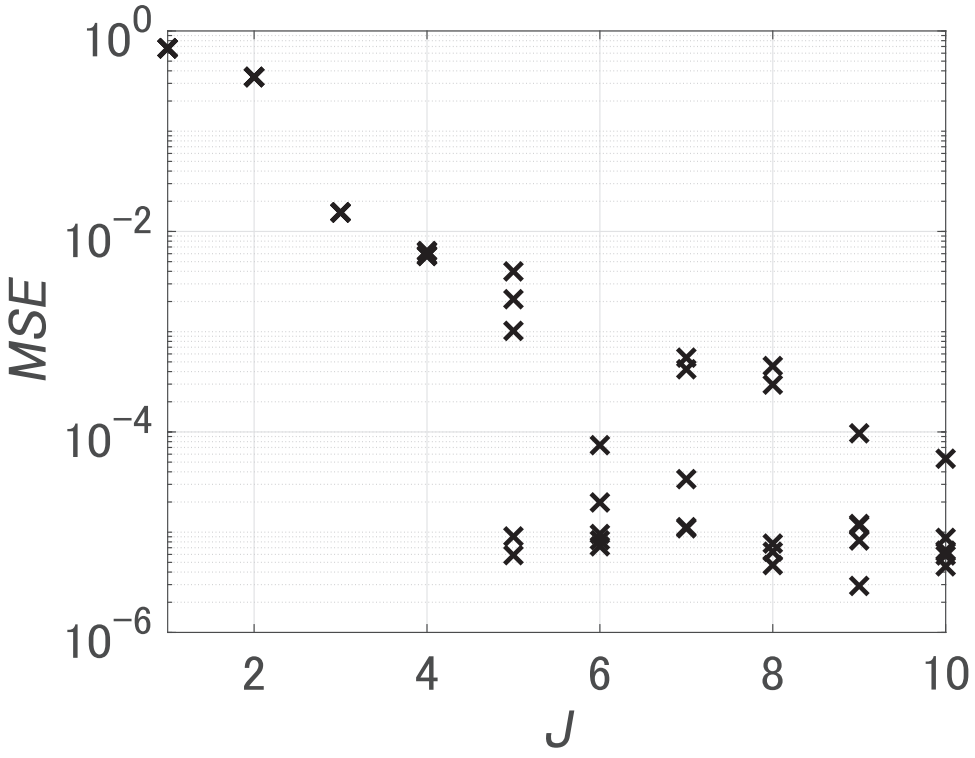
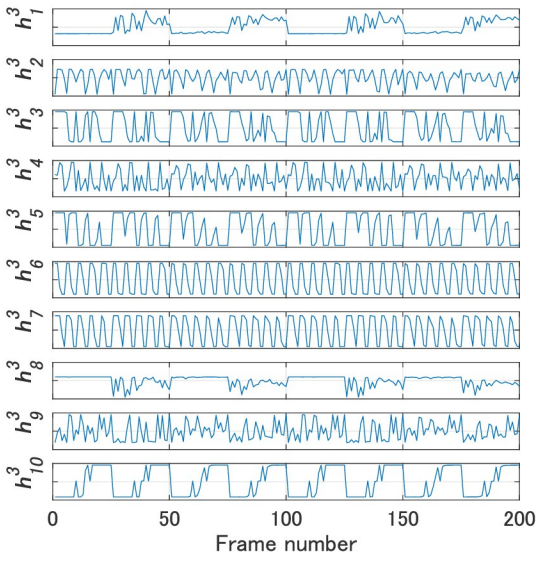
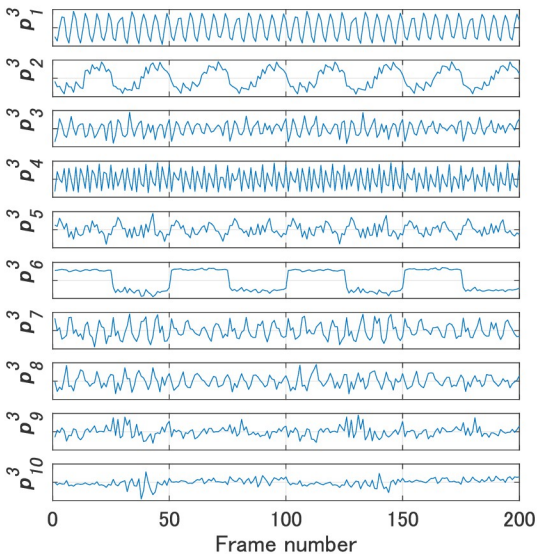


Fig 7. MSEs for dataset 2. J is the number of hidden units.

https://doi.org/10.1371/journal.pone.0290435.g007



(a) Activations. $\boldsymbol{h}_j^3$ represents the activations of the jth hidden unit.



