Signal Processing 141 (2017) 137–143

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

LMAE: A large margin Auto-Encoders for classification

Weifeng Liu[a,*], Tengzhou Ma[a], Qiangsheng Xie[b], Dapeng Tao[c], Jun Cheng[d]

[a] China University of Petroleum (East China), China
[b] Shandong Institute for Food and Drug Control, China
[c] Yunnan University, China
[d] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 5 April 2017
Revised 23 May 2017
Accepted 30 May 2017
Available online 9 June 2017

Keywords:
Auto-Encoder
Large Margin
kNN
Classification

ABSTRACT

Auto-Encoders, as one representative deep learning method, has demonstrated to achieve superior performance in many applications. Hence, it is drawing more and more attentions and variants of Auto-Encoders have been reported including Contractive Auto-Encoders, Denoising Auto-Encoders, Sparse Auto-Encoders and Nonnegativity Constraints Auto-Encoders. Recently, a Discriminative Auto-Encoders is reported to improve the performance by considering the within class and between class information. In this paper, we propose the Large Margin Auto-Encoders (LMAE) to further boost the discriminability by enforcing different class samples to be large marginally distributed in hidden feature space. Particularly, we stack the single-layer LMAE to construct a deep neural network to learn proper features. And finally we put these features into a softmax classifier for classification. Extensive experiments are conducted on the MNIST dataset and the CIFAR-10 dataset for classification respectively. The experimental results demonstrate that the proposed LMAE outperforms the traditional Auto-Encoders algorithm.

1. Introduction

Feature extraction plays a key role in computer vision applications such as image annotation, face recognition, action analysis, object detection, target tracking and video retrieval. Auto-Encoders, as one representative deep learning method, has demonstrated to achieve superior performance for feature representation learning [1–5]. Recently, various variants of Auto-Encoders have been brought up. They are Sparse Auto-Encoders (SAE) [6–8], Denoising Auto-Encoders (DAE) [9–13], Contractive Auto-Encoders (CAE) [14–16], Nonnegativity Constraints Auto-Encoders (NCAE) [17,18], Laplacian Regularized Auto-Encoders (LAE) [19,20], Hessian Regularized Sparse Auto-Encoders (HSAE) [21], Bayesian Auto-Encoder (BAE) [22], Coupled Deep Auto-Encoder (CDA) [23], Multimodal Deep Auto-Encoder (MDA) [24] and Discriminative Auto-Encoder [25]. SAE introduced the sparsity regularization into the code vector of the hidden layer [6,7] or the output layer [8]. DAE was trained to make the learned representations robust to partial corruption of the input pattern [9–12]. A step further, the Denoising Auto-Encoder was trained in a convolutional way that can abstract hierarchical feature representations from raw visual data [13]. CAE added a penalty term computed by the Forbenius norm of the Jacobian matrix on the hidden layer features [14]. And then Rifai et al. extended the method by adding a penalizing term on second order derivatives of the encoders' output with respect to the input [15]. NCAE trained the Sparse Auto-Encoder by applying nonnegativity constraints [18] on the weight matrix [17]. LAE added a Laplacian regularization penalty term to enhance the locality-preserving property of learned encoders for data points [19]. HSAE applied the Hessian regularization to SAE, which can well preserve local geometry for data points [21]. In the algorithm of BAE, the author combined Auto-Encoder with Bayesian Net, and constructed multi-layer Bayes Net as a recognition system [22]. CDA was based on an individual architecture that can simultaneously learns the intrinsic representations of low-resolution and high-resolution image patches for single image super-resolution [23]. MDA extracted features with multimodal fusion and back-propagation deep learning [24].

All the abovementioned variants of Auto-Encoders learnt the feature representation without considering the label information in the pre-training phase. It is undeniable that the optimization process will be more effective with an unsupervised pre-training to initialize the model [26]. And if we enforce the discriminability of the features by using the labels, that will promote the efficiency of the classifier [27,28]. The discriminative Auto-Encoder aimed to boost the discriminability of the hidden layer features by minimizing the within class scatter and maximizing the between class scatter of samples [25], the method was efficacious. In this paper, we propose a Large Margin Auto-Encoders (LMAE) to fur-

* Corresponding author.
E-mail addresses: liuwf@upc.edu.cn, liuwfxy@gmail.com (W. Liu).

