



Boosting Sparsity-Induced Autoencoder: A Novel Sparse Feature Ensemble Learning for Image Classification

Rui Shi, Jian Ji^(✉), Chunhui Zhang, and Qiguang Miao

School of Computer Science and Technology, Xidian University,
Xi'an 710071, China
jji@xidian.edu.cn

Abstract. As a model of unsupervised learning, autoencoder is often employed to perform the pre-training of the deep neural networks. However, autoencoder and its variants have not taken the statistical characteristics and the domain knowledge of training set into the design of deep neural networks and have abandoned a lot of features learned from different levels at the pre-training process. In this paper, we propose a novel sparse feature ensemble learning method for natural image classification, named boosting sparsity-induced autoencoder, to fully utilize hierarchical and diverse features. Firstly, a sparsity encourage method is introduced by adding an extra sparsity-induced layer to exploit the representative and intrinsic features of the input. And then, the ensemble learning is taken into consideration of the construction of the model to improve and boost the accuracy and stability of a single model. The classification results on three datasets demonstrate the effectiveness of the proposed method.

Keywords: Sparse representation · Sparsity-induced method
Ensemble learning · Image classification

1 Introduction

The performance of a generic learning algorithm, especially adopted to the classification problem, extremely relies on the quality of learned feature representation of raw input data. Good features not only could remove irrelevant or redundant features coexisting in the original input space, but preserve the essential information for the target tasks. A good feature extractor built for input space, especially using unsupervised learning methods, can be further utilized for computer vision tasks. Deep hierarchical features produced by stacked unsupervised models have been demonstrated to be a powerful tool and appeal to emerging focus [1, 2].

In recent years, the study found that deep learning constructed by the multiple non-linear transformations can be a powerful feature learning tool. Deep learning has already been broadly used to address image classification tasks [3–6]. As a tool of deep learning with a special architecture, the autoencoder has already been stacked to pre-train a deep neural network using a greedy layer-wise means [7], where each layer is separately initialized by unsupervised pre-training method, and then a fine-tuning way

based on backpropagation is used by a supervised learning algorithm [8, 9], leading to solving the lack of expression ability of shallow network.

By restricting the output of the model identical to the input data, autoencoder can be regarded as an identity function which could reconstruct the raw input data composed of an encoding phase and a decoding phase. Meanwhile, sparse representation has proven its significant impact on computer vision [10–12]. The performance of an image classifier can be improved if the input image can be represented by a sparse representation. Ghifary [12] demonstrated that, in most cases, sparse network structures have better classification performance than dense structures. In recent years, the sparse deep model is proposed based on the sparse encoding strategy, sparse regularization term and sparse filtering that have taken the input samples into sparse depth related neural network model.

However, autoencoder and its variants have not taken the statistical characteristics and the domain knowledge of training set into the design of deep networks, and they have abandoned a lot of features learned from different levels. Therefore, autoencoder can only provide a relatively coarse parameters setting and serves as a pre-training method because of the large variance and low generalization ability on the unknown testing dataset. So, how to fully utilize the features existed in the input is one of the most important points in our work. It is well known that an ensemble of multiple classifiers is considered as a practical technique for improving accuracy and stability with comparisons to a single classifier. Ensemble learning employs some weak classifiers, according to some combination rule, to construct a stronger one to obtain significantly reduced generalization error than any weak one. But, two key issues, namely the diversity and accuracy of each classifier and the combination rules of fusion rules [13], are required to be taken into consideration to ensure a better performance.

In this paper, we introduce a novel sparsity-induced autoencoder that can further exploit the representative and intrinsic features of the input. Then, to benefit the ability of the ensemble learning, an ensemble sparse feature learning algorithm based on the novel sparse autoencoder mentioned above, named BoostingAE, is proposed. On the one hand, the completion of the pre-training sparsity-induced autoencoder can obtain a plurality of different levels of abstraction of sparse features; on the other hand, ensemble learning could effectively improve and enhance the recognition rate and stability of single classifier. Experimental results on three different datasets show that the proposed ensemble feature learning method can significantly improve the overall performance.

