A Graph Regularized Deep Neural Network for Unsupervised Image Representation Learning

Shijie Yang[1,2], Liang Li[2*], Shuhui Wang[2], Weigang Zhang[1,3] ,Qingming Huang[1,2*]
[1]University of Chinese Academy of Sciences, Beijing, 100049, China
[2]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[3]School of Computer Science and Technology, Harbin Institute of Technology, Weihai, 264209, China
shijie.yang@vipl.ict.ac.cn, ✉liang.li@vipl.ict.ac.cn, wangshuhui@ict.ac.cn
wgzhang@hit.edu.cn, qmhuang@ucas.ac.cn

Abstract

Deep Auto-Encoder (DAE) has shown its promising power in high-level representation learning. From the perspective of manifold learning, we propose a graph regularized deep neural network (GR-DNN) to endue traditional DAEs with the ability of retaining local geometric structure. A deep-structured regularizer is formulated upon multi-layer perceptions to capture this structure. The robust and discriminative embedding space is learned to simultaneously preserve the high-level semantics and the geometric structure within local manifold tangent space. Theoretical analysis presents the close relationship between the proposed graph regularizer and the graph Laplacian regularizer in terms of the optimization objective. We also alleviate the growth of the network complexity by introducing the anchor-based bipartite graph, which guarantees the good scalability for large scale data. The experiments on four datasets show the comparable results of the proposed GR-DNN with the state-of-the-art methods.

1. Introduction

Unsupervised representation learning via Deep Auto-Encoder (DAE) has shown its promising power in computer vision. Basically, an under-complete encoder compresses the input data into low-dimensional codes, and a similar inverted decoder reconstructs the input from the codes. Equipped with multiple layers of non-linear transformations, DAEs can simulate the perception of human brain to extract high-level semantic abstractions from low-level input signals [19, 24, 9].

To capture the high-level semantics and avoid learning the trivial suboptimal function, regularized DAEs make a tradeoff-decision between attaining some specific properties and preserving the input information $\mathbf{X}$. A regularization term Φ is added to the reconstruction cost as follows.

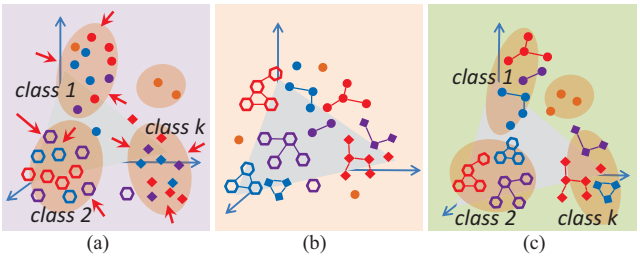


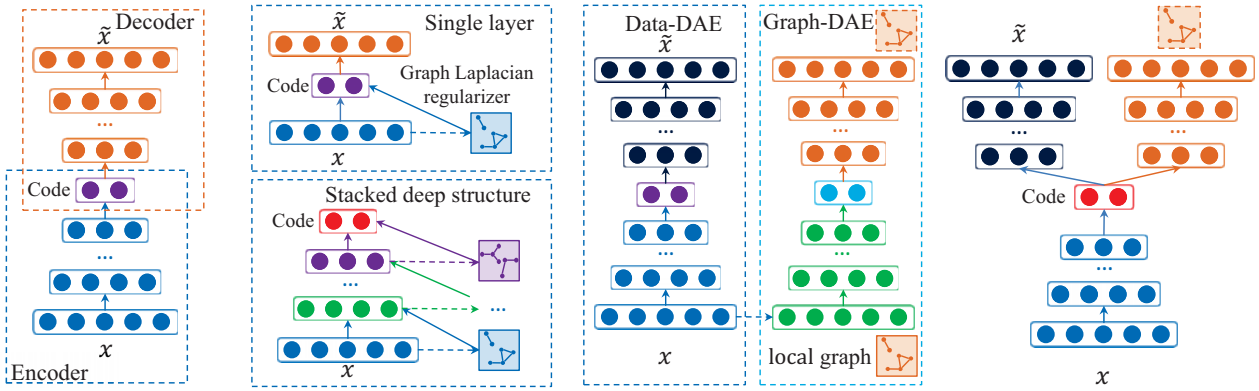
Figure 1: Toy examples of the learned embedding subspace: (a) traditional DAEs, (b) local invariance learners, (c) our GR-DNN. Different shapes denote the labels, and different colors denote the neighbor relations.

$$Cost = \Delta(\mathbf{X}, \tilde{\mathbf{X}}) + \gamma\Phi. \tag{1}$$

The regularizations include the sparsity of the code [6], the robustness to noise inputs [24] and the insensitivity of the input signals [19]. These strategies penalize the sensitivity of the code to small perturbations of the input signals, which makes crucial contribution to the success of DAEs.

Meanwhile, a stream of successful manifold learning methods benefit from the local invariance theory, such as the Locally Linear Embedding (LLE) [20], Laplacian Eigenmap [1], and Locality Preserving Projections (LPP) [18]. They emphasize that the geometric relation among neighboring points on the original manifold should be maintained in the learned embedding space, which has been widely used in unsupervised dimensional reduction [4] and semi-supervised learning [27].