(b) ICs $\boldsymbol{p}_1^3, ..., \boldsymbol{p}_{10}^3$.

Fig 8. Activations of the hidden units of FNN 3 and their ICs for Experiment 2.

https://doi.org/10.1371/journal.pone.0290435.g008

9c, while all the values were greater than 5 in the comparisons of the activations, $d(\boldsymbol{p}_5^2, \boldsymbol{p}_6^3)$ was a small value in the comparisons of the ICs. These ICs are considered to contribute to shape recognition.

When simply comparing weights as shown in Fig 9a, the smallest Euclidean distance was between weights $\boldsymbol{w}_1^2$ of FNN 2 and $\boldsymbol{w}_4^3$ of FNN 3. However, the value was not below 0.5 as in Experiment 1, but was 2.53. It was found that the distribution of values did not clearly distinguish between those that are similar and those that are not, as in the comparison of ICs in our method.

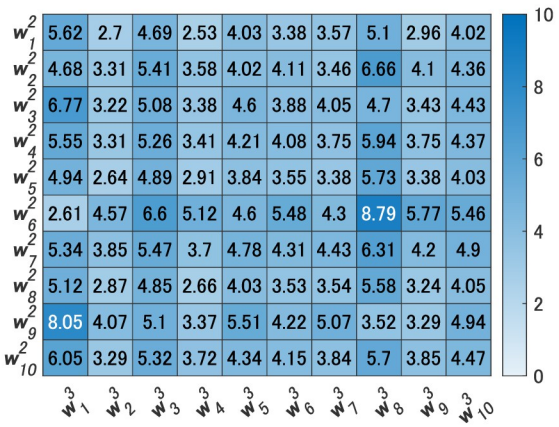
From the above, we consider that our method could extract similar ICs from two FNNs that had learned from partially different teacher signals. However, simple weight or activation comparison between the two FNNs did not result in meaningful discovery of similar relationships.

Experiment 3: Different activation functions

In Experiment 3 we verified whether two FNNs could be compared by the proposed method when the activation functions were different. We used the logistic sigmoid and softplus functions as the activation functions of the hidden layers. We used FNN 2 as an FNN with the logistic sigmoid function, which we used in Experiments 1 and 2.

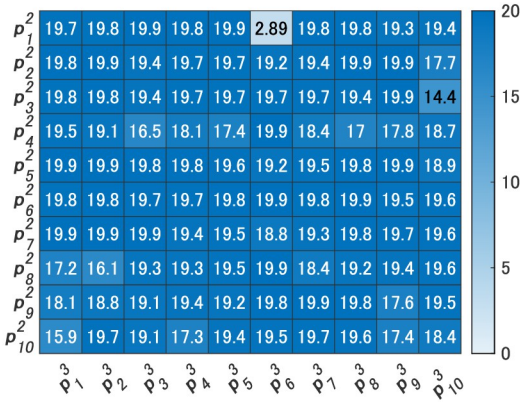
In order to prepare a trained FNN with the softplus function, we first set the number of hidden units J from 1 to 10 and performed 5 trials to learn dataset 1 for each J changing initial weights. Fig 10 shows MSEs after learning. We used the FNN with the smallest MSE at $J = 3$ as FNN 4.

We compared FNN 2 and FNN 4 by our method. We used the same video used in Experiments 1 and 2 as the video for Steps 2 and 3 of Algorithm 1. Fig 11 shows the activations $\boldsymbol{h}_1^4, \boldsymbol{h}_2^4, \boldsymbol{h}_3^4$ of the hidden units of FNN 4 and their ICs $\boldsymbol{p}_1^4, \boldsymbol{p}_2^4, \boldsymbol{p}_3^4$. Fig 12c shows the dissimilarities between the ICs of FNNs 2 and 4. In addition, Fig 12a shows the Euclidean distances between



(a) Euclidean distances between the weights of FNNs 2 and 3.

(b) Dissimilarities between the activations of FNNs 2 and 3.