2 Related Work

2.1 Sparse Representation

Sparse Coding. Sparse coding provides a family of methods for acquiring the condense features in the input. Given only the unlabeled dataset, it can discover the basic functions aimed to capture the higher-level features in the data itself. Despite its close relationship to the traditional sparse coding techniques on image denoising, the main

drawback of sparse coding is its high computation cost. Moreover, it is well-known that the sparse coding is not “smooth” [14, 15], which means a tiny variation in input space might result in a significant difference in code space.

Sparse Filtering [16]. In contrast to many existing feature learning models, one of the important properties of sparse filtering is that it only requires one hyper-parameter rather than extensive hyper-parameters tuning for its very simple cost function:

$$\min \sum_{i=1}^M \|\hat{f}^{(i)}\|_1 = \sum_{i=1}^M \left\| \frac{\tilde{f}^{(i)}}{\|\tilde{f}^{(i)}\|_2} \right\|_1 \tag{1}$$

where f represents the learned feature value for input sample, \tilde{f} is defined by ℓ_2 norm of f , and M indicates the sample’s number.

Sparse Regularization. Compared with sparse coding, sparse regularization needs to perform an extra separate stage to induce sparsity and encourage sparse representations of input. Various methods of sparsity regularization either employed in deep belief network or autoencoder [17], similar to sparse coding, each of which has been proved the beneficial effects for some particular scene.

2.2 Softmax Regression

Softmax regression is a generalized version of logistic regression applied to classification problems where the class label y can be chosen from more than two values. Assume that there are k labels and m training samples: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(i)}, y^{(i)}), \dots, (x^{(m)}, y^{(m)})\}$ ($i = 1, 2, \dots, m$), where x is the input sample, and $y \in \{1, 2, \dots, k\}$ is the corresponding label.

For every input, the output probability function can be defined as follows:

$$h_{\theta}(x^i) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_j^T x^{(i)}} \end{bmatrix} \tag{2}$$

where θ is the parameter of the softmax model. For each input, the probability of its category is estimated to be:

$$p(y = k|x^{(i)}; \theta) = \frac{e^{\theta_k^T x^{(i)}}}{\sum_k e^{\theta_k^T x^{(i)}}} \tag{3}$$

2.3 Ensemble Learning

According to certain combination rule, ensemble learning employs some weak classifiers to construct a stronger one to obtain significantly reduced generalization error

than any weak one. Weak learner refers to whose generalization performance on the unknown testing dataset is only slightly better than random guessing. From mathematics, ensemble learning can significantly reduce the variance to achieve more stable performance. In order to get a better integration result, it is necessary to make the individual learner as different as possible, that is to say, there is a high degree of diversity between the base learners, which will be helpful to the performance of ensemble learning.

Boosting method is a widely used method for statistical learning, and serves as an important means of ensemble learning. By changing the weights of training samples, boosting method trains a group of individual learners and gets final decision results with a combination rule of voting.

3 Boosting Sparsity-Induced Autoencoder

To learn more representative and intrinsic features of input, a novel sparsity encourage method is first introduced to build a new autoencoder, called sparsity-induced autoencoder (SparsityAE). Based on SparsityAE and ensemble learning, we further proposed a boosting sparsity-induced autoencoder (BoostingAE), which is capable of utilizing the hierarchical and diverse features, ensuring the accuracy and diversity, and boosting the performance of the single SparsityAE on computer vision tasks.

3.1 Sparsity-Induced Autoencoder

Inspired by the assumptions of the sparse representation and the efficient reconstruction of low-dimension feature representation obtained in the encoding phase of deep models, SparsityAE is proposed, whose structure is shown in Fig. 1.