Comparing the above DAEs (Fig. 1a) with the manifold learning models (Fig. 1b), the former follow a global reconstruction criterion and extract the high-level salient factors



(a) DAE (b) Laplacian auto-encoder (LAE) (c) Pre-training of GR-DNN (d) Joint fine-tuning of GR-DNN

Figure 2: Illustrations of the network structures. Dashed arrows represent the construction of local similarity graph, while solid arrows denote deterministic connections.

which can best approximate the whole data space, while the latter emphasize on preserving the local geometric structure and infer the subspace according to the affinity propagations across the local manifold tangents. In fact, the two learning strategies complement each other. DAEs are devoted to extracting the inter-class structure, while the local invariance learners focus more on modeling the intra-class correlations. If the two learners were combined as illustrated in Fig. 1c, more discriminative representation could be theoretically obtained, with both the global and local structure well preserved.

In recent literatures, the most widely used regularizer for local invariance is the graph Laplacian [4, 14, 28], which can be transformed to generalized eigenvalue problems. As a representative work, Cai et al. [4] propose the Graph regularized Nonnegative Matrix Factorization (G-NMF), which incorporates graph Laplacian regularizer into a matrix factorization objective. However, despite the popularity among shallow models, the graph Laplacian shows its limitation when integrating with deep models [10]. Due to the shallow-structure nature, it has to be treated as a basic building block for the pre-training of each layer as shown in Fig. 2b. This indicates that a completely new graph Laplacian should be constructed based on the last hidden-layer's outputs, which is not scalable for deeper hierarchical architecture. As a result, it remains an open question to design a local invariance regularizer for practical use in deep models.

This paper proposes a graph regularized deep neural network (GR-DNN) for unsupervised image representation learning, where both the high-level semantics and local geometric structure of the data manifold are simultaneously learned. In details, a deep-structured regularizer is formulated upon multi-layer perception (MLP) to leverage the DAEs with the local invariant theory to explicitly reconstruct the geometric similarity graph. As illustrated in Fig. 2d, GR-DNN is a dual-pathway network composed of one encoder and two decoders. The encoder transforms the input data into low-dimensional codes, and one data-decoder reconstructs the original input. Moreover, an additional graph-decoder is introduced as the local invariance regularizer to reconstruct a pre-constructed local similarity graph. The two decoders share a bottleneck code layer, and the loss function of GR-DNN is the weighted sum of reconstruction errors from both the decoders. Further, we alleviate the complexity-growth of the network structure by introducing the anchor-based bipartite graph, while the complexity of traditionally reconstructing the similarity graph grows dramatically w.r.t. the data volume. The experiments on four public datasets show the promising performance of our proposed model, and demonstrate that the proposed graph regularizer could be an effective module to enhance traditional DAEs. To summarize, our main contributions are as follows:

- A graph regularized deep neural network is proposed to effectively leverage DAEs with the local invariant theory for unsupervised image representation learning, where both the high-level semantics and local geometric structure of the embedding subspace are simultaneously learned.
- A deep-structured graph regularizer is introduced with solid theoretical analysis. Compared with traditional graph Laplacian regularizer, it achieves both the lower computational complexity and superior learning performance.

2. Background

Representation learning attempts to transform the original input to new feature representation which is more robust and compact to explain the data structure. Given a matrix containing n data points $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]^\top \in \mathbb{R}^{n \times m}$,

where m is the dimension of the input space, our task is to find $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_n]^\top \in \mathbb{R}^{n\times k}$ with lower dimension k, i.e., $k < m$.

2.1. Deep Auto-Encoders

DAE attempts to extract high-level semantic abstractions using architectures composed of multiple layer perceptions (MLP). Typically, deeper layers are expected to represent more abstract features, i.e., better capturing the high-level data distribution. As shown in Fig. 2a, the network contains two symmetrical sub-networks. The encoder transforms the input data $\mathbf{X}$ into low-dimensional code matrix $\mathbf{H}$, and a decoder network reconstructs the data from the code.

To learn more robust features and avoid the non-smooth suboptimal input-feature mappings, regularized DAEs attain some nice properties by introducing a regularization term Φ to the reconstruction cost shown in Eq. (1). Specifically, the denoising DAE (D-DAE) [24] is trained to discover salient features which can recover the original data from partially corrupted signals. A penalty term is added in Contractive DAE (C-DAE) [19] which minimizes the Frobenius norm of the Jacobian matrix of the code with respect to the input. The learned representation better captures the local manifold directions dictated by the data. To sum up, these techniques obtain the robustness by encouraging the insensitivity of the code to small perturbations of the input signals. In contrast, our work aims to enhance the DAEs by explicitly enforcing the internal geometric structure, which is performed based on the similarity metrics.

2.2. Local Invariance Regularizer

The local invariance theory [1, 8, 18] requires that the points on the manifold with short geodesic distances should be mapped close. Most of the existing methods adopt the graph Laplacian regularizer [4, 10, 15], which is defined based on an undirected weighted graph $G = (V, E)$, where $V = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ is the node set and $E = \{e_{ij}\}$ is the edge set. The graph structure is encoded into a similarity matrix denoted by $\mathbf{S} \in \mathbb{R}^{n\times n}$ where $[\mathbf{S}]_{ij} \geq 0$ denotes the similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$. In practice, the K-Nearest Neighbor (K-NN) graph is constructed, where the similarity is defined using Gaussian kernels with the bandwidth parameter σ:

$$[\mathbf{S}]_{ij}=\begin{cases} \exp\left(\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|}{\sigma}\right) & \text{, if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected} \\ 0 & \text{, otherwise} \end{cases} \tag{2}$$

