Manifold Regularized Stacked Autoencoder for Feature Learning

Sicong Lu, Huaping Liu, Chunwen Li
School of Information Science and Technology
Tsinghua University
Beijing, China
lusicong@gmail.com, hpliu@mail.tsinghua.edu.cn, lcw@mail.tsinghua.edu.cn

***Abstract*—Stacked autoencoders enjoy their popularization with the prosperity of deep learning in recent years. However, relative studies seldom exploit the intrinsic information buried in the interrelations between the samples with respect to deep networks. Regarding this, the manifold regularization is introduced to analyze the neighborhood of each training sample, which leads to a manifold regularized stacked autoencoder hierarchical framework with deep multilayer substructures. A series of experiments are conducted upon MNIST and YaleFaces using locally linear embedding as the manifold regularization module. The results show that neighborhood analysis should be combined with stacked autoencoders to achieve some notable promotions of their performances.**

Keywords-manifold regularization; stacked autoencoder; feature learning; locally linear embedding

I. INTRODUCTION

As is revealed by some neuroscience findings around the beginning of this century, human brain tends to process the massive information received everyday by a hierarchical architecture within the neocortex. Hence, deep machine learning comes from the classical neural network (NN) theory by mimicking the core principles of the working mechanism of human nervous system and enjoys its prosperity shortly afterwards.

Deep learning theory attempts to tackle the fundamental problem of "the curse of dimensionality" proposed by Richard Bellman in [1] in the 1950s, which is one of the cardinal obstacles within many fields relating to the artificial intelligence domain. Among various deep methods proposed, autoencoders, the concept of which is based on the notion of sparse coding proposed by Bruno Olshausen in [2], form a typical family of deep networks. However, although gifted researchers as Geoffrey Hinton [3], Cheng-Yuan Liou [4], Yoshua Bengio [5] et al. address themselves to this field and obtain many valuable results, more problems are exposed and modifications are performed by Xinwei Jiang [6], Wen Wang [7], Stanislas Lauly [8], Xi Zhang [9], Mathieu Germain [10] et al.

On the other hand, the well-established structures of many deep learning models are designed for image processing, which indeed exploit the interrelations among the data within a single sample but fail to discover more information, such as convolutional neural networks. Furthermore, some methods are deliberately modified to fulfill the analysis of the time-series data, but they basically consider the similarity among the input samples in a rigid way.

None of the present adaptations and derivatives of autoencoder manages to take into account the interrelationships among the training samples, although they are conspicuous and are elaborated many times. Hence, this motivates us to introduce the manifold regularization to be combined with stacked autoencoders (SAEs) to help describe the intrinsic information carried by the sample set. In fact, manifold learning is fairly well known and can be illustrated by Fig. 1. It assumes that the training samples are mainly distributed over some low-dimensional manifolds in the seemingly high-dimensional feature space and tries to seek out a local circumstance preserving method to characterize and learn the information deeply buried in the interrelations among the samples.

The main contributions of our research work can be summarized as:

- A manifold regularized stacked autoencoder (MRSAE) framework is proposed for feature learning.

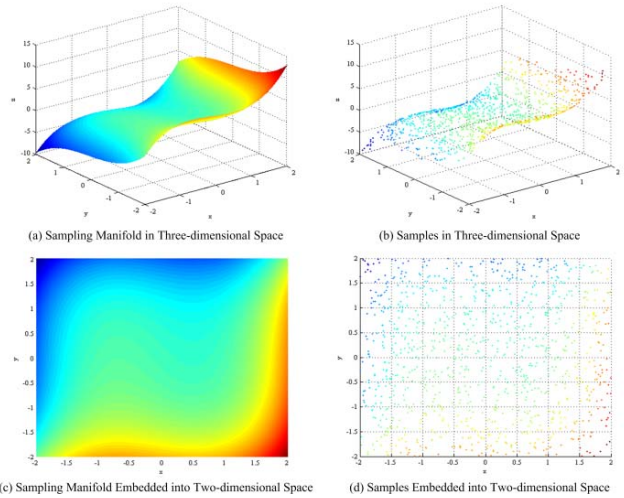


Figure 1. Manifold Learning

978-1-4799-8697-2/15 $31.00 © 2015 IEEE
DOI 10.1109/SMC.2015.513

- A complete set of structure of the framework and the corresponding learning algorithms are described for technical implementations.
- Some experiments to demonstrate and support this concept are performed and discussed afterwards.

