mathematics

MDPI

Article

Learning Neural Representations and Local Embedding for Nonlinear Dimensionality Reduction Mapping

Sheng-Shiung Wu, Sing-Jie Jong, Kai Hu and Jiann-Ming Wu *

Department of Applied Mathematics, National Dong Hwa University, Hualien 947301, Taiwan; ett2012@gmail.com (S.-S.W.); 810611001@gms.ndhu.edu.tw (S.-J.J.); khu@gms.ndhu.edu.tw (K.H.)
* Correspondence: jmwu@gms.ndhu.edu.tw; Tel.: +886-03-8903531

Abstract: This work explores neural approximation for nonlinear dimensionality reduction mapping based on internal representations of graph-organized regular data supports. Given training observations are assumed as a sample from a high-dimensional space with an embedding low-dimensional manifold. An approximating function consisting of adaptable built-in parameters is optimized subject to given training observations by the proposed learning process, and verified for transformation of novel testing observations to images in the low-dimensional output space. Optimized internal representations sketch graph-organized supports of distributed data clusters and their representative images in the output space. On the basis, the approximating function is able to operate for testing without reserving original massive training observations. The neural approximating model contains multiple modules. Each activates a non-zero output for mapping in response to an input inside its correspondent local support. Graph-organized data supports have lateral interconnections for representing neighboring relations, inferring the minimal path between centroids of any two data supports, and proposing distance constraints for mapping all centroids to images in the output space. Following the distance-preserving principle, this work proposes Levenberg-Marquardt learning for optimizing images of centroids in the output space subject to given distance constraints, and further develops local embedding constraints for mapping during execution phase. Numerical simulations show the proposed neural approximation effective and reliable for nonlinear dimensionality reduction mapping.

Keywords: unsupervised learning; distance preserving mapping; nonlinear dimensionality red uction mapping; data visualization; topology preservation; data support approximation; nonli near system solving; Levenberg-Marquardt learning; clustering analysis; principle component analysis; locally nonlinear embedding

Citation: Wu, S.-S.; Jong, S.-J.; Hu, K.; Wu, J.-M. Learning Neural Representations and Local Embedding for Nonlinear Dimensionality Reduction Mapping. *Mathematics* **2021**, *9*, 1017. https://doi.org/10.3390/math9091017

Academic Editor: Miguel Atencia

Received: 30 March 2021
Accepted: 28 April 2021
Published: 30 April 2021

1. Introduction

Nonlinear dimensionality reduction (NDR) mapping [1–4] addresses transforming high dimensional observations to an embedded lower dimensional manifold. NDR mapping has attracted many attentions for analyzing large volume of high dimensional observations, such as genomics [5], images [6,7], video [8] and audio signals. The goal is to preserve and visualize neighborhood relations of observations by displaying transformed images in the low dimensional output space. Principle component analysis (PCA) [9,10] extracts orthogonal eigenvectors, termed as principle components, which serve as internal representations of given training observations for linearly transforming high-dimensional observations to an output space. Linear projections of observations on selected principle components can be determined without reserving original training observations. Linear transformation by selected principle components can operate as an online process that transforms one observation at a time, but it has been shown infeasible for topology preserving [1–3] and can't be directly applied for NDR mapping.

Locally linear embedding (LLE) [1] has been presented for NDR mapping and data visualization. LLE is a batch process that simultaneously determines images of all given observations in a training set X. Applying the k-nearest neighboring method, it recruits k closest observations to form the neighborhood $N_k(\boldsymbol{x})$ of each observation $\boldsymbol{x}$. It assumes a locally linear relation within $N_k(\boldsymbol{x})$ such that observation x is a linear combination of observations in $N_k(\boldsymbol{x})$. For topology preserving, images of observations in $N_k(\boldsymbol{x})$ are regarded as neighbors of the image r_x of $\boldsymbol{x}$. After optimizing coefficients c_x of a correspondent linear relation that expresses each $\boldsymbol{x}$, LLE further poses a linear relation within images of k observations in $N_k(\boldsymbol{x})$. Based on the assumption that r_x is a linear combination of images of observations in $N_k(\boldsymbol{x})$ using c_x, solving linear relations that express all r_x simultaneously attains images of all observations.

In LLE, expressing r_x by a linear relation makes use of the neighborhood and coefficients of the linear relation that expresses $\boldsymbol{x}$. Inferring the image of any novel observation during execution phase hence needs neighbors defined over all training observations. LLE cannot operate with only internal representations extracted from X for image inference during testing phase. This limits portability and computational efficiency of LLE due to massive memory access to all training observations. To overcome the difficulty, this work extends LLE to locally nonlinear embedding (LNE) for NDR mapping. LNE adopts nonlinear relations for inferring images of novel observations during execution phase. LNE stands within a larger scope than LLE and can operate with only extracted internal representations for neural approximation of NDR mapping.

Similar to LLE, Isomap [2] and Laplacian Eigenmaps [11,12] maintain $N_k(\boldsymbol{x})$ for each $\boldsymbol{x}$. Isomap [2] applies the k nearest neighboring method to calculate geodesic distances and applying the traditional multi-dimensional scaling method [13–15], equivalently PCA, to infer images of all observations. Laplacian Eigenmaps sketch the k-nearest-neighbor graph based on $N_k(\boldsymbol{x})$ for all $\boldsymbol{x}$ and solve the generalized eigenvalue problem for inference of images of all observations. Both Isomap and Laplacian Eigenmaps require reserving all training observations for inferring images of novel observations during execution phase.

Self-organization maps (SOM) [16–19] as well as elastic nets (EN) [20,21] use grid-organized receptive fields as adaptable internal representations for inferring images of observations. Unsupervised learning is a process that extracts internal representations subject to training observations. Equipped with well extracted receptive fields, SOM emulates a cortex-like map, and attains a two-dimensional embedding for topology preserving mapping. It activates one and only one node in response to an observation following the winner-take-all principle. The active neural node must have a receptive field that is closest to the given observation and its geometrical location on a grid refers to the inferred image in the low-dimensional embedding. SOM infers images of novel observations during execution phase without reserving training observations. Since unsupervised learning of SOM makes use of updating operations, which directly adopt Euclidean distances among observations, it needs further improvement for NDR mapping.

The NDR mapping proposed in this work ensures properties of extracting essential internal representations and recovering the low-dimensional embedded manifold. Based on the extracted internal representations and locally nonlinear embedding, the NDR mapping infers images of novel observations during testing phase, requiring no reservation of training observations.

This work proposes graph-organized data supports to scope training observations. The union of graph-organized data supports well sketches the underlying global density support of raw observations. Internal representations of the proposed NDR mapping contain a set of receptive fields and built-in parameters of adalines (adaptive linear elements) [22,23], where receptive fields are related to represent centroids of distributed data supports. The scope of each local data support is a K-dimensional regular box, where K is less than or equals the dimension of the input space. A neural module consisting of

K pairs of adalines is employed to determine the membership of observations to a correspondent data support. An adaline neural module is an indicator to the scope of a correspondent data support. There are M neural modules, respectively determining individual scopes of M data supports as well as their neighboring relations. Neighboring relations among data supports are related to edges of a graph. Derived from training observations, the graph configuration describes neighboring relations among data supports. All neural modules are further extended for NDR mapping. The extension simply equips every neural module with a posterior weight that represents the image of the centroid of a correspondent data support. The image of every centroid and images of its neighboring centroids following the property of locally nonlinear embedding induce nonlinear constraints for optimizing all posterior weights.

Internal representations extracted from training observations include features well characterizing the membership to every data support. Based on extracted internal representations of M neural modules as well as posterior weights, the NDR mapping following locally nonlinear embedding can infer images of novel observations during testing phase without reserving original training observations. This property highly increases portability of the proposed NDR mapping. The size of adaptable built-in parameters for the proposed NDR mapping depends on the number of neural modules and the dimension of every data support. Massive training observations are no more required during execution of the proposed NDR mapping for testing.

The challenge is to optimize adaptable built-in parameters and posterior weights of joint adaline neural modules for the proposed NDR mapping. The union of graph-organized data supports sketches a bounded domain of the proposed neural approximation for NDR mapping. The NDR mapping explored in this work transforms high dimensional observations to images in the output space that recovers the manifold embedded within the input space. It is realized by adaline neural modules extended with posterior weights. The learning process mainly contains stages respectively constructing graph-organized cluster supports and optimizing posterior weights by the Levenberg-Marquardt algorithm [24–27]. The first learning stage is aimed to optimize centroids, bulit-in parameters of adaline modules and graph interconnections for representing graph-organized data supports. The second stage is to determine posterior weights by solving a nonlinear system that characterizes distance preserving mapping of centroids to images in the output space. The proposed neural approximation realizes NDR mapping without reserving training observations, depending only on adaptable feature representations or built-in parameters. Equipped with well trained built-in parameters, the proposed NDR mapping can determine the image of a novel testing observation during execution phase by resolving constraints of locally nonlinear embedding.

