IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 19, 2020, accepted May 28, 2020, date of publication June 1, 2020, date of current version June 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999155

Mixing Autoencoder With Classifier: Conceptual Data Visualization

PITOYO HARTONO, (Member, IEEE)
School of Engineering, Chukyo University, Nagoya 466-8666, Japan
e-mail: hartono@ieee.org

ABSTRACT In this paper, a neural network that is able to form a low-dimensional topological hidden representation is explained. The neural network can be trained as an autoencoder, as a classifier or as a mixture of both and produces a different low-dimensional topological map for each. When it is trained as an autoencoder, the inherent topological structure of the data can be visualized, while when it is trained as a classifier, a topological structure that is further constrained by a given concept, for example, the labels of the data, can be formed. Here, the resulting visualization is not only structural but also conceptual. The proposed neural network significantly differs from many dimensional reduction models, primarily in its ability to execute both supervised and unsupervised dimensional reduction and its ability to visualize not only the structure of high-dimensional data but also the concept assigned to them at various levels of abstraction.

INDEX TERMS Autoencoder, concept visualization, dimensional reduction, learning representations, neural network, self-organizing maps, topological representations.

I. INTRODUCTION

In this study, a neural network that is able to build a contextual topological map in its hidden layer is explained. Over the past few years, rich collections of machine learning methods for visualizing high-dimensional data through dimensional reduction have been proposed. Many of them form low-dimensional representations while preserving some inherent characteristics of high-dimensional data. For example, stochastic neighborhood embedding (SNE) [1] and its variants [2], [3] map high-dimensional data into a low-dimensional space while preserving the stochastic neighborhood structure of the data. Locally linear embedding (LLE) [4] is a dimensional reduction method that locally preserves the linear dependency of high-dimensional data, while isometric mapping (ISOMAP) [5], [6] is also a nonlinear dimensional reduction mapping that preserves the geodesic structure of high-dimensional data. Kohonen's self-organizing maps (SOM) [7], [8] is a popular dimensional reduction and visualization method that preserves the topological structure of high-dimensional data in a low-dimensional space. Recently, uniform manifold approximation and projection (UMAP) [9], a manifold learning technique for dimensional reduction based on Riemannian geometry, was proposed; it results in high-quality visualization with a scalable calculation time. All these methods execute unsupervised mapping primarily for visualizing the application-relevant structure of high-dimensional data but ignore the contexts (for example, labels) of the data. There are also many supervised dimensional reduction algorithms that take the context of the data into account. These methods form low-dimensional representations of high-dimensional data by preserving their inherent structures that are relevant to their labels. Thus, the representation is not only structural but also conceptual. Some examples of supervised dimensional reduction methods are as follows. Neighborhood component analysis (NCA) [10] forms low-dimensional representations on which the classification accuracy of the k-nearest neighbors is maximized. A semisupervised version of ISOMAP was proposed in [11], and a combination of multidimensional scaling (MDS) [12], [13] and SOMs that can either be supervised or unsupervised was proposed in [14].

While the methods above are able to achieve visualization on high-dimensional data, they are either supervised or unsupervised. However, data analysis sometimes requires multiple perspectives to extract insightful information. Changing the methods to learn different aspects of data is often problematic, since all the methods execute different criteria in reducing the dimensions of the data.

This study proposes a new hierarchical neural network that, during its learning process, builds a low-dimensional and hence displayable topological representation in its

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

hidden layer. Different from most dimensional reduction methods, this neural network is able to execute supervised learning, unsupervised learning or mixture of the two in one learning framework by controlling a single coefficient in the learning process. The proposed neural network is built based on the previously proposed restricted radial basis function (r-RBF) network [15], [16], which is a hierarchical supervised neural network that generates a two-dimensional topological representation in its hidden layer. Here, the output layer and the learning process are modified so that the network can be trained as an autoencoder, a classifier or a mix of both. When the network is trained as an autoencoder, it forms a low-dimensional representation that encodes a relevant topological structure to reconstruct the high-dimensional input and thus allows visualization of the inherent structure of the data. When it is trained as a classifier, the hidden representation is constrained by the labels of the data; hence, the visualization is not only structural but also conceptual in that different labeling of the same data will produce different representations. The network can also be trained by mixing the autoencoder and classifier, resulting in flexible representations, where the difference between the inherent vectorial characteristics and the characteristics conceptualized by the labels of the data can be learned.

The supervised autoencoder was proposed in [17] to improve the generalizations of neural networks across an array of architectures. The proposed model differs from previous models in that it has the flexibility to be trained as an autoencoder, a classifier, or a mix of both while also being able to form a two-dimensional representation that is visualizable and thus may help in understanding the inherent characteristics of a problem.

In the past few years, new studies on autoencoders have been proposed [18]–[20]. The proposed work differs from those past studies in that it investigates the topological constraints in forming the hidden layers under different contexts. Because the hidden layer here is visualizable, it offers an intuitive understanding of how neural networks embed concepts attached to the data in its hidden representations.

Due to its learning flexibility, the neural network is named the soft-supervised topological autoencoder (STA). The flexibility of this new neural network allows the multilevel concept visualization of high-dimensional data and thus enables discoveries of deeper characteristics of the data that so far have been difficult to achieve within a single learning framework.

The primarily claim for the STA's strength here is not in its generalization ability but its ability in infusing the topological signatures of high-dimensional data under various context into its hidden layer, and thus allowing intuitive understanding on the context-oriented data structure. This strength is due to the topological constraint that intrinsically presents in its hidden layer. Unlike many layered neural networks that distribute the activations in unorganized manners, here the activations are constrained, and thus forcing the hidden representations to reflect the topological features of the data. Further, the STA allows the context of the learning data to interact with the formed topological organization under a well-defined objective function, resulting in unique visualization that gives intuitive understanding for recognizing the context-oriented structure of high-dimensional data. The STA offers a simple method for an initial attempt in contributing to topological data analysis, a new concept in machine learning community [21]–[23].

II. SOFT-SUPERVISED TOPOLOGICAL AUTOENCODER

The framework of the STA is illustrated in Fig. 1. Here, a three-layered STA is presented, in which the hidden layer is a topological layer, where the neurons are aligned in a two-dimensional grid similar to Kohonen's SOMs. The output layer is composed of two parts: a decoder part that reconstructs the encoded input and a classifier part that predicts the labels of the input. In the training process, a mixing coefficient is set to control the weightings of the decoder part and the classifier part; hence, the STA can be trained as an autoencoder, a classifier or a mixture of both.