Specifically, we connect $\mathbf{x}_i$ and $\mathbf{x}_j$ if one of them is among the K-Nearest Neighbors of the other according to a given distance measurement, i.e., usually Euclidean distance [13]. With the above defined similarity matrix $\mathbf{S}$, the local geometrical structure of the learned representation $\mathbf{H}$ can be preserved by minimizing the following term:

$$\begin{aligned} \Omega(\mathbf{H}) &= \tfrac{1}{2}\sum_{i,j=1}^{n}\|\mathbf{h}_i-\mathbf{h}_j\|^2[\mathbf{S}]_{ij} \\ &= \sum_{i=1}^{n}\mathbf{h}_i^\top\mathbf{h}_i[\mathbf{D}]_{ii} - \sum_{i,j=1}^{n}\mathbf{h}_i^\top\mathbf{h}_j[\mathbf{S}]_{ij} \\ &= Tr(\mathbf{H}^\top\mathbf{D}\mathbf{H}) - Tr(\mathbf{H}^\top\mathbf{S}\mathbf{H}) = Tr(\mathbf{H}^\top\mathbf{L}\mathbf{H}), \end{aligned} \tag{3}$$

where $Tr(.)$ denotes the trace of a matrix and $\mathbf{D}$ is a diagonal matrix whose entries are $[\mathbf{D}]_{ii} = \sum_j [\mathbf{S}]_{ij}$. Moreover, $\mathbf{L} = \mathbf{D} - \mathbf{S}$, which is called graph Laplacian. By minimizing the term $\Omega(\mathbf{H})$, we expect that if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close (i.e., $[\mathbf{S}]_{ij}$ is large), $\mathbf{h}_i$ and $\mathbf{h}_j$ are mapped close to each other in the new space.

As a representative work, Cai et al. [4] proposes the Graph regularized Nonnegative Matrix Factorization (G-NMF), which incorporates graph Laplacian regularizer into a matrix reconstruction objective as shown in Eq. (4).

$$Cost = \Delta(\mathbf{X}, \tilde{\mathbf{X}}) + \gamma\Omega(\mathbf{H}). \tag{4}$$

It has been shown that the learning performance can be significantly enhanced if the geometric structure is exploited.

As the most related work (shown in Fig. 2b), Laplacian regularized auto-encoder (LAE) [10, 15] formalizes the term $\Delta(\mathbf{X}, \tilde{\mathbf{X}})$ of Eq. (4) using DAEs. However, the shallow-structured regularizer $\Omega(\mathbf{H})$ suffers from the following limitation. When integrating with deep models, the shallow-structured regularizer has to be treated as a basic building block for pre-training each layer. As shown in Fig. 2b, a completely new graph Laplacian should be constructed based on the last hidden-layer's outputs. Note that both constructing and solving the graph Laplacian need a quadratic computational complexity w.r.t. the data volume, traditional graph Laplacian regularizer is not scalable for deeper architectures.

3. Methodology

To endue traditional DAEs with the extra ability to preserve the local geometric structure, we propose a graph regularized deep neural network (GR-DNN). In details, we formulate a deep-structured regularizer on the multi-layer perception to explicitly capture the local geometric structure.

GR-DNN is composed of one encoder and two decoders as shown in Fig. 2d. The encoder transforms a input data item into a low-dimensional code, and the first data-decoder reconstructs the original input. Inspired by the graph auto-encoders [23], we introduce a graph-decoder which explicitly enforces the internal geometric structure by reconstructing the similarity matrix $\mathbf{S}$. The two decoders are connected by a middle-bottleneck code layer. As shown in Eq. (5), the loss function is the weighted sum of reconstruction errors of the dual pathways. The learned codes simultaneously capture the high-level global abstractions, and attains the latent

geometric structure within local manifold tangents.

$$Cost = \underbrace{\Delta(\mathbf{X}, \tilde{\mathbf{X}}) + \gamma\Phi}_{traditional\,(regularized)\,DAE} + \underbrace{\eta\Delta(\,\mathbf{S}\,,\tilde{\mathbf{S}}\,)}_{graph\ regularizer} \quad (5)$$

To learn the geometrical-aware representation, good initial codes are first generated based on the global reconstructive objective. Then, the codes are further refined according to the affinity propagation within local manifold regions. From the perspective of manifold learning, the graph-decoder exerts a force to repel or attract the mapped points depending on whether they are geometrically close. From the perspective of multi-modal learning [17, 21, 25], the original image data and the local geometric graph are taken as two views or modalities to complement each other. Accordingly, compared with traditional DAEs in Fig. 2a, more compact and discriminative representation could be learned by the proposed structure.

Comparing GR-DNN with regularized denoising DAE and contractive DAE, all of them could capture the local manifold structure of the input data space. Differently, the denoising DAE and contractive DAE achieve this by enforcing the robustness and insensitiveness of the learned code w.r.t. small perturbations of the input signals. Our GR-DNN directly encodes the latent manifold structure by performing affinity propagation on the similarity graph, which captures the geometric information more accurately.

Moreover, comparing with the siamese-network [5], both of them could achieve weighted metric learning. However, the siamese-network is usually adopted in supervised learning, which requires strong supervised information on whether pairs of input samples are close or not. In contrast, our deep graph regularizer softly encodes the local geometric structure of the original data space, and usually cooperates with reconstructive criterion for unsupervised learning.

3.0.1 Relationship with graph Laplacian regularizer

We first analyze the close relationship in terms of the optimization objective. Then, we discuss the superior property of the proposed model.