2. Materials and Methods

2.1. Adaline Neural Modules for Representing Distributed Data Supports

Adaline neural modules are proposed for NDR mapping from the space of high dimensional observations to the output space of images. Those neural modules are with lateral interconnections and posterior weights for performing composite linear and nonlinear transformation. Every neural module is equipped with a receptive field as well as K pairs of adalines [22,23], where K denotes the projection dimension, in reception of observations. An NDR mapping model is composed of many neural modules. Like SOM and EN, there exist lateral interconnections among neural modules. The NDR mapping is realized by graph-organized neural modules, where the graph configuration represents neighboring relations among extracted data supports with interconnections dynamically derived subject to training observations.

Let $\boldsymbol{x}$ denote an observation in the input space R^L and $\mu \in R^L$ denote a receptive field. The deference $\boldsymbol{x} - \mu$ is a result of de-mean, as μ is related to the centroid of a data cluster. The difference $\boldsymbol{x} - \mu$ propagates forward through K projective fields of adalines

as shown in Figure 1, where $a_k \in R^L$ denotes a projective field and matrix $A = [a_k^T]$ collects K receptive fields. The projection $h_k = (\boldsymbol{x} - \mu)^T a_k$ forms an external field to paired threshold elements of adalines,

$$\theta_l(h) = \begin{cases} 1, & if\ h \geq l \\ 0, & otherwise \end{cases}$$

and

$$\theta^u(h) = \begin{cases} 1, & if\ h \leq u \\ 0, & otherwise \end{cases}$$

when both threshold elements are active in response to h, it means that $h \in [l, u]$.

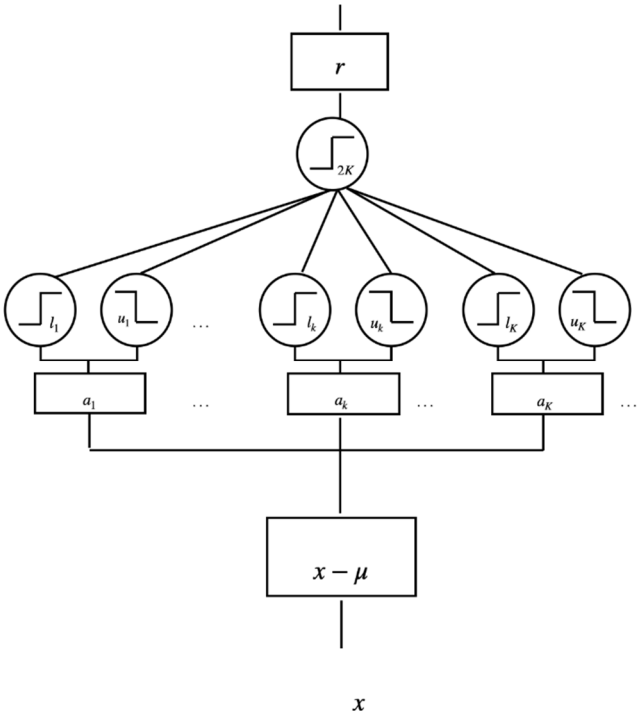

Figure 1. A feedforward neural module for translation of a high-dimensional observation to an image in the output space.

Each adaline neural module for receipting observations is equipped with K projective fields and K pairs of threshold elements in additional to a receptive field. Let $I_k = [l_k, u_k]$, where l_k and u_k respectively denote the lower and upper bounds of projection $h_k = a_k^T(\boldsymbol{x} - \mu)$. When all K pairs of threshold elements are active, the tuple $(h_1, \ldots h_K)$ is within a K-dimensional cuboid or box defined by the Cartesian product of K projection intervals, $I = I_1 \times \ldots \times I_K$, expressed by

$$I = \{(h_1, \ldots, h_K) | h_k = a_k^T(\boldsymbol{x} - \mu) \in I_k, k = 1, \ldots, K\}$$

For $K = L$, in the original observation space, a bounded region termed as a data support is expressed by

$$\Lambda = \{\boldsymbol{x} | h = A(\boldsymbol{x} - \mu) \in I, \boldsymbol{x} \in R^L\}$$

A neural module is equipped with a threshold element that has a lower bound $2K$ for indicating membership to Λ as shown in Figure 1. This threshold element activates if all of $2K$ threshold elements contribute positive responses to given $\boldsymbol{x}$. Internal representations of a neural module in Figure 1 include a receptive field μ, K projective fields and K pairs of lower and upper bounds, for reception of observations.

When $\boldsymbol{x} \in \Lambda$, equivalently $A(\boldsymbol{x} - \mu) \in I$, an adaline module summarizes the membership to the cluster support Λ and is activated if the external field reaches the lower

threshold level, $2K$. If the membership threshold element is active, the attached posterior weight r produces a non-zero image in the output space R^d, where d is less than L for dimensionality reduction, otherwise the output is a zero vector.

The membership threshold element in an adaline neural module activates to indicate the membership of the input to a correspondent data support. The proposed NDR model consists of multiple graph-organized neural modules, each possessing its own data support. The union of all cluster supports presents an approximation to the global density support underlying training observations in the original input space for modeling the embedded manifold. An NDR neural model contains not only receptive fields for input perception but also posterior weights for output generation.

The NDR neural model shown in Figure 2 is composed of M adaline neural modules. Each neural module is with internal representations, including a receptive field μ_m, K projective fields in matrix form A_m, lower and upper bounds, respectively denoted by $\{l_{mk}\}_k$ and $\{u_{mk}\}_k$, and a posterior weight r_m. Figure 3 shows training observations from the Swiss roll and edges of vertices in a graph. Each vertex m in the graph has a set of neighboring vertices, denoted by NB_m, according to the graph configuration derived by the learning process. It is notable that neighboring relations of nodes exactly concise with those among correspondent data supports and lateral interconnections of joint adaline neural modules.

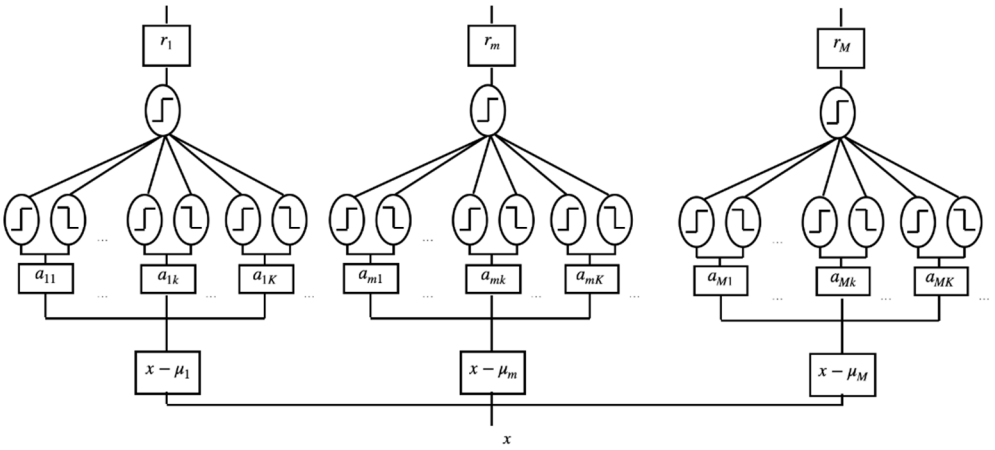

Figure 2. A deep neural network consisting of multiple neural modules for dimensionality reduction mapping.