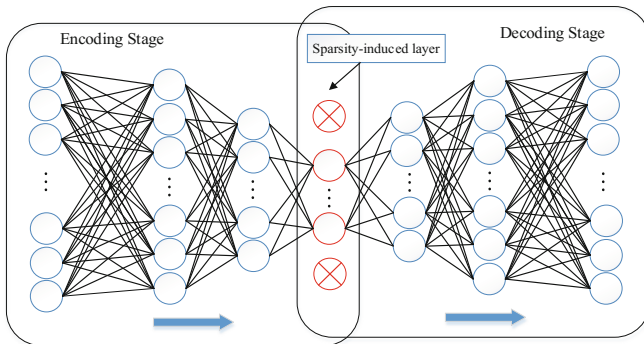


Fig. 1. The topology of the proposed sparsity-induced autoencoder

We feed the encoder by the dataset of high-dimension space. The length of the codes learned by each layer gets less along with the deepening of the encoder. At the end of the encoding phase, we employ a sparsity-induced layer to generate sparse

codes. Conversely, the decoding phase deals with the compressed and sparse codes given by sparsity-induced layer. In sparsity-induced layer, the neurons without significant activation value will be set to zero, which could decrease the number of neurons, remove the correlation between attributes and compress the raw inputs.

Let $y_i (i = 1, 2, \dots, N)$ be the original data and x_i be its degraded version, so the input can be mapped to a hidden representation by the formulations as follows:

$$\hat{y}(x_i) = \sigma(W'h(x_i) + b') \tag{4}$$

where \hat{y}_i is an approximation of y_i , and $\sigma(\bullet)$ is the mapping function.

To benefit both from the virtues of sparse representation and deep neural networks, we optimize the reconstruction loss regularized by a weight decay and a sparsity-inducing term. The cost function can be designed as follows:

$$L(X, Y; \theta) = \|y_i - \hat{y}(x_i)\|_2^2 / N + \beta \bullet KL(\hat{\rho} || \rho) \tag{5}$$

where $\theta = (W, b, W', b')$ represents weights and bias, $KL(\hat{\rho} || \rho)$ is the sparse regularization to extract sparse representation, and $\hat{\rho}$ is the average output of hidden neurons:

$$KL(\hat{\rho} || \rho) = \sum_{j=1}^{|\hat{\rho}|} \rho \log\left(\frac{\rho}{\hat{\rho}_j}\right) + (1 - \rho) \log\left(\frac{1 - \rho}{1 - \hat{\rho}_j}\right) \tag{6}$$

$$\hat{\rho} = (1/N) \sum_i^N h(x_i) \tag{7}$$

In the training process, for the intractability of the whole image, the model is provided with the original overlapping patches $y_i (i = 1, 2, \dots, N)$ as the reconstruction, and their corrupted image patches x_i as the polluted input. As long as the training is completed, the learned model could reconstruct the corresponding clean image given any polluted observation. The detailed process is shown in Algorithm 1.

Algorithm 1 SparsityAE

Notation: Ω_i is reconstruction error for x_i , S_i is the reconstruction coefficient for x_i , and $\{y_i\}_1^N$ is the hidden representation for every input.

Input: training set $D = \{x_i\}_1^N$, parameters k (constant) and $\theta = \{W, b, W', b'\}$.

Process:

- (1) Compute the S_i for each input $\{x_i\}_1^N$;
- (2) Minimize the cost function by the stochastic gradient descent and update θ ;
- (3) Compute the hidden representation $\{y_i\}_1^N$ for each input, keep the k biggest activation value and others are set to zero, and update S_i and Ω_i ;
- (4) Repeat the step (2) and (3) until convergence.

Output: reconstruction representation of the input.

3.2 Feature Ensemble Method

Multiple sparse features with different abstraction levels will be obtained using the SparsityAE introduced above; ensemble feature learning could effectively improve the accuracy and stability of a single classifier. Together two points above, a BoostingAE is proposed that uses hierarchical feature obtained in pre-training stage to train multiple classifiers, and integrates the outputs of classifiers with specific fusion rules to get the final prediction of image classification.

To make the whole structure easy to understand, Fig. 2 gives a clear and detailed understanding of BoostingAE, which indicates that by cascading multiple SparsityAEs, BoostingAE is theoretically possible to obtain N compressed sparse features derived from the output of SparsityAEs.