http://dx.doi.org/10.1016/j.sigpro.2017.05.030
0165-1684/

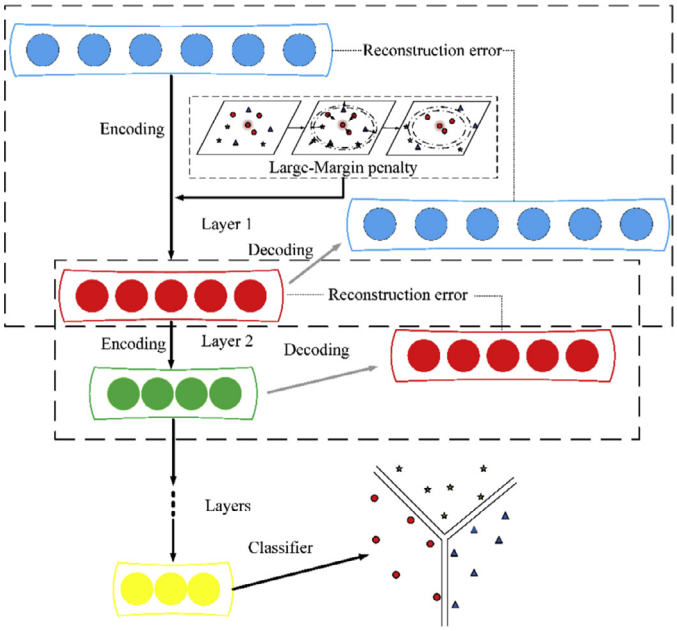

Fig. 1. The framework of LMAE for classification.

ther enforce the discriminability by enforcing different class samples to be large marginally distributed in hidden feature space. Particularly, we employ a large margin penalizing term that constrains the samples with different class labels to be distanced by an safety margin in the k-nearest neighborhood [29–32]. Fig. 1 shows the deep architecture by stacking multiple layers of LMAE for classification. In each layer, LMAE attempts to minimize the reconstruction error between the inputs and outputs while separating the different class samples with a large margin. Then, a deep architecture is constructed by stacking multiple layers of LMAE. Finally, the learnt feature representations are put into a classifier for recognition.

To assess the effectiveness of our method, we apply LMAE on two popular datasets including the MNIST dataset and the CIFAR-10 dataset for classification. And we also compare the proposed LMAE with the traditional Auto-Encoders. The experimental results demonstrate that LMAE always outperforms the traditional Auto-Encoders algorithm. In summary, our contribution in this paper is threefold: (1) we integrate the large margin penalty into the framework of Auto-Encoders that boost the discriminability significantly, (2) we provide the optimization of the proposed Large Margin Auto-Encoders (LMAE) algorithm, and (3) we conduct comparing experiments on two popular datasets respectively to demonstrate the advantages of LMAE.

We organize the rest of this paper as follows. In Section 2, we briefly review the related works including the traditional Auto-Encoders, the Discriminative Auto-Encoder and the Large-Margin kNN classification method. In Section 3, we present the proposed LMAE and the corresponding optimization. In Section 4, we describe our extensive experiments and discuss the experimental results. Finally, in Section 5, we conclude the paper with some discussions and propose possible extensions of our current method.

2. Related works

In this section, we briefly review the related works including the traditional Auto-Encoders, Discriminative Auto-Encoders and the Large-Margin Nearest Neighbor classification.

2.1. Auto-Encoders

The basic Auto-Encoder [33] aims to find a parameter vector θ for the encoder and decoder by minimizing the reconstruction error $J_{AE}(\theta)$. The objective function can be expressed as the following problem:

$$\min_{\theta} J_{AE}(\theta) = \min_{\theta} \sum_{k} L(x_k, g_\theta(f_\theta(x_k))), \tag{1}$$

where x_k is a training sample, $L(x, r) = \|x - r\|^2$ is the reconstruction error of the input and output data. $f_\theta(x) = s_f(b^e + Wx)$ and $g_\theta(x) = s_g(b^z + W'x)$ are the encoder and decoder mapping functions respectively. Usually, s_f and s_g can be the general activation functions such as the sigmoid function. The parameter vector $\theta = \{b^e, W, b^z, W'\}$, where b^e and b^z are bias vectors of the encoder and decoder, and W and W' are weight matrices of the encoder and decoder.

For the traditional Auto-Encoder, a weight decay term J_{wd} can be added into the overall objective function to control the decreasing of the weight magnitudes [34,36]. Significantly, it can improve the generalization and avoid the overfitting in the neural network by suppressing the effects of static noise on the targets and the irrelevant components of the weight vector [35].

A deep architecture can be constructed by stacking the abovementioned basis Auto-Encoders, in which the output of the hidden layer in the first encoder is treated as the input of the second encoder. And then the last layer of the deep architecture obtains the final representation of the samples that can be used for classification tasks.

2.2. Discriminative Auto-Encoders

The Discriminative Auto-Encoder tries to boost the discriminative of the hidden layer features [25]. Denote $e_{i,j}$ as the hidden layer features of the j^{th} sample from the class i. The discriminative Auto-Encoders simultaneously minimizes the within-class scatter $S_w(e)$ and maximizes the between-class scatter $S_b(e)$. The $S_w(e)$ and $S_b(e)$ can be defined as:

$$S_w(e) = \sum_{i=1}^{c} \sum_{e_{i,j} \in i} \left(e_{i,j} - \bar{e}_i\right)\left(e_{i,j} - \bar{e}_i\right)^T, \tag{2}$$

$$S_b(e) = \sum_{i=1}^{c} m_i(\bar{e}_i - \bar{e})(\bar{e}_i - \bar{e})^T, \tag{3}$$

where $\bar{e}_i$ and $\bar{e}$ are denoted as the mean vector of e_i and e, respectively. And m_i is the samples number of class i. The discriminative regularization term can be defined as:

$$L(e) = tr(S_w(e)) - tr(S_b(e)). \tag{4}$$

Then, the objective function of the Discriminative Auto-Encoder can be expressed as the following problem:

$$\min_{\theta} J_{Dis-AE}(\theta) = \min_{\theta} \left[J_{AE}(\theta) + \frac{1}{2}\lambda J_{wd} + \frac{1}{2}\gamma L(e)\right], \tag{5}$$

where λ and γ are parameters to balance the different penalty terms respectively.