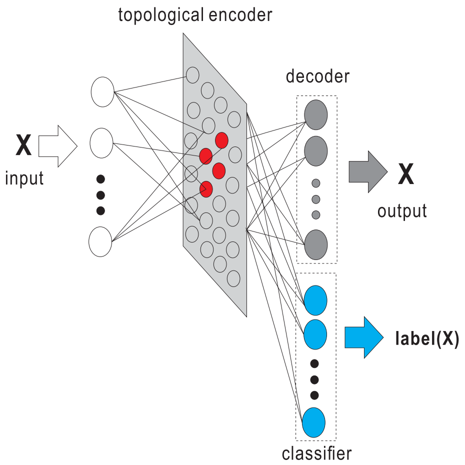


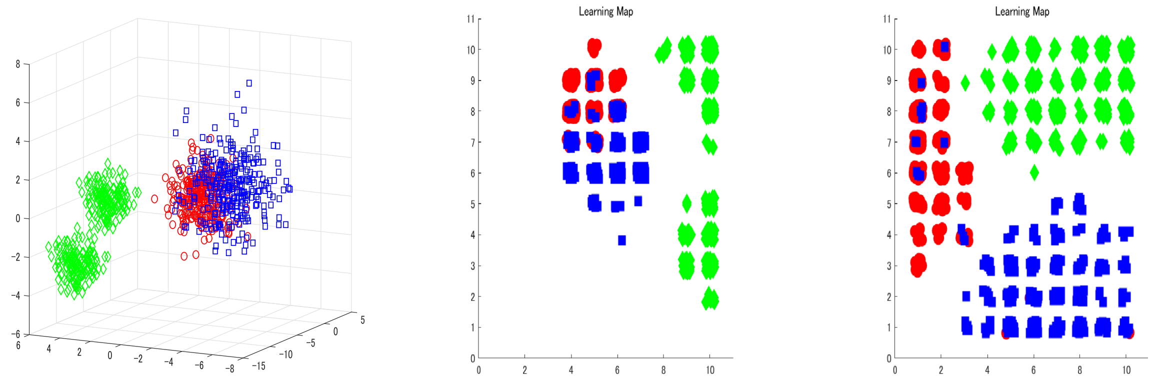
FIGURE 1. Framework of soft-supervised topological autoencoder.

Here, similar to Kohonen's SOMs, due to its low dimensions, it is possible to visualize the topological representation of the inputs and discover their characteristics. However, different from SOMs, which preserve the topological structures of the high-dimensional inputs in their two-dimensional representations, in the STA, the hidden representations are also regulated by the error signals backpropagated from the output layer. In the case where the STA is trained as an autoencoder, the hidden topological representation is formed to encode a topological structure that enables the STA to reconstruct the inputs. In the case of a classifier, the topological structure is further constrained by the requirement to predict the output.

The dynamics of the STA is explained as follows.

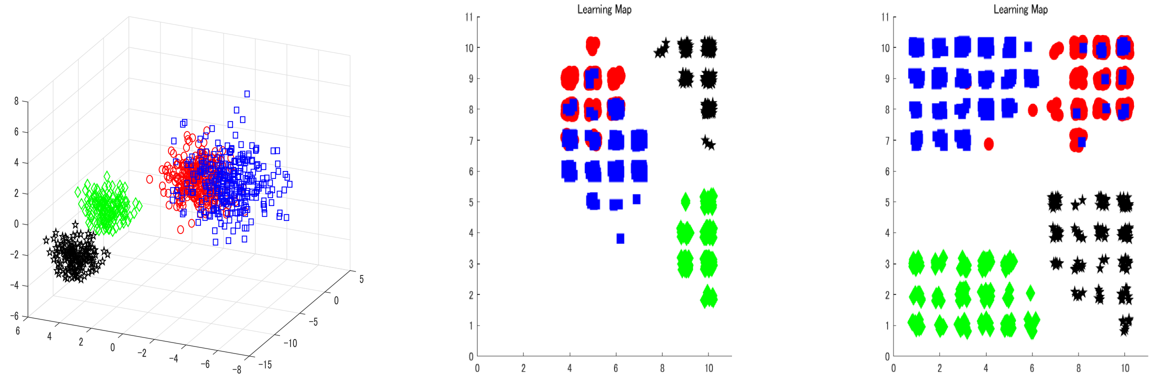
$$win = \underset{j}{arg\ min} \|X - W_j\| \tag{1}$$

$$H_j = N(j, win, t)\mathrm{e}^{-\frac{\|X-W_j\|^2}{2\sigma^2}} \tag{2}$$



(a) 3 Clusters (b) 3 Clusters ($\kappa = 0$) (c) 3 Clusters ($\kappa = 1$)

FIGURE 2. Three clusters. The original data in (a) show that there are many overlapping points in the two clusters represented by the ●s and ■s, while the remaining two clusters are identically labeled. Map in (b) visualizes the hidden representation of the STA when it was trained as an autoencoder ($\kappa = 0$), while (c) visualizes the hidden representation of the STA when it was trained as a classifier ($\kappa = 1$). Figure 2b visualizes the natural distribution of the data in their original high-dimensional space, in which many instances, indicated by the ●s and ■s, overlap, while data points belonging to the same class, marked with a ♦, are separate. In (c), the data points that originally overlapped are more distinctively separated, while the data originally separated into clusters containing data points from the same class are merged. It can be seen from these two figures that the contexts of the data play important roles in shaping the different topological representations. The STA's setting for this experiment: $N_{hid} = 10 \times 10$, $N_0 = 20, N_\infty = 2, t_\infty = 2000$.



(a) 4 Clusters (b) 4 Clusters ($\kappa = 0$) (c) 4 Clusters ($\kappa = 1$)

FIGURE 3. Four clusters. The data distribution is identical to that in the previous problem, but the identically labeled double clusters in the previous problem are labeled differently, being indicated by the ♦s and ★s in (a). (b) visualizes the hidden representation of the STA as an autoencoder. Naturally, the structure of this representation as shown in is exactly the same as the one in 2b, as the labels of the data do not play any role in the learning process. However, when the STA was trained as a classifier, the context of the data changed from that of the previous problem, thus altering the distribution of the hidden representation. The large cluster representing the two different normal distributions with the same labels is now replaced by two adjacent but separated clusters with different labels as shown in (c). The STA's setting for this experiment: $N_{hid} = 10 \times 10$, $N_0 = 20, N_\infty = 2, t_\infty = 2000$.