The remainder of this paper is organized as follows. Section II mainly provides an outline of MRSAE while Section III and Section IV focus on the modules of the framework respectively. Then Section V presents the experiments with their results and some further discussions and Section VI states the conclusions drawn from this research.

II. Framework of Manifold Regularized Stacked Autoencoder

The conceptual structure of MRSAE framework is shown in Fig. 2. The architecture is complicated and hierarchical, each "layer" of which still has its multilayer substructures respectively. The main network of each single layer (marked with a box and "A Single 'Layer'" in Fig. 2) is derived from SAE with a highlighted regularization module that will be elaborated in the next section.

As is shown in Fig. 2, the raw training features of each "layer" are firstly put through the manifold regularization module to generate the results of their neighborhood analysis. Thereafter, the transformed samples are treated with a deep NN that can be trained with a greedy layer-wise algorithm (resembling SAE but obviously not the same). Then a single "layer" of the entire system is passed through and the first-layer results are obtained. Such operations can be iterated to form a deep and generally linear training process with local feedbacks.

III. Manifold Regularization

As is stated in the introduction section, SAEs do not explicitly investigate and make use of the information buried in the neighborhoods of the training samples. On the other hand, however, the interrelationships among the samples (especially when they are from the same class) are conspicuous. Thus the manifold regularization module is introduced to help describe and extract the intrinsic characteristics carried by the sample set.

Here we choose locally linear embedding (LLE) to perform as the neighborhood analyzer, which is proposed and further modified by Sam Roweis [11], Dick de Ridder [12] [13], Pietro Perona [14] et al. It assumes that although the sample vectors appear to be in a fairly high-dimensional feature space, they are actually distributed on or at least close to a virtually low-dimensional manifold, which can be identified through the data and hence feature learning can be achieved. To technically apply this idea, LLE is designed to exploit the latent information buried in the neighborhoods of each sample, which is exactly what is desired according to the regularization concept proposed previously.

Yet it must be pointed out that the manifold regularization module needed here is not simply limited to LLE, which can be substituted by the other neighborhood preserving methods as well (e.g. Isomap), for the core notion of this paper is that SAEs should be combined with some local circumstance analysis or regularization technique to exploit the interrelations among the training samples so as to eventually achieve better performances.

A. Locally Linear Embedding

Now let us shed some light upon the method of LLE. For each training sample $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N$, it firstly seeks out its neighbors $\boldsymbol{x}_{ni(i,1)}, \boldsymbol{x}_{ni(i,2)}, \ldots, \boldsymbol{x}_{ni(i,m)}$, where $ni(i, j)$ denotes the function to get the index of the j^{th} neighbor of $\boldsymbol{x}_i$, $j = 1, 2, \ldots, m$, and m is a parameter given in advance to denote the number of the neighbors taken into consideration for a single sample; then it performs a locally linear reconstruction

$$\boldsymbol{x}_i \approx \sum_{j=1}^{m} w_{i,ni(i,j)} \boldsymbol{x}_{ni(i,j)},$$

where $w_{i,ni(i,j)}$ is the reconstructive weight of $\boldsymbol{x}_{ni(i,j)}$ upon $\boldsymbol{x}_i$ and they should obey the constraint

$$\sum_{j=1}^{m} w_{i,ni(i,j)} = 1, i = 1, 2, \ldots, N;$$

furthermore, if we let $\boldsymbol{W} = [w_{i,ni(i,j)}]_{N \times N}$, the final optimization problem can be expressed as

$$\min_{\boldsymbol{W}} \left\| \boldsymbol{X} - \boldsymbol{X}\boldsymbol{W}^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \quad \text{s.t.}\ \boldsymbol{W}\boldsymbol{1} = \boldsymbol{1}, \tag{1}$$