2.3. Large-Margin kNN classification

The Large-Margin kNN classification method (LMNN) attempts to shrink distances of neighboring same labeled points and to separate points in different classes [30,37]. The LMNN optimization problem can be formulated as the following minimization problem:

Table 1
List of important notations.

Notation	Description	Notation	Description
m	Number of samples	x_k	kth input training sample
n	Dimensionality of sample space	e_k	The data in hidden layer of the kth training sample
d	Dimensionality of hidden space	z_k	Reconstruction data from the kth training sample
W	The weight matrix of encoder	W'	The weight matrix of decoder
b^e	The bias of encoder	b^z	The bias of decoder
λ,β	Parameters of objective function terms	α	Learning rate in gradient decent method
$\eta_{k_1k_2}$	Target neighbors parameter	$\tau_{k_1k_3}$	Same class identifier parameter

$$\min_W \sum_{il} \eta_{il} d_W(i,l) + \sigma \sum_{ilj} \eta_{il}\left(1-\tau_{ij}\right) h(1+d_W(i,j)-d_W(i,l))_+, \tag{6}$$

where $d_W(i,j) = \|W(x_i - x_j)\|^2$ is the square distance between the projections of sample x_i and x_j, W is the transformation matrix of the mapping functions. $\eta_{il} = 1$ is to indicate that x_l is one of the target neighbors of x_i (*i.e.* x_l is one of the k nearest neighbors of x_i and share the same label with x_i), and otherwise $\eta_{il} = 0$. $\tau_{ij} = 1$ is to indicate that x_j has the same label with x_i, and otherwise $\tau_{ij} = 0$. $h(\cdot)$ is a hinge loss function that is denoted as $h(a)_+ = max(a, 0)$. σ is a positive constant that controls the relative importance of the two terms. In problem (6), the first term is used to penalize the large distances between each input and its target neighbors, while the second term penalizes small distances between each input and all other inputs that do not share the same label [30].

3. Large Margin Auto-Encoders

In this section, we introduce our proposed the Large Margin Auto-Encoders (LMAE) and its optimization. For both feature extraction process of Auto-Encoders and LMNN are mapping processes for the samples from original space to feature space. They can be optimized by minimizing their objective functions. We blend their objective functions in a same optimization process to achieve the purpose of the fusion of Auto-Encoders and LMNN.

Suppose we are given m training samples in the n-dimensional space: $X = \{x_k\}_{k=1}^m \in R^n$. LMAE aims to find an effective representation $E = \{e_k\}_{k=1}^m \in R^d$ for the training samples with a proper encoding function $e = f(x) = s(Wx + b^e)$ by minimizing the reconstruction error between the inputs and outputs. The reconstruction outputs are got by the corresponding decoding function $z = g(e) = s(W'e + b^z)$, where $s(\theta) = \frac{1}{1+e^{-\theta}}$ is a sigmoid function. In the following sections, we use $X = [x_1, \cdots, x_m] \in R^{n\times m}$ to denote the training sample matrix, $E = [e_1, \cdots, e_m] \in R^{d\times m}$ to denote the representation matrix of the training samples and $Z = [z_1, \cdots, z_m] \in R^{n\times m}$ to denote the reconstruction matrix of the training samples. Then we have the objective function of the LMAE as below:

$$J_{LMAE} = J_{AE} + \lambda J_{wd} + \beta J_{Large-Margin}, \tag{7}$$

where λ and β are the parameters to balance the different penalty terms respectively.

Here, the first term $J_{AE} = [\frac{1}{m}\sum_{k=1}^m \|x_k - g(f(x_k))\|^2]$ is the average reconstruction errors between the input data and the output data for all samples. The second term $J_{wd} = \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^d (W_{ji})^2$ is the weight decay term. The last term $J_{Large-Margin}$ is the Large-Margin penalty and can be expressed as:

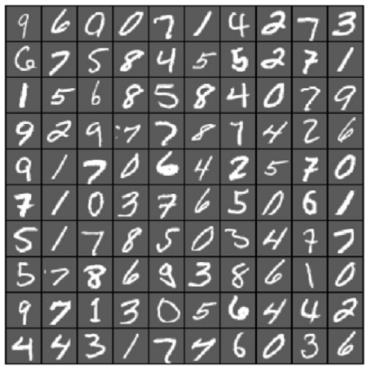

Fig. 2. Some samples of hand-written digits from the MNIST dataset.