$$\begin{aligned} N(j, win, t) &= \frac{e^{-dist^2(win,j)}}{S(t)} \\ S(t) &= N_\infty + \frac{1}{2}(N_0 - N_\infty)(1 + cos\frac{\pi t}{t_\infty}) \end{aligned} \tag{3}$$

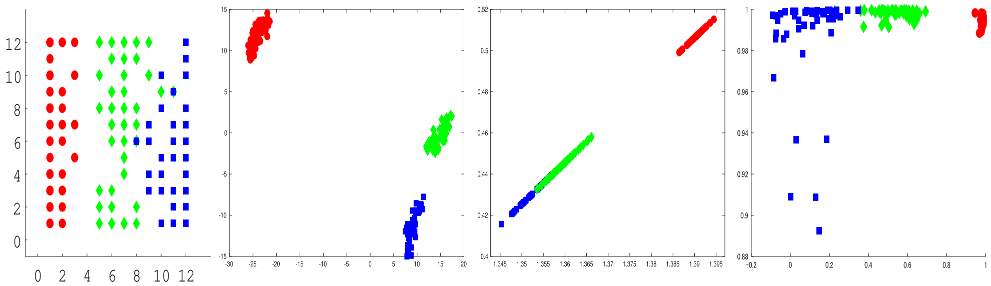
For a high-dimensional input $X \in \mathbb{R}^d$, the STA selects the best matching unit (BMU), win, among all the reference vectors associated with the hidden units of the STA, as shown in Eq. 1, where $W_j \in \mathbb{R}^d$ is the reference vector associated with the j-th hidden unit. The output of the j-th hidden neuron, H_j, is shown in Eq. 2, where $N(j, win, t)$ is the neighborhood function defined in Eq. 3, with an annealing function $S(t)$.

Here, $dist(win.j)$ is the Euclidean distance between the BMU and the j-th neuron on the low-dimensional hidden layer, $N_0 > N_\infty > 0$ are the initial and final values of the annealing term, t is the current epoch, and t_∞ is the termination epoch.

The values of the k-th decoder neuron, O_k^{dec}, and the l-th label neuron, O_l^{cls}, in the output layer are defined in Eq. 4, where $f(x) = \frac{1}{1+e^{-x}}$:

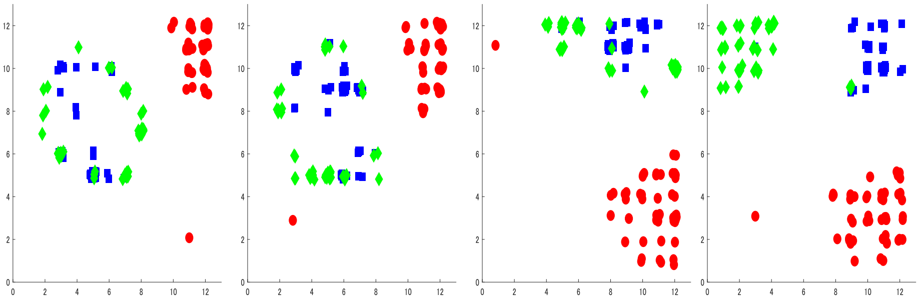
$$\begin{aligned} O_k^{dec} &= f((V_k^{enc})^\top H) \\ O_l^{cls} &= f((V_l^{cls})^\top H) \end{aligned} \tag{4}$$

Here, V_k^{enc} denotes the weight vector leading from the hidden layer to the k-th decoder neuron, which also includes



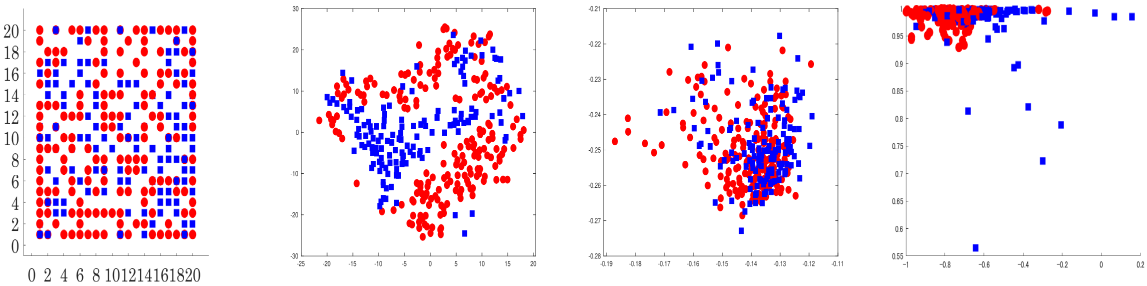
(a) SOM (b) t-SNE (c) MLP-encoder (d) MLP-classifier

FIGURE 4. **Low-dimensional representations: Iris data. The Iris data set is four-dimensional and comprises three labels, where it is known that one of the classes, marked with a ●, is linearly separable from the other two, while those two, marked with a ■ and ♦, are not linearly separable. The low-dimensional representations of the SOM are shown in (a), the representations of the t-SNE are shown in (b), the representation of the MLP-autoencoder is shown in (c), and the two-dimensional supervised representation of the MLP-classifier is presented in (d). Although with different visual appearances, all the representations indicate the intrinsic separability of this problem, in which one of the classes has a large margin of separability relative to the two other classes, while for those two, this is not necessarily so.**



(a) Iris ($\kappa = 0$) (b) Iris ($\kappa = 0.1$) (c) Iris ($\kappa = 0.5$) (d) Iris ($\kappa = 1$)

FIGURE 5. **Iris data. From this experiment, it can be observed that the class represented by the ●s is linearly separable from the other two, as shown in (a) when $\kappa = 0$. By increasing κ, as observed in (b)-(d), the two originally overlapping classes are gradually separated with a large margin, except for a few outliers. The STA's setting for this experiment: $N_{hid} = 12 \times 12$, $N_0 = 20$, $N_\infty = 2$, $t_\infty = 2000$.**



(a) SOM (b) t-SNE (c) MLP-encoder (d) MLP-classifier

FIGURE 6. **Low-dimensional representations: Bupa data. The low-dimensional representations of the (a) SOM, (b) t-SNE, (c) MLP-encoder and (d) MLP-classifier. It can be observed from all the representations that there are many overlapping points for the two contrasting classes.**

bias toward the decoder neuron. V_l^{cls} denotes the weight vector leading from the hidden layer to the l-th class neuron in the output layer, which also includes its bias, and $H = (H_1, H_2, \cdots, H_{N_{hid}}, -1)^\top$ is the hidden layer output vector, in which N_{hid} is the number of hidden neurons.