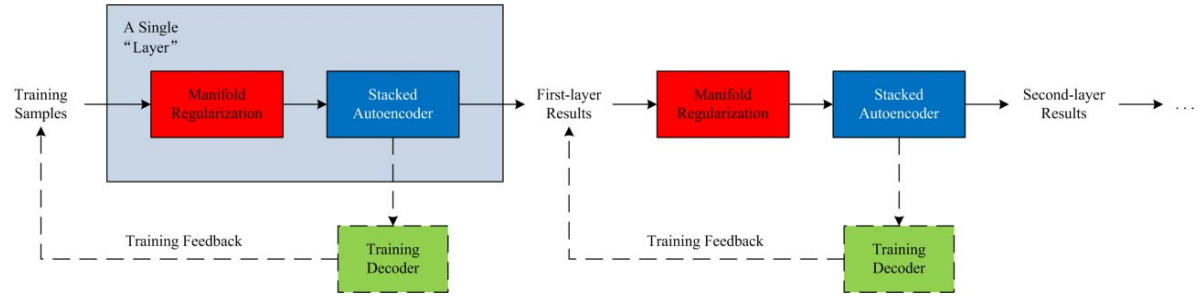


Figure 2. Framework of Manifold Regularized Stacked Autoencoder

where $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)_{D \times N}$ is the sample matrix generated by piling all the training feature vectors up. Note that (1) has omitted an important set of constraints, the very ones that describe the "neighboring relationships", i.e. $w_{i,j} = 0$ if $\boldsymbol{x}_j$ is not one of the neighbors of $\boldsymbol{x}_i$. However, it should be noticed that $\boldsymbol{W}$ itself is a sparse matrix and (1) will lead to a series of independent least-squares problems, which can be solved analytically and row-wise with respect to $\boldsymbol{W}$, and in this sense the creation and operation mechanisms of sparse matrices in MATLAB can ensure the satisfaction of the neighboring constraints by default.

We should like to emphasize that the matrix $\boldsymbol{W}$ calculated above is independent of the linear transformations (e.g. rescaling, rotation and their combinations) applied to the sample set, although LLE itself is a nonlinear method. Thus it is reasonable to deem that $\boldsymbol{W}$ contains some intrinsic information buried deep in the training set. Hence, it is natural to maintain the neighborhood interrelationships by letting $\boldsymbol{W}$ remain the same after the embedding work.

Based on this idea, the mathematical model built to describe the second and final step of LLE can be summarized as

$$\min_{\boldsymbol{Y}} \left\| \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{W}^{\mathrm{T}} \right\|_{\mathrm{F}}^2. \tag{2}$$

Here $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)_{d \times N}$ is the desired embedding matrix. In addition, a constraint on $\boldsymbol{Y}$ is needed for the regularization, i.e.

$$\frac{1}{N}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} = \boldsymbol{I}.$$

Hence, the model of this step is

$$\begin{aligned} &\min_{\boldsymbol{Y}} \left\| \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{W}^{\mathrm{T}} \right\|_{\mathrm{F}}^2 \\ &\text{s.t.} \frac{1}{N}\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}} = \boldsymbol{I}, \end{aligned}$$

which is a quadratic optimization problem that can be solved analytically by the eigen-decomposition of

$$\boldsymbol{M} = (\boldsymbol{I} - \boldsymbol{W})^{\mathrm{T}} (\boldsymbol{I} - \boldsymbol{W})$$

and the globally optimal solution can be obtained.