$$\begin{aligned} J_{Large-Margin} = &\sum_{k_1=1}^m \sum_{k_2=1}^m \eta_{k_1k_2} \|W\left(x_{k_1} - x_{k_2}\right)\|^2 \\ &+\sigma \sum_{k_1=1}^m \sum_{k_2=1}^m \sum_{k_3=1}^m \eta_{k_1k_2}\left(1-\tau_{k_1k_3}\right) h\left(s_{k_1k_2k_3}\right)_+. \end{aligned} \tag{8}$$

Here, $\eta_{k_1k_2} \in \{0,1\}$, $\eta_{k_1k_2} = 1$ means that x_{k_2} is one of the target neighbors of x_{k_1}, and $\eta_{k_1k_2} = 0$ otherwise. $\tau_{k_1k_3} \in \{0,1\}$, $\tau_{k_1k_3} = 1$ means that x_{k_3} shared the same label with x_{k_1}, and $\tau_{k_1k_3} = 0$ otherwise. σ is a positive constant to control the relative importance of the two competing terms in $J_{Large-Margin}$ and we set $\sigma = 1$ in this paper. $s_{k_1k_2k_3} = 1 + \|W(x_{k_1} - x_{k_2})\|^2 - \|W(x_{k_1} - x_{k_3})\|^2$ is the slack variable. And $h(s)_+ = \max(s, 0)$ is the hinge function. For the optimization, it is difficult to find the global optimum of the hinge function. Then we use a substitution function $h(s)$ to approximate it [38]. And $h(s) = \frac{1}{\gamma}\ln(1 + \exp(\gamma s))$, where γ is a constant that controls the approximation and we set $\gamma = 500$ in this paper.

Then the formula of Large-Margin has the following expression

$$\begin{aligned} J_{Large-Margin} = &\sum_{k_1=1}^m \sum_{k_2=1}^m \eta_{k_1k_2} \|W\left(x_{k_1} - x_{k_2}\right)\|^2 \\ &+\sigma \sum_{k_1=1}^m \sum_{k_2=1}^m \sum_{k_3=1}^m \eta_{k_1k_2}\left(1-\tau_{k_1k_3}\right) h\left(s_{k_1k_2k_3}\right). \end{aligned} \tag{9}$$

For convenience, we list the important notations with brief description in Table 1.

In order to optimize the objective function (7) w.r.t. W, b^e, and b^z, we employ the gradient decent method and hence use the updating iteration as follows [29]:

$$W_{ij}(t+1) := W_{ij}(t) - \alpha \frac{\partial J_{LMAE}}{\partial W_{ij}(t)}, \tag{10}$$

$$b_i^e(t+1) := b_i^e(t) - \alpha \frac{\partial J_{LMAE}}{\partial b_i^e(t)}, \tag{11}$$

$$b_j^z(t+1) := b_j^z(t) - \alpha \frac{\partial J_{LMAE}}{\partial b_j^z(t)}, \tag{12}$$

where α is the learning rate.

Fig. 3. Some samples of CIFAR-10 dataset.

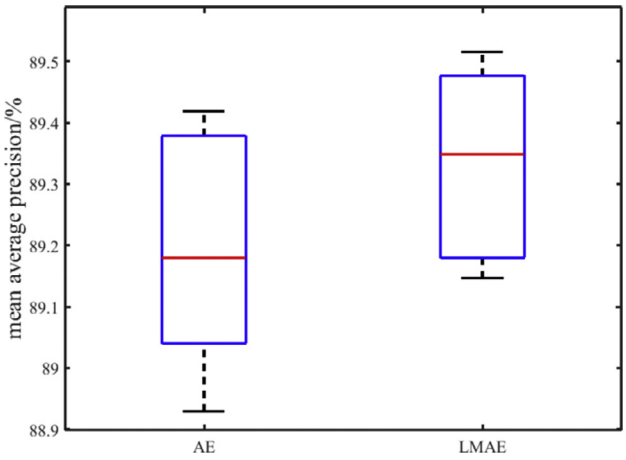

Fig. 4. The mean average precision of standard AE and LMAE on MNIST.

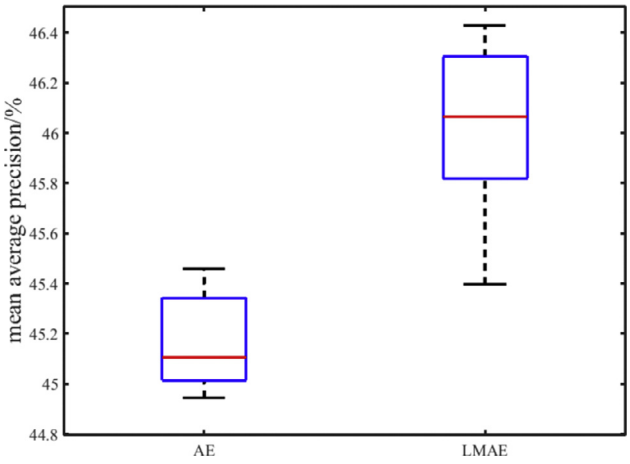

Fig. 5. The mean average precision of standard AE and LMAE on CIFAR-10.