(c) Dissimilarities between the ICs for FNNs 2 and 3 (our method).

Fig 9. Heatmaps of the Euclidean distances and dissimilarities for Experiment 2.

https://doi.org/10.1371/journal.pone.0290435.g009

the weights $\boldsymbol{w}^2_1, \ldots, \boldsymbol{w}^2_{10}$ of FNN 2 and those $\boldsymbol{w}^4_1, \boldsymbol{w}^4_2, \boldsymbol{w}^4_3$ of FNN 4, and Fig 12b shows the dissimilarities between the activations of FNNs 2 and 4. As shown in Fig 12b and 12c, while all the values were greater than 5 in the comparisons of the activations, the values $d(\boldsymbol{p}^2_3, \boldsymbol{p}^4_1)$, $d(\boldsymbol{p}^2_1, \boldsymbol{p}^4_2)$, and $d(\boldsymbol{p}^2_2, \boldsymbol{p}^4_3)$ were less than 5 in the comparisons of ICs. Activations of FNN 2 were not similar to those of FNN 4, but three ICs of FNN 2 were similar to three ICs of FNN 4, each of which was considered an IC that contributes to size, color, and shape recognition, as in Experiment 1.

As shown in Fig 12a, simply comparing weights did not lead to the meaningful discovery of similar relationships. similar ICs were extracted in two FNNs with different activation functions.

Conclusions

In this paper we proposed a novel method of comparing FNNs based on the ICs obtained by performing ICA on the hidden layers of the FNNs. In our three experiments we showed that

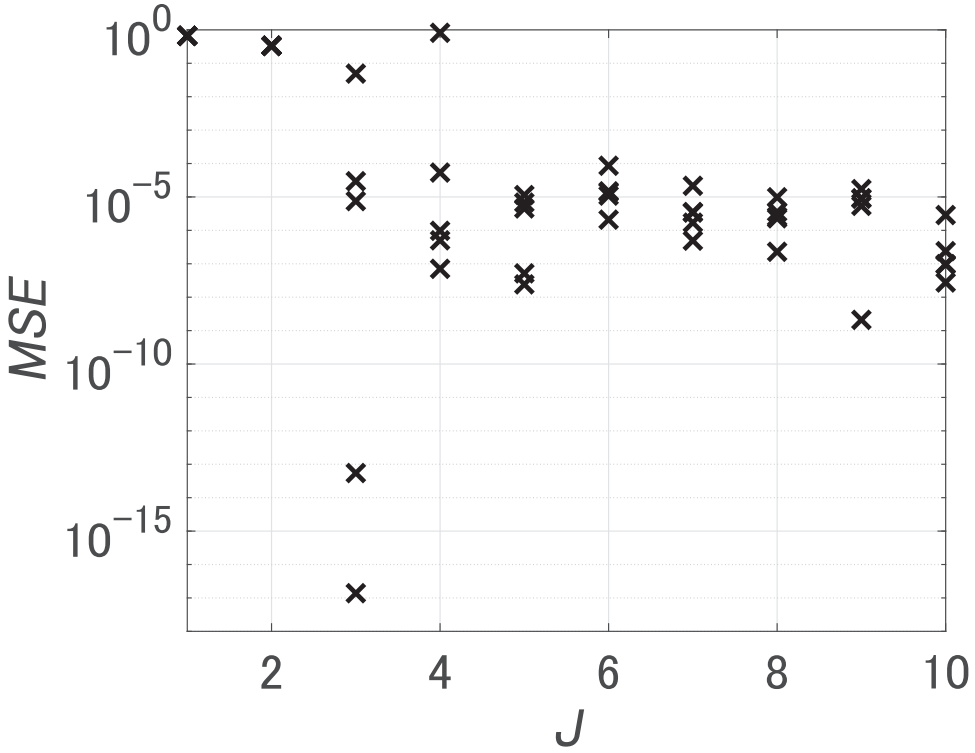
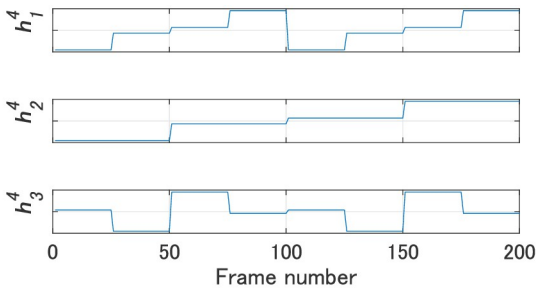