In addition, the generalization method of LLE can be derived directly from its inherent idea: when a new or test sample arrives, we can firstly compute its reconstruction weight vector upon $\boldsymbol{X}$ and then apply the same weights to $\boldsymbol{Y}$ to achieve its low-dimensional embedding.

B. Supervised Locally Linear Embedding

It is not difficult to infer that LLE is firstly an unsupervised manifold learning technique. Yet it can be easily adapted for supervised learning. If the procedure of choosing neighbors is characterized more theoretically, it can be described as: firstly construct the distance matrix $\boldsymbol{D} = (d_{i,j})_{N \times N}$, where $d_{i,j}$ denotes the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$; then select the indices of the m smallest elements of each row or column ($\boldsymbol{D}$ is symmetric by definition) and the corresponding samples are considered the neighbors of the target one. Now when the labels of the training samples are taken into account, the above algorithm can be adapted in this way: let

$$\boldsymbol{D}' = \boldsymbol{D} + \alpha \, max\,(\boldsymbol{D})\, \boldsymbol{L},$$

where $\alpha \in [0, 1]$, $max\,(\cdot)$ stands for the maximum function returning the largest element of $\boldsymbol{D}$ and $\boldsymbol{L}$ is an N-by-N logical matrix with

$$l_{i,j} = \begin{cases} 0, \text{ if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the same class,} \\ 1, \text{ otherwise.} \end{cases}$$

In this sense $\boldsymbol{D}$ can be substituted with $\boldsymbol{D}'$ to accomplish the neighbor selections and all the subsequent steps are the same as what are described previously. Then α-supervised locally linear embedding (α-SLLE) is achieved.

In fact, in order to reduce computation complexity, 1-SLLE is chosen to be implemented that the neighbors of some sample $\boldsymbol{x}$ are all selected from the same class, i.e. those from the different classes are regarded as infinitely far from $\boldsymbol{x}$.

IV. Manifold Regularized Stacked Autoencoder

With the widespread use of deep machine learning, SAEs become a fundamental and significant family of deep networks. As is shown in Fig. 3, their architecture in fact fairly resembles that of the classical NNs. By definition, an NN can be regarded as an SAE if it is designed to approximate the identity function at the output layer, i.e. it is expected in Fig. 3 that

$$\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{p}) \approx \boldsymbol{x}, \tag{3}$$

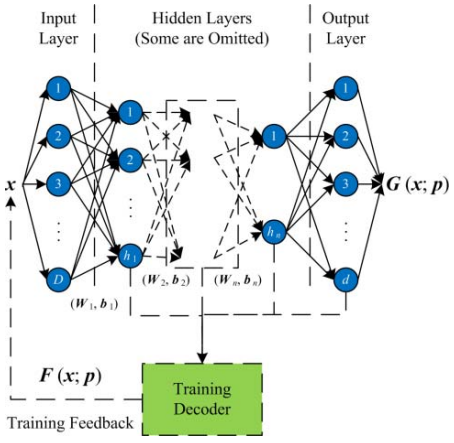


Figure 3. Stacked Autoencoder

where $\boldsymbol{F}(\cdot; \boldsymbol{p})$ means the transfer function of the whole network including the training decoder (actually one layer of nodes itself) and $\boldsymbol{p}$ stands for its parameters.

Now let us consider the learning problem of SAE. Apparently, the objective function can be derived from (3), so here we mainly focus on the sparsity constraints to be taken. This idea is quite understandable because SAE itself is intended for dimensionality reduction and feature learning. But mathematically, it should be described with the "zero norm", which results immediately in a highly nonconvex problem with the complexity of nondeterministic polynomial time. Hence, the genuine sparsity constraints are substituted by the approximations to a parameter close to zero, which is referred to as ρ in the following parts.