Theorem 1 ([1]). *The solution of finding*

$$\arg \min_{\mathbf{H}^\top \mathbf{D}\mathbf{H}=\mathbf{I}} Tr(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \quad (6)$$

is provided by the matrix of eigenvectors corresponding to the lowest eigenvalues of the generalized eigenvalue problem $\mathbf{L}\mathbf{h} = \lambda\mathbf{D}\mathbf{h}$.

In the above theorem, the constraint $\mathbf{H}^\top\mathbf{D}\mathbf{H} = \mathbf{I}$ removes an arbitrary scaling factor in the embedding space, which can be omitted when used as regularization. Since $\mathbf{L} = \mathbf{D} - \mathbf{S}$ and based on the Eckart-Young-Mirsky theorem [7], we obtain the same corollary as in [23].

Corollary 2. $\mathbf{H} \in \mathbb{R}^{n\times k}$ *contains the* k *eigenvectors which provide the best rank-*k *reconstruction of* $\mathbf{S}$ *under the Frobenuis norm.*

Thus, we get the following conclusion for the graph Laplacian regularizer, where g_2 denotes the quadric function of $\mathbf{H}$:

$$\begin{aligned} &\min_{\mathbf{H}^\top\mathbf{D}\mathbf{H}=\mathbf{I}} Tr(\mathbf{H}^\top\mathbf{L}\mathbf{H}) \rightarrow \min_{rank(\mathbf{H})=k} \left\|\mathbf{S} - \mathbf{H}\mathbf{H}^\top\right\|_F^2 \\ &\rightarrow \min_{rank(\mathbf{H})=k, g_2} \|\mathbf{S} - g_2(\mathbf{H})\|_F^2 \end{aligned} \quad (7)$$

On the other hand, for a DAE which reconstructs $\mathbf{S}$ by minimizing the Frobenuis norm, we get Eq. (8), where f and g are multiple layers of nonlinear encoder and decoder respectively:

$$\min_{f,g} \|\mathbf{S} - g(f(\mathbf{S}))\|_F^2 \rightarrow \min_{\mathbf{H}=f(\mathbf{S}),g} \|\mathbf{S} - g(\mathbf{H})\|_F^2 \quad (8)$$

As we can see, both of the proposed graph-decoder and the graph Laplacian regularizer aim to find the best reconstruction of the input graph similarity matrix. With the more flexible non-linear decoder g rather than the quadric g_2, the graph-decoder can be regarded as a more generalized version of the graph Laplacian regularizer, and can uncover the intrinsic graph structure with theoretical guarantees.

Compared with the graph Laplacian regularizer [10], the proposed graph regularizer has the following advantages. *First, the MLP-based graph-decoder is more flexible for parameterizing complex non-linear functions, and provides a more smooth way of enforcing the local geometric structure. Second, the similarity matrix* $\mathbf{S} \in \mathbb{R}^{n\times n}$ *can be regarded as* n *regular* n*-dimensional input samples to perform the standard layer-wise pre-training. The local affinity graph only needs to be constructed once, which is more applicable in deep models with arbitrary number of layers.*

3.1. Implementation Details

3.1.1 Anchor graph (AG) for large-scale data

As conventional graph-based methods, the similarity graph is not directly applicable for large-scale data. In GR-DNN, the K-NN similarity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ is regarded as n data samples: $[\mathbf{s}_1, ..., \mathbf{s}_n]^\top$, where each sample $\mathbf{s}_i \in \mathbb{R}^n$ is an n-dimensional input data vector. This indicates that the size of the input and output layer of a DAE grows linearly w.r.t. the data volume n. Considering this, we adopt the efficient approximation method in [13] and construct the anchor-based graph (AG) instead. The key idea is to use a small set of representative anchor points $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{N_a}\}$ ($\mathbf{a}_i \in \mathbb{R}^m$) to approximate the graph structure. First, the anchor points can be sampled through many strategies, e.g., randomly selected or chosen to be the k-means cluster centroids. We adopt the latter because of its effectiveness and convenience. Then, the approximated K-NN graph

$\hat{\mathbf{S}} \in \mathbb{R}^{n \times N_a}$ is constructed between the original data points $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ and the anchor points $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{N_a}\}$ according to Eq. (2). As a result, the size of the input and output layer of a DAE can be fixed to N_a, and the construction of the graph becomes extremely efficient since now we only need to consider $O(nN_a)$ distances.

3.1.2 Training

As the most popular strategy for training deep network, the layer-wise pre-training [2] has been proven to be helpful in achieving local optimal solutions.

The training of our model is composed of three parts, i.e., *(1) The layer-wise pre-training, (2) The path-wise fine-tuning (Fig. 2c) and (3) The joint fine-tuning (Fig. 2d).* For the dual-pathway, we define a set of input pairs $\{\mathbf{x}_d^i, \hat{\mathbf{s}}_g^i\}_{i=1}^{N_a}$, where $\mathbf{x}_d^i$ is the i-th data and $\hat{\mathbf{s}}_g^i$ is the corresponding row vector of pre-constructed anchor graph $\hat{\mathbf{S}}$. First, as shown in Fig. 2c, we layer-wisely pre-train a regular data-DAE and a graph-DAE as in [2]. Then, we unroll and fine-tune them separately. In the end, as shown in Fig. 2d, a joint fine-tuning procedure is performed according to Eq. (5) to get the optimal solution. We only choose the encoder of data-DAE as the final encoder of GR-DNN. Compared with multi-modal networks [17, 21], it effectively extends our network to handle out-of-sample data, i.e., no graph needs to be constructed when encoding the test data.