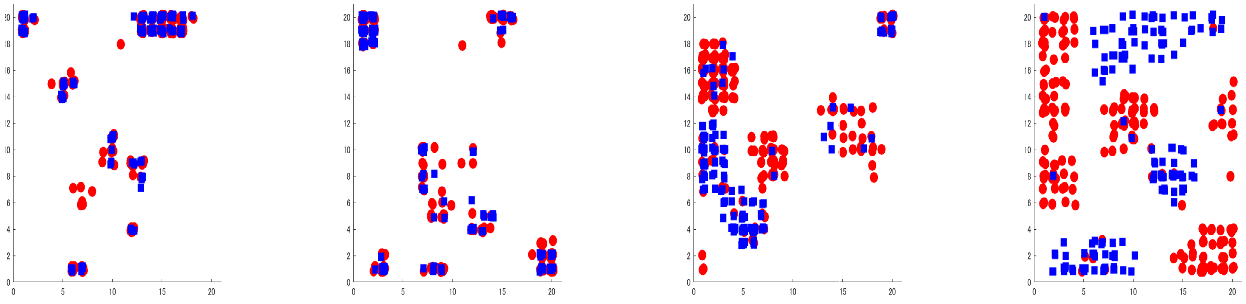
The cost function is defined in Eq. 5, in which $0 \leq \kappa \leq 1$ is the mixing coefficient. Here, $\kappa = 0$ generates an autoencoder, while $\kappa = 1$ generates a classifier.

$$L = \frac{(1-\kappa)}{2d} \sum_k (O_k^{dec} - X_k)^2 + \frac{\kappa}{2d_{cls}} \sum_l (O_l^{cls} - T_l)^2 \quad (5)$$

It has to be noted here that while many deep autoencoders have symmetric structures between their encoder part and their decoder part, the symmetry is not implemented in the STA when it is trained as an autoencoder, as the input is encoded in a topological process but the final output is generated by a sigmoidal perceptron.

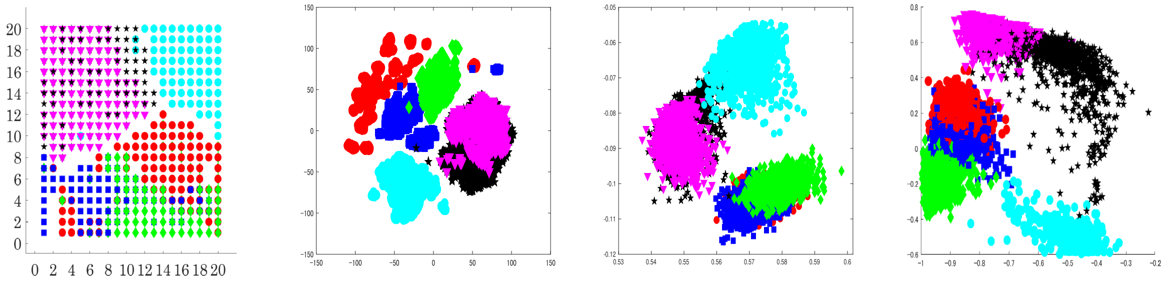
In Eq. 5, d is the dimensions of the input, d_{cls} is the number of classes and thus the number of label neurons, and T_l denotes the l-th component of the teacher signal.

Applying stochastic gradient descent, the modifications of the connection weights from the hidden layer to the output



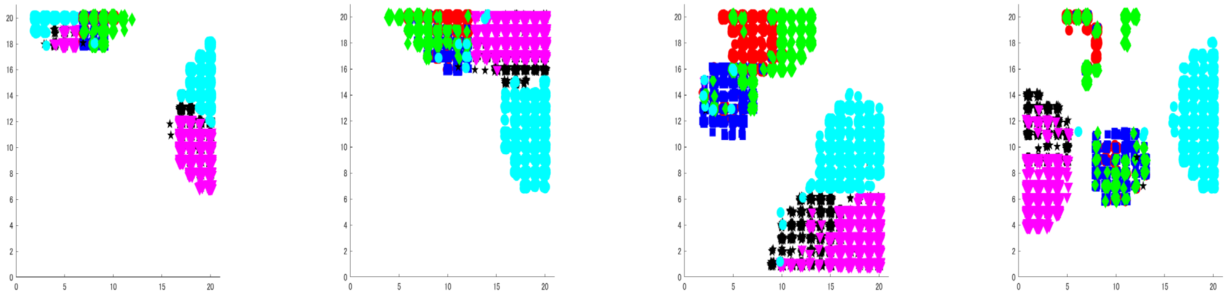
(a) Bupa ($\kappa = 0$) (b) Bupa ($\kappa = 0.1$) (c) Bupa ($\kappa = 0.5$) (d) Bupa ($\kappa = 1$)

FIGURE 7. Bupa data. In (a), it can be observed that the topological autoencoder representation of the STA for this problem agrees with the other dimensional reduction methods in that there are many overlapping samples. Figures (b)-(d) show how the two conflicting classes are disentangled with the increase in κ. The STA's setting for this experiment: $N_{hid} = 20 \times 20$, $N_0 = 20$, $N_\infty = 1$, $t_\infty = 6000$.



(a) SOM (b) t-SNE (c) MLP-encoder (d) MLP-classifier

FIGURE 8. Low-dimensional representations: Activity recognition data. The low-dimensional representations of this problem obtained by the SOM, t-SNE, MLP-encoder and MLP-classifier are shown in (a)-(d). It is interesting to note here that different dimensional reduction methods generate noticeably different two-dimensional maps. This difference is due to the multiple inherent aspects characterizing these data, which are partially captured by the different methods here, resulting in the different appearances of the maps.



(a) Activity ($\kappa = 0$) (b) Activity ($\kappa = 0.1$) (c) Activity ($\kappa = 0.5$) (d) Activity ($\kappa = 1$)

FIGURE 9. Activity recognition data. The autoencoder representations for these data generate two large clusters, as shown in (a). Here, some of the classes are included in those two different clusters. With the increase in the mixing coefficient κ, it is obvious that there are two different forces, one working to merge samples belonging to the same class and another working to separate dissimilar samples. The classifier representations in (d) indicate that there are overlapping areas that cannot be disentangled and thus likely to contribute to the classification errors. The STA's setting for this experiment: $N_{hid} = 10 \times 10$, $N_0 = 20$, $N_\infty = 4$, $t_\infty = 2000$.