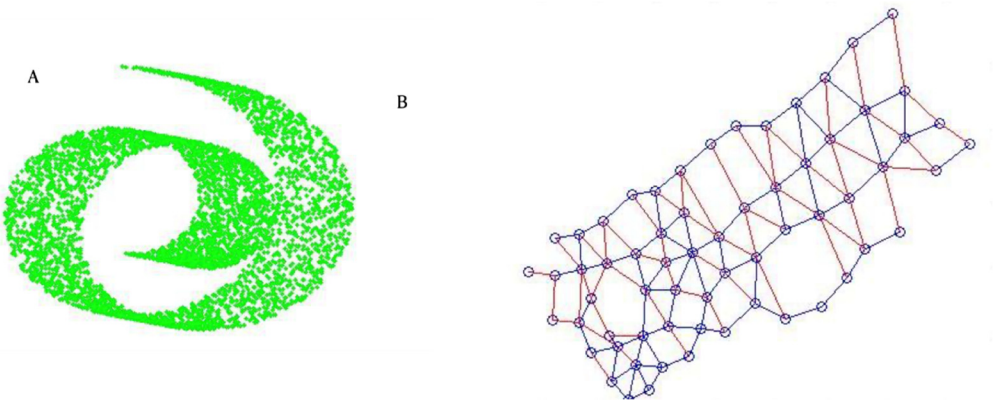

Figure 3. (**A**) Swiss-roll data. (**B**) A graph with derived edges for organizing neural modules.

2.2. NDR Model Learning

The proposed neural model for NDR mapping inherits receptive fields and graph-organized neural nodes of SOM and EN. The model design further recruits neural organization of dynamical graph configuration, projective fields, threshold elements and posterior weights. For data analysis, an NDR model has adaptable built-in parameters for representing distributed cluster supports, neighboring relations of local supports, and images of centroids. NDR model learning is with objectives of well approximating the global density support underlying training observations by the union of distributed cluster supports, structuring neighborhood relations of distributed cluster supports by the dynamic graph configuration and determining images of centroids based on distance preserving mapping.

2.2.1. Clustering Analysis for Learning Receptive Fields

Given high dimensional observations are patterns without labels for NDR mapping. Let $S = \{\boldsymbol{x}_i \in R^L\}_i$ be a training set. NDR model learning is subject to unlabeled training observations and is hence unsupervised. An NDR model transforms an observation to an image that is not explicitly provided in advance. This work relates receptive fields to centroids of clusters derived by clustering analysis subject to training observations in S.

Mathematical modeling [28–30] for clustering analysis involves formulation of constraints and the objective using mixed integer and continuous variables. The mixed integer programming leads to the annealed clustering algorithm [28,29] in Appendix A, where M receptive fields eventually partition training observations in S to M non-overlapping subsets. Let S_i collect observations that are closest to μ_i among M receptive fields,

$$S_i = \left\{\boldsymbol{x}\middle| i = \underset{m}{argmin} \parallel \boldsymbol{x} - \mu_m \parallel, \boldsymbol{x} \in S\right\}. \quad (1)$$

The annealed clustering algorithm attains not only all μ_m but also the exclusive membership of each $\boldsymbol{x}_t$ to M subsets, denoted by a unitary vector of binary elements, $\delta[t] = (\delta_1[t], \ldots, \delta_M[t])$, where $\delta_m[t] \in \{0,1\}$. There is one only one active bit among M binary bits in $\delta[t]$. It is ensured by the annealed clustering algorithm that $\delta_i[t]$ is the only active bit if and only if $\boldsymbol{x}_t$ belongs to S_i. $\delta[t]$ represents the membership of $\boldsymbol{x}_t$ to M subsets partitioned by M receptive fields.

The objective function [28,29] for optimizing continuous receptive fields and discrete memberships could have different forms under variant computation objectives. The objective function is commonly not differentiable with respect to discrete $\delta_i[t]$. Minimization of the objective function with respect to discrete and continuous variables by the annealed clustering algorithm has been extensively verified in previous works [28,29]. Appendix A gives a simplified version of the annealed clustering algorithm. Numerical simulations that show its effectiveness and reliability for clustering analysis have been extensively given in previous works [28,29].

2.2.2. Deriving Cluster Supports by Optimizing Adaline Modules

Each adaline module in Figure 2 plays a role of indicating the membership to a cluster support. Clustering analysis partitions training observations in S to M disjoint subsets. It follows $S_i \cap S_j = \emptyset$ and $S = \cup_m S_m$. Both SOM and EN make use of receptive fields to determine the exclusive membership of an observation following the winner-take-all (WTA) principle. This work further constructs cluster supports by optimizing adaline modules. A local cluster support is related to a region Λ_m that is characterized by K orthogonal projective fields derived from training observations in S_m, where $K \leq L$. The membership to Λ_m is determined by lower and upper thresholds of projections on K projective fields.

The receptive field μ_m derived by clustering analysis well represents the mean of training observations in S_m. Subtracting μ_m attains demeaned observations in S_m. Let

matrix C_m denote the covariance matrix of demeaned observations in S_m. Orthogonal projective fields can be obtained by solving the following eigenvalue problem,

$$C_m a_m = \lambda a_m$$

Orthogonal projective fields, denoted by $\{a_{mk}\}_k$, are set to eigenvectors corresponding to K largest eigenvalues. Let l_{mk} and u_{mk} respectively denote the lower bound and the upper bound of projections on a_{mk} over S_m, where

$$\begin{cases} l_{mk} & = min_{x \in S_m} (\boldsymbol{x} - \mu_m)^T a_{mk} \\ u_{mk} & = max_{x \in S_m} (\boldsymbol{x} - \mu_m)^T a_{mk} \end{cases}$$

Built-in parameters in an adaline module including K projective fields and K pairs of lower and upper thresholds have been determined for constructing cluster supports. The projection interval $I_{mk} = [l_{mk}, u_{mk}]$ denotes the minimal interval that covers projections of all demeaned observations in S_m on a_{mk}.

Let $f_m(\boldsymbol{x})$ denote the result of transferring $\boldsymbol{x}$ to vector $(h_{m1}, \ldots, h_{mK})^T$, where $h_{mk} = (\boldsymbol{x} - \mu_m)^T a_{mk}$. The local support Λ_m is defined by

$$\Lambda_m = \{\boldsymbol{x} | f_m(\boldsymbol{x}) \in I_{m1} \times \ldots \times I_{mK}\}. \quad (2)$$

If $f_m(\boldsymbol{x})$ belongs to the Cartesian product $I_{m1} \times \ldots \times I_{mK}$, $\boldsymbol{x}$ belongs to Λ_m.

If $\boldsymbol{x}$ belongs to S_m, by definition of l_{mk} and u_{mk}, the component h_{mk} of $f_m(\boldsymbol{x})$ must belong to I_{mk} for any k, and $f_m(\boldsymbol{x})$ belongs to Λ_m. If K equals L, Λ_m is simply a geometry box with orthogonal edges and is centered at μ_m, containing training observations in S_m. The union of all Λ_m is an approximation to the real global density support that covers all training observations in S. The volume of the local support corresponding to S_m can be calculated by

$$V_m = \prod_k (u_{mk} - l_{mk}) \quad (3)$$

The ratio of cluster size to the volume can be calculated by $\frac{|S_m|}{V_m}$ that represents the uniform density of Λ_m and the global uniform density is estimated by $\frac{|S|}{\sum_m V_m}$.

It is notable that the membership of the projection $f_m(\boldsymbol{x})$ to $I_{m1} \times \ldots \times I_{mK}$ is calculated by K pairs of threshold elements shown in Figure 1. The top threshold element in Figure 1 that has a lower bound, $2K$, is active if K pairs of threshold elements are active.

2.2.3. Graph Configuration and Neighboring Relations of Cluster Supports

The membership of $\boldsymbol{x}$ to Λ_m can be determined by the adaline neural module in Figure 2. It takes one subtraction and K projections on projective fields and $2K + 1$ comparisons by threshold elements to determine the membership. It is extended to determine the membership of a line segment between two points in the space of R^L to Λ_m in case of $K = L$. The membership of a line segment to a cluster support is significant for calculating the distance between centroids of Λ_m and Λ_n. The line segment between μ_m and μ_n is expressed by

$$\mu_{mn}(t) = \mu_m + t(\mu_n - \mu_m)$$

where $t \in [0, 1]$. By linearly spacing the interval $[0, 1]$ to a partition, one can obtain a lot of equally spaced points on the segment. If all sampled points on the segment belong to $\Lambda_m \cup \Lambda_n$, it is inferred that the line segment belongs to $\Lambda_m \cup \Lambda_n$ as shown in Figure 4C. In the occasion, the distance between μ_m and μ_n is defined by

$$D_{mn} = \sqrt{(\mu_m - \mu_n)^T (\mu_m - \mu_n)} \quad (4)$$

If there exists some t such that $u_{mn}(t)$ does not belong to $\Lambda_m \cup \Lambda_n$, the above definition is not feasible for determining the distance between centroids of Λ_m and Λ_n.