3.2. Complexity Analysis

Training the GR-DNN consists of three stages, i.e., (1) generating the anchor points using k-means, (2) constructing a AG graph $\hat{\mathbf{S}}$ and (3) training the network. The first stage takes $O(t_1 n N_a m)$ time, where t_1 is the iterations for running k-means, m is the original feature dimension, and n is the total data volume and N_a is the number of anchor-points. The second stage takes $O(nN_a m)$ to construct the graph matrix.For the training stage, we suppose there are t_2 epochs. The computational cost of updating the parameters for the l-th layer in each encoder and decoder is $O(nt_2(s_p^l s_p^{l+1}))$, where $p \in \{d, g\}$ represents the data-pathway and graph-pathway respectively, and s_p^l is the size of the l-th layer. Then the overall complexity is

$$\begin{aligned} &O(t_1 n N_a m) + O(nN_a m) + O(nt_2 \sum_{p\in\{d,g\}} \sum_l s_p^l s_p^{l+1}) \\ &\approx O(n(t_1 N_a m + t_2 \sum_{p\in\{d,g\}} \sum_l s_p^l s_p^{l+1})) \end{aligned},$$

whose complexity is nearly $O(n)$, since $N_a, m \ll n$. Noticing that the pre-training and path-wise fine-tuning of the two pathways can be parallelized, the additional computational cost of GR-DNN over traditional DAEs mainly lies in the first two stages. Comparing with GR-DNN, the additional computational cost of LAE is $O((L-1)n^2)$, where L is the depth of the network.

Table 1: Summary of the datasets

Dataset	Size: train,test	Dimension	# of classes
COIL20	$1440:1000, 440$	1024	20
YaleB	$5850:5000, 850$	1200	10
MNIST	$70k:60k, 10k$	784	10
MNIST2	$120k:100k, 20k$	784	10

4. Experiments

To verify the performance of the proposed GR-DNN, we respectively conduct k-means clustering and nearest-neighbor search experiments based on the learned representation. For the clustering task, two standard evaluation metrics, i.e., Accuracy (ACC) [3] and Normalized Mutual Information(NMI), are reported. For the nearest-neighbor search task, given a query sample, we retrieve the top k nearest neighbors according to Euclidean distance. The retrieval performance is evaluated using the mean average precision (mAP), e.g., mAP@10 measures the mAP for the top 10 retrieved samples. In all the evaluations, the number of clusters is set to be the true number of classes of the dataset. For each method, all the experiments are repeated for 20 times and the best mean results are reported. Our implementation is publicly available [1] based on Theano [22].

4.1. Dataset and Parameter Setting

As in Table 1, four benchmark datasets are used. (1) The COIL20 image library [16] contains 32×32 gray images viewed from varying angles. (2) The YaleB dataset contains face images over 10 categories. (3) The MNIST handwritten digits dataset [11] contains 28×28 gray scale images. (4) MNIST2: Following [26], we add 50000 noisy MNIST digits which are rotated at angles uniformly sampled from between $[-\frac{\pi}{4}, \frac{\pi}{4}]$ of the original MNIST. We take the normalized pixel intensities (in the interval of $[0, 1]$) as the input image for all the datasets. Specifically, the validation sets are used for watching the early stopping, and we evaluate all the performances on test set. For all the auto-encoders, we adopt the more widely-used binary cross-entropy instead of the Frobenuis norm for reconstruction.

4.1.1 Comparison Methods

(1) KMEANS: K-means clustering on raw inputs. (2) NMF: Nonnegative matrix factorization [12]. (3) GNMF: Graph-regularized nonnegative matrix factorization [4]. (4) DAE: standard DAE without any regularizer. (5) D-DAE: Denoising-DAE [24]. (6) C-DAE: Contractive-DAE [19]. (7) LAE: Laplacian auto-encoders [10]. (8) GR-DNN(DAE): The proposed model built upon a standard DAE. (9) GR-DNN(D-DAE): The proposed model built upon a D-DAE (Eq. (5)). (10) GR-DNN(C-DAE): The proposed model built upon a C-DAE (Eq. (5)).

[1] https://github.com/ysjakking/GR-DNN

Table 2: Clustering and Nearest-neighbor search results(%): NMI, ACC and mAP

	YaleB			COIL20			MNIST			MNIST2		
Methods	NMI	ACC	mAP	NMI	ACC	mAP	NMI	ACC	mAP	NMI	ACC	mAP
KMEANS (Raw)	70.69	65.47	70.56	73.43	63.78	70.46	49.78	53.70	66.21	47.92	46.74	63.72
NMF	70.72	66.80	71.14	72.68	64.81	72.59	50.82	53.76	67.97	48.90	46.49	64.24
GNMF	75.43	67.81	75.25	83.51	71.33	75.17	59.27	61.09	69.52	52.85	53.70	68.92
DAE	85.53	86.34	81.21	83.05	71.53	76.54	65.65	62.36	74.91	58.75	57.36	74.39
D-DAE	90.66	87.71	82.61	83.78	72.04	77.61	68.87	64.10	75.82	59.22	59.87	75.71
C-DAE	91.32	86.94	81.70	83.70	71.58	77.49	68.69	63.78	76.12	61.12	58.81	75.82
LAE	91.71	88.21	82.29	84.13	71.71	80.18	69.76	65.22	77.82	64.46	60.63	76.47
GR-DNN(DAE)	92.21	89.97	83.32	84.95	71.88	81.12	70.61	65.76	78.98	65.75	60.85	76.89
GR-DNN(D-DAE)	**92.55**	89.77	**83.73**	**85.73**	72.02	**82.23**	**70.94**	**66.02**	79.15	**65.86**	60.76	**77.43**
GR-DNN(C-DAE)	92.51	**90.56**	83.52	84.70	**72.11**	81.18	70.32	65.90	**79.20**	65.73	**61.94**	77.31