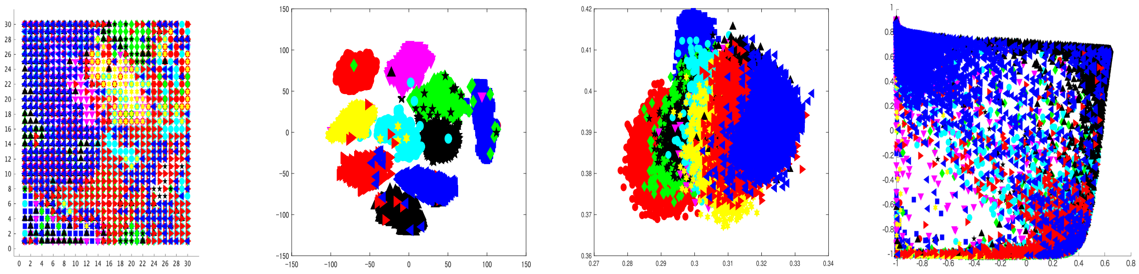
layers are calculated from the gradients as follows.

$$\begin{aligned} \Delta V_k^{dec} &= \frac{\partial L}{\partial V_k^{dec}} \\ &= \frac{(1-\kappa)}{d}(O_k^{dec} - X_k)O_k^{dec}(1 - O_k^{dec})H \\ &= (1-\kappa)\delta_k^{dec} H \end{aligned} \tag{6}$$

$$\delta_k^{dec} = \frac{1}{d}(O_k^{dec} - X_k)O_k^{dec}(1 - O_k^{dec}) \tag{7}$$

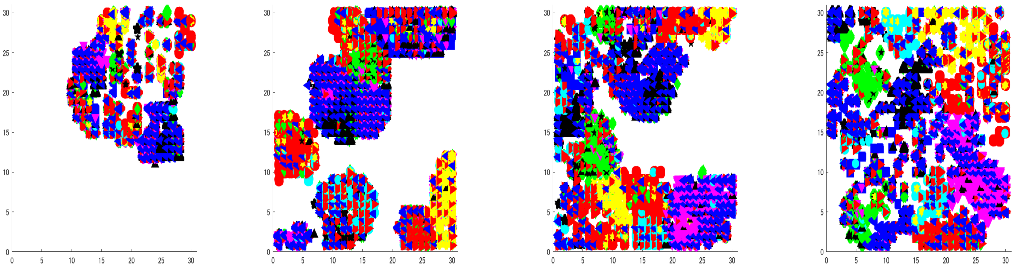
$$\begin{aligned} \Delta V_l^{cls} &= \frac{\partial L}{\partial V_l^{cls}} \\ &= \frac{\kappa}{d_{cls}}(O_l^{cls} - T_l)O_l^{cls}(1 - O_l^{cls})H \\ &= \kappa \delta_l^{cls} H \end{aligned} \tag{8}$$

$$\delta_l^{cls} = \frac{1}{d_{cls}}(O_l^{cls} - T_l)O_l^{cls}(1 - O_l^{cls}) \tag{9}$$



(a) SOM (b) t-SNE (c) MLP-encoder (d) MLP-classifier

FIGURE 10. Low-dimensional representations: MNIST Here, the low-dimensional representations for these data are shown. Although the appearances of the SOM representations (a), t-SNE representations (b), MLP-encoder (c) and MLP-classifier (d) differ, overlapping areas are observable in all the representations. This is due to the natural ways in which digits are handwritten, in that different digits may be written very similarly, while there are many different ways to express the same digit. The classifier internal representations in (d) indicate that there are some overlapping areas, which, due to the structural similarities for a single digit, will naturally contribute to the classification errors.



(a) MNIST ($\kappa = 0$) (b) MNIST ($\kappa = 0.1$) (c) MNIST ($\kappa = 0.5$) (d) MNIST ($\kappa = 1$)

FIGURE 11. MNIST. The gradual effect of the infusions of context into the handwriting can be observed in (a)-(d). Here, (a) shows the intrinsic characteristics of the handwriting of the digits. The intrinsic structure is then gradually modified (b)-(d) with the increase in the mixing coefficient. The STA's setting for this experiment: $N_{hid} = 30 \times 30$, $N_0 = 100, N_\infty = 2, t_\infty = 2000$.

In Eq. 7 and Eq. 9, δ_k^{dec} and δ_l^{cls} are the error signals backpropagated from the k-th decoder neuron and the l-th label neuron, respectively.

The modifications to the reference vectors associated with the

j-th hidden neuron can be calculated from the gradient as follows.

$$\begin{aligned}\frac{\partial L}{\partial W_j} &= \frac{\partial L}{\partial O_k^{dec}}\frac{\partial O_k^{dec}}{\partial W_j} + \frac{\partial L}{\partial O_k^{cls}}\frac{\partial O_k^{cls}}{\partial W_j} \\ &= \delta_j^{hid} H_j(X - W_j) \qquad (10)\\ \delta_j^{hid} &= \frac{1}{\sigma^2}\{(1-\kappa)\sum_k \delta_k^{dec} v_{jk}^{dec} \\ &\quad + \kappa \sum_l \delta_l^{cls} v_{jl}^{cls}\} \qquad (11)\end{aligned}$$