Now, we introduce the computing of the partial derivatives w.r.t. each variable. For the term of reconstruction error $J_{AE} = [\frac{1}{m}\sum_{k=1}^{m}\|x_k - z_k\|^2]$, the partial derivatives w.r.t. Z can be written as $\partial J_{AE}/\partial Z = -\frac{2}{m}(X - Z)$. Then we derive the partial derivatives of J_{AE} w.r.t. decoder variables as follows:

$$\frac{\partial J_{AE}}{\partial b^z} = \left[\frac{\partial J_{AE}}{\partial z} \circ s'\left(W'E + b^z \otimes 1_m^T\right)\right]1_m, \tag{13}$$

$$\frac{\partial J_{AE}}{\partial W'} = \left[\frac{\partial J_{AE}}{\partial z} \circ s'\left(W'E + b^z \otimes 1_m^T\right)\right]E^T, \tag{14}$$

where $\circ$ is Hadamard product, and $\otimes$ is outer product, 1_m denotes a m-dimensional column vector with all entry values of 1. $s'(\theta)$ is the derivatives of the sigmoid function, it can be deduced as $s'(\theta) = s(\theta)[1 - s(\theta)]$ by applying to matrices element-wisely. Similarly, the partial derivatives of J_{AE} w.r.t. hidden representation E can be written as $\frac{\partial J_{AE}}{\partial E} = W[\frac{\partial J_{AE}}{\partial z} \circ s'(W'E + b^z \otimes 1_m^T)]$. And then we can express the derivatives of J_{AE} w.r.t. the encoder variables as follows:

$$\frac{\partial J_{AE}}{\partial b^e} = \left[\frac{\partial J_{AE}}{\partial E} \circ s'\left(WX + b^e \otimes 1_m^T\right)\right]1_m, \tag{15}$$

$$\frac{\partial J_{AE}}{\partial W} = \left[\frac{\partial J_{AE}}{\partial E} \circ s'\left(WX + b^e \otimes 1_m^T\right)\right]X^T. \tag{16}$$

Note that in formulas (14) and (16), the derivatives of ∂J_{AE} w.r.t. the weight matrices W and W' of encoder and decoder functions are decoupled. To ensure tied weight between encoder and decoder, formulas (14) and (16) can simply be combined as:

$$\frac{\partial J_{AE}}{\partial W} := \frac{1}{2}\left\{\left[\frac{\partial J_{AE}}{\partial z} \circ s'\left(W'E + b^z \otimes 1_m^T\right)\right]E^T + \left[\frac{\partial J_{AE}}{\partial E} \circ s'\left(WX + b^e \otimes 1_m^T\right)\right]X^T\right\}. \tag{17}$$

There is only the weight matrix W involved in the second term of formula (7). So the partial derivatives of J_{wd} w.r.t. W can be computed as $\frac{\partial J_{wd}}{\partial W} = W$.

For the Large-Margin term $J_{Large-Margin}$ of the objective function, the partial derivatives of $J_{Large-Margin}$ w.r.t. the encoder variables are as follows:

$$\frac{\partial J_{Large-Margin}}{\partial b^e} = 0, \tag{18}$$

$$\begin{aligned}\frac{\partial J_{Large-Margin}}{\partial W} &= 2W\sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\eta_{k_1k_2}\left(x_{k_1} - x_{k_2}\right)\left(x_{k_1} - x_{k_2}\right)^T \\ &+2cW\sum_{k_1=1}^{m}\sum_{k_2=1}^{m}\sum_{k_3=1}^{m}\eta_{k_1k_2}\left(1 - \tau_{k_1k_3}\right) \\ &\times\begin{bmatrix}\left(x_{k_1} - x_{k_2}\right)\left(x_{k_1} - x_{k_2}\right)^T \\ -\left(x_{k_1} - x_{k_3}\right)\left(x_{k_1} - x_{k_3}\right)^T\end{bmatrix}\left[1 - \frac{1}{1-\exp\left(\gamma s_{k_1k_2k_3}\right)}\right]\end{aligned} \tag{19}$$

Combining the aforementioned partial derivatives of reconstruction and weight decay terms, we have the partial derivatives of the LMAE objective J_{LMAE} w.r.t. the involved variables as follows:

$$\frac{\partial J_{LMAE}}{\partial W} = \frac{\partial J_{AE}}{\partial W} + \lambda\frac{\partial J_{wd}}{\partial W} + \beta\frac{\partial J_{Large-Margin}}{\partial W}, \tag{20}$$

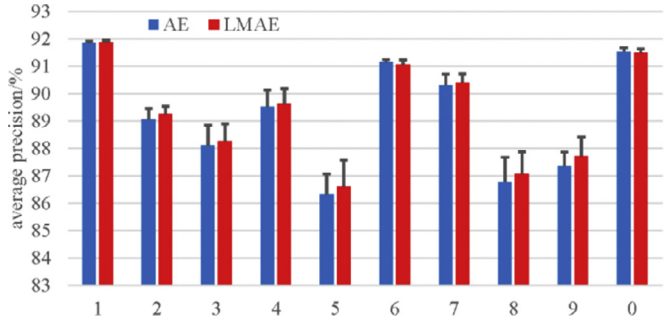

Fig. 6. The average precision on single digit class of MNIST.