Let $a_{i,j}(\boldsymbol{x})$ denote the activation of the j^{th} unit of the i^{th} hidden layer when the input sample is $\boldsymbol{x}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, h_i$, and the modified sparsity constraints (simply termed the sparsity constraints if no ambiguity occurs) can be written as

$$\bar{a}_{i,j} = \frac{1}{N}\sum_{k=1}^{N} a_{i,j}(\boldsymbol{x}_k) \to \rho, \tag{4}$$

where $\boldsymbol{x}_k$ denotes the k^{th} sample vector of the training set. Furthermore, (4) should be transformed to its penalty form as a regularization term in the objective function and thus a preferred unconstrained problem is obtained. This actually considers the distance between the average activation value of a hidden unit and the sparsity parameter and can be explicitly characterized by many measures. Here, Kullback-Leibler (KL) divergence is raised for the judge of the numerical similarity. Thus the optimization problem is eventually modeled as

$$\begin{aligned}\min \frac{1}{N}\sum_{k=1}^{N}\left\|\boldsymbol{F}(\boldsymbol{x}_k;\boldsymbol{p})-\boldsymbol{x}_k\right\|_2^2 \\ + p_s\sum_{j=1}^{h_i} KL\left(\rho \,\|\, \bar{a}_{i,j}\right) + p_w\left(\left\|\boldsymbol{W}_i\right\|_{\mathrm{F}}^2 + \frac{\left\|\boldsymbol{W}_o\right\|_{\mathrm{F}}^2}{2}\right),\end{aligned} \tag{5}$$

where p_s and p_w are the sparsity and the weight penalty factors and $KL(\cdot \,\|\, \cdot)$ and $\|\cdot\|_{\mathrm{F}}$ stand for KL divergence function and Frobenius norm of matrices respectively. The meanings of the other symbols are in accordance with those cited in (3) (4) and Fig. 3. In addition, the subscript i here implies that (5) is intended for learning the weights of the i^{th} hidden layer of the whole SAE, i.e. it is only responsible for one iteration of the layer-wise training process.

After finishing the greedy training steps, backpropagation algorithm is employed to accomplish the fine-tuning of the whole network. Then the outputs after the last layer, regardless of the decoder layer obviously, i.e. $\boldsymbol{G}(\boldsymbol{x}; \boldsymbol{p})$ in Fig. 3, exactly present the low-dimensional features that are desired by the subsequent classifiers.

Now when the manifold regularization module is taken into consideration, SAE can be adapted by changing (3) into

$$\boldsymbol{F}(\boldsymbol{x}^{'}; \boldsymbol{p}) \approx \boldsymbol{x}, \tag{6}$$

where $\boldsymbol{x}^{'}$ denotes the transformation of $\boldsymbol{x}$ after being treated by the manifold learning technique. Then (5) is consequently modified as

$$\begin{aligned}\min \frac{1}{N}\sum_{k=1}^{N}\left\|\boldsymbol{F}\left(\boldsymbol{x}_k^{'};\boldsymbol{p}\right)-\boldsymbol{x}_k\right\|_2^2 \\ + p_s\sum_{j=1}^{h_i} KL\left(\rho \,\|\, \bar{a}_{i,j}\right) + p_w\left(\left\|\boldsymbol{W}_i\right\|_{\mathrm{F}}^2 + \frac{\left\|\boldsymbol{W}_o\right\|_{\mathrm{F}}^2}{2}\right).\end{aligned} \tag{7}$$

It may appear that (7) is quite similar to (5), but the significant fact should be noticed that the kernel part of the cost function (the self-reconstruction error) is changed with (6), meaning that the core conception of SAE is adapted and we now expect the new NN to reconstruct the raw training features with their corresponding transformations achieved by the manifold regularization module rather than merely the simple self-reconstruction (i.e. the NN we propose can hardly be named as an SAE by definition). This seemingly little modification may be regarded as a trivial change of SAE, but the results to be shown in the next section are able to present a considerable promotion of the performance and approve our concept of combining SAE with the neighborhood analysis.