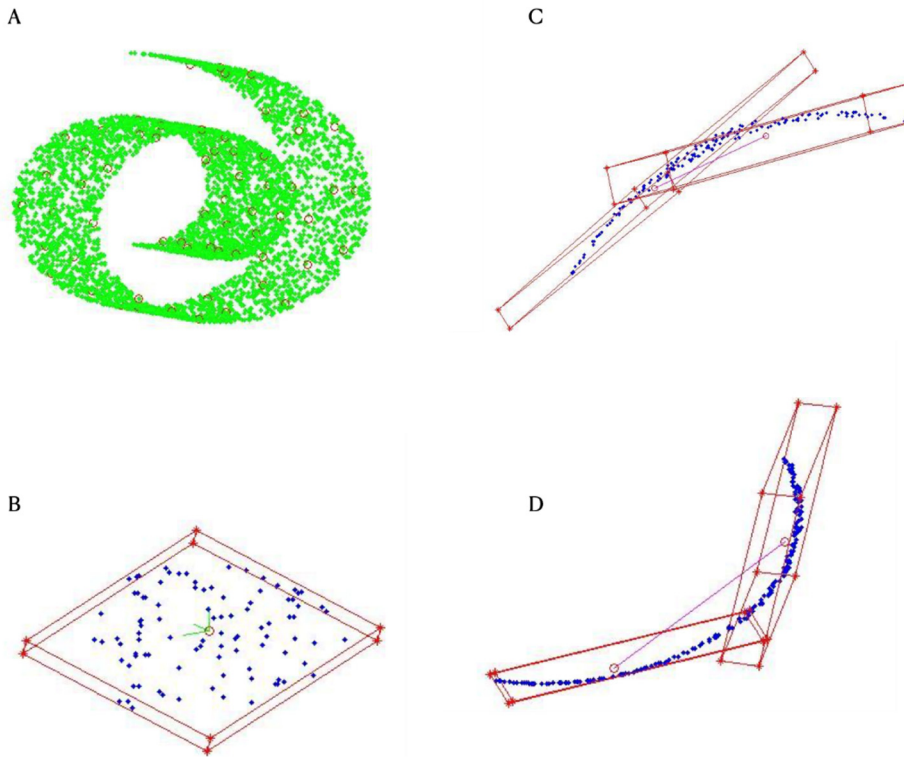

Figure 4. (**A**) Centroids of partitioned subsets. (**B**) A regular box for representing a local support. (**C**) The line segment that connects two centers totally belongs to union of two local supports. (**D**) The line segment that connects two centers partially belonged to union of two local supports.

When K equals L, the mapping $f_m(\boldsymbol{x})$ is invertible. Λ_m is a geometrical box with 2^K vertices defined by the Cartesian product $\{l_{m1}, u_{m1}\} \times \ldots \times \{l_{mK}, u_{mK}\}$. Two vertices with $K-1$ same coordinates and only one different coordinate are neighboring. Applying the inverse of f_m to two neighboring vertices induces two end points of an edge of Λ_m in R^L. Again the membership of each edge of Λ_m to Λ_n can be examined by the neural circuit in Figure 2. If there exists one vertex or one point sampled from an edge of Λ_m belonging to Λ_n, or inversely, from an edge of Λ_n belonging to Λ_m, $\Lambda_m \cap \Lambda_n$ is not empty. Two cluster supports are neighboring, if their intersection is not empty. The neighboring relations are maintained by a symmetric interconnection matrix G with element $G_{mn} \in \{0,1\}$. Both G_{mn} and G_{nm} equal one if Λ_m and Λ_n are neighboring, and are zero otherwise.

The distance between μ_m and μ_n can be also determined if some part in the whole segment does not belong to $\Lambda_m \cup \Lambda_n$ as shown in Figure 4D and G_{mn} equals one. The following definition is employed if $\mu_{mn}(t) \notin \Lambda_m \cup \Lambda_n$ for some t and $\Lambda_m \cap \Lambda_n \neq \emptyset$,

$$D_{mn} = min_{\{z \in \Lambda_m \cap \Lambda_n\}} \| \mu_m - z \| + \| \mu_n - z \| \quad (5)$$

Calculation of D_{mn} is approximated by searching for the best z among vertices and edges of two neighboring cluster supports for simplification. For $K < L$, subject to given training observations from R^L, the vertices of Λ_m can not be determined from the Cartesian product $\{l_{m1}, u_{m1}\} \times \ldots \times \{l_{mK}, u_{mK}\}$, since f_m is no more invertible. If K is less than L, the nearest distance between S_m and S_n is employed to determine the neighboring relation of Λ_m and Λ_n.

When $K < L$, $G_{mn} = 1$ if d_{mn} is less than a predetermined small positive number ϵ, and $G_{mn} = 0$, otherwise, where d_{mn} denotes the nearest distance between S_m and S_n defined by the minimal distance between any $\boldsymbol{x}_m \in S_m$ and $\boldsymbol{x}_n \in S_n$. The two observations, respectively belonging to S_m and S_n, whose distance induces d_{mn} less than ϵ could be reserved for recalculating d_{mn} during testing phase. The distance D_{mn} between

centroids μ_m and μ_n, if $G_{mn} = 1$, is determined by the sum of three distances, respectively between μ_m and x_m, μ_n and $\boldsymbol{x}_n$, and $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$, where $\boldsymbol{x}_m \in S_m$, $\boldsymbol{x}_n \in S_n$, and the distance between $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ is less than ϵ. Both $\boldsymbol{x}_m$ and $\boldsymbol{x}_n$ in accompany with G_{mn} for $K < L$, are reserved in order to calculate the distance between two observations respectively belonging to two neighboring local supports.

For the case with $K = L$, the connectivity G_{mn} is one if intersection of Λ_m and Λ_n is non-empty and is zero otherwise. Since the projection function f_m is invertible, vertices and edges of each local support can be sketched and their memberships to other supports can be checked directly. However, in case with $K < L$, the projection function f_m is not invertible. The connectivity G_{mn} is checked by comparing the set distance between S_m and S_n with a predetermined threshold. Figure 3B shows the connectivity derived subject to the Swiss roll data. The graph configuration, denoted by G, is not predetermined and is a result of checking neighboring relations between cluster supports. Neighboring relations among training observations in LLE have been extended to neighboring relations of data supports.

The distance between centroids of two cluster supports that are not neighboring is calculated based on given graph configuration and all D_{mn} with $G_{mn} = 1$. The problem now is to determine D_{mn} between μ_m and μ_n for $G_{mn} = 0$. Generally, G denotes edges among M nodes on a graph such that nodes m and n are connected if and only if $G_{mn} = 1$. The distance D_{mn} corresponding to an edge that connects nodes m and n is well defined. Two arbitrary nodes on a graph are path-connected if there exists a path between them. The distance between two path-connected nodes can be determined by the shortest path algorithm of Dijkstra [31].

If M nodes on the graph are path-connected, all distances among M nodes or all entries in matrix D are determined.

2.3. Levenberg-Marquardt Learning for Distance-Preserving Mapping and Optimal Posterior Weights

The distance matrix D provides clues for determining images of centroids of cluster supports, where images of centroids are regarded as optimal posterior weights of graph-organized neural modules for NDR mapping. The dimension of the output space is typically less than or equals three for data visualization by computer graphics. If the output dimension equals one, without losing generality, the output space is related to the interval $[0, 2\pi]$ for visualization and the posterior weight r_m refers to the radian of a unitary circle. The problem of determining all posterior weights turns to find images on a unitary circle based on the distance matrix D. On a unitary circle, each image has two neighbors. The goal is to minimize the total distance of neighboring images subject to given D. The neighboring relation defines a cycle path that visits each node exactly once and returns to where it starts. The task is different from a typical TSP (traveling salesman problem) that is provided with city cites within a Euclidean space.

The Hopfield neural network [32,33] and Potts neural networks [34,35] for solving TSP with only the distance matrix D can be directly applied for optimizing images of centroids, equivalently posterior weights. Given distance matrix D, neural approaches [34,35] attain a circular sequence that visits M centroids. Let p_i denote the index of a node at stop i on the determined circular sequence, where $1 \le i \le M$+1 and $p_{M+1} = p_1$ for circular representation. Let q_i denote the difference between images of node p_i and p_{i+1}. Let q_i be proportional to $D_{p_i p_{i+1}}$ and the sum of all q_i be 2π. Then

$$q_i = \frac{2\pi D_{p_i p_{i+1}}}{\sum_j {}^{D_{p_j p_{j+1}}}}. \quad (6)$$

By setting the image of node p_1 to zero, one can determine the image of node p_{i+1} by adding q_i to the image of node p_i for all i.