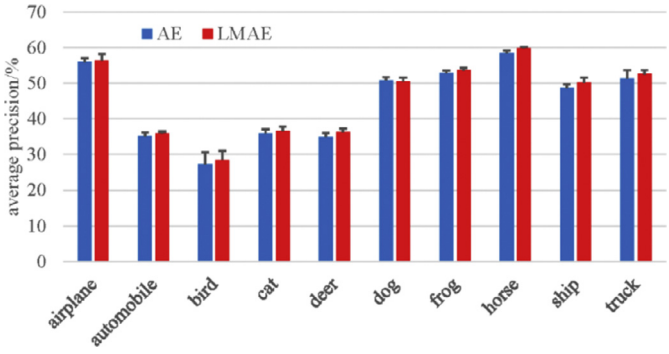

Fig. 7. The average precision on single object class of CIFAR-10.

$$\frac{\partial J_{LMAE}}{\partial b^e} = \frac{\partial J_{AE}}{\partial b^e} + \lambda \frac{\partial J_{wd}}{\partial b^e}, \tag{21}$$

$$\frac{\partial J_{LMAE}}{\partial b^z} = \frac{\partial J_{AE}}{\partial b^z} + \lambda \frac{\partial J_{wd}}{\partial b^z}. \tag{22}$$

We stack the basic Auto-Encoders abovementioned to form the deep architecture. For optimization, we employ the L-BFGS algorithm to update the parameters of LMAE layer by layer. In particular, we firstly learn the encoder and decoder of the first layer. Then we put the output of the encoder into the next layer. When all the encoders are obtained, the learnt features from the multilayer LMAE are put into a softmax classifier. At last we employ fine-tuning to slightly modify the features of the LMAE according to the boundaries between the classification classes.

In each iteration of LMAE, the time complexity of computing the reconstruction error term J_{AE} and its gradient ∂J_{AE} is $O(dmn)$. The time complexity of computing the weight decay term J_{wd} and its gradient ∂J_{wd} is $O(dn)$. The time complexity of computing the first term of formula (9) is $O(km)$, and the second term is $O(km^2)$. Finally, the time complexity of computing LMNN term $J_{Large-Margin}$ and its gradient $\partial J_{Large-Margin}$ is $O(km^2)$. Therefore, the overall time complexity of computing the proposed LMAE is $O(km^2 + dmn)$ for each iteration. k is the number of nearest neighbor samples here.

4. Experiments

To evaluate the performance of the proposed LMAE, we carry out experiments on the MNIST dataset [39] for digits recognition and the CIFAR-10 dataset [40] for object classification respectively.

The MNIST dataset consists of 70,000 gray-scale 28 × 28 images of hand-written digits. Fig. 2 shows some samples of MNIST. We randomly select a subset of 1000 images that contains about 100 images for each digit for training and the rest 69,000 images for testing. We conduct the experiment for 5 times.

The CIFAR-10 dataset consists of 60,000 32 × 32 color images. There are 10 object classes in CIFAR-10 dataset including airplane, deer, automobile, dog, bird, cat, truck, frog, horse and ship with 6000 images per class. Some image samples are shown in Fig. 3. We randomly select a subset of 10,000 images for training and the rest for testing. And the experiment is also carried out for 5 times.

For different problem or different dataset, we can set more or less layers for the model. In this paper, we stack two layers of Auto-Encoders to form the LMAE architecture. This setting is enough to react the problem of the datasets and to prove the effectiveness of our method. Specifically, we set 600 for the number of the first hidden layer units and 200 for the second hidden layer for the MNIST dataset. And we set 2000 for the number of the first hidden layer units and 800 for the second hidden layer for the CIFAR-10 dataset. The parameters λ and β are tuned on a range of value based on the standard Auto-Encoders. λ is tuned from the candidate set $\{1 \times 10^{\alpha} | \alpha = -5, -4, -3, -2, -1\}$, and β is tuned from the set $\{1 \times 10^{\alpha} | \alpha = -15, -14, \ldots, -2, -1\}$. For the neighborhood indication, we set $k = 3$. And we use the average precision (AP) for each single class and mean average precision over all classes for evaluation.

Figs. 4 and 5 are the box plot of the mean average precision of LMAE and the standard AE algorithm on the MNIST dataset and CIFAR-10 dataset respectively. We can see that the proposed LMAE performs significantly better than the compared AE algorithm for both two experiments.