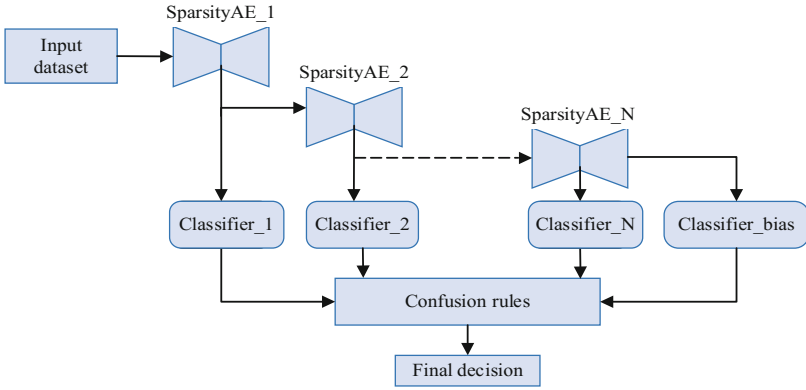


Fig. 2. The topology of the proposed BoostingAE

In our work, training three SparsityAEs and utilizing softmax regression to perform the classification task. First, we train SparsityAE_1 in Fig. 2. Assuming that the original input sample is x , the weight matrix connecting the input layer and the hidden layer is $W^{(1)}$, the bias vector is $b^{(1)}$, so its output can be mapped by Eq. (4):

$$\hat{y}_1 = \sigma(x \bullet W^{(1)} + b^{(1)}) \quad (8)$$

Regard \hat{y}_1 as the input of the second SparsityAE, thus further train SparsityAE_2. Its weight matrix connecting the input layer and the hidden layer is $W^{(2)}$, and bias vector is $b^{(2)}$. With reference to the above operation, SparsityAE_2's output can be further obtained, which also be used as the input of next SparsityAE:

$$\hat{y}_2 = \sigma(\hat{y}_1 \bullet W^{(2)} + b^{(2)}) \quad (9)$$

Along with the cascaded sparsity-induced autoencoder network, the characteristic attributes which are trained from the current layer will be passed to the next layer by

above process, and therefore three SparsityAEs can be trained. At the same time, in the longitudinal direction, the trained classifier model and optimal parameters of each classifier are obtained by training the characteristic attribute at the encoding stage. Further, three classifiers are obtained.

3.3 Combination Method of Voting

After training all base classifiers, final prediction is given by results of three classifiers after integrating with some fusion rules. Here, the Naïve Bayes combination rules [18] are applied which assume that individual classifiers are mutually independent.

We adopt three Naïve Bayes combination methods, namely MAX, MIN, and AVG rules. Given a sample x , and its label y has C possible values. Assuming that the current BoostingAE model consists of N base classifiers, $P_{nj}(x)$ is the probability that the category of x is j in the n th classifier. So label y can be defined as follows:

- MAX rule: $y = \arg \max_{j=1,2,\dots,C} \max_{n=1,2,\dots,N} P_{nj}(x)$,
- MIN rule: $y = \arg \max_{j=1,2,\dots,C} \min_{n=1,2,\dots,N} P_{nj}(x)$,
- AVG rule: $y = \arg \max_{j=1,2,\dots,C} \frac{1}{N} \sum_{n=1}^N P_{nj}(x)$.

3.4 BoostingAE Algorithm

From Fig. 2, a multi-layer architecture based on ensemble learning consists of an input layer, some hidden layers and an output layer to carry out specific tasks.

Here, how to measure the importance of each layer's feature and how to select the optimal models for each layer are two key issues, which will directly influence the performance of the model. As a main contribution of this paper, we employ Adaboost to supervise the adjustment of parameters and weight coefficients. Algorithm 2 gives the detailed process of the proposed BoostingAE.

Algorithm 2 BoostingAE

Notation: T is the number of base sparse autoencoder, and the algorithm of base learner is SparsityAE.

Input: training set $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$.

Process:

Initialize: assign the default parameters.

Loop: for $k = 1, 2, \dots, k$

Generate the new dataset D' by bootstrapping the original dataset;

Obtain the trained individual classifier, $g_k = \text{SparsityAE}(D')$.

Output: Obtain ensemble classifier $D' = \text{Uniform}(\{g_k\})$.