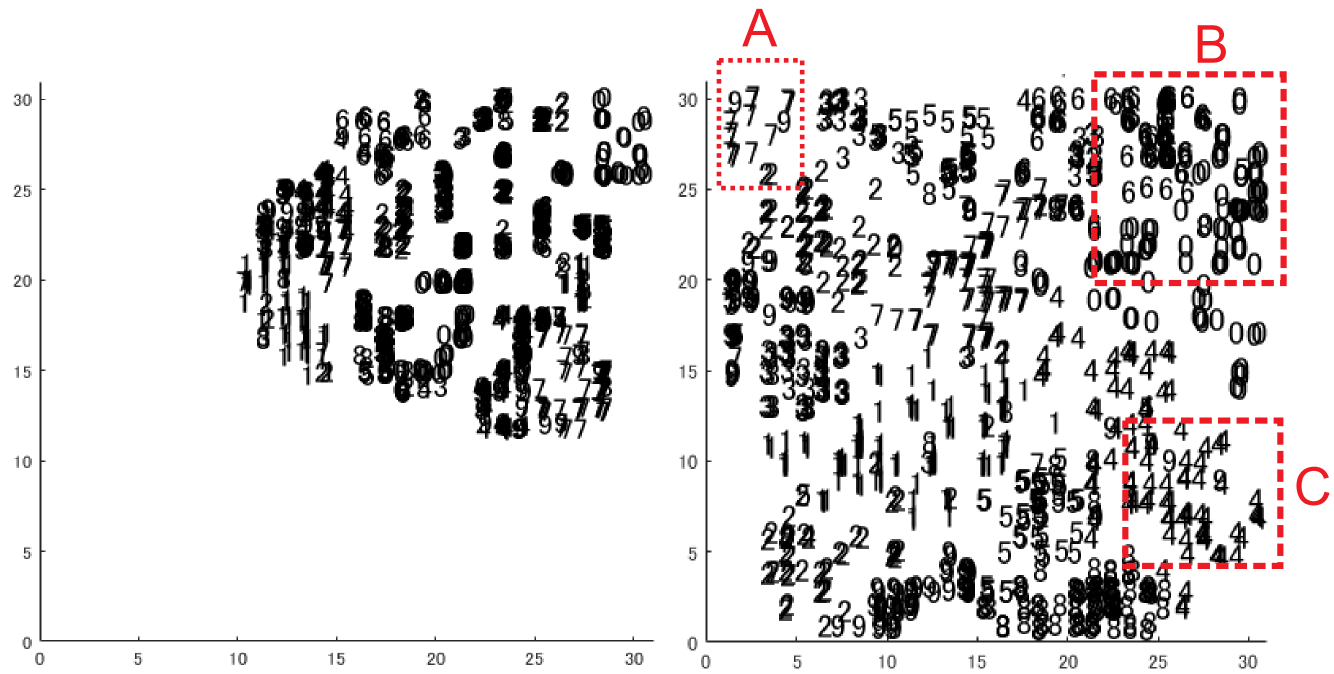
In Eq. 11, δ_j^{hid} is the error signal backpropagated to the j-th hidden layer.

The reference vector modification in Eq. 10 is similar to that of SOM in that the difference between the input and the reference vector drives the modification and, because H_j includes the neighborhood function, the proximity of the hidden neuron to the best matching unit *win* ensures the formation of the topological structure. In SOM, the modification is always directed toward the input X, while in the STA, the direction is controlled by the sign of δ_j^{hid}, where in the case of a positive δ_j^{hid}, the modification is identical to that of SOM, while a negative δ_j^{hid} repulses the reference vector away from the input vector. Because δ_j^{hid} is the error signal backpropagated from the output layer, the two-dimensional hidden layer in the STA is self-organized based not only on the topological structure of the inputs but also on their contexts, which have to be generated in the output layer. It is obvious that for the same inputs, different contexts, such as labels, or cost functions will generate different topological representations in the hidden layer. Hence, unlike SOMs, the STA generates maps that visualize the topological structure of the inputs in their given context.

The training complexity of the proposed STA is similar to that of r-RBF in [15], which is linearly scalable with that of SOM.

III. EXPERIMENTS

In the preliminary experiment, the STA is tested against the 3-dimensional artificial problem shown in Fig. 2a, where four normally distributed clusters are assigned to three classes denoted by three different colors and markers, i.e., ●, ■ and ♦. The distribution of the original 3-dimensional data and the



(a) MNIST ($\kappa = 0$)

(b) MNIST ($\kappa = 1$)

FIGURE 12. MNIST generalization. (a) The autoencoder's internal representations of MNIST. Obviously, there are many overlapping samples due to the similarity in the writing of different digits. (b) The classifier's representations of MNIST, in which the STA attempts to disentangle the overlapping samples belonging to different digits. Here, the clusters of particular digits are more distinctive. The structural similarities and the ways in which they are handwritten are well expressed on the map. For example, area A on the map reflects the similarities between digits 7 and 9, area B reflects the similarities between digits 0 and 6, and area C reflects the inherent similarities between 4 and 9. The generalization error for (a) is 88.05%, while for (b) is 9.49%.

resulting internal representations of the STA are explained in Fig. 2.

The next experiment is conducted on the same data distribution, but in this case, 4 classes are assigned for these data. The class distributions of the data and the resulting internal representations of the STA are shown in Fig. 3.

The two preliminary experiments indicate that the low-dimensional representations of the STA are influenced by the context of the data, enabling it to visualize not only the topological structure of high-dimensional data, as in SOM, but also their topological structure under various contexts.

For visualization clarity, in the two previous toy problems and the subsequent problems throughout this paper, in displaying the internal representations of the STA, different inputs sharing the same BMU are plotted separately by adding small random noise to the coordinate of the BMU. With these plotting styles, all the samples can be displayed rather than just a few BMUs, thus not only better displaying their distribution but also more clearly showing the conflicting classes among similar samples.

To better illustrate the visualization characteristics of the STA, it was trained on the well-known Iris data. Figure 4 shows various low-dimensional representations of these data generated by SOM, a t-SNE and an MLP trained as a soft autoencoder in the same way as for the STA, producing an autoencoder (denoted with MLP-encoder) for $\kappa = 0$ and a classifier (denoted with MLP-classifier) for $\kappa = 1$, in which one of the hidden layers contains two neurons that are used to generate a two-dimensional map, as used in [24]. The resulting maps for the STA for various values of the mixing coefficient κ are shown in Fig. 5

The next experiment is conducted on 6-dimensional two-classed Bupa data, the various low-dimensional representations of which are shown in Fig. 6, and their STA representations for various mixing coefficients are shown in Fig. 7.

The next experiment is conducted on 561-dimensional, six-classed activity recognition data, the low-dimensional representations of which are shown in Fig. 8, while the STA representations with regard to various values of the mixing coefficient are shown in Fig. 9.

The data sets for the last three problems were obtained from the UCI Machine Learning Repository [25].

For the final experiment, the STA is tested against MNIST, a 28 × 28 pixel handwritten digit classification problem [26]. Figure 10 shows different dimensional reduction representations for this problem. Figure 11 shows the representations of the STA under different values of the mixing coefficient, and Fig. 12 shows the representations of 10000 test data that were not used during the training process.

To quantitatively evaluate the generated hidden representation, the class similarity index (CSI) is defined as follows.