Table 3: Summary of the network structures for GR-DNN

Dataset	Data pathway	Graph pathway
COIL20	1024–1200–500–250–20	1000–1200–500–100–20
YaleB	1200–1300–500–250–10	5000–5100–1000–100–10
MNIST	784–1000–500–250–10	1000–1200–500–100–10
MNIST2	784–1000–500–250–10	2000–2200–1000–100–10

4.1.2 Parameter Setting

Since GNMF, LAE and C-DAE share the framework of Eq. (1), the trade-off parameter of the regularization term is set by searching the grid of $\{0.01, 0.1, 1, 10, 100\}$. The K-NN and AG graphs with the bandwidth parameter $\sigma = \underset{i,j}{mean}([\mathbf{S}]_{ij}/\log 0.5)$ are built for GNMF, LAE and GR-DNN. The value of K is selected to be $\{20, 20, 30, 30\}$ respectively for COIL20, YaleB, MNIST and MNIST2 by experimental search. Specifically, regular K-NN graphs are built on COIL20 and YaleB, and we built AG graphs with 1000 and 2000 anchor points on MNIST and MNIST2 as the input of GR-DNN. GR-DNN(*) share all the hyper-parameter settings with LAE, D-DAE and C-DAE during training. In greedy layer-wise pre-training, each layer is pre-trained for 50 epochs and the corruption rate is 20% for denoising networks, For all networks, the SGD batch-size and the learning rate are set by searching the grid of $\{50, 100, 200\}$ and $\{0.001, 0.005, 0.01, 0.05, 0.1\}$, respectively.

Due to the large freedom of deep structures, we empirically design a series of 5-layer structures for all the datasets shown in Table 3, and avoid dataset-specific tuning as much as possible. As the work of [9], we put a bit more neurones on the first hidden layer than inputs, and decrease slowly until the last hidden layer. All the last hidden layer's size is set to be the true number of classes of the datasets, and all the baseline networks (i.e., D-DAE, C-DAE and LAE) and the data-pathway network of GR-DNN share the same structure at all the time. For GR-DNN, the output-size of the graph-decoder equals to the node-volume of the anchor graph, and the impact of its size will be further discussed.

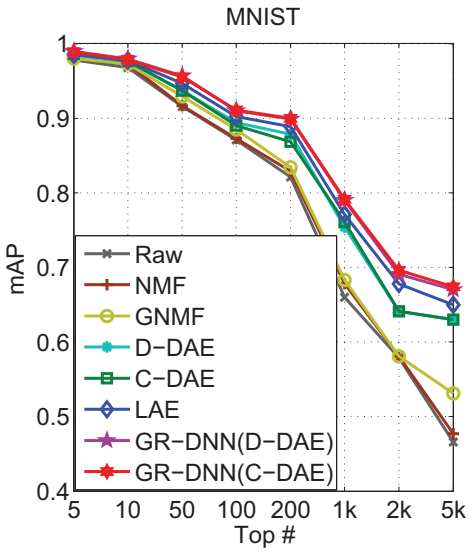


Figure 3: The top # mAP score on MNIST.

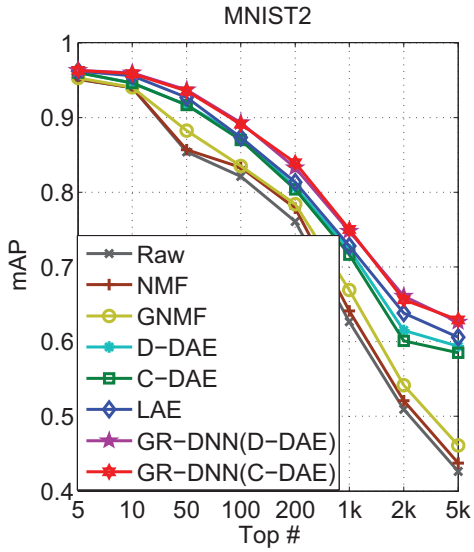
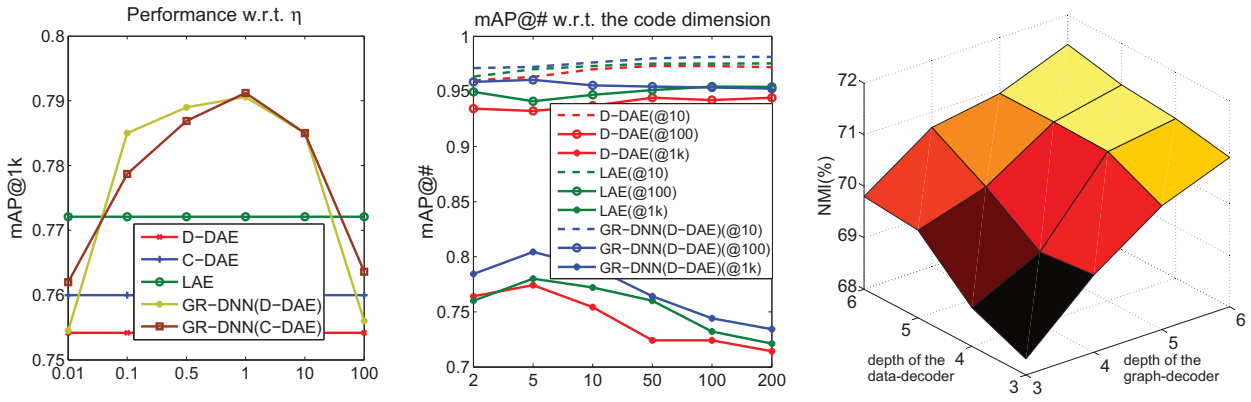