Compared with traditional sparse stacked autoencoder, BoostingAE’s characteristics are shown in following aspects:

- whose construction of base learners is similar to that of AdaBoost, BoostingAE utilizes the cascade serialization mechanism among the base learners, which makes the individual learners are related to each other and also maintain the difference;
- the subsequent layer takes the output of the previous layer as the input to obtain rich feature representation, which makes each learners receive various “training input” at the same time and avoids the waste of computing and storage resources;
- when design individual learners, the topology of each model can be specified separately rather than by a unified model topology. This makes it possible to further increase the diversity of base learners while maintaining the homogeneity of them.

4 Experiments

First of all, we verify the performance of SparsityAE in sparse feature learning. Next, to unbiasedly and accurately show the performance and the stability of BoostingAE on real-world image classification, the experiments are carried out on three widely employed datasets, i.e., MNIST, CIFAR-10, and SVHN. Moreover, some state-of-art methods are employed to provide the comparable results on the same datasets.

4.1 The Sparse Feature Learning of SparsityAE

To validate the performance of SparsityAE in sparse feature learning, we mainly focus on denoising of grey-scale images. From <http://decsai.ugr.es/cvg/dbimagenes>, a set of natural images are employed as the training set, and a set of standard natural images as the testing set which has been widely used in the image processing.

When it comes to the training process, we randomly pick a clean image y from the dataset and generate its corresponding noisy patch x by corrupting it with a specific strength of additive white Gaussian noise. The training performed, the learned model will be capable of reconstructing the corresponding clean image given any noisy observation. To avoid the local minimum, we adopt the layer-wise pre-training procedure introduced in [7].

Figure 3 shows the comparison between SparsityAE and classic image denoising methods: KSVD [19], BM3D [20], on standard testing images degraded by various noise levels. We tell that when $\sigma = 25$, SparsityAE (magenta line) is competitive, while corrupts for other different, i.e., higher noise strengthens, which is owing to that our model knowing nothing about the noise of level but other methods were provided with such information. The green line shows that if we train the proposed model on several different noise levels, our SparsityAE is more robust to the change of noise levels which means that it can generalize significantly better to higher noise levels.

What’s more, we compared the SparsityAE with several state-of-art denoising methods: WNNM [21] and two training based methods: MLP [22], TNRD [23]. The numerical results are shown in Table 1, which is measured by the peak signal to noise ratio (PSNR in dB). The best PSNR result for each image is highlighted in bold.

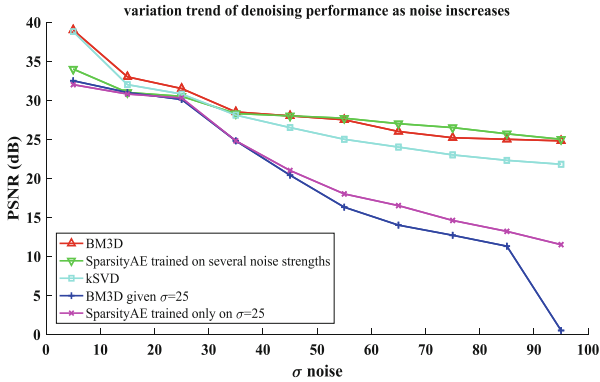


Fig. 3. Denoising performance comparison of various methods with various noise levels (Color figure online)

Table 1. Comparison of the various methods' denoising performance measured by PSNR.

Image	KSVD	BM3D	WNNM	MLP	TNRD	Ours
Lena	31.35	32.08	32.24	32.25	32.00	32.46
House	32.14	32.86	33.23	32.56	32.53	32.96
C.man	28.72	29.45	29.64	29.61	29.72	29.98
Moarch	28.81	29.25	29.85	29.61	29.85	29.65
Couple	28.84	29.71	29.82	29.76	29.69	29.68
Man	29.09	29.61	29.76	29.88	30.11	29.78
Babara	29.60	30.72	31.24	29.54	29.41	29.20
Boat	29.32	29.91	30.03	29.97	30.21	29.91
Pepper	29.71	30.16	30.42	30.30	30.57	30.37

Although images with a lot of repeating structure are ideal for both KSVD and BM3D, we do outperform KSVD, BM3D, and WNNM on every image except Barbara. It is also shown that our SparsityAE is able to compete with MLP and TNPd.

The results illustrated that SparsityAE can not only project the original high dimensional space to a lower dimensional and more intrinsic space from the perspective of dimension-reduction, but capture the more representative sparse feature from multiple layers to make the best use of the information contained in original space.