Figs. 6 and 7 demonstrate the bar plot of the two methods on each single class of MNIST and CIFAR-10. The horizontal axis shows each class label, and the vertical axis shows the mean average precision. We can see that for most classes, the LMAE outperforms the standard AE. Furthermore, the standard deviation shows the results of LMAE is more stable.

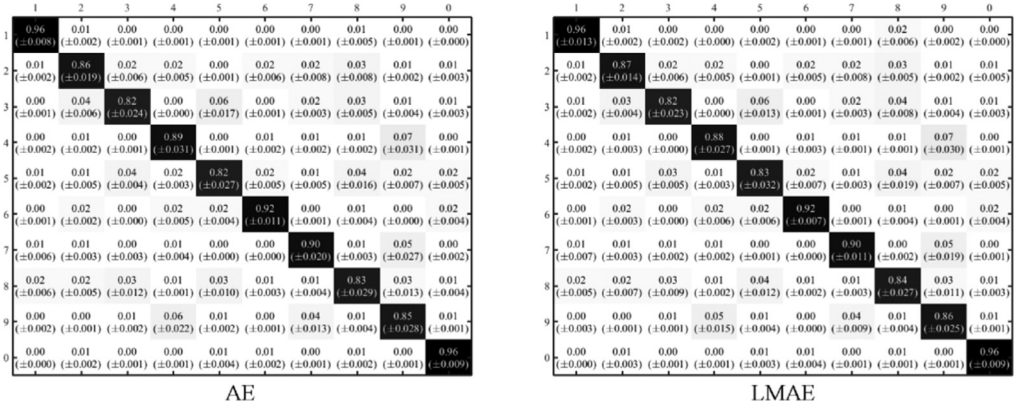

Fig. 8. The confusion matrix on MNIST.

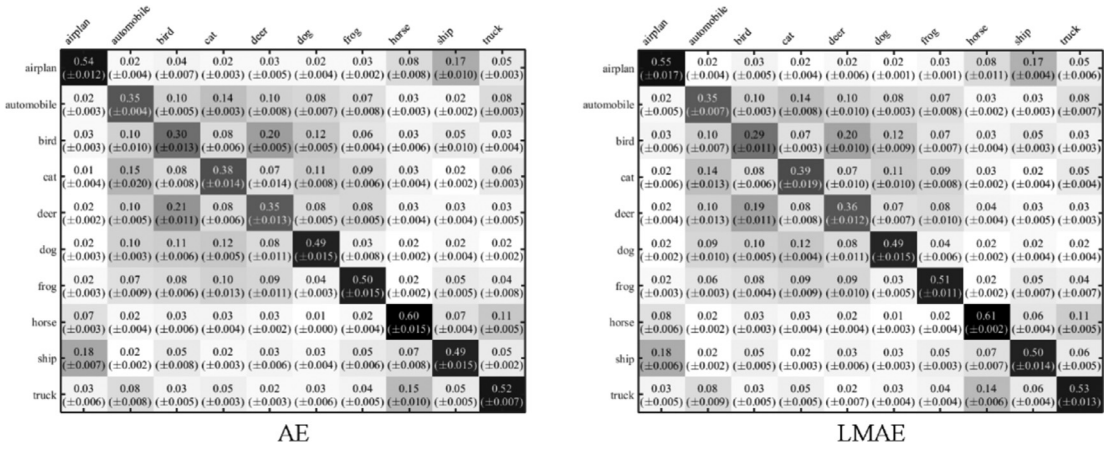

Fig. 9. The confusion matrix on CIFAR-10.

Figs. 8 and 9 further illustrate the average confusion matrix of different algorithms on each dataset. We can see that from the confusion matrix in Figs. 8 and 9 the proposed LMAE performs better than the compared AE algorithm in most cases.

5. Conclusion

In this paper, we integrate large-margin into Auto-Encoders and propose the Large Margin Auto-Encoders (LMAE). By adding large-margin penalty in hidden feature space, LMAE boost the discriminability for different classes than the traditional Auto-Encoders. We stack the single-layer LMAE to construct a deep neural network architecture for feature learning. Extensive classification experiments on the MNIST dataset and the CIFAR-10 dataset demonstrate the advantages of the proposed LMAE algorithm.

Acknowledgments

This paper is partly supported by the National Natural Science Foundation of China (Grant No. 61671480, 61572486), the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) (Grant No. 14CX02203A), the Yunnan Natural Science Funds (Grant No. 2016FB105).