V. Experiments and Discussions

A. Setup

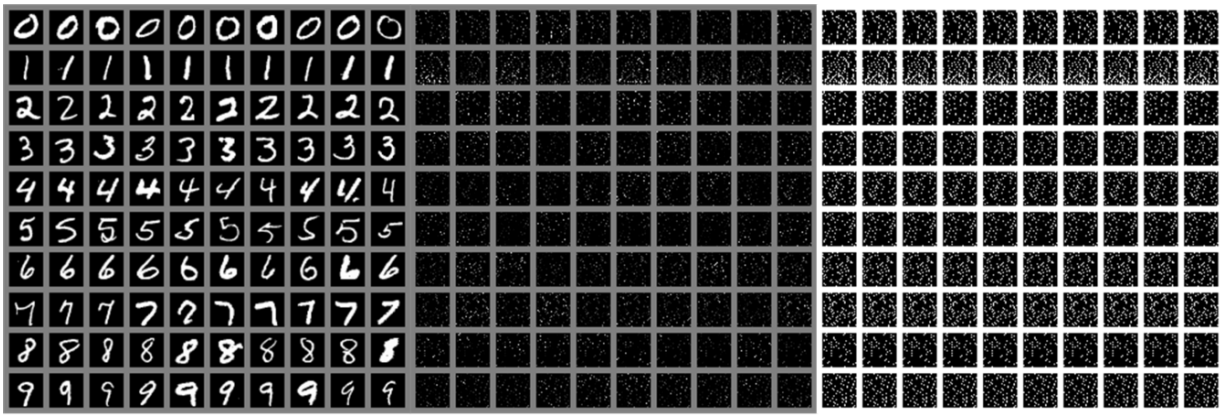
We choose to conduct the experiments on a laptop with an Intel(R) Core(TM) i7-3630QM CPU (2.40 GHz) and 4.00 GB RAM and the software platform is MATLAB R2012b in Windows 7 (64 bit).

In order to demonstrate the generalization of this framework, we assign MNIST (handwriting digits, 60, 000 training samples and 10, 000 testing samples) and YaleFaces (face images, 165 samples from 15 people) as the data sets to be tested upon (YaleFaces is randomly divided into two parts of 120 and 45 samples for training and testing respectively) and choose support vector machine to be the final supervised classifiers.

The architectures applied include: one-layer autoencoder, one-layer autoencoder with 1-SLLE regularization, three-layer SAE, three-layer SAE with 1-SLLE regularization and two layers of two-layer SAE with 1-SLLE regularization.

B. MNIST

Fig. 4 shows some randomly selected samples in MNIST and the corresponding 1-SLLE results. It can be inferred that after the manifold regularization, the handwriting digits are transformed into some much less straightforward manner. However, if enough attention is paid to the awkwardly written digits (e.g. the ninth "4"), it is not hard to discover especially through the binary map that LLE manages to extract some intrinsic features from the raw samples as the elusive patterns in the same row look much more similar to each other than the digit images.



(a) Randomly Selected Samples in MNIST (b) Corresponding 1-SLLE Results (Grayscale) (c) Corresponding 1-SLLE Results (Binary)

Figure 4. Randomly Selected Samples in MNIST and Corresponding 1-SLLE Results

Furthermore, Fig. 5 shows the visualizations of the first-layer weights of MRSAE. Seemingly recondite to understand, when they are compared with Fig. 4, it is not difficult to catch the fact that this deep network does learn the distinct features of the SLLE-treated samples (the patterns of "8" are probably the most conspicuous evidences) and tries to reconstruct the raw inputs with their combinations. In addition, about thirty of the one hundred blocks in Fig. 5 are nearly black, which suggests that the handwriting digits can be principally embedded into a seventy-dimensional feature space (manifold) although MNIST itself is a seven-hundred-and-eighty-four-dimensional data set. Such results exactly comply with our proposition as well.