Figure 4: The top # mAP score on MNIST2



(a) (b) (c)

Figure 5: (a) Parameter analysis on MNIST. (b) The mAP of different methods on MNIST. (c) Clustering results w.r.t. number of layers.

4.2. K-NN Search Results

The K-NN search experiments aim to verify that the proposed GR-DNN can preserve the local geometric structure of the data manifolds. We show the mAP score with respect to different nearest-search scales in Fig. 3 and 4. Compared with other methods, the mAP of GR-DNN(*) decreases more slowly with the increase of retrieved samples, which indicates that cleaner neighborhoods within local regions are better obtained. We present the statistics of mAP@100,@100,@1000,@1000 on four datasets respectively on Table 2. The mAP of GR-DNN(*) has an comprehensive improvement compared with all the baselines. For example, the mAP scores of GR-DNN(C-DAE) are 0.8352, 0.7920 and 0.7731 on YaleB, MNIST and MNIST2 respectively, while C-DAE achieves 0.8170, 0.7612 and 0.7582, respectively. The relative improvements are 2.23%,4.05% and 1.97%, which shows that the proposed graph regularizer could be an effective module to enhance C-DAE. Moreover, GR-DNN(DAE) outperforms over C-DAE, D-DAE and LAE. The superior performance indicates that the deep graph regularizer provides more effective regularization influence and a more compact way to enforce the local structure than existing regularizer.

4.3. Clustering Results

The clustering experiments are conducted to demonstrate that GR-DNN not only can preserve the local geometric structure but also can better extract the global high-level semantics. Table 2 shows that the performance of GR-DNN(*) has an comprehensive improvement in terms of both NMI and ACC. Comparing GR-DNN(*) with DAE, D-DAE and C-DAE, both the average NMI and ACC are improved with a margin on YaleB, MNIST, and MNIST2. For example, GR-DNN(D-DAE) achieves relatively 3.01% NMI improvement over D-DAE on MNIST, and 2.17% NMI improvement over LAE on MNIST2. It reveals that better semantic structures are captured with the help of the deep graph regularizer, and the preserved local geometric structure gives rise to the better clustering quality and more discriminative embedding space. Comparing GR-DNN(DAE) with D-DAE and C-DAE, we see that explicitly performing affinity propagation can better capture the geometric structure than implicitly enforcing the insensitivity of the code. The superior performance of GR-DNN(DAE) over LAE shows that the proposed deep graph regularizer has more merit in capturing complex geometric structure than the shallow-structured Laplacian regularizer.

4.4. Discussions on the anchor graphs

Anchor graph (AG) is an approximation method for building regular K-Nearest Neighbor (K-NN) graphs. When building an AG, the parameter K and the number of anchor points N_a have an impact on its "locality" property. We formulate the improvement of mAP as $\Delta(mAP@\#) = mAP@\#\text{(GR-DNN)} - mAP@\#\text{(D-DAE)}$ to evaluate this impact. First, as shown in Fig. 6a, we investigate different Ks of K-NN graphs while fixing $N_a = 1k$. With the increment of K, $\Delta(mAP@10)$ and $\Delta(mAP@100)$ are gradually decreased while $\Delta(mAP@2k)$ and $\Delta(mAP@5k)$ are increased a lot. It indicates that small K emphasizes more on the local structure, and large value respects more on the global structure. However, there is a trade-off between these two impacts. Then we investigate different N_as while fixing $K = 30$ as illustrated in Fig. 6b. On the contrary, the more local structure is captured as the increase of N_a, and the enforcement of the global structure is gradually lost. Moreover, both the overly small and overly large values of N_a show relatively poor performance of GR-DNN. Overly small value of N_a overemphasizes the global structure, while overly large value results in highly sparse AG, and the affinity information is lost too much.

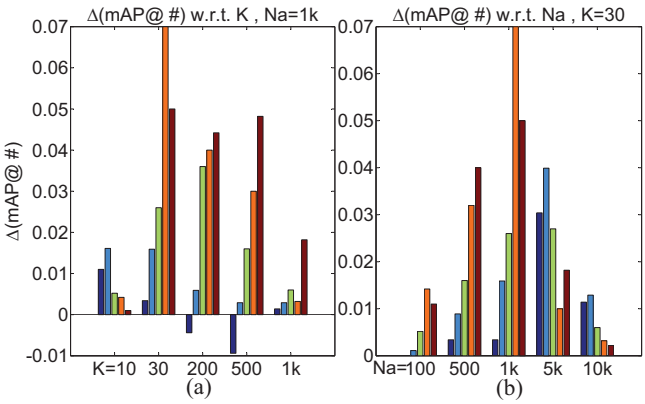


Figure 6: (a) Relative mAP improvement w.r.t. K. (b) Relative mAP improvement w.r.t. N_a.