For $K = L$, the matrix D contains distances defined inside the union of cluster supports with the non-negative, symmetric and triangle properties for distance measure. The

first two properties are trivial. The third triangle property holds for distances in D. Let Q_{ik} denote the shortest path from node i to node k. According to the shortest path algorithm, path Q_{ik} contains no cycle. Let Q_{ij} and Q_{jk} respectively denote the shortest path between node i and node j, and node j and node k. Let node j be absent in path Q_{ik}. The triangle property holds if $D_{ik} \leq D_{ij} + D_{jk}$. Assume D_{ik} greater than $D_{ij} + D_{jk}$. It follows that path Q_{ik} is not the shortest one, since it could be improved by the path that concatenates Q_{ij} with Q_{jk}. This leads to a contradiction. So, the length of Q_{ik} must be less than or equal to $D_{ij} + D_{jk}$.

Now the task extends to find images of centroids or posterior weights, where the dimension of the output space is two or three and $K = L$. The purpose is to find and display the image r_m of every μ_m. The distance D_{mn} is determined by Equations (4) and (5) if $G_{mn} = 1$, and by the shortest path algorithm otherwise. For distance preserving mapping, distances in D propose the following constraint for determining r_m and r_n in the output space,

$$\| r_m - r_n \| - D_{mn} = 0 \tag{7}$$

The constraint (7) for any path-connected node m and n constitutes a nonlinear system. There are $\frac{1}{2}M(M-1)$ constraints. The LM (Levenberg-Marquardt) algorithm has been extensively applied for learning multilayer neural networks [26,27] and solving nonlinear system. This work applies the LM algorithm for solving the nonlinear system (7) and determining images of centroids for distance preserve mapping. The following criterion is formulated for least square fitting,

$$E(\boldsymbol{r}) = \sum_{m,n} (\| r_n - r_m \| - D_{nm})^2$$

where $\boldsymbol{r}$ denotes a collection of all posterior weights. The outstanding performance of the LM algorithm has been extensively explored in previous works [26,27] for learning neural networks. The LM algorithm is applied to find optimal $\boldsymbol{r}_{opt}$ defined by

$$\boldsymbol{r}_{opt} = \underset{\boldsymbol{r}}{argmin} E(\boldsymbol{r})$$

If G contains unconnected subgraphs, solving the nonlinear system (7) by the LM algorithm can be separately applied to every unconnected subgraph. There will be multiple isolated subgraphs and multiple sets of images, each corresponding to one subgraph whose nodes are path-connected.

2.4. Locally Nonlinear Embedding for NDR Mapping

Learning the proposed neural model subject to training observations in S achieves optimal built-in parameters, including receptive fields, projective fields, projection bounds and posterior weights. This section presents locally nonlinear embedding (LNE) for NDR mapping based on a neural model that has been equipped with optimal built-in parameters and posterior weights.

In response to an observation in the training set S, a well-trained NDR model is expected to generate an image in the output space. Let $\boldsymbol{x}$ belong to some subset S_m. Then $\boldsymbol{x}$ belongs to cluster support Λ_m and a correspondent neural module is active. Let $NB_m = \{n | G_{mn} = 1\}$ and l_n with $n \in NB_m$ denote the distance between $\boldsymbol{x}$ and μ_n similar to distance measure described in Equation (5). If the line segment between $\boldsymbol{x}$ and μ_n belongs to $\Lambda_m \cup \Lambda_n$, since $\boldsymbol{x} \in \Lambda_m$, l_n is calculated by the Euclidean distance between $\boldsymbol{x}$ and μ_n, otherwise by the following equation

$$l_n = min_{\{\boldsymbol{z} \in \Lambda_m \cap \Lambda_n\}} \| \boldsymbol{x} - z \| + \| \mu_n - z \| \tag{8}$$

The calculation of l_n is approximated by seeking the best $\boldsymbol{z}$ on vertices and edges of Λ_m for simplicity. Similar to Equation (7), the LNE distance-preserving constraint is given by

$$\| r - r_n \| = l_n \quad (9)$$

where $n \in NB_m \cup \{m\}$. Equation (9) poses locally nonlinear constraints for inferring image r subject to given distances. Following locally nonlinear embedding, the constraint (9) specifies a nonlinear system, where the only unknown is r and the constraint number is only $|NB_m| + 1$. By the strategy of distance preservation, optimal r can be trivially resolved by the LM algorithm with random initialization near r_m. A well trained NDR mapping can be applied to determine the image of each observation in S.

A well trained NDR model is employed to generate locally nonlinear constraints (9) for mapping a novel observation. The graph-organized neural modules simultaneously operate to encode a novel observation $\boldsymbol{x}$. If these is no active module that generates an image in the output space, it is implied that $\boldsymbol{x}$ is out of all local supports and is not within the domain of NDR mapping, otherwise there exists at least one local support that covers $\boldsymbol{x}$. Let H denote a set that collects indices of local supports commonly covering $\boldsymbol{x}$. These local supports must be overlapping and H is a small set only with several elements. Let α denote a vector with element l_n that measures the distance between $\boldsymbol{x}$ and μ_n according to Equation (8), and the image vector β consist of r_n, where $n \in NB_m \cup \{m\}$ for each $m \in H$. Locally nonlinear constraints (9) are well defined by given vectors α and β. The calculation of l_n in case of $K = L$ and $K < L$ has been given in contexts of Section 2.2.3.

Inferring the image r of $\boldsymbol{x}$ is based on distances in α and images in β. According to Equation (9), locally nonlinear constraints are in number identical to the length of vector α or β, or the sum of $|NB_m| + 1$ over $m \in H$. Since H is a small set and there is only one unknown, locally nonlinear constraints (9) constitute a nonlinear system that can be easily solved by the LM algorithm.

3. Results

Numerical simulations verify the proposed neural model and learning process for NDR mapping of the 3-dimensional Swiss-roll data [1,2]. Given Swiss-roll data contain N = 5000 3-dimensional points, $S = \{\boldsymbol{x_i} \in R^3\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$. The training data S are embedded within a 2-dimensional surface as shown in Figure 3A. The Swiss-roll data are suitable for evaluating the effectiveness of the proposed neural model for NDR mapping.

The first step applies the annealed clustering algorithm [28,29] in Appendix A to partition S to M non-overlapping subsets as described in Section 2.2.1. This step obtains receptive fields $\{\mu_m\}_m$ as centroids of cluster supports.

The second step completes cluster support construction for optimal projective fields and projection intervals as described in Section 2.2.2. The projection dimension K is set to three for current example. The centroid of a correspondent local support Λ_m is set to the mean of a cluster. Figures 4A and 5A show all centroids. Subtracting μ_m from observations in S_m leads to demeaned observations, subsequently a covariance matrix C_m. Solving the eigenvalue problem corresponding to C_m attains K orthogonal projective fields, $\{a_{mk}\}_k$, and K projection intervals, each being denoted by I_{mk}. As shown in Figure 4B, these parameters sketch a local support as a 3-dimensional box. The union of all local supports approximates the global density support underlying observations in S.

If M = 1, there is only one cluster support. Let S_{test} collect novel 20,000 data points randomly sampled from the sole regular cluster support that covers all observations in S. A well-trained neural model with M = 64 can evaluate memberships of newly sampled observations in S_{test} to the union of M cluster supports. The membership is verified if there exists at least one activated module in the neural model. Reserving those inside the union of cluster supports attain results in Figure 5B, which displays similar structure to the original Swiss-roll.

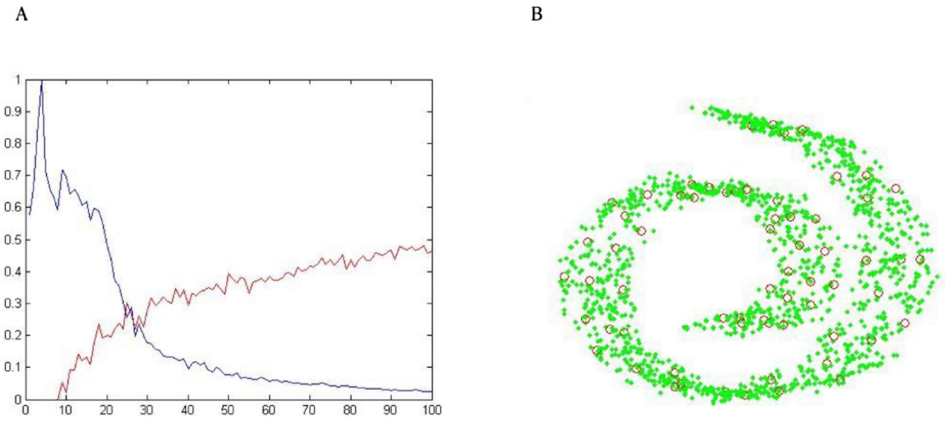

Figure 5. (**A**) Volume *V(M)* and connectivity *C(M)* empirically change as cluster number *M* increases. (**B**) Centroids.