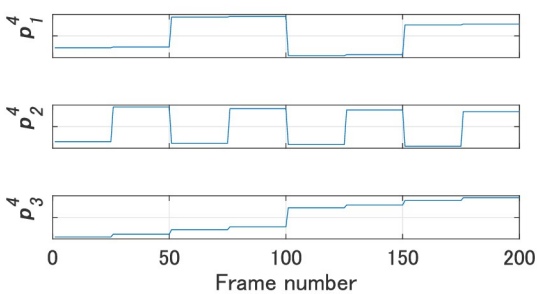
Fig 10. MSEs for dataset 1 (FNNs with softmax function). J is the number of hidden units.

https://doi.org/10.1371/journal.pone.0290435.g010

similar ICs were extracted from two FNNs, even when the numbers of hidden units were different; that similar ICs were extracted from two FNNs, even when the teacher signals were partially different; and that similar ICs were extracted from two FNNs, even when the activation functions were different. Our findings also revealed that direct comparisons of weights or activations did not uncover meaningful similarities. Note that we used only FNNs with a single hidden layer, utilizing either the logistic sigmoid function or the softplus function as the activation function. Also note that we used only SOBI as ICA and datasets generated using dSprites.



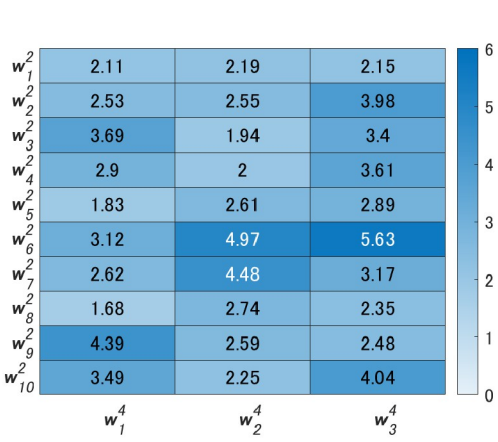
(a) Activations. $\boldsymbol{h}_j^4$ represents the activations of the jth hidden unit.



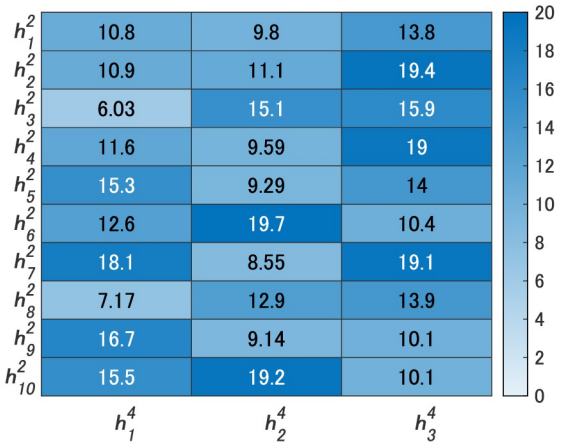
(b) ICs $\boldsymbol{p}_1^4, \boldsymbol{p}_2^4, \boldsymbol{p}_3^4$.

Fig 11. Activations of the hidden units of FNN 4 and their ICs for Experiment 3.

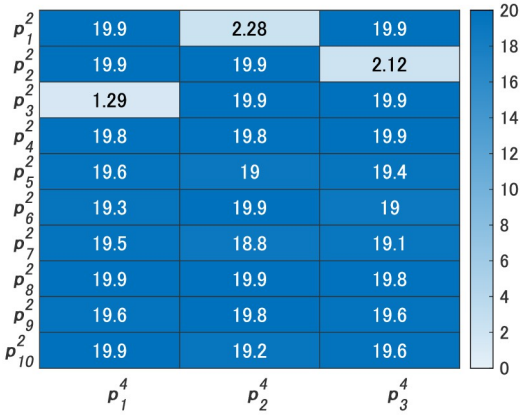
https://doi.org/10.1371/journal.pone.0290435.g011



(a) Euclidean distances between the weights of FNNs 2 and 4.



(b) Dissimilarities between the activations of FNNs 2 and 4.



(c) Dissimilarities between the ICs for FNNs 2 and 4 (our method).