Moreover, a series of experiments are performed with different neighborhood and sparsity parameters of MRSAE and the trends of the testing accuracies are shown in Fig. 6. It can be reasonably inferred that the manifold regularization module matters with its existence rather than the parameter, i.e. the number of the neighbors considered in 1-SLLE does not have a significant impact on the performance of MRSAE, but the introduction of itself and the combination with SAE play much more important roles with respect to the outcomes. Thus it is

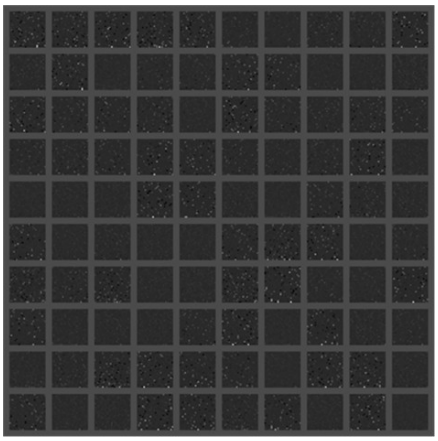
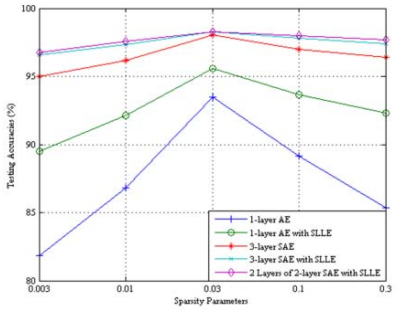
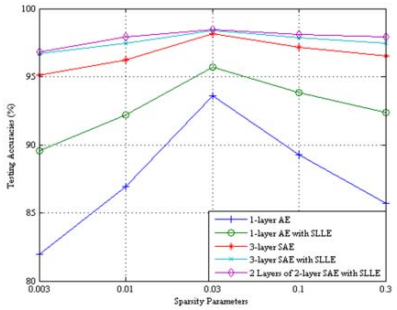


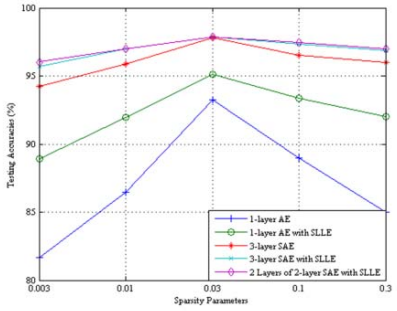
Figure 5. First-layer Weights of MRSAE upon MNIST



(a) Five Neighbors Considered



(b) Ten Neighbors Considered



(c) Twenty Neighbors Considered

Figure 6. Testing Accuracy Curves with Different Parameters

suggested that when SLLE is implemented as the manifold regularization technique, the neighborhood size can be chosen as five to twenty by experience or 0.2% to 0.5% if there are sufficient training samples. On the other hand, Fig. 6 clearly shows as well that the neighborhood analysis reduces SAE's sensitivity to certain parameters, which is one of the minor defects of deep networks and is long debated in this field, and generally raises the testing accuracies.

Finally, the best performances (i.e. the parameters are elaborately adjusted) are listed in Table I. Despite the fact that the considerable promotions of the performances of one-layer autoencoders are more likely to be caused by the addition of a feature learning module, the manifold regularization still decreases the testing error rates by over 10% regarding three-layer SAEs. This outcome can be explained as: the manifold learning process is able to perceive some intrinsic information carried by the interrelationships among the samples and more importantly, the supervised algorithm of the neighborhood analysis is an effective complement to the unsupervised nature of SAEs.

C. YaleFaces

The best performances are provided in Table II and this set of results are exactly better support evidences of our proposition of the manifold regularization regarding the decrease percentages of the testing error rates it brings about (around 15%). But unfortunately, YaleFaces is a small data set with insufficient samples. Therefore, experiments on different parameters are omitted here.