Let $R(j) = \{X|\forall X, \forall k|X - W_j| \leq |X - W_k|\}$ be a set of inputs represented by the j-th hidden neuron and $C(j, k) = \{X|\forall X \in R(j), class(X) = k\}$, in which $class(X)$ is the class

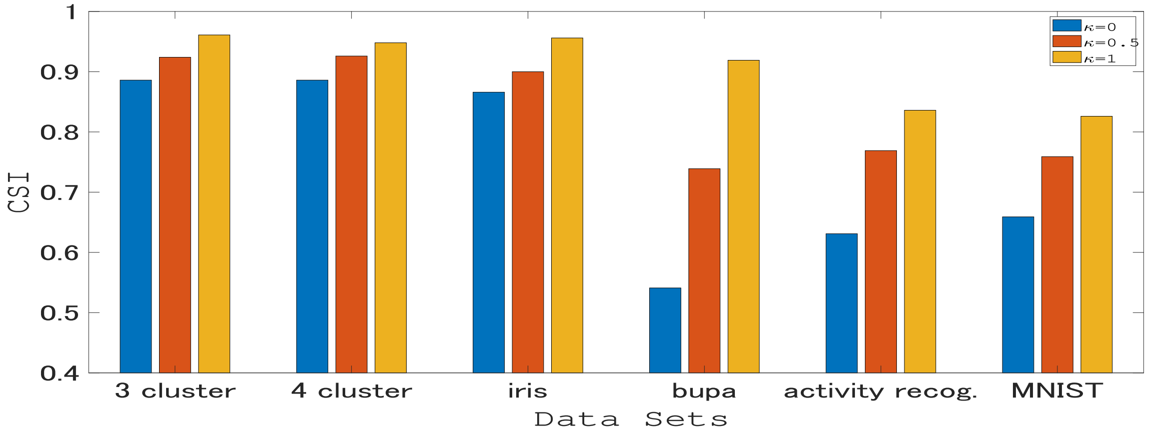


FIGURE 13. Class similarity index. It can be observed here that the CSI for the respective problem increases as the value of the mixing coefficient κ increases. For the first three problems, the increase in κ does not significantly increase their CSI values, as their data distributions essentially match their class distributions. For the final three problems, the increase in κ has a stronger influence on their CSI values. This indicates that the STA has attempted to embed the context of the data into their original context-free topological structure.

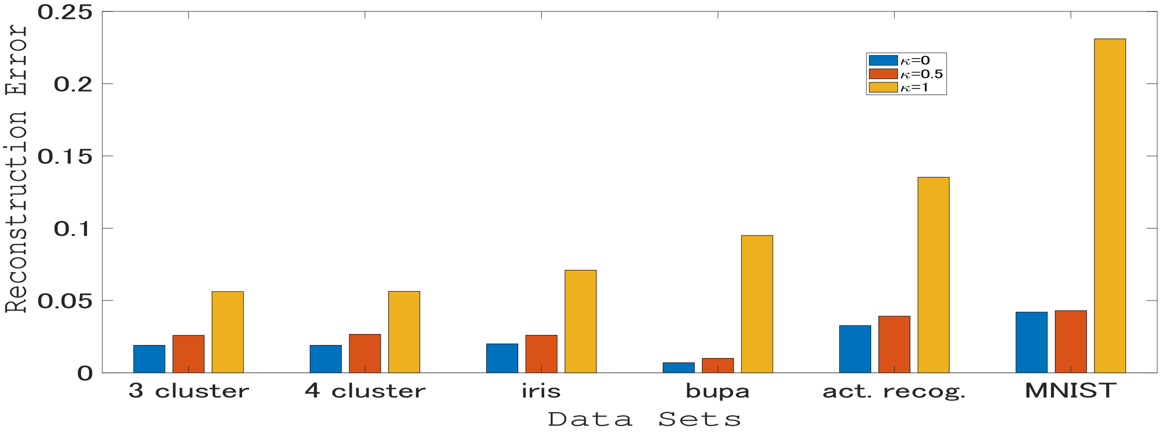


FIGURE 14. Reconstruction error. It can be observed that the value of the reconstruction error for each problem increases with increasing κ. However, it is obvious that the increase in the reconstruction error is not significant in the first three problems. As for the CSI, this lack of impact is due to the similarities between the context-free structure and context-infused structure for these problems. For the last three problems, the reconstruction errors increase more drastically as the mixing coefficient increases.

label of input X. The class ratio, which is the ratio of inputs belonging to class k, represented by the j-th hidden neuron, $CR(j,k)$, is calculated as follows.

$$CR(j,k) = \frac{|C(j,k)|}{|R(j)|} \tag{12}$$

When $\tilde{CR}(j,k)$ is regarded as the average of the class ratio of inputs belonging to class k contained in the j-th hidden neuron and its nearest neighbors with a distance of 1, the class similarity index of the STA over all inputs, CSI, is defined as follows.

$$CSI = \frac{1}{N}\sum_i \tilde{CR}(win(X^i), class(X^i)) \tag{13}$$

In Eq. 13, $win(X^i)$ is the best matching unit for the input X^i, and N is the size of the data.

Here, the CSI measures the relation between the topological proximity and the class similarity. The infusion of contexts in the label part of the output layer influences this relation, in that topological representations of inputs belonging to the same class are organized in adjacent areas due to the similar error signals received during the learning process. The CSI values for the problems in the experiments for different values of the mixing coefficient are shown in Fig. 13.

The reconstruction errors of the STA, defined in Eq. 14, under different values of the mixing coefficient are shown in Fig. 14, while the classification errors are shown in Fig. 15.

$$Err_{recon} = \frac{1}{N}\frac{1}{d}\sum_i\sum_k (O_k^{i_{dec}} - X_k^i)^2 \tag{14}$$

In Eq. 14, $O_k^{i_{dec}}$ is the value of the k-th decoder neuron for the i-th input, X_k^i is the k-th element of the

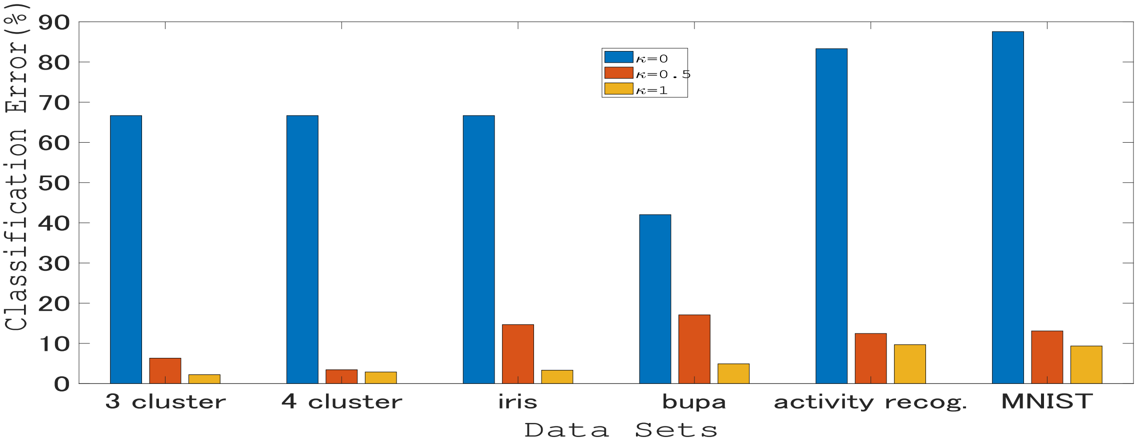


FIGURE 15. Classification error. Here, the classification error for the respective problem is shown. As is obvious from the graph, the classification error decreases as the mixing coefficient increases, as the STA transitions from an autoencoder to a classifier.

i-th input vector, N is the number of inputs and d is the dimensionality of each input. The source code for the research in this paper is published in Code Ocean: https://codeocean.com/capsule/0392390/tree/v1.