Fig 12. Heatmaps of the Euclidean distances and dissimilarities for Experiment 3.

https://doi.org/10.1371/journal.pone.0290435.g012

In the future, we will further investigate the availability of our method by conducting more complex experiments, such as experiments with FNNs that have two or more hidden layers, experiments with more complex data, and experiments with nonlinear ICA.

Author Contributions

Conceptualization: Seiya Satoh, Tatsuji Takahashi.

Data curation: Seiya Satoh, Kenta Yamagishi.

Formal analysis: Seiya Satoh, Kenta Yamagishi.

Funding acquisition: Seiya Satoh.

Investigation: Seiya Satoh, Kenta Yamagishi.

Methodology: Seiya Satoh, Kenta Yamagishi.

Project administration: Seiya Satoh, Tatsuji Takahashi.

Resources: Seiya Satoh.

Software: Seiya Satoh, Kenta Yamagishi.

Supervision: Seiya Satoh, Tatsuji Takahashi.

Validation: Seiya Satoh, Tatsuji Takahashi.

Visualization: Seiya Satoh, Kenta Yamagishi.

Writing – original draft: Seiya Satoh.

Writing – review & editing: Seiya Satoh, Tatsuji Takahashi.

References

1. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009. p. 248–255.
2. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. International journal of computer vision. 2015; 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
3. 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; 2016. p. 770–778.
4. Karayılan T, Kılıç Ö. Prediction of heart disease using neural network. In: 2017 International Conference on Computer Science and Engineering (UBMK). IEEE; 2017. p. 719–723.
5. Yadav SS, Jadhav SM. Deep convolutional neural network based medical image classification for disease diagnosis. Journal of Big Data. 2019; 6(1):1–18. https://doi.org/10.1186/s40537-019-0276-2
6. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE access. 2018; 6:52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052
7. Das A, Rad P. Opportunities and challenges in explainable artificial intelligence (xai): A survey. arXiv preprint arXiv:200611371. 2020;.
8. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. In: International Conference on Machine Learning. PMLR; 2017. p. 3319–3328.
9. Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: International Conference on Machine Learning. PMLR; 2017. p. 3145–3153.
10. Ghorbani A, Abid A, Zou J. Interpretation of neural networks is fragile. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 3681–3688.
11. Kindermans PJ, Hooker S, Adebayo J, Alber M, Schütt KT, Dähne S, et al. The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer; 2019. p. 267–280.
12. Salahuddin Z, Woodruff HC, Chatterjee A, Lambin P. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. Computers in biology and medicine. 2022; 140:105111. https://doi.org/10.1016/j.compbiomed.2021.105111
13. Minh D, Wang HX, Li YF, Nguyen TN. Explainable artificial intelligence: a comprehensive review. Artificial Intelligence Review. 2022; p. 1–66.
14. Zhang Y, Tiňo P, Leonardis A, Tang K. A survey on neural network interpretability. IEEE Transactions on Emerging Topics in Computational Intelligence. 2021; 5(5):726–742. https://doi.org/10.1109/TETCI.2021.3100641
15. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2011. p. 315–323.
16. Sussmann HJ. Uniqueness of the weights for minimal feedforward nets with a given input-output map. Neural networks. 1992; 5(4):589–593. https://doi.org/10.1016/S0893-6080(05)80037-1
17. Fukumizu K, Amari Si. Local minima and plateaus in hierarchical structures of multilayer perceptrons. Neural networks. 2000; 13(3):317–327. https://doi.org/10.1016/S0893-6080(00)00009-5 PMID: 10937965
18. Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. Neural networks. 2000; 13(4-5):411–430. https://doi.org/10.1016/S0893-6080(00)00026-5 PMID: 10946390

19. Belouchrani A, Abed-Meraim K, Cardoso JF, Moulines E. A blind source separation technique using second-order statistics. IEEE Transactions on signal processing. 1997; 45(2):434–444. https://doi.org/10.1109/78.554307

20. Sahonero-Alvarez G, Calderon H. A comparison of SOBI, FastICA, JADE and Infomax algorithms. In: Proceedings of the 8th International Multi-Conference on Complexity, Informatics and Cybernetics; 2017. p. 17–22.

21. Matthey L, Higgins I, Hassabis D, Lerchner A. dSprites: Disentanglement testing Sprites dataset; 2017. https://github.com/deepmind/dsprites-dataset/.

22. Nabney I. NETLAB: algorithms for pattern recognition. Springer Science & Business Media; 2002.