TABLE I. BEST PERFORMANCES UPON MNIST

Structures Applied	Testing Accuracies (%) (Averages of Ten Repetitions)
1-layer Autoencoder	93.62
1-layer Autoencoder with 1-SLLE	95.71
3-layer SAE	98.15
3-layer SAE with 1-SLLE	98.39
2 Layers of 2-layer SAE with 1-SLLE	98.46

TABLE II. BEST PERFORMANCES UPON YALEFACES

Structures Applied	Testing Accuracies (%) (Averages of Ten Repetitions)
1-layer Autoencoder	75.33
1-layer Autoencoder with 1-SLLE	84.22
3-layer SAE	91.33
3-layer SAE with 1-SLLE	92.67
2 Layers of 2-layer SAE with 1-SLLE	93.11

VI. CONCLUSIONS

The architecture of SAEs is modified by introducing a manifold regularization module and a new framework of MRSAE is proposed to tackle the feature learning problem. The central point of this paper is to seek out the intrinsic information buried deeply in the interrelations among the training samples and to combine it with SAEs to achieve some conspicuous promotions of their performances.

Several sets of experiments are conducted and they turn out to successfully demonstrate the idea above, especially when the samples well satisfy the manifold assumption. In addition, the results show that the supervised manifold regularization technique is probably complementary to unsupervised SAEs thus MRSAE may enjoy a better performance in this sense as well.

Further research work may focus on the other manifold learning methods that can be taken into consideration as well as many deep networks that can be modified and regularized.

REFERENCES

[1] R. Bellman, *Dynamic Programming.* Princeton, NJ: Princeton Univ. Press, 1957.

[2] B. A. Olshausen. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". *Nature* 381.6583 (1996): 607-609.

[3] G. E. Hinton and R. R. Salakhutdinov. "Reducing the dimensionality of data with neural networks". *Science* 313.5786 (2006): 504-507.

[4] C-Y Liou, J-C Huang, and W-C Yang. "Modeling word perception using the Elman network". *Neurocomputing* 71.16 (2008): 3150-3157.

[5] Y. Bengio. "Learning deep architectures for AI". *Foundations and trends® in Machine Learning* 2.1 (2009): 1-127.

[6] X.W. Jiang, J.B. Gao, X. Hong and Z.H. Cai. "Gaussian processes autoencoder for dimensionality reduction". *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, 2014. 62-73.

[7] W. Wang, Z. Cui, H. Chang, S.G. Shan and X.L. Chen. "Deeply coupled auto-encoder networks for cross-view classification". *arXiv preprint arXiv:1402.2031* (2014).

[8] S. Lauly et al. "An autoencoder approach to learning bilingual word representations". *Advances in Neural Information Processing Systems*. 2014.

[9] X. Zhang, Y.W. Fu, A. Zang, L. Sigal and G. Agam. "Learning classifiers from synthetic data using a multichannel autoencoder". *arXiv preprint arXiv:1503.03163* (2015).

[10] M. Germain, K. Gregor, I. Murray and H. Larochelle. "MADE: masked autoencoder for distribution estimation". *arXiv preprint arXiv:1502.03509* (2015).

[11] S. T. Roweis and L. K. Saul. "Nonlinear dimensionality reduction by locally linear embedding". *Science* 290.5500 (2000): 2323-2326.

[12] D. de Ridder and R. P.W. Duin. "Locally linear embedding for classification". *Pattern Recognition Group, Dept. of Imaging Science & Technology, Delft University of Technology, Delft, The Netherlands, Tech. Rep. PH-2002-01* (2002): 1-12.

[13] D. de Ridder, O. Kouropteva, O. Okun, M. Pietikäinen, and R. P.W. Duin. "Supervised locally linear embedding". *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*. Springer Berlin Heidelberg, 2003. 333-341.

[14] M. Polito and P. Perona. "Grouping and dimensionality reduction by locally linear embedding". (2002): 1255-1262.