IV. CONCLUSIONS

In this study, a neural network that is able to form a two-dimensional topological map based not only on the high-dimensional structure of the data but also on their contexts is proposed. The ability to visualize high-dimensional data under different contexts adds flexibility to the process of discovering obscure characteristics of the data. As opposed to many dimensional reduction and visualization methods, which are either supervised or unsupervised, the STA can be flexibly trained under different contexts. In this paper, the STA is trained as an autoencoder, where the inherent label-free characteristics of the data are captured, as a classifier, where the topological characteristics under the contextual relation of the labels are captured, or as a mixture of the two. When the STA is trained as an autoencoder, the hidden representation encodes the natural topological structure of the data, which is required to reconstruct the high-dimensional input in the output layer. When the STA is trained as a classifier, the hidden representation encodes a contextual topological structure that is needed to predict the labels of the high-dimensional input. By controlling the mixture coefficient, an intermediate representation is formed. When the hidden representations are observed under different training contexts, some insights into how the infusion of contexts changes the internal representation can be learned. For example, the degree of difficulty in training a classifier can be intuitively understood. It is especially interesting here to observe how the topological constraint shapes the internal representations under different concepts. A quantitative index to measure the class similarity in the topological hidden representation is defined to complement the visualization for discovering the inherent characteristics of various high-dimensional data.

In this paper, the framework for context-flexible visualization and some basic experiments are presented. The primary strength of the proposed STA is in its ability to manifest the topological signature of the high-dimensional data into its internal representations under different context. The low-dimensionality of the hidden representations allows intuitive understand on the context-oriented structure of the data and at the same time understanding on how the data are related to the output of the neural network.

In future works, the STA will be utilized for multi-context data visualization analysis and also for topological data analaysis, a young field in machine learning communities. For example, in an educational setting, where the learning characteristics of students can be interpreted in different contexts, the STA can be utilized to further support their learning activities. As an analytical tool of explainable AI, a method for explaining the topological map in a human-friendly form will also be developed.

REFERENCES

[1] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA, USA: MIT Press, 2003, pp. 857–864. [Online]. Available: http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf

[2] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[3] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014. [Online]. Available: http://jmlr.org/papers/v15/vandermaaten14a.html

[4] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000. [Online]. Available: https://science.sciencemag.org/content/290/5500/2323

[5] J. B. Tenenbaum, "Mapping a manifold of perceptual observations," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds. Cambridge, MA, USA: MIT Press, 1998, pp. 682–688. [Online]. Available: http://papers.nips.cc/paper/1332-mapping-a-manifold-of-perceptual-observations.pdf

[6] J. B. Tenenbaum, V. D. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000. [Online]. Available: https://science.sciencemag.org/content/290/5500/2319

[7] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[8] T. Kohonen, "Essentials of the self-organizing map," *Neural Netw.*, vol. 37, pp. 52–65, Jan. 2013.

[9] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*. [Online]. Available: https://arxiv.org/abs/1802.03426

[10] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, 2004, pp. 513–520. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976040.2976105

[11] Y. Zhang, Z. Zhang, J. Qin, L. Zhang, B. Li, and F. Li, "Semi-supervised local multi-manifold isomap by linear embedding for feature extraction," *Pattern Recognit.*, vol. 76, pp. 662–678, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320317303977

[12] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-18, no. 5, pp. 401–409, May 1969.

[13] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964, doi: 10.1007/BF02289565.

[14] P. Hartono and Y. Take, "Pairwise elastic self-organizing maps," in *Proc. 12th Int. Workshop Self-Organizing Maps Learn. Vector Quantization, Clustering Data Vis. (WSOM)*, Jun. 2017, pp. 1–7.

[15] P. Hartono, P. Hollensen, and T. Trappenberg, "Learning-regulated context relevant topographical map," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2323–2335, Oct. 2015.

[16] P. Hartono, "Classification and dimensional reduction using restricted radial basis function networks," *Neural Comput. Appl.*, vol. 30, no. 3, pp. 905–915, Aug. 2018.

[17] L. Le, A. Patterson, and M. White, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2018, pp. 107–117.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[19] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2016, p. 2360–2368.

[20] D. Bouchacourt, R. Tomioka, and S. Nowozin, "Multi-level variational autoencoder: Learning disentangled representations from grouped observations," in *Proc. 32nd AAAI Conf. Artif. Intell., (AAAI), 30th Innov. Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, S. A. McIlraith and K. Q. Weinberger, Eds., New Orleans, LA, USA, Feb. 2018, pp. 2095–2102.

[21] C. Hofer, R. Kwitt, M. Niethammer, and A. Uhl, "Deep learning with topological signatures," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2017, pp. 1633–1643.

[22] C. Li, M. Ovsjanikov, and F. Chazal, "Persistence-based structural recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2003–2010.

[23] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4741–4748.

[24] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[25] *UCI Machine Learning Repository*. Accessed: Nov. 5, 2019. [Online]. Available: http://archive.ics.uci.edu/ml/

[26] *MNIST*. Accessed: Nov. 5, 2019. [Online]. Available: http://yann.lecun.com/exdb/mnist/

PITOYO HARTONO (Member, IEEE) received the B.Eng., M.Eng., and D.Eng. degrees from the Department of Pure and Applied Physics, Waseda University, Tokyo, Japan, in 1993, 1995, and 2002, respectively.

He was a Software Engineer with Hitachi Ltd., from 1995 to 1998. From 2001 to 2005, he was a Research Associate and a Visiting Lecturer with Waseda University. He was an Associate Professor with Future University Hakodate, Hakodate, Japan, from 2005 to 2010. Since 2010, he has been a Professor with the School of Engineering, Chukyo University, Nagoya, Japan. His research interests include the theory and application of neural networks, explainable AI (especially in medical fields), intelligent robotics, and human interface.