4.5. Discussions on the graph regularizer depth

Here we give some discussion on how the depth of data and graph pathway affect the learning performance. For this purpose, we empirically tune the number of layers for each pathway with the searching grid of $\{3, 4, 5, 6\}$ and the corresponding performance is shown in Fig. 5c. We observe that more layers lead to a better performance. It is also shown that the depth of the graph-decoder has stronger impact on the performance than the data-decoder. This reconfirms the regularizing influence of the graph regularizer.

4.6. Parameter Analysis

For the parameters to be tuned for GR-DNN, we analyze the most crucial η and the code dimension due to the space limit. From Fig. 5a, we see that both overly small and overly large values of η show relatively poor performance of GR-DNN. This is consistent with the impact of η in the presented models. Overly small values of η eliminates the influence of local invariance criterion, while too large values overemphasize the local correlations and ignores the individuality of the data. As a result, η can be chosen between interval of $[0.5, 10]$ in practice. Fig. 5b shows that GR-DNN consistently outperforms other methods w.r.t. different code dimensions. Unlike the stable mAP@10 and mAP@100, the mAP@1k decreases significantly as we increase the code dimension. This indicates that large code size introduces more noise, and additional codes tend to model less discriminative visual information (e.g., reconstructing background pixels), which degrades the clustering performance.

4.7. Visualization

We qualitatively investigate the learned embedding by projecting them into a 2D space. The resulting visualizations are given in Fig. 7. In comparison, the proposed graph regularizer exerts a force to repel or attract the mapped points depending on whether they are geometrically close within local regions, and thus a discriminative embedding result can be achieved. Visualizations of the reconstructed samples of different methods are presented in Fig. 8. Our method achieves more clear and accurate reconstruction.

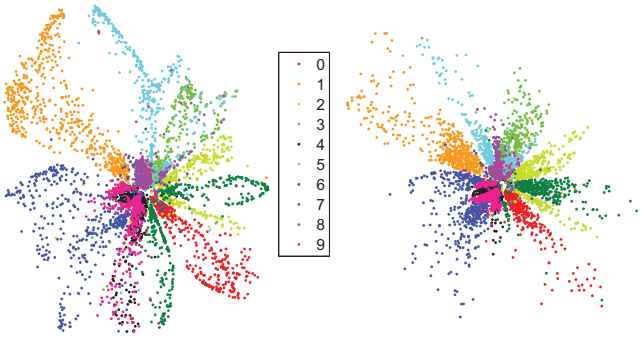


Figure 7: The left and right panel shows the 2-dimensional codes produced by D-DAE and GR-DNN(D-DAE) on the MNIST test data using a 784-1000-500-250-2 encoder respectively.

Figure 8: The top, middle and bottom panel respectively shows the original samples, the reconstructed samples by D-DAE and GR-DNN(D-DAE).

5. Conclusion

In this work, we propose a graph regularized deep neural network (GR-DNN) to endow the DAEs with the ability of retaining the local geometric structure. A robust and compact embedding space is learned to simultaneously preserve the high-level semantics and the geometric structure within local manifolds. Theoretical analysis presents the close relationship between the proposed graph regularizer and the graph Laplacian regularizer in terms of the optimization objective. The proposed model achieves linear computational complexity and empirical study shows the promising learning performance. In future work, we will focus on tailoring the model to learn compact hash codes for retrieval task and extending it to multi-view scenarios.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China: 61402431, 61332016, 61620106009, 61672497, 61650202, U1636214 and 61572488, and in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[2] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[3] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE TKDE*, 17(12):1624–1637, 2005.

[4] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[5] S. Chopra, R. Hadsell, and Y. Lecun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546 vol. 1, 2005.

[6] J. Deng, Z. Zhang, E. Marchi, and B. Schuller. Sparse autoencoder-based feature transfer learning for speech emotion recognition. In *Affective Computing and Intelligent Interaction*, pages 511–516, 2013.

[7] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[8] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742, 2006.

[9] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[10] K. Jia, L. Sun, S. Gao, Z. Song, and B. E. Shi. Laplacian auto-encoders: An explicit learning of nonlinear data manifold. *Neurocomputing*, 160:250–260, 2015.

[11] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998.

[12] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[13] Y. Li, F. Nie, H. Huang, and J. Huang. Large-scale multi-view spectral clustering via bipartite graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

[14] Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of the International Conference on Artificial Intelligence*, 2012.

[15] Y. Liao, Y. Wang, and Y. Liu. Image representation learning using graph regularized auto-encoders. *arXiv preprint arXiv:1312.0786*, 2013.

[16] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20). Technical report, technical report CUCS-005-96, 1996.

[17] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[18] X. Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153. MIT, 2004.

[19] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

[20] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[21] C. Silberer and M. Lapata. Learning grounded meaning representations with autoencoders. In *Proceedings of ACL*, pages 721–732, 2014.

[22] Theano. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[23] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1293–1299, 2014.

[24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[25] D. Wang, P. Cui, M. Ou, and W. Zhu. Deep multimodal hashing with orthogonal units. In *International Joint Conference on Artificial Intelligence,IJCAI*, 2015.

[26] W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. *In Proceedings of the international conference on Machine learning,ICML*, 2015.

[27] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012.

[28] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.