4.2 BoostingAE for Classification on MNIST

MNIST is a large dataset of handwritten digits that is widely used for image processing and computer vision tasks. It contains 60,000 training images and 10,000 testing images with labels, and the size of a single image is 28×28 .

The topology of the SparsityAE on MNIST with three hidden layers is first determined, i.e., $784 \times 500 \times 250 \times 100 \times 10$, which 784 is the size of the image and

10 is the label number. Then, Classifier_1 in Fig. 2 is obtained by pre-training and fine-tuning SparsityAE_1. After that, SparsityAE_2 takes the feature representations learned from SparsityAE_1 as input to get Classifier_2. With the process above, we get the final three base classifiers that will be integrated when all base learners achieve convergence after fine-tuning. With this, the whole BoostingAE model is constructed and trained completely. When it comes to predicting the real samples, three Naïve Bayes combination rules will be respectively used for voting the integrative result of three classifiers to improve the performance.

Table 2 shows that three individual classifiers have better classification results than KNN and SVM because of the introduction of the sparsity-induced layer in SparsityAE. And the BoostingAE with three different fusion rules gets better performance than any individual classifier and achieves 98.37%, 98.43% and 98.87% accuracy rate respectively, which is very close to the result of L_p -norm AE [24]. Moreover, stacked CAE [25] and CASE [26] employ more feature maps obtained by convolutional operations and hidden layers, so our performance is slightly worse than these.

Table 2. Classification results on three datasets.

	MNIST	CIFAR-10	SVHN
KNN	91.32%	84.47%	78.32%
SVM	94.02%	88.45%	83.24%
L_p -norm AE(KNN) [24]	97.44%	/	67.23%
L_p -norm AE(SVM) [24]	98.64%	/	71.19%
Stacked CAE [25]	99.29%	79.20%	/
CSAE [26]	99.39%	/	/
CDSAe [27]	/	74.18%	/
Classifier_1	96.73%	91.46%	88.46%
Classifier_2	96.35%	91.83%	88.93%
Classifier_3	95.89%	90.96%	87.69%
BoostingAE(MAX)	98.37%	92.32%	89.94%
BoostingAE(MIN)	98.43%	91.49%	90.35%
BoostingAE(AVG)	98.87%	92.63%	90.87%

4.3 BoostingAE for Classification on CIFAR-10 and SVHN

CIFAR-10 is a dataset contains ten kinds of color images, each category contains 6000 color images. The training set contains 5000 images of each category, the remaining is used for testing. SVHN dataset can be regarded as the upgrade of MNIST and also contains ten kinds of color images. Both are captured from the real life so the background is more complex and the images are difficult to identify. SVHN is divided into training set, testing set and extra set; the validation set is constructed in a random way: the 2/3 of them is derived from the training set (400 samples per class), and the remaining samples come from the extra set (200 samples per class).

Before the experiment, the original images of CIFAR-10 and SVHN should be transformed from RGB space into grey space, and then normalized. To improve the training efficiency, the mini-batch gradient descent algorithm is used when pre-training and fine-tuning. Considering the unsupervised learning mechanism of autoencoder, both CIFAR-10 and SVHN use a certain proportion of unlabeled samples as training set in pre-training; and in the process of fine-tuning, two datasets require ground-truth to implement the classification. Next, the topology of SparsityAE is determined as $1024 \times 500 \times 250 \times 100 \times 10$. The subsequent operations are similar to those performed on MNIST.

We report the results of comparison methods, individual classifiers and the proposed method in Table 2 and get the similar conclusion as MNIST. Our methods achieve the best results among comparison methods. The results illustrated the BoostingAE could capture more sparse representation and utilize multi-layer features, resulting in the improvement of accuracy and diversity of overall.

5 Conclusion

In this work, we first built SparsityAE by adding an extra sparsity-induced layer, which efficiently abstract the sparse feature representations, and then based on SparsityAE and ensemble learning, we further proposed a BoostingAE model to integrate sparse feature learned from multi-layer, so as to improve the performance of individual sparse encoder, which has been successfully applied to image classification.