The third step figures out lateral interconnections of neural modules, equivalently neighboring relations of cluster supports. For current situation, f_m is invertible such that vertices and edges of cluster support Λ_m can be efficiently tracked. Figure 6A shows neighborhood relations of cluster supports in the input space. On the basis, the distance between centers of two neighboring cluster supports has two different ways for calculation, depending on the straight line between two centroids μ_i and μ_j totally within two neighboring cluster supports as shown in Figure 4C or partially within $\Lambda_i \cup \Lambda_j$ as shown in Figure 4D. Based on distances of centroids of neighboring cluster supports, as described in Section 2.2.3, the shortest path algorithm of Dijkstra [12] is further applied to determine the distance between every pair of path-connected nodes m and n, where $G_{mn} = 0$. Now all entries in matrix D have been determined for inferring images of all centroids.

Based on distances of centroids in matrix D, distance preserving constraints in Equation (7) constitute a large scaled nonlinear system, where all images of centroids in $\boldsymbol{r}$ are unknown. As described in Section 2.3, the LM algorithm minimizes the least square criterion for optimizing all posterior weights. As a result, images of all μ_m on a 2D plane are shown in Figure 6B. The centroids in Figure 6A are mapped to images according to the distance preserving criterion. This illustrates significance of applying the LM algorithm for topology-preserving dimensionality reduction.

A well trained NDR neural model is employed to map each training or testing observation. As described in Section 2.4, locally nonlinear constraints (9) are given for each training or testing observation, where the only unknown denotes the generated image in the output space. Figure 6C shows the generated images of all observations. The LM algorithm is effective and reliable for seeking the image of each testing observation that satisfies locally nonlinear constraints (9). It is notable that a well-trained NDR neural model is semi-parametric. It operates without reserving full training observations and is able to map novel testing observations to an embedded low dimensional manifold faithfully.

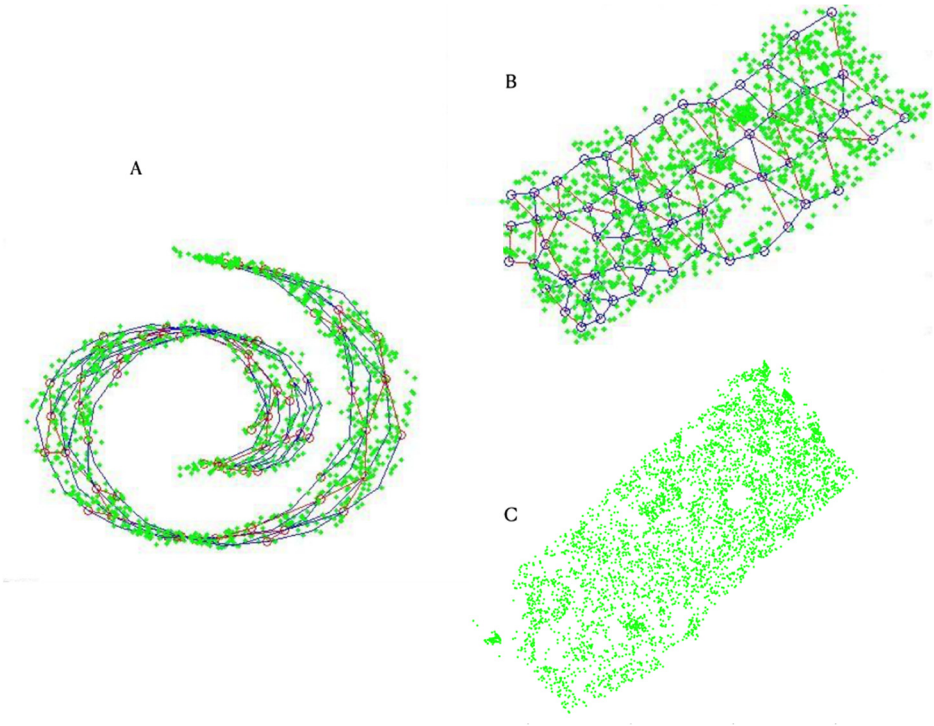

Figure 6. (**A**) Edges on a graph and neighboring relations of cluster supports in the input space. (**B**) Images of centroids and training observations. (**C**) Images of all observations in the output space.

4. Discussion

Let *V(M)* be the sum of all V_i, where V_i denotes the volume of cluster support Λ_i determined by Equation (3). *V(M)* is expected to be related to an upper bound of the volume of the the global density support that covers all data points in *S*. However, as shown in Figure 5A, *V(M)* decreases as *M* increases. Hence minimal *V(M)* cannot serve as the only criterion of setting *M*. For current example, two local supports are neighboring if $\Lambda_i \cap \Lambda_j \neq \emptyset$. The connectivity of the graph *G* tends to increase as *M* increases. Following the minimal volume and maximal connectivity principle, the optimal number of cluster supports derived by advanced clustering analysis is determined by

$$M_{opt} = \min_m (V(m) - \lambda C(m)) \quad (20)$$

where *C(M)* is the ratio of the sum of elements in *G* to the number, $\frac{1}{2}M(M\text{-}1)$, of full connections. Figure 5A shows empirical volume *V(M)* and connectivity *C(M)*, where the horizontal axis denotes the number of clusters. Here λ is set to $V(1)$ for balancing two quantities. As the number of clusters increases to $M = 64$, it tends to balance volume and connectivity. Advanced clustering analysis partitions *S* into *M* clusters, as shown in Figure 5B. For this case, nodes on the graph are connected and there is no isolated node.

The proposed NDR model is applied for neighborhood preservation mapping. Consisting of two hidden layers in additional to input and output layers, the proposed NDR model is a deep neural network [36–38]. For generalization, learning an NDR model subject to training observations is verified by a testing set that is not provided during learning phase. Let S_{test} denote the testing set. The first step is to check if given testing observations are within the union of local supports defined by the trained NDR model. Let S'_{test} collect screened observations whose images could be inferred by the trained NDR model. The reservation ratio of testing observations is expressed by $\frac{|S_{test}|}{|S'_{test}|}$.

By a well-trained NDR model and locally nonlinear embedding, each observation $\mathbf{x}$ in S'_{test} is transformed to an image $\mathbf{r}$ in the output space. Let $NB_k(\mathbf{x})$ denote k nearest neighbors in the input space and $NB_{k'}(r)$ denote k′ nearest neighbors of the image $\mathbf{r}$ in

the output space. For $z \in NB_k(\mathbf{x})$, an indicator hit(z) is set to one if the image of z belongs to $NB_{k'}(r)$, and zero otherwise. An error percentage for testing is defined by,

$$\varepsilon(S'_{test}) = 1 - \frac{1}{|S'_{test}| \times k} \sum_{x \in S'_{test}} \sum_{z \in NB_k(x)} \text{hit}(z) \quad (11)$$

Determining images of testing observations can be realized by parallel computation. Inferences of two testing observations to images are independent and can be performed simultaneously. Numerical simulations employ a notebook equipped with four 2.3 GHz Intel Core i5 workers that can simultaneously execute to speed up NDR mapping of testing observations under Matlab environment.

Numerical simulations independently generate two "brokenswiss" datasets respectively for training and testing. Each dataset consists of $N = 5000$ three dimensional observations. The testing "brokenswiss" dataset is shown in Figure 7. Learning an NDR model with $M = 91$ internal nodes subject to the training set ten times attains entries in the first row of Table 1, where mean and variance of reservation ratio and error percentage over ten executions are listed. Low variances in Table 1 reflect high reliability of the proposed learning process. For current example, $k = 12$ and $k' = 36$. The mean of reservation ratio is near 90% and the mean of error percentage is less than 4%, which shows effectiveness of the proposed learning process.