References

[1] S. Su, Z. Liu, S. Xu, S. Li, R. Ji, Sparse auto-encoder based feature learning for human body detection in depth image, Signal Process. 112 (2015) 43–52.
[2] C. Hong, X. Chen, X. Wang, C. Tang, Hypergraph regularized autoencoder for image-based 3D human pose recovery, Signal Process. 124 (2016) 132–140.
[3] H. Cheng, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: theory and applications, Signal Process. 93 (6) (2013) 1408–1425.
[4] T. Liu, D. Tao, D. Xu, Dimensionality-dependent generalization bounds for k-dimensional coding schemes, Neural Comput. 28 (10) (2016) 2213–2249.
[5] C. Hong, J. Yu, X. Chen, Image-based 3D human pose recovery with locality sensitive sparse retrieval, in: Systems, Man, and Cybernetics (SMC), IEEE International Conference, 2013, pp. 2103–2108.
[6] C.P. MarcAurelio Ranzato, S. Chopra, Y. LeCun, Efficient learning of sparse representations with an energy-based model, in: Proceedings of the 19th International Conference on Neural Information Processing Systems, 2006, pp. 1137–1144.
[7] H. Lee, C. Ekanadham, A.Y. Ng, Sparse deep belief net model for visual area V2, Advances in Neural Information Processing Systems (2008) 873–880.
[8] Y. Boureau, Y. LeCun, Sparse feature learning for deep belief networks, in: Advances in Neural Information Processing Systems, 2008, pp. 1185–1192.
[9] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[10] P. Vincent, A connection between score matching and denoising autoencoders, Neural Comput. 23 (7) (2011) 1661–1674.
[11] K. Swersky, D. Buchman, N.D. Freitas, B.M. Marlin, On autoencoders and score matching for energy based models, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 1201–1208.
[12] P. Vincent, H. Larochelle, Y. Bengio, P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.
[13] B. Du, W. Xiong, J. Wu, L. Zhang, D. Tao, Stacked convolutional denoising auto-encoders for feature representation, IEEE Trans. Cybern. (2016) 1–11.
[14] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 833–840.
[15] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, X. Glorot, Higher order contractive auto-encoder, Machine Learning and Knowledge Discovery in Databases (2011) 645–660.
[16] Y. Liu, X. Feng, Z. Zhou, Multimodal video classification with stacked contractive autoencoders, Signal Process. 120 (2016) 761–766.
[17] E. Hosseini-Asl, J.M. Zurada, O. Nasraoui, Deep learning of part-based representation of data using sparse Autoencoders with nonnegativity constraints, IEEE Trans. Neural Networks Learn. Syst. 27 (12) (2016) 2486–2498.
[18] J. Chorowski, J.M. Zurada, Learning understandable neural networks with nonnegative weight constraints, IEEE Trans. Neural Networks Learn. Syst. 26 (1) (2015) 62–69.
[19] K. Jia, L. Sun, S. Gao, Z. Song, B.E. Shi, Laplacian Auto-Encoders: an explicit learning of nonlinear data manifold, Neurocomputing 160 (2015) 250–260.
[20] Q. Liu, Y. Huang, D.N. Metaxas, Hypergraph with sampling for image retrieval, Pattern Recognit. 44 (10) (2011) 2255–2262.
[21] W. Liu, T. Ma, D. Tao, J. You, HSAE: a Hessian Regularized Sparse Auto-Encoders, Neurocomputing 187 (2016) 59–65.
[22] K. Nishino, M. Inaba, Bayesian AutoEncoder: generation of Bayesian networks with hidden nodes for features, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 4244–4245.
[23] K. Zeng, J. Yu, R. Wang, C. Li, D. Tao, Coupled deep autoencoder for single image super-resolution, IEEE Trans. Cybern. 47 (1) (2017) 27–37.
[24] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep Autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[25] J. Xie, Y. Fang, F. Zhu, E. Wong, Deepshape: deep learned shape descriptor for 3D shape matching and retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1275–1283.
[26] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (Feb) (2010) 625–660.
[27] Q. Liu, J. Yang, K. Zhang, Y. Wu, Adaptive compressive tracking via online vector boosting feature selection, IEEE Trans. Cybern. (2016).
[28] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 447–461.
[29] L. Torresani, K.C. Lee, Large margin component analysis, advances in neural information processing systems, (2006) 1385–1392.
[30] K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (2009) 207–244.
[31] D. Tao, C. Xu, Y. Rui, Large-margin weakly supervised dimensionality reduction, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 865–873.
[32] J. Chai, H. Liu, B. Chen, Z. Bao, Large margin nearest local mean classifier, Signal Process. 90 (1) (2010) 236–248.
[33] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.
[34] C. Gravelines, Electronic Thesis and Dissertation Repository, 2014.
[35] J. Moody, S. Hanson, A. Krogh, J.A. Hertz, A simple weight decay can improve generalization, Adv. Neural Inf. Process. Syst. 4 (1991) 950–957.
[36] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[37] R. Min, D. Stanley, Z. Yuan, A. Bonner, Z. Zhang, A deep non-linear feature mapping for large-margin knn classification, in: 2009 Ninth IEEE International Conference on Data Mining, 2009, pp. 357–366.

[38] J.D. Rennie, N. Srebro, Fast maximum margin matrix factorization for collaborative prediction, in: Proceedings of the 22nd international conference on Machine learning, 2005, pp. 713–719.
[39] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 86, 1998, pp. 2278–2324. http://yann.lecun.com/exdb/mnist/.
[40] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, 2009. http://www.cs.toronto.edu/~kriz/cifar.html.