The main advantage of our approach is that it could abstract more significantly sparse representations that reflect the distribution of original data better and make full use of the features learned from multi-layer to improve the diversity of base learners. What's more, it also promotes the overall performance after integrating multiple weak learners. Additional experiments on three different datasets validate the effectiveness of the proposed algorithm in image classification.

References

1. Goh, H., Thome, N., Cord, M., Lim, J.H.: Learning deep hierarchical visual feature coding. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(12), 2212–2225 (2014)
2. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: *IEEE International Conference on Computer Vision*, pp. 3074–3082. IEEE Computer Society Press, Santiago (2015)
3. Yang, X., Ye, W., Li, X., Lau, R.Y.K., Zhang, X., Huang, X.: Hyperspectral image classification with deep learning models. *IEEE Trans. Geosci. Remote Sens.* **56**(9), 1–16 (2018)
4. Mei, S., Jiang, R., Ji, J., Sun, J., Peng, Y.: Invariant feature extraction for image classification via multi-channel convolutional neural network. In: *International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 491–495. IEEE, Japan (2018)

5. Durand, T., Mordan, T., Thome, N., Cord, M.: WILDCAT: weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 642–651. IEEE Computer Society, Hawaii (2017)
6. Marino, K., Salakhutdinov, R., Gupta, A.: The more you know: using knowledge graphs for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2673–2681. IEEE Computer Society Press, Hawaii (2017)
7. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: International conference on Neural Information Processing Systems, pp. 153–160. MIT Press, Vancouver (2006)
8. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
9. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: International conference on Neural Information Processing Systems, pp. 2074–2082. MIT Press, Barcelona (2016)
10. Shahnawazuddin, S., Sinha, R.: Sparse coding over redundant dictionaries for fast adaptation of speech recognition system. *Comput. Speech Lang.* **43**, 1–17 (2017)
11. Srinivas, M., Lin, Y., Liao, H.Y.M.: Learning deep and sparse feature representation for fine-grained object recognition. In: IEEE International Conference on Multimedia and Expo, pp. 1458–1463. IEEE Press, Hong Kong (2017)
12. Ghifary, M., Kleijn, W.B., Zhang, M.: Sparse representations in deep learning for noise-robust digit classification. In: International Conference on Image and Vision Computing New Zealand, pp. 340–345. IEEE Press, Wellington (2013)
13. Zhang, L., Zhou, W.: Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recogn.* **44**(1), 97–106 (2011)
14. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3360–3367. IEEE Press, San Francisco (2010)
15. Gao, S., Tsang, I.W., Chia, L., Zhao, P.: Local features are not Lonely–Laplacian sparse coding for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3555–3561. IEEE Computer Society Press, San Francisco (2010)
16. Ngiam, J., Koh, P.W., Chen, Z., Bhaskar, S., Ng, A.Y.: Sparse filtering. In: International Conference on Neural Information Processing Systems, pp. 1125–1133. MIT Press, Granada (2011)
17. Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Ng, A.Y.: On optimization methods for deep learning. In: 28th International Conference on Machine Learning, pp. 265–272. Omnipress, Bellevue (2011)
18. Kuncheva, L.I.: *Combining Pattern Classifiers: Methods and Algorithms*, 1st edn. Wiley, Hoboken (2004)
19. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.* **15**(12), 3736–3745 (2016)
20. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**(8), 2080–2095 (2007)
21. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2862–2869. IEEE Computer Society Press, Columbus (2014)
22. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2392–2399. IEEE Computer Society Press, Providence (2012)

23. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1256–1272 (2017)
24. Mehta, J., Gupta, K., Gogna, A., Majumdar, A., Anand, S.: Stacked robust autoencoder for classification. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (eds.) *ICONIP 2016*. LNCS, vol. 9949, pp. 600–607. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46675-0_66
25. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) *ICANN 2011*. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_7
26. Luo, W., Li, J., Yang, J.: Convolutional sparse autoencoders for image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**(99), 1–6 (2017)
27. Chen, S., Liu, H., Zheng, X., Qian, S., Yu, J., Guo, W.: Image classification based on convolutional denoising sparse autoencoder. *Math. Probl. Eng.* **2017**, 1–16 (2017)