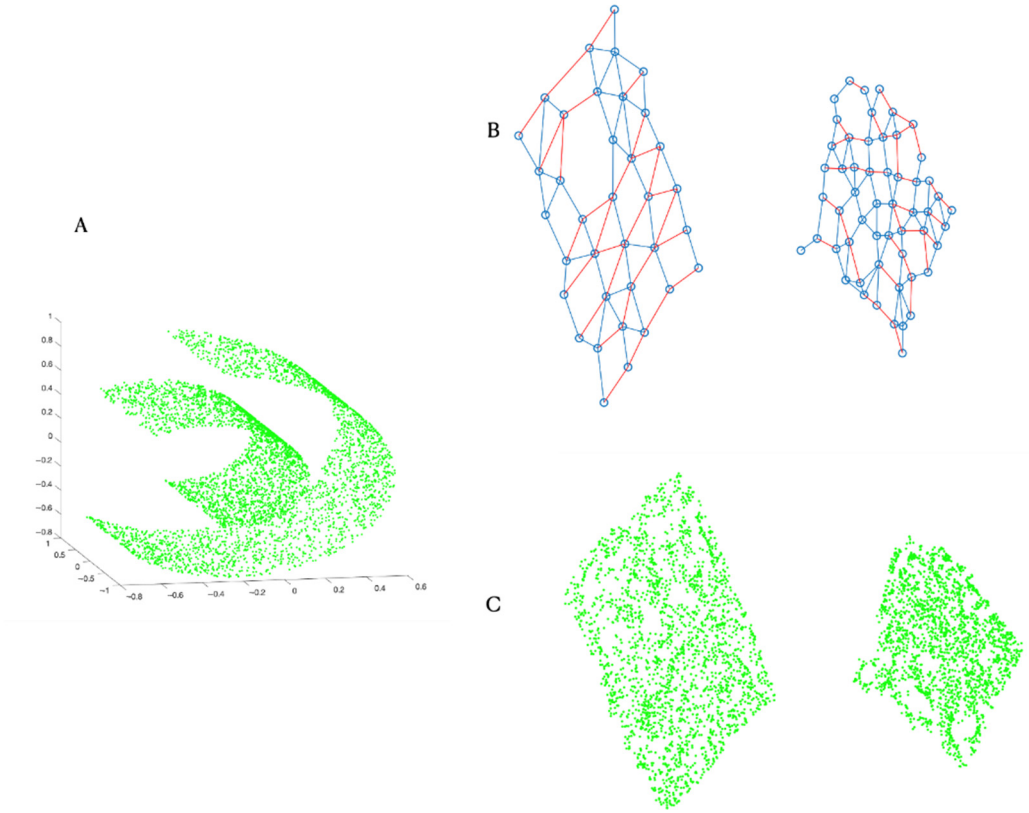

Figure 7. (**A**) A broken Swiss dataset for testing. (**B**) Isolated images of centroids and their edges. (**C**) Images of testing observations.

Table 1. Mean and variance of reservation ratio and error percentage for testing.

	Reservation Ratio (Testing)		Error Percentage (Testing)	
Dataset	**Mean**	**Var**	**Mean**	**Var**
Brokenswiss	89.78%	1.73×10^{-5}	3.92%	6.03×10^{-5}
Colorball	95.00%	5.35×10^{-6}	16.51%	2.83×10^{-5}

Both LLE and Isomap methods are not model based. Without further improvement, these two methods are unable to produce models for testing, hence inducing neither reservation ratio nor error percentage for testing in Table 1. LLE and Isomap codes [39] induce execution results that miss images of half training observations in the output space for this example. Both LLE and Isomap are not able to reduce the training error percentage for current example. Their training error percentage is about 50%.

Figure 8A shows observations sampled from the two-colored surface of a ball. The training set contains four-dimensional observations, since except for 3D coordinates an attribute is recruited for representing two different colors. The dimension L of the input space is four for this example. Figure 8B shows a graph with edges in the output space, where the positions of nodes refer to images of centroids of local supports. These edges represent neighboring relations of extracted local supports. The graph contains two isolated subgraphs. It is observed that two nodes respectively belonging to two isolated subgraphs are not connected. It is verified that local supports corresponding to nodes on each isolated subgraph contain observations of the same color and two subgraphs separately reflect two different colors of observations. The proposed NDR mapping successfully translates 4D observations to an embedded low-dimensional manifold. The encouraging results motive further applications of the proposed NDR neural model to high dimensional observations oriented from spectral features of sounds and handwritten patterns of characters.

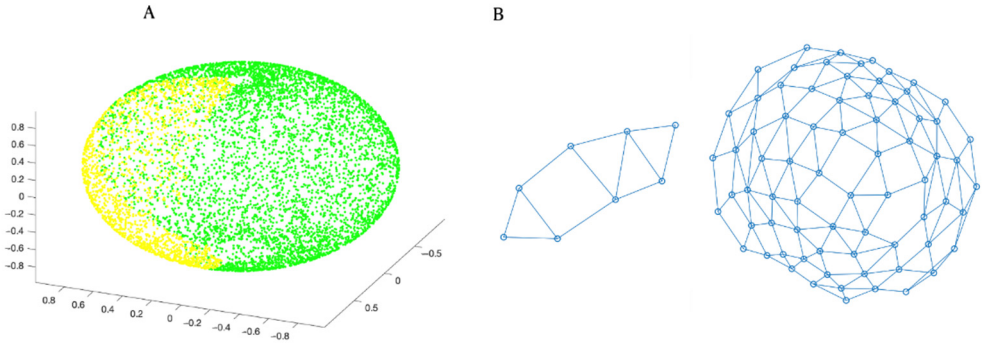

Figure 8. (**A**) Observations sampled from the two-colored ball surface for testing. (**B**) Isolated subgraphs and their edges.

The second row of Table 1 lists numerical results of evaluating the proposed learning process for the testing dataset in Figure 8. Either the training set S or the testing set S_{test} contains $N = 8000$ four dimensional observations. For this example, $k = 12$, $k' = 48$. The proposed learning process executes ten times for training an NDR model with $M = 81$ subject to S. Each time the trained NDR model is verified with S_{test}. Statistics over ten executions are summarized in Tables 1 and 2, where the reservation ratio is 95% and the error percentage is 16.51% in average. For the same reason stated previously, testing results of LLE and Isomapp are not listed in Table 1. As shown in Table 2, the proposed learning process in average takes 166 s to train an NDR model, and 190 s to map 8000 testing observations to images through parallel computation. Matlab environment can further extend parallel computation with more cores. Both LLE and Isomap are also unable to reduce training error percentage effectively for this example. It is notable that Isomap takes 1336 s to process training observations for this example. The proposed learning process significantly improves computational efficiency for training and accuracy for mapping testing observations.

Table 2. Mean and variance of execution time for training and testing.

	Execution Time			
	Training (Seconds)		Testing (Seconds)	
Dataset	**Mean**	**Var**	**Mean**	**Var**
Colorball	166.0	66.0	190.0	43.7

5. Conclusions

Numerical simulations show the proposed NDR neural model feasible for nonlinear dimensionality reduction and visualization of the embedded low dimensional manifold. A well trained NDR neural model needs no more training observations for mapping novel testing observations during execution phase. Traditional LLE and Isomap require reserving full training observations for mapping a novel testing observation. Compared with LLE and Isomap, the proposed NDR neural model possesses better portability for mapping testing observations during execution phase. It only requires built-in parameters for further execution.

All built-in parameters of a well-trained NDR model are comprehensible and meaningful for abstractive data structures. The optimized built-in parameters subject to training observations constitute extracted local supports, serving as internal representations of the centroid and projected PCA box of each local support. The obtained posterior weights refer to images of all centroids of local supports. The dynamic graph configuration states neighboring relations of local supports. Edges on the graph refer to lateral interconnections of adaline neural modules in a well-trained NDR model.

The LM algorithm has been shown effective for solving a large scaled nonlinear system toward transferring centroids to images in the output space. The nonlinear system is derived following the principle of distance preserving mapping. The LM algorithm has been also shown for seeking the image of a novel observation that satisfies locally nonlinear embedding constraints (9) during execution phase. Since there is only one unknown in a small set of locally nonlinear constraints (9), the LM algorithm is fast for NDR mapping of a single testing observation. This work has successfully extended LLE to locally nonlinear embedding constraints. The LNE constraints hold within a larger scope in comparison with traditional LLE, extensively enhancing quality of NDR mapping. Learning an NDR neural model is a process of integrating advanced clustering analysis, optimizing adaline neural modules and solving large-scale distance-preserving nonlinear constraints. The integrating learning process has been shown effective and reliable for optimizing built-in parameters of the proposed NDR neural model. An NDR neural model performs a standalone approximation that infers images of testing observations without maintaining huge volume of training observations.

Author Contributions: Conceptualization of locally nonlinear embedding, K.H. and J.-M.W.; methodology, S.-S.W. and J.-M.W.; software, S.-S.W. and J.-M.W.; validation, S.-J.J., K.H. and J.-M.W.; data curation, S.-S.W. and S.-J.J.; writing—original draft preparation, S.-S.W. and J.-M.W.; writing—review and editing, S.-S.W., K.H. and J.-M.W.; visualization, S.-S.W. and J.-M.W.; supervision, J.-M.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The data presented in this study are openly available in [39].

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

A simplified version of the annealed clustering algorithm [28,29] is summarized by the following iterative steps.

i. Randomly set all μ_j near the mean of all data points, $\{\boldsymbol{x_i}\}_i$, initialize the inverse temperature parameter β to sufficiently low value, and set ε to a small positive value.
ii. Calculate the distance d_{ij} between each point $\boldsymbol{x}_i$ and μ_j.
iii. Update the expectation of each element in exclusive memberships. $q_j[i] = \frac{exp(-\beta d_{ij})}{\sum_k exp(-\beta d_{ik})}$.
iv. Set stability to the mean of $\lambda_i = \sum_j q_j^2[i]$ over i. If stability is less than the sum of $\frac{1}{M}$ and ε, add each $q_j[i]$ with a small andom noise for perturbation.
v. Fix all $q_k[i]$ and minimize the $\frac{1}{N}\sum_i \sum_k q_k[i] \parallel \boldsymbol{x}_i - \mu_k \parallel^2$ with respect to all μ_m.
vi. If stability < 0.98, set β to β/0.995 and repeat Step ii-v, otherwise halt.

As in previous works [28,29], the algorithm operates under a physical-like annealing process. Annealing schedules a temperature-like parameter $\frac{1}{\beta}$ from sufficiently high to low values. The expectation of each element in stochastic membership variables and each adaptive centroid μ_j are maintained during the physical-like annealing process. At each intermediate temperature, the algorithm iteratively updates each $q_j[i]$ and each centroid μ_j. At the end of the annealing process, the temperature-like parameter is scheduled to sufficiently low values, where the algorithm eventually attains optimal exclusive memberships and centroids. The mean of λ_i over i denotes the stability. This quantity is calculated at step iv. If it is less than the sum of $\frac{1}{M}$ and a pre-determined small positive value ε, each $q_j[i]$ is added with a noise to escape a trivial state at step iv. If it approaches one at step vi, the algorithm halts.

References

1. Roweis, S.; Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326.
2. Tenenbaum, J.; Silva, d.V.; Langford, J. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323.
3. Hinton, G.; Salakhutdinov, R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.
4. Sorzano, C.O.S.; Vargas, J.; Pascual-Montano, A.D. A Survey of Dimensionality Reduction Techniques. *arXiv* **2014**, arXiv:1403.2877.
5. Afshar, M.; Usefi, H. High-dimensional feature selection for genomic datasets. *Knowl. Based Syst.* **2020**, *206*, 106370, doi:10.1016/j.knosys.2020.106370.
6. Rabin, N.; Kahlon, M.; Malayev, S. Classification of human hand movements based on EMG signals using nonlinear dimensionality reduction and data fusion techniques. *Expert Syst. Appl.* **2020**, *149*, 113281.
7. Taskin, G.; Crawford, M.M. An Out-of-Sample Extension to Manifold Learning via Meta-Modelling. *IEEE Trans. Image Process.* **2019**, *28*, 5227–5237, doi:10.1109/TIP.2019.2915162.
8. Li, H. 1D representation of Laplacian eigenmaps and dual k-nearest neighbours for unified video coding. *IET Image Process.* **2020**, *14*, 2156–2165, doi:10.1049/iet-ipr.2019.1119.
9. Pearson, K.F.R.S. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572, doi:10.1080/14786440109462720.
10. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Edu. Psychol.* **1933**, *24*, 417–441.
11. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems (NIPS 2001)*; Dietterich, T., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2002.
12. Donoho, D.; Grimes, C. Hessian eigenmaps: Locally linear em-bedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 5591–5596.
13. Torgerson, W.S. Multidimensional scaling: I. Theory and method. *Psychometrika* **1952**, *17*, 401–419, doi:10.1007/bf02288916.
14. Young, G.; Householder, A.S. Discussion of a set of points in terms of their mutual distances. *Psychometrika* **1938**, *3*, 19–22, doi:10.1007/bf02287916.
15. Sammon, J. A nonlinear mapping algorithm for data structure analysis. *IEEE Trans. Comput.* **1969**, *100*, 401–409.
16. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* **1982**, *43*, 59–69, doi:10.1007/bf00337288.

17. Ritter, H.; Martinetz, T.; Schulten, K. Reading. *Neural Computation and Self-Organizing Maps*; Addison-Wesley: Boston, MA, USA, 1992.
18. Kohonen, T. *Self-Organizing Maps*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1995.
19. Hu, R.; Ratner, K.; Ratner, E. ELM-SOM plus: A continuous mapping for visualization. *Neurocomputing* **2019**, *365*, 147–156.
20. Durbin, R.; Willshaw, G. An analogue approach to the traveling salesman problem using an elastic net method. *Nature* **1987**, *326*, 689–691.
21. Durbin, R.; Mitchison, G. A dimension reduction framework for cortical maps. *Nature* **1990**, *343*, 644–647.
22. Widrow, B.; Lehr, M. 30 years of adaptive neural networks: Perceptron, Madaline, and backpropagation. *Proc. IEEE* **1990**, *78*, 1415–1442, doi:10.1109/5.58323.
23. Wu, J.-M.; Lin, Z.-H.; Hsu, P.-H. Function approximation using generalized adalines. *IEEE Trans. Neural Netw.* **2006**, *17*, 541–558, doi:10.1109/tnn.2006.873284.
24. Hagan, M.; Menhaj, M. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993, doi:10.1109/72.329697.
25. Ljung, L. *System Identification—Theory for the User*; Prentice-Hall: Hoboken, NJ, USA; Englewood Cliffs: Bergen, NJ, USA, 1987.
26. NØrgaard, M.; Ravn, O.; Poulsen, N.K.; Hansen, L.K. *Neural Networks for Modelling and Control of Dynamic Systems*; Springer: Berlin/Heidelberg, Germany, 2000.
27. Wu, J.-M. Multilayer Potts Perceptrons with Levenberg–Marquardt Learning. *IEEE Trans. Neural Netw.* **2008**, *19*, 2032–2043, doi:10.1109/tnn.2008.2003271.
28. Wu, J.-M.; Hsu, P.-H. Annealed Kullback—Leibler divergence minimization for generalized TSP, spot identification and gene sorting. *Neurocomputing* **2011**, *74*, 2228–2240, doi:10.1016/j.neucom.2011.03.002.
29. Wu, J.-M.; Lin, Z.-H. Learning generative models of natural images. *Neural Netw.* **2002**, *15*, 337–347, doi:10.1016/s0893-6080(02)00018-7.
30. Tasoulis, S.; Pavlidis, N.G.; Roos, T. Nonlinear Dimensionality Reduction for Clustering. *Pattern Recognit.* **2020**, *107*, 107508, doi:10.1016/j.patcog.2020.107508.
31. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271.
32. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558.
33. Hopfield, J.J.; Tank, D.W. "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **1985**, *52*, 141–152.
34. Peterson, C.; Söderberg, B. A New Method for Mapping Optimization Problems onto Neural Networks. *Int. J. Neural Syst.* **1989**, *1*, 3–22, doi:10.1142/s0129065789000414.
35. Wu, J.-M. Potts models with two sets of interactive dynamics. *Neurocomputing* **2000**, *34*, 55–77, doi:10.1016/s0925-2312(00)00303-9.
36. Martin, B.; Jens, L.; André, S.; Thomas, Z. Robust dimensionality reduction for data visualization with deep neural networks. *Graph. Models* **2020**, *108*, 101060.
37. Ding, J.; Condon, A.; Shah, S.P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat. Commun.* **2018**, *9*, 1–13, doi:10.1038/s41467-018-04368-5.
38. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
39. Available online: https://lvdmaaten.github.io/drtoolbox/ (Matlab Toolbox for Dimensionality Reduction accessed on